

Reverse Engineering the Apollo 11 Guidance Computer (AGC) Source Code

Gary Young

proudindiv@gmail.com

July 12, 2016

Contents

1	Introduction and Notes	9
1.1	References	11
2	Oddities Spotted	13
3	Command Modules (CM) Apollo Guidance Computer (AGC), for Apollo 11	15
3.1	complete program	15
3.2	main headers	15
3.2.1	contract and approvals	15
3.2.2	assembly and operation information	17
3.2.3	stags for relative setloc	48
3.3	comerase routine	48
3.3.1	erasable assignments	48
3.4	comaid routines	49
3.4.1	interrupt lead ins	49
3.4.2	st4rupt program	49
3.4.3	downlink lists	49
3.4.4	fresh start and restart	49
3.4.5	restart tables	49
3.4.6	sctmark code	49
3.4.7	extended verbs	50
3.4.8	pinball noun tables	50
3.4.9	csn geometry	50
3.4.10	imu compensation package	50
3.4.11	pinball game buttons and lights	50
3.4.12	r60-62 code	50
3.4.13	anglfind code	50
3.4.14	gimbal lock avoidance	50
3.4.15	kalcmanu steering	50
3.4.16	system test standard lead ins	50
3.4.17	imu calibration and alignment	50
3.5	comekiss routines	51
3.5.1	ground tracking determination program	51

3.5.2	p34-35 p74-75 code	51
3.5.3	r31 code	51
3.5.4	p76 code	51
3.5.5	r30 code	51
3.5.6	stable orbit	51
3.6	trouble routines	52
3.6.1	p11 code	52
3.6.2	tpi search	52
3.6.3	p20 p25 code	52
3.6.4	p30 p37 code	52
3.6.5	p32-p33 p72-p73 code	52
3.6.6	p40-p47 code	52
3.6.7	p51-p53 code	53
3.6.8	lunar and solar ephemerides subroutines	53
3.6.9	p61-p67 code	53
3.6.10	servicer207 code	53
3.6.11	entry lexicon	53
3.6.12	reentry control	53
3.6.13	cm body attitude	53
3.6.14	p37 p70 code	53
3.6.15	s-band antenna for cm	53
3.6.16	lunar landmark selection for cm	53
3.7	tvcdaps routines	54
3.7.1	tvcinitalize code	54
3.7.2	tvcexecutive code	54
3.7.3	tvcmassprop code	54
3.7.4	tvcrestarts code	54
3.7.5	tvcdaps code	54
3.7.6	tvcrestoketest code	54
3.7.7	tvcrestolldap code	54
3.7.8	mysubs code	55
3.7.9	rsc-csm digital autopilot	55
3.7.10	automatic maneuvers	55
3.7.11	rsc-csm dap executive programs	55
3.7.12	jet selection logic	55
3.7.13	cm entry digital autopilot	55
3.8	chieftan routines	56
3.8.1	down-telemetry program	56
3.8.2	inter-bank communication	56
3.8.3	interpreter code	56
3.8.4	fixed fixed constant pool	57
3.8.5	interpretive constants	57
3.8.6	single precision subroutines	57
3.8.7	executive code	57
3.8.8	waitlist code	57
3.8.9	latitude longitude subroutines	57

3.8.10	planetary inertial orientation	57
3.8.11	measurement incorporation	57
3.8.12	conic subroutines	57
3.8.13	integration initialization	57
3.8.14	orbital integration	57
3.8.15	inflight alignment routines	58
3.8.16	powered flight subroutines	58
3.8.17	time of free fall	58
3.8.18	star tables	58
3.8.19	agc block two self check	58
3.8.20	phase table maintenance	58
3.8.21	restarts routine	58
3.8.22	imu mode switching routines	58
3.8.23	keyrupt uprupt	58
3.8.24	display interface routines	58
3.8.25	service routines	58
3.8.26	alarm and abort	59
3.8.27	update program	59
3.8.28	rtb op codes	59

Appendices **61**

A labels **63**

B Original Files **65**

B.1	README.md	65
B.2	HeaderTemplate	67
B.3	AGC BLOCK TWO SELF CHECK	68
B.4	AGC BLOCK TWO SELF-CHECK	80
B.5	AGS INITIALIZATION	92
B.6	ALARM AND ABORT	98
B.7	ANGLFIND	104
B.8	AOSTASK AND AOSJOB	118
B.9	AOTMARK	142
B.10	ASCENT GUIDANCE	158
B.11	ASSEMBLY AND OPERATION INFORMATION	173
B.12	ATTITUDE MANEUVER ROUTINE	176
B.13	AUTOMATIC MANEUVERS	199
B.14	BURN BABY BURN-MASTER IGNITION ROUTINE	210
B.15	CM BODY ATTITUDE	234
B.16	CM ENTRY DIGITAL AUTOPILOT	241
B.17	CONIC SUBROUTINES	269
B.18	CONTRACT AND APPROVALS	311
B.19	CONTROLLED CONSTANTS	312
B.20	CSM GEOMETRY	325
B.21	DAPIDLER PROGRAM	335

B.22 DAP INTERFACE SUBROUTINES	346
B.23 DISPLAY INTERFACE ROUTINES	350
B.24 DOWNLINK LISTS	383
B.25 DOWN TELEMETRY PROGRAM	392
B.26 DOWN-TELEMETRY PROGRAM	402
B.27 ENTRY LEXICON	413
B.28 ERASABLE ASSIGNMENTS	422
B.29 EXECUTIVE	505
B.30 EXTENDED VERBS	516
B.31 FINDCDUW-GUIDAP INTERFACE	546
B.32 FIXED FIXED CONSTANT POOL	563
B.33 FLAGWORD ASSIGNMENTS	569
B.34 FRESH START AND RESTART	597
B.35 GIMBAL LOCK AVOIDANCE	630
B.36 GROUND TRACKING DETERMINATION PROGRAM	633
B.37 IMU CALIBRATION AND ALIGNMENT	638
B.38 IMU COMPENSATION PACKAGE	669
B.39 IMU MODE SWITCHING ROUTINES	678
B.40 IMU PERFORMANCE TEST 2	702
B.41 IMU PERFORMANCE TESTS 4	712
B.42 INFLIGHT ALIGNMENT ROUTINES	720
B.43 INPUT OUTPUT CHANNEL BIT DESCRIPTIONS	727
B.44 INTEGRATION INITIALIZATION	732
B.45 INTER-BANK COMMUNICATION	758
B.46 INTERPRETER	763
B.47 INTERPRETIVE CONSTANT	830
B.48 INTERPRETIVE CONSTANTS	832
B.49 INTERRUPT LEAD INS	834
B.50 JET SELECTION LOGIC	837
B.51 KALCMANU STEERING	858
B.52 KALMAN FILTER	864
B.53 KEYRUPT UPRUPT	867
B.54 LAMBERT AIMPOINT GUIDANCE	871
B.55 LANDING ANALOG DISPLAYS	875
B.56 LATITUDE LONGITUDE SUBROUTINES	887
B.57 LEM GEOMETRY	894
B.58 LUNAR AND SOLAR EPHEMERIDES SUBROUTINES	899
B.59 LUNAR LANDING GUIDANCE EQUATIONS	904
B.60 LUNAR LANDMARK SELECTION FOR CM	937
B.61 MAIN	938
B.62 MEASUREMENT INCORPORATION	942
B.63 MYSUBS	954
B.64 ORBITAL INTEGRATION	957
B.65 P11	978
B.66 P12	999
B.67 P20-P25	1005

B.68 P30 P37	1082
B.69 P30-P37	1087
B.70 P32-P33 P72-P73	1101
B.71 P32-P35 P72-P75	1132
B.72 P34-35 P74-75	1163
B.73 P37 P70	1201
B.74 P40-P47	1244
B.75 P51-P53	1297
B.76 P61-P67	1346
B.77 P70-P71	1373
B.78 P76	1383
B.79 P-AXIS RCS AUTOPILOT	1387
B.80 PHASE TABLE MAINTENANCE	1411
B.81 PINBALL GAME BUTTONS AND LIGHTS	1420
B.82 PINBALL NOUN TABLES	1503
B.83 PLANETARY INERTIAL ORIENTATION	1522
B.84 POWERED FLIGHT SUBROUTINES	1531
B.85 Q R-AXIS RCS AUTOPILOT	1539
B.86 R30	1558
B.87 R31	1569
B.88 R60 62	1576
B.89 R63	1585
B.90 RADAR LEADIN ROUTINES	1589
B.91 RCS-CSM DAP EXECUTIVE PROGRAMS	1592
B.92 RCS-CSM DIGITAL AUTOPILOT	1594
B.93 RCS FAILURE MONITOR	1616
B.94 REENTRY CONTROL	1620
B.95 RESTARTS ROUTINE	1656
B.96 RESTART TABLES	1664
B.97 RT8 OP CODES	1677
B.98 RTB OP CODES	1685
B.99 S-BAND ANTENNA FOR CM	1691
B.100 S-BAND ANTENNA FOR LM	1694
B.101 SERVICER207	1699
B.102 SERVICE ROUTINES	1717
B.103 SERVICER	1723
B.104 SINGLE PRECISION SUBROUTINES	1761
B.105 SPS BACK-UP RCS CONTROL	1763
B.106 STABLE ORBIT	1768
B.107 STAR TABLES	1778
B.108 SXTMARK	1783
B.109 SYSTEM TEST STANDARD LEAD INS	1799
B.110 T4RUPT PROGRAM	1803
B.111 T6-RUPT PROGRAMS	1835
B.112 TAGS FOR RELATIVE SETLOC	1839
B.113 THE LUNAR LANDING	1849

B.114	THROTTLE CONTROL ROUTINES	1857
B.115	TIME OF FREE FALL	1862
B.116	JET LAW	1878
B.117	TPI SEARCH	1890
B.118	TRIM GIMBAL CNTROL SYSTEM	1903
B.119	TVC DAPS	1917
B.120	TVC EXECUTIVE	1935
B.121	TVC INITIALIZE	1942
B.122	TVC MASSPROP	1952
B.123	TVC RESTARTS	1958
B.124	TVC ROLL DAP	1964
B.125	TVC STROKE TEST	1978
B.126	UPDATE PROGRAM	1984
B.127	WAITLIST	1997
C	Miscellaneous Files	2011
D	Notes, Bibliography and Indexes	2015
D.1	Chunks	2015
D.2	Index	2020

Chapter 1

Introduction and Notes

I don't know how much time I'm going to spend on this, nor how far I'll take it, but I'll collect my notes here. I started putting comments on my facebook account, and I'll initially copy them here to keep them together.

- Original Apollo 11 guidance computer (AGC) source code, I wrote programs like this in the same time period, however I started using more abstraction with macros and defines in the assembler. ([chrislgarry/Apollo-11](#))
- Today, I would create a literate program in noweb to reverse engineer the code, integrating the documentation and code into a version of the Forth programming language. Forth would be more compact to fit in the small memories of the computers of those days, but I would have to emulate the stack because those old computers hadn't implemented them in the hardware yet. ([wikipedia](#))
- I wrote assemblers in forth to compile (assemble) this type of code on machines before personal computers that didn't have operating systems.
- Such a macro-assembler could be written in several pages of machine code almost directly from the processor reference card.
- I've cloned the Apollo-11 repository and made my first change to the README.md: ([link](#))
- Well, I've got my first stub reverse engineering of the Apollo-11 code in last night. ([link](#))
- There were no file systems with 80 (or more) characters in a file name. There were no file systems at all. The source

had to have been on punch cards or paper tape. I worked at the Federal Reserve Bank in Seattle then on an IBM 360 main frame and we had to order a select chain to put on the printer for the characters that we could print. We had 26 capital letters "A-Z", 10 digits "0-9", and 6 "special characters". In our case the special characters were chosen for accounting applications: "\$", "#", "_", period, comma, and space. (I think, I can't remember if we had the pound sign.) Notice that there are no lower case letters. That came later, at Bell Labs with UNIX and such, I think.

- There was no floating point arithmetic, only 16 bit integers, from 0 to 65535, or -32767 to 0 to 32767 or possibly -32768 to 32767 depending on whether 0 has a sign and is different from minus zero. Heck there may not have been an integer "divide" operation implemented in the hardware, and probably not even a 16 bit integer multiply. I'll have to look through the source to see. We had to implement multiply and divide in software. Also there was a distinction between "scientific" and "business" computers. My memory is not too good on the details and I have no reference in front of me.
- Aha! I just looked at the first file in my "Apollo-11" book that I just started AND ... "S-ZERO EQUALS NEG0" so plus zero and negative zero were two different numbers!

- I could clone the entire repository and then download it to my Intel Skull Canyon computer from the internet in a fraction of a second. It's 2MB, including all the boiler plate that has been recently added.
- It takes about five seconds to typeset this book. It probably took all night to print the original source listing once.
- The release of the Apollo 11 software and the extreme popularity, with over 2000 clones already, has much larger significance than just a fad. If you look at the IoT phenomenon and the computers that are used there, the Apollo 11 software is a training guide and a huge inspiration on what can be done. Imagine how you could program a self driving car modeling it on the Apollo 11. The Apollo 11 software shows a new generation how computers work at the low level and how they integrate into the real world. The next huge changes to society aren't coming from the corporate world.

1.1 References

- QUARTZ: "BURN, BABY! BURN!" The code that took America to the moon was just published to GitHub, and its like a 1960s time capsule, Written by Keith Collins, July 09, 2016.
- Apollo Guidance Computer emulation by Dean Koska This Palm Centro program is running the computer software used on the Apollo moon landings. It utilizes Ron Burkey's Virtual AGC engine for the emulation (www.ibiblio.org/apollo).
- Virtual AGC Home Page - Ibiblio
- slashdot: Assembly Code That Took America to the Moon Now Published On GitHub
- Skylab quick-reference card
- Space Guidance Analysis (SGA) memos

Chapter 2

Oddities Spotted

Files that are versions of each other:

- AGC_BLOCK_TWO_SELF_CHECK.s and AGC_BLOCK_TWO_SELF-CHECK.s
- DOWN_TELEMETRY_PROGRAM.s and DOWN-TELEMETRY_PROGRAM.s
- P30_P37.s and P30-P37.s (Luminary vs Colossus)

Chapter 3

Command Modules (CM) Apollo Guidance Computer (AGC), for Apollo 11

3.1 complete program

15a $\langle \textit{Apollo-11.s} \ 15a \rangle \equiv$
 $\langle \textit{main headers} \ 15b \rangle$
 $\langle \textit{comerasc routine} \ 48c \rangle$
 $\langle \textit{comaid routines} \ 49a \rangle$
 $\langle \textit{comekiss routines} \ 51a \rangle$
 $\langle \textit{trouble routines} \ 52a \rangle$
 $\langle \textit{tvcdaps routines} \ 54a \rangle$
 $\langle \textit{chieftan routines} \ 56a \rangle$

This code is written to file `Apollo-11.s`.

3.2 main headers

15b $\langle \textit{main headers} \ 15b \rangle \equiv$ (15a)
 $\langle \textit{contract and approvals} \ 15c \rangle$
 $\langle \textit{assembly and operation information} \ 17 \rangle$
 $\langle \textit{stags for relative setloc} \ 48b \rangle$

3.2.1 contract and approvals

15c $\langle \textit{contract and approvals} \ 15c \rangle \equiv$ (15b)
 $\langle \textit{Page 1} \ 16 \rangle$

16 (Page 1 16)≡

(15c 311)

```
*****
#
#          THIS AGC PROGRAM SHALL ALSO BE REFERRED TO AS:
#
#
#          COLOSSUS 2A
#
#
#          THIS PROGRAM IS INTENDED FOR USE IN THE CM AS SPECIFIED
#          IN REPORT R-577.  THIS PROGRAM WAS PREPARED UNDER DSR
#          PROJECT 55-23870, SPONSORED BY THE MANNED SPACECRAFT
#          CENTER OF THE NATIONAL AERONAUTICS AND SPACE
#          ADMINISTRATION THROUGH CONTRACT NAS 9-4065 WITH THE
#          INSTRUMENTATION LABORATORY, MASSACHUSETTS INSTITUTE OF
#          TECHNOLOGY, CAMBRIDGE, MASS.
#
#
*****

#          SUBMITTED:      MARGARET H. HAMILTON          DATE:   28 MAR 69
#          M.H.HAMILTON, COLOSSUS PROGRAMMING LEADER
#          APOLLO GUIDANCE AND NAVIGATION

#          APPROVED:       DANIEL J. LICKLY              DATE:   28 MAR 69
#          D.J.LICKLY, DIRECTOR, MISSION PROGRAM DEVELOPMENT
#          APOLLO GUIDANCE AND NAVIGATION PROGRAM

#          APPROVED:       FRED H. MARTIN                DATE:   28 MAR 69
#          FRED H. MARTIN, COLOSSUS PROJECT MANAGER
#          APOLLO GUIDANCE AND NAVIGATION PROGRAM

#          APPROVED:       NORMAN E. SEARS               DATE:   28 MAR 69
#          N.E. SEARS, DIRECTOR, MISSION DEVELOPMENT
#          APOLLO GUIDANCE AND NAVIGATION PROGRAM

#          APPROVED:       RICHARD H. BATTIN             DATE:   28 MAR 69
#          R.H. BATTIN, DIRECTOR, MISSION DEVELOPMENT
#          APOLLO GUIDANCE AND NAVIGATION PROGRAM

#          APPROVED:       DAVID G. HOAG                DATE:   28 MAR 69
#          D.G. HOAG, DIRECTOR
#          APOLLO GUIDANCE AND NAVIGATION PROGRAM

#          APPROVED:       RALPH R. RAGAN               DATE:   28 MAR 69
#          R.R. RAGAN, DEPUTY DIRECTOR
```


#

INSTRUMENTATION LABORATORY

3.2.2 assembly and operation information

17 $\langle \text{assembly and operation information } 17 \rangle \equiv$ (15b)

$\langle \text{Page } 2 \text{ } 18 \rangle$
 $\langle \text{Page } 3 \text{ } 20 \rangle$
 $\langle \text{Page } 4 \text{ } 21 \rangle$
 $\langle \text{Page } 5 \text{ } 22 \rangle$
 $\langle \text{Page } 6 \text{ } 23 \rangle$
 $\langle \text{Page } 7 \text{ } 24 \rangle$
 $\langle \text{Page } 8 \text{ } 25 \rangle$
 $\langle \text{Page } 9 \text{ } 26\text{a} \rangle$
 $\langle \text{Page } 10 \text{ } 26\text{b} \rangle$
 $\langle \text{Page } 11 \text{ } 27 \rangle$
 $\langle \text{Page } 12 \text{ } 29 \rangle$
 $\langle \text{Page } 13 \text{ } 30 \rangle$
 $\langle \text{Page } 14 \text{ } 31 \rangle$
 $\langle \text{Page } 15 \text{ } 32 \rangle$
 $\langle \text{Page } 16 \text{ } 34 \rangle$
 $\langle \text{Page } 17 \text{ } 36 \rangle$
 $\langle \text{Page } 18 \text{ } 37 \rangle$
 $\langle \text{Page } 19 \text{ } 38 \rangle$
 $\langle \text{Page } 20 \text{ } 40 \rangle$
 $\langle \text{Page } 21 \text{ } 42 \rangle$
 $\langle \text{Page } 22 \text{ } 43 \rangle$
 $\langle \text{Page } 23 \text{ } 44 \rangle$
 $\langle \text{Page } 24 \text{ } 46 \rangle$
 $\langle \text{Page } 25 \text{ } 47 \rangle$
 $\langle \text{Page } 26 \text{ } 48\text{a} \rangle$

18 (*Page 2 18*)≡ (17 173)

```

# ASSEMBLY AND OPERATIONS INFORMATION
# TAGS FOR RELATIVE SETLOC AND BLANK BANK CARDS
# SUBROUTINE CALLS
#   COMERASE
#       ERASABLE ASSIGNMENTS
#   COMAID
#       INTERRUPT LEAD INS
#       T4RUPT PROGRAM
#       DOWNLINK LISTS
#       FRESH START AND RESTART
#       RESTART TABLES
#       SXTMARK
#       EXTENDED VERBS
#       PINBALL NOUN TABLES
#       CSM GEOMETRY
#       IMU COMPENSATION PACKAGE
#       PINBALL GAME BUTTONS AND LIGHTS
#       R60,R62
#       ANGLFIND
#       GIMBAL LOCK AVOIDANCE
#       KALCMANU STEERING
#       SYSTEM TEST STANDARD LEAD INS
#       IMU CALIBRATION AND ALIGNMENT
#   COMEKISS
#       GROUND TRACKING DETERMINATION PROGRAM -- P21
#       P34-P35, P74-P75
#       R31
#       P76
#       R30
#       STABLE ORBIT -- P38-P39
#   TROUBLE
#       P11
#       TP1 SEARCH
#       P20-P25
#       P30,P37
#       P40-P47
#       P51-P53
#       LUNAR AND SOLAR EPHEMERIDES SUBROUTINES
#       P61-P67
#       SERVICER207
#       ENTRY LEXICON
#       REENTRY CONTROL
#       CM BODY ATTITUDE
#       P37,P70
#       S-BAND ANTENNA FOR CM

```

July 12, 2016

Apollo-11.nw 19

```
#          LUNAR LANDMARK SELECTION FOR CM
#          TVCDAPS
#          TVC INITIALIZE
```

20 (Page 3 20)≡

(17 173)

```

# TVC EXECUTIVE
# TVC MASSPROP
# TVC RESTARTS
# TVC DAPS
# TVC STROKE TEST
# TVC ROLLDAP
# MYSUBS
# RCS-CSM DIGITAL AUTOPILOT
# AUTOMATIC MANEUVERS
# RCS-CSM DAP EXECUTIVE PROGRAMS
# JET SELECTION LOGIC
# CM ENTRY DIGITAL AUTOPILOT
# CHIEFTAN
# DOWN-TELEMETRY PROGRAM
# INTER-BANK COMMUNICATION
# INTERPRETER
# FIXED-FIXED CONSTANT POOL
# INTERPRETIVE CONSTANTS
# SINGLE PRECISION SUBROUTINES
# EXECUTIVE
# WAITLIST
# LATITUDE LONGITUDE SUBROUTINES
# PLANETARY INERTIAL ORIENTATION
# MEASUREMENT INCORPORATION
# CONIC SUBROUTINES
# INTEGRATION INITIALIZATION
# ORBITAL INTEGRATION
# INFLIGHT ALIGNMENT ROUTINES
# POWERED FLIGHT SUBROUTINES
# TIME OF FREE FALL
# STAR TABLES
# AGC BLOCK TWO SELF-CHECK
# PHASE TABLE MAINTENANCE
# RESTARTS ROUTINE
# IMU MODE SWITCHING ROUTINES
# KEYRUPT, UPRUPT
# DISPLAY INTERFACE ROUTINES
# SERVICE ROUTINES
# ALARM AND ABORT
# UPDATE PROGRAM
# RTB OP CODES
# SYMBOL TABLE LISTING
# UNREFERANCES SYMBOL LISTING
# ERASABLE & EQUALS CROSS-REFERENCE TABLE
# SUMMARY OF SYMBOL TABLE LISTINGS

```

MEMORY TYPE & AVAILABILITY DISPLAY
COUNT TABLE
PARAGRAPHS GENERATED FOR THIS DISPLAY

21 $\langle Page\ 4\ 21 \rangle \equiv$ (17 173)
OCTAL LISTING
OCCUPIED LOCATIONS TABLE
SUBROS CALLED & PROGRAM STATUS

22 (Page 5 22)≡

(17 173)

VERB LIST FOR CSM

REGULAR VERBS

```
# 00 NOT IN USE
# 01 DISPLAY OCTAL COMP 1 IN R1
# 02 DISPLAY OCTAL COMP 2 IN R1
# 03 DISPLAY OCTAL COMP 3 IN R1
# 04 DISPLAY OCTAL COMP 1,2 IN R1,R2
# 05 DISPLAY OCTAL COMP 1,2,3 IN R1,R2,R3
# 06 DISPLAY DECIMAL IN R1 OR R1,R2 OR R1,R2,R3
# 07 DISPLAY DP DECIMAL IN R1,R2 (TEST ONLY)
# 08
# 09
# 10
# 11 MONITOR OCTAL COMP 1 IN R1
# 12 MONITOR OCTAL COMP 2 IN R1
# 13 MONITOR OCTAL COMP 3 IN R1
# 14 MONITOR OCTAL COMP 1,2, IN R1,R2
# 15 MONITOR OCTAL COMP 1,2,3 IN R1,R2,R3
# 16 MONITOR DECIMAL IN R1 OR R1,R2 OR R1,R2,R3
# 17 MONITOR DP DECIMAL IN R1,R2 (TEST ONLY)
# 18
# 19
# 20
# 21 LOAD COMPONENT 1 INTO R1
# 22 LOAD COMPONENT 2 INTO R2
# 23 LOAD COMPONENT 3 INTO R3
# 24 LOAD COMPONENT 1,2 INTO R1,R2
# 25 LOAD COMPONENT 1,2,3 INTO R1,R2,R3
# 26
# 27 DISPLAY FIXED MEMORY
# 28
# 29
# 30 REQUEST EXECUTIVE
# 31 REQUEST WAITLIST
# 32 RECYCLE PROGRAM
# 33 PROCEED WITHOUT DSKY INPUTS
# 34 TERMINATE FUNCTION
# 35 TEST LIGHTS
# 36 REQUEST FRESH START
# 37 CHANGE PROGRAM (MAJOR MODE)
# 38
# 39
```

23 (Page 6 23)≡

(17 173)

EXTENDED VERBS

```
# 40    ZERO CDU'S
# 41    COARSE ALIGN CDU'S
# 42    FINE ALIGN IMU'S
# 43    LOAD IMU ATT ERROR METERS
# 44    SET    SURFACE FLAG
# 45    RESET SURFACE FLAG
# 46    ESTABLISH G&C CONTROL
# 47    MOVE LM STATE VECTOR INTO CM STATE VECTOR
# 48    REQUEST DAP DATA LOAD ROUTINE (R03)
# 49    REQUES CREW DEFINED MANEUVER ROUTINE (R62)
# 50    PLEASE PERFORM
# 51    PLEASE MARK
# 52    MARK ON OFFSET LANDING SITE
# 53    PLEASE PERFORM ALTERNATE LOS MARK
# 54    REQUEST RENDEZVOUS BACKUP SIGHTING MARK ROUTINE (R23)
# 55    INCREMENT AGC TIME (DECIMAL)
# 56    TERMINATE TRACKING (P20 & P25)
# 57    REQUEST RENDEZVOUS SIGHTING MARK ROUTINE (R21)
# 58    RESET STICK FLAG
# 59    PLEASE CALIBRATE
# 60    SET ASTRONAUT TOTAL ATTITUDE (N17) TO PRESENT ATTITUDE
# 61    DISPLAY DAP ATTITUDE ERROR
# 62    DISPLAY TOTAL ATTITUDE ERROR (W.R.T. N22 (THETAD))
# 63    DISPLAY TOTAL ASTRONAUT ATTITUDE ERROR (W.R.T. N17 (CPHIX))
# 64    REQUEST S-BAND ANTENNA ROUTINE
# 65    OPTICAL VERIFICATION OF PRELAUNCH ALIGNMENT
# 66    VEHICLES ARE ATTACHED.  MOVE THIS VEHICLE STATE TO OTHER VEHICLE.
# 67
# 68    CSM STROKE TEST ON
# 69    CAUSE RESTART
# 70    UPDATE LIFTOFF TIME
# 71    UNIVERSAL UPDATE - BLOCK ADR
# 72    UNIVERSAL UPDATE - SINGLE ADR
# 73    UPDATE AGC TIME (OCTAL)
# 74    INITIALIZE ERASABLE DUMP VIA DOWNLINK
# 75    BACKUP LIFTOFF
# 76    SET PREFERRED ATTITUDE FLAG
# 77    RESET PREFERRED ATTITUDE FLAG
# 78    UPDATE PRELAUNCH AZIMUTH
# 79    REQUEST LUNAR LANDMARK SELECTION ROUTINE (R35)
# 80    UPDATE LEM STATE VECTOR
# 81    UPDATE CSM STATE VECTOR
# 82    REQUEST ORBIT PARAM DISPLAY (R30)
```

```
# 83 REQUEST REND PARAM DISPLAY (R31)
# 84 START TARGET DELTA V (R32)
# 85 REQUEST RENDEZVOUS PARAMETER DISPLAY NO. 2 (R34)
# 86 REJECT RENDEZVOUS BACKUP SIGHTING MARK
# 87 SET VHF RANGE FLAG
```

24 \langle Page 7 24 $\rangle \equiv$

(17 173)

```
# 88 RESET VHF RANGE FLAG
# 89 REQUEST RENDEZVOUS FINAL ATTITUDE ROUTINE (R63)
# 90 REQUEST RENDEZVOUS OUT OF PLANE DISPLAY ROUTINE (R36)
# 91 DISPLAY BANK SUM
# 92 OPERATE IMU PERFORMANCE TEST (P07)
# 93 ENABLE W MATRIX INITIALIZATION
# 94 PERFORM SYSLUNAR ATTITUDE MANEUVER (P23)
# 95 NO UPDATE OF EITHER STATE VECTOR (P20 OR P22)
# 96 TERMINATE INTEGRATION AND GO TO P00
# 97 PERFORM ENGINE FAIL PROCEDURE
# 98 ENABLE TRANSLUNAR INJECT
# 99 PLEASE ENABLE ENGINE
```


25 (Page 8 25)≡

(17 173)

IN THE FOLLOWING NOUN LIST THE 'NO LOAD' RESTRICTION MEANS THE NOUN
CONTAINS AT LEAST ONE COMONENT WHICH CANNOT BE LOADED, I.E. OF
SCALE TYPE L (MIN/SEC) OR PP (2 INTEGERS).

IN THIS CASE VERBS 24 AND 25 ARE NOT ALLOWED, BUT VERBS 21, 22, OR 23
MAY BE USED TO LOAD ANY OF THE NOUN'S COMPONENTS WHICH ARE NOT OF THE
ABOVE SCALE TYPES.

THE 'DEC ONLY' RESTRICTION MEANS ONLY DECIMAL OPERATION IS ALLOWED ON
EVERY COMPONENT IN THENOUN. (NOT THAT 'NO LOAD' IMPLIES 'DEC ONLY'.)

#	NORMAL NOUNS	COMPONENTS	SCALE & DECIMAL POINT	RESTRICTION
# 00	NOT IN USE			
# 01	SPECIFY MACHINE ADDRESS (FRACTIONAL)	3COMP	.XXXXX FOR EACH	
# 02	SPECIFY MACHINE ADDRESS (WHOLE)	3COMP	XXXXX. FOR EACH	
# 03	SPECIFY MACHINE ADDRESS (DEGREES)	3COMP	XXX.XX DEG FOR EACH	
# 04	SPARE			
# 05	ANGULAR ERROR/DIFFERENCE	1COMP	XXX.XX DEG	
# 06	OPTION CODE	2COMP	OCTAL ONLY FOR EACH	
# 07	LOADING NOUN 07 WILL SET OR RESET SELECTED BITS IN ANY ERASABLE REGISTER.			
#	ECADR OF WORD TO BE MODIFIED	3COMP	OCTAL ONLY FOR EACH	
#	ONES FOR BITS TO BE MODIFIED			
#	1 TO SET OR 0 TO RESET SELECTED BITS			
# 08	ALARM DATA	3COMP	OCTAL ONLY FOR EACH	
# 09	ALARM CODES	3COMP	OCTAL ONLY FOR EACH	
# 10	CHANNEL TO BE SPECIFIED	1COMP	OCTAL ONLY	
# 11	TIG OF CSI	3COMP	00XXX. HRS	DEC ONLY
#			000XX. MIN	MUST LOAD 3 COM
#			0XX.XX SEC	
# 12	OPTION CODE	2COMP	OCTAL ONLY FOR EACH	
#	(USED BY EXTENDED VERBS ONLY)			
# 13	TIG OF CDH	3COMP	00XXX. HRS	DEC ONLY
#			000XX. MIN	MUST LOAD 3 COM
#			0XX.XX SEC	
# 14	SPARE			
# 15	INCREMENT MACHINE ADDRESS	1COMP	OCTAL ONLY	
# 16	TIME OF EVENT	3COMP	00XXX. HRS	DEC ONLY
#	(USED BY EXTENDED VERBS ONLY)		000XX. MIN	MUST LOAD 3 COM
#			0XX.XX SEC	
# 17	ASTRONAUT TOTAL ATTITUDE	3COMP	XXX.XX DEG FOR EACH	
# 18	AUTO MANEUVER BALL ANGLES	3COMP	XXX.XX DEG FOR EACH	
# 19	BYPASS ATTITUDE TRIM MANEUVER	3COMP	XXX.XX DEG FOR EACH	
# 20	ICDU ANGLES	3COMP	XXX.XX DEG FOR EACH	
# 21	PIPAS	3COMP	XXXXX. PULSES FOR EACH	

	# 22	NEW ICDU ANGLES	3COMP	XXX.XX DEG FOR EACH	
	# 23	SPARE			
	# 24	DELTA TIME FOR AGC CLOCK	3COMP	00XXX. HRS.	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 25	CHECKLIST	3COMP	XXXXX. FOR EACH	
	#	(USED WITH PLEASE PERFORM ONLY)			
26a	<Page 9 26a>≡			(17 173)	
	# 26	PRIORITY/DELAY, ADRES, BBON	3COMP	OCTAL ONLY FOR EACH	
	# 27	SELF TEST ON/OFF SWITCH	1COMP	XXXXX.	
	# 28	SPARE			
	# 29	XSM LAUNCH AZIMUTH	1COMP	XXX.XX DEG	DEC 0
26b	<Page 10 26b>≡			(17 173)	
	# 30	TARGET CODES	3COMP	XXXXX. FOR EACH	
	# 31	TIME OF LANDING SITE	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 32	TIME FROM PERIGEE	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 33	TIME OF IGNITION	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 34	TIME OF EVENT	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 35	TIME FROM EVENT	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 36	TIME OF AGC CLOCK	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 37	TIG OF TPI	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 38	TIME OF STATE VECTOR	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	
	# 39	DELTA TIME FOR TRANSFER	3COMP	00XXX. HRS	DEC 0
	#			000XX. MIN	MUST
	#			0XX.XX SEC	

July 12, 2016

Apollo-11.nw 27

27 (Page 11 27)≡

#	MIXED NOUNS	COMPONENTS	(17 173) SCALE & DECIMAL POINT	RESTRICTION
#				
# 40	TIME FROM IGNITION/CUTOFF	3COMP	XXBXX MIN/SEC	NO LOAD, DEC ON
#	VG		XXXX.X FT/SEC	
#	DELTA V (ACCUMULATED)		XXXX.X FT/SEC	
# 41	TARGET AZIMUTH	2COMP	XXX.XX DEG	
#	ELEVATION		XX.XXX DEG	
# 42	APOGEE	3COMP	XXXX.X NAUT MI	DEC ONLY
#	PERIGEE		XXXX.X NAUT MI	
#	DELTA V (REQUIRED)		XXXX.X FT/SEC	
# 43	LATITUDE	3COMP	XXX.XX DEG	DEC ONLY
#	LONGITUDE		XXX.XX DEG	
#	ALTITUDE		XXXX.X NAUT MI	
# 44	APOGEE	3COMP	XXXX.X NAUT MI	NO LOAD, DEC ON
#	PERIGEE		XXXX.X NAUT MI	
#	TFF		XXBXX MIN/SEC	
# 45	MARKS (VHF - OPTICS)	3COMP	+XXBXX	NO LOAD, DEC ON
#	TFI OF NEXT BURN		XXBXX MIN/SEC	
#	MGA		XXX.XX DEG	
# 46	AUTOPILOT CONFIGURATION	2COMP	OCTAL ONLY FOR EACH	
# 47	THIS VEHICLE WEIGHT	2COMP	XXXXX. LBS	DEC ONLY
#	OTHER VEHICLE WEIGHT		XXXXX. LBS	
# 48	PITCH TRIM	2COMP	XXX.XX DEG	DEC ONLY
#	YAW TRIM		XXX.XX DEG	
# 49	DELTA R	3COMP	XXXX.X NAUT MI	DEC ONLY
#	DELTA V		XXXX.X FT/SEC	
#	VHF OR OPTICS CODE		XXXXX.	
# 50	SPLASH ERROR	3COMP	XXXX.X NAUT MI	NO LOAD, DEC ON
#	PERIGEE		XXXX.X NAUT MI	
#	TFF		XXBXX MIN/SEC	
# 51	S-BAND ANTENNA ANGLES PITCH	2COMP	XXX.XX DEG	DEC ONLY
#	YAW		XXX.XX DEG	
# 52	CENTRAL ANGLE OF ACTIVE VEHICLE	1COMP	XXX.XX DEG	
# 53	RANGE	3COMP	XXX.XX NAUT MI	DEC ONLY
#	RANGE RATE		XXXX.X FT/SEC	
#	PHI		XXX.X DEG	
# 54	RANGE	3COMP	XXX.XX NAUT MI	DEC ONLY
#	RANGE RATE		XXXX.X FT/SEC	
#	THETA		XXX.XX DEG	
# 55	PERIGEE CODE	3COMP	XXXXX.	DEC ONLY
#	ELEVATION ANGLE		XXX.XX DEG	
#	CENTRAL ANGLE OF PASSIVE VEHICLE		XXX.XX DEG	
# 56	REENTRY ANGLE	2COMP	XXX.XX DEG	DEC ONLY
#	DELTA V		XXXXX. FT/SEC	
# 57	DELTA R	1COMP	XXXX.X NAUT MI	DEC ONLY

28 Apollo-11.nw

July 12, 2016

# 58	PERIGEE ALT (POST TPI)	3COMP	XXXX.X NAUT MI	DEC 0
#	DELTA V TPI		XXXX.X FT/SEC	
#	DELTA V TPF		XXXX.X FT/SEC	
# 59	DELTA VELOCITY LOS	3COMP	XXXX.X FT/SEC FOR EACH	DEC 0
# 60	GMAX	3COMP	XXX.XX G	DEC 0

29 (Page 12 29)≡

(17 173)

#	VPRED		XXXXX. FT/SEC	
#	GAMMA EI		XXX.XX DEG	
# 61	IMPACT LATITUDE	3COMP	XXX.XX DEG	DEC ONLY
#	IMPACT LONGITUDE		XXX.XX DEG	
#	HEADS UP/DOWN		+/- 00001	
# 62	INERTIAL VEL MAG (VI)	3COMP	XXXXX. FT/SEC	DEC ONLY
#	ALT RATE CHANGE (HDOT)		XXXXX. FT/SEC	
#	ALT ABOVE PAD RADIUS (H)		XXXX.X NAUT MI	
# 63	RANGE 297,431 TO SPLASH (RTGO)	3COMP	XXXX.X NAUT MI	NO LOAD, DEC ON
#	PREDICTED INERT VEL (VIO)		XXXXX. FT/SEC	
#	TIME FROM 297,431 (TFE)		XXBXX MIN/SEC	
# 64	DRAG ACCELERATION	3COMP	XXX.XX G	DEC ONLY
#	INERTIAL VELOCITY (VI)		XXXXX. FT/SEC	
#	RANGE TO SPLASH		XXXX.X NAUT MI	
# 65	SAMPLED AGC TIME	3COMP	00XXX. HRS	DEC ONLY
#	(FETCHED IN INTERRUPT)		000XX. MIN	
#			0XX.XX SEC	
# 66	COMMAND BANK ANGLE (BETA)	3COMP	XXX.XX DEG	DEC ONLY
#	CROSS RANGE ERROR		XXXX.X NAUT MI	
#	DOWN RANGE ERROR		XXXX.X NAUT MI	
# 67	RANGE TO TARGET	3COMP	XXXX.X NAUT MI	DEC ONLY
#	PRESENT LATITUDE		XXX.XX DEG	
#	PRESENT LONGITUDE		XXX.XX DEG	
# 68	COMMAND BANK ANGLE (BETA)	3COMP	XXX.XX DEG	DEC ONLY
#	INERTIAL VELOCITY (VI)		XXXXX. FT/SEC	
#	ALT RATE CHANGE (RDOT)		XXXXX. FT/SEC	
# 69	BETA	3COMP	XXX.XX DEG	
#	DL		XXX.XX G	
#	VL		XXXXX. FT/SEC	
# 70	STAR CODE	3COMP	OCTAL ONLY	
#	LANDMARK DATA		OCTAL ONLY	
#	HORIZON DATA		OCTAL ONLY	
# 71	STAR CODE	3COMP	OCTAL ONLY	
#	LANDMARK DATA		OCTAL ONLY	
#	HORIZON DATA		OCTAL ONLY	
# 72	DELT ANG	3COMP	XXX.XX DEG	DEC ONLY
# 73	ALTITUDE	3COMP	XXXXXB. NAUT MI	
#	VELOCITY		XXXXX. FT/SEC	
#	FLIGHT PATH ANGLE		XXX.XX DEG	
# 74	COMMAND BANK ANGLE (BETA)	3COMP	XXX.XX DEG	
#	INERTIAL VELOCITY (VI)		XXXXX. FT/SEC	
#	DRAG ACCELERATION		XXX.XX G	
# 75	DELTA ALTITUDE CDH	3COMP	XXXX.X NAUT MI	NO LOAD, DEC ON
#	DELTA TIME (CDH-CSI OR TPI-CDH)		XXBXX MIN/SEC	
#	DELTA TIME (TPI-CDH OR TPI-NOMTPI)		XXBXX MIN/SEC	

# 76	SPARE				
# 77	SPARE				
# 78	SPARE				
# 79	SPARE				
# 80	TIME FROM IGNITION/CUTOFF	3COMP	XXBXX MIN/SEC		NO L
30	\langle Page 13 30 $\rangle \equiv$		(17 173)		
#	VG		XXXXX. FT/SEC		
#	DELTA V (ACCUMULATED)		XXXXX. FT/SEC		
# 81	DELTA V (LV)	3COMP	XXXX.X FT/SEC FOR EACH		DEC 0
# 82	DELTA V (LV)	3COMP	XXXX.X FT/SEC FOR EACH		DEC 0
# 83	DELTA V (BODY)	3COMP	XXXX.X FT/SEC FOR EACH		DEC 0
# 84	DELTA V (OTHER VEHICLE)	3COMP	XXXX.X FT/SEC FOR EACH		DEC 0
# 85	VG (BODY)	3COMP	XXXX.X FT/SEC FOR EACH		DEC 0
# 86	DELTA V (LV)	3COMP	XXXXX. FT/SEC FOR EACH		DEC 0
# 87	MARK DATA	2COMP	XXX.XX DEG		
#			XX.XXX DEG		
# 88	HALF UNIT SUN OR PLANET VECTOR	3COMP	.XXXXX FOR EACH		DEC 0
# 89	LANDMARK	3COMP	XX.XXX DEG		DEC 0
#			XX.XXX DEG		
#			XXX.XX NAUT MI		
# 90	Y	3COMP	XXX.XX NM		DEC 0
#	Y DOT		XXXX.X FPS		
#	PSI		XXX.XX DEG		
# 91	OCDU ANGLES	2COMP	XXX.XX DEG		
#			XX.XXX DEG		
# 92	NEW OPTICS ANGLES	2COMP	XXX.XX DEG		
#			XX.XXX DEG		
# 93	DELTA GYRO ANGLES	3COMP	XX.XXX DEG FOR EACH		
# 94	NEW OPTICS ANGLES	2COMP	XXX.XX DEG		
#			XX.XXX DEG		
# 95	PREFERRED ATTITUDE ICDU ANGLES	3COMP	XXX.XX FOR EACH		
# 96	+X-AXIS ATTITUDE ICDU ANGLES	3COMP	XXX.XX DEG FOR EACH		
# 97	SYSTEM TEST INPUTS	3COMP	XXXXX. FOR EACH		
# 98	SYSTEM TEST RESULTS AND INPUTS	3COMP	XXXXX.		
#			.XXXXX		
#			XXXXX.		
# 99	RMS IN POSITION	3COMP	XXX.XX NAUT MI		DEC 0
#	RMS IN VELOCITY		XXXX.X FT/SEC		
#	RMS OPTION		XXXXX.		

31 (Page 14 31)≡

(17 173)

```

# REGISTERS AND SCALING FOR NORMAL NOUNS
#
# NOUN  REGISTER          SCALE TYPE
#
# 00    NOT IN USE
# 01    SPECIFY ADDRESS    H
# 02    SPECIFY ADDRESS    C
# 03    SPECIFY ADDRESS    D
# 04    SPARE
# 05          DSPTEM1      H
# 06          OPTION1      A
# 07          XREG         A
# 08          ALMCADR      A
# 09          FAILREG      A
# 10    SPECIFY CHANNEL    A
# 11          TCSI         K
# 12          OPTIONX      A
# 13          TCDH         K
# 14    SPARE
# 15    INCREMENT ADDRESS  A
# 16          DSPTEMX      C
# 17          CPHIX        D
# 18          THETAD       D
# 19          THETAD       D
# 20          CDUX         D
# 21          PIPAX        C
# 22          THETAD       D
# 23    SPARE
# 24          DSPTEM2 +1    K
# 25          DSPTEM1      C
# 26          DSPTEM1      A
# 27          SMODE        C
# 28    SPARE
# 29          DSPTEM1      D
# 30          DSPTEM1      C
# 31          DSPTEM1      K
# 32          -TPER        K
# 33          TIG          K
# 34          DSPTEM1      K
# 35          TTOGO        K
# 36          TIME2        K
# 37          TTP1         K
# 38          TET          K
# 39          T3TOT4       K

```

32 (Page 15 32)≡

(17 173)

REGISTERS AND SCALING FOR MIXED NOUNS

#	NOUN	COMP	REGISTER	SCALE TYPE
#				
# 40	1		TTOGO	L
#	2		VGDISP	S
#	3		DVTOTAL	S
# 41	1		DSPTM1	D
#	2		DSPTM1 +1	E
# 42	1		HAPO	Q
#	2		HPER	Q
#	3		VGDISP	S
# 43	1		LAT	H
#	2		LONG	H
#	3		ALT	Q
# 44	1		HAPOX	Q
#	2		HPERX	Q
#	3		TFF	L
# 45	1		VHFCNT	PP
#	2		TTOGO	L
#	3		+MGA	H
# 46	1		DAPDATR1	A
#	2		DAPDATR2	A
# 47	1		CSMMASS	KK
#	2		LEMMASS	KK
# 48	1		PACTOFF	FF
#	2		YACTOFF	FF
# 49	1		N49DISP	Q
#	2		N49DISP +2	S
#	3		N49DISP +4	C
# 50	1		RSP-RREC	LL
#	2		HPERX	Q
#	3		TFF	L
# 51	1		RHOSB	H
#	2		GAMMASB	H
# 52	1		ACTCENT	H
# 53	1		RANGE	JJ
#	2		RRATE	S
#	3		RTHETA	H
# 54	1		RANGE	JJ
#	2		RRATE	S
#	3		RTHETA	H
# 55	1		NN1	C
#	2		ELEV	H
#	3		CENTANG	H

# 56	1	RTEGAM2D	H
#	2	RTEDVD	P
# 57	1	DELTAR	Q
# 58	1	POSTTPI	Q
#	2	DELVTPI	S

34 (Page 16 34)≡

(17 173)

#	3	DELVTPF	S
# 59	1	DVLOS	S
#	2	DVLOS +2	S
#	3	DVLOS +4	S
# 60	1	GMAX	T
#	2	VPRED	P
#	3	GAMMAEI	H
# 61	1	LAT (SPL)	H
#	2	LNG (SPL)	H
#	3	HEADSUP	C
# 62	1	VMAGI	P
#	2	HDOT	P
#	3	ALTI	Q
# 63	1	RTGO	LL
#	2	VIO	P
#	3	TTE	L
# 64	1	D	MM
#	2	VMAGI	P
#	3	RTGON64	LL
# 65	1	SAMPTIME	K
#	2	SAMPTIME	K
#	3	SAMPTIME	K
# 66	1	ROLLC	H
#	2	XRNGERR	VV
#	3	DNRNGERR	LL
# 67	1	RTGON67	LL
#	2	LAT	H
#	3	LONG	H
# 68	1	ROLLC	H
#	2	VMAGI	P
#	3	RDOT	UU
# 69	1	ROLLC	H
#	2	Q7	MM
#	3	VL	UU
# 70	1	STARCODE	A
#	2	LANDMARK	A
#	3	HORIZON	A
# 71	1	STARCODE	A
#	2	LANDMARK	A
#	3	HORIZON	A
# 72	1	THETZERO	H
# 73	1	P21ALT	Q (MEMORY/100 TO DISPLAY TENS N.M.)
#	2	P21VEL	P
#	3	P21GAM	H
# 74	1	ROLLC	H

July 12, 2016

Apollo-11.nw 35

#	2	VMAGI	P
#	3	D	MM
# 75	1	DIFFALT	Q
#	2	T1TOT2	L
#	3	T2TOT3	L

36 (Page 17 36)≡

(17 173)

# 76	SPARE		
# 77	SPARE		
# 78	SPARE		
# 79	SPARE		
# 80	1	TTOGO	L
#	2	VGDISP	P
#	3	DVTOTAL	P
# 81	1	DEVLVC	S
#	2	DEVLVC +2	S
#	3	DEVLVC +4	S
# 82	1	DEVLVC	S
#	2	DEVLVC +2	S
#	3	DEVLVC +4	S
# 83	1	DELVIMU	S
#	2	DELVIMU +2	S
#	3	DELVIMU +4	S
# 84	1	DELVOV	S
#	2	DELVOV +2	S
#	3	DELVOV +4	S
# 85	1	VGBODY	S
#	2	VGBODY +2	S
#	3	VGBODY +4	S
# 86	1	DEVLVC	P
#	2	DEVLVC +2	P
#	3	DEVLVC +4	P
# 87	1	MRKBUF1 +3	D
#	2	MRKBUF1 +5	J
# 88	1	STARSAV	ZZ
#	2	STARSAV +2	ZZ
#	3	STARSAV +4	ZZ
# 89	1	LANDLAT	G
#	2	LANDLONG	G
#	3	LANDALT	JJ
# 90	1	RANGE	JJ
#	2	RRATE	S
#	3	RTHETA	H
# 91	1	CDUS	D
#	2	CDUT	J
# 92	1	SAC	D
#	2	PAC	J
# 93	1	OGC	G
#	2	OGC +2	G
#	3	OGC +4	G
# 94	1	MRKBUF1 +3	D
#	2	MRKBUF1 +5	J

# 95	1	PRAXIS	D
#	2	PRAXIS +1	D
#	3	PRAXIS +2	D
# 96	1	CPHIX	D
#	2	CPHIX +1	D

37 $\langle \text{Page } 18 \text{ } 37 \rangle \equiv$ (17 173)

#	3	CPHIX +2	D
# 97	1	DSPTM1	C
#	2	DSPTM1 +1	C
#	3	DSPTM1 +2	C
# 98	1	DSPTM2	C
#	2	DSPTM2 +1	B
#	3	DSPTM2 +2	C
# 99	1	WWPOS	XX
#	2	WWVEL	YY
#	3	WWOPT	C

38 (Page 19 38)≡

(17 173)

NOUN SCALES AND FORMATS

#

-SCALE TYPE-

PRECISION

UNITS

DECIMAL FORMAT

--

AGC FORMAT

--

#

-A-

OCTAL

XXXXX

SP

OCTAL

#

-B-

FRACTIONAL

.XXXXX
(MAX .99996)

SP

-14
BIT 1 = 2 UNITS

#

-C-

WHOLE

XXXXX.
(MAX 16383.)

SP

BIT 1 = 1 UNIT

#

-D-

CDU DEGREES

XXX.XX DEGREES
(MAX 359.99)

SP

15
BIT 1 = 360/2 DEGREES
(USES 15 BITS FOR MAGNITUDE AND 2'S COMP.)

#

-E-

ELEVATION DEGREES

XX.XXX DEGREES
(MAX 89.999)

SP

14
BIT 1 = 90/2 DEGREES

#

-F-

DEGREES (180)

XXX.XX DEGREES
(MAX 179.99)

SP

14
BIT 1 = 180/2 DEGREES

#

-G-

DP DEGREES (90)

XX.XXX DEGREES

DP

BIT 1 OF LOW REGISTER =
28
360/2 DEGREES

#

#

-H-

DP DEGREES (360)

XXX.XX DEGREES
(MAX 359.99)

DP

BIT 1 OF LOW REGISTER =
28
360/2 DEGREES

#

-J-

Y OPTICS DEGREES

XX.XXX DEGREES
(BIAS OF 19.775
DEGREES ADDED FOR
DISPLAY, SUBTRACTED

SP

15
BIT 1 = 90/2 DEGREES
(USES 15 BITS FOR MAGNITUDE AND S'S COMP.)

July 12, 2016

Apollo-11.nw 39

```
#          FOR LOAD.)
#          NOTE:  NEGATIVE NUM-
#          BERS CANNOT BE
#          LOADED.
#
# -K-
```

40 (Page 20 40)≡

# TIME (HR, MIN, SEC)	00XXX. HR	DP	(17 173) BIT 1 OF LOW REGISTER =
#	000XX. MIN		-2
#	OXX.XX SEC		10 SEC
#	(DECIMAL ONLY.		
#	MAX MIN COMP = 59		
#	MAX SEC COMP = 59.99		
#	MAX CAPACITY = 745 HRS		
#	39 MINS		
#	14.55 SECS.		
#	WHEN LOADING, ALL 3		
#	COMPONENTS MUST BE		
#	SUPPLIED.)		
# -L-			
# TIME (MIN/SEC)	XXBXX MIN/SEC	DP	BIT 1 OF LOW REGISTER =
#	(B IS A BLANK		-2
#	POSITION, DECIMAL		10 SEC
#	ONLY, DISPLAY OR		
#	MONITOR ONLY. CANNOT		
#	BE LOADED.		
#	MAX MIN COMP = 59		
#	MAX SEC COMP = 59		
#	VALUES GREATER THAN		
#	59 MIN 59 SEC		
#	ARE DISPLAYED AS		
#	59 MIN 59 SEC.)		
#			
# -M-			-2
# TIME (SEC)	XXX.XX SEC	SP	BIT 1 = 10 SEC
#	(MAX 163.83)		
#			
# -N-			
# TIME (SEC) DP	XXX.XX SEC	DP	BIT 1 OF LOW REGISTER =
#			-2
#			10 SEC
#			
# -P-			
# VELOCITY 2	XXXXX. FEET/SEC	DP	BIT 1 OF HIGH REGISTER =
#	(MAX 41994.)		-7
#			2 METERS/CENTI-SEC
#			
# -Q-			
# POSITION 4	XXXX.XX NAUTICAL MILES	DP	BIT 1 OF LOW REGISTER =
#			2 METERS.
#			

July 12, 2016

Apollo-11.nw 41

# -S-			
# VELOCITY 3	XXXX.X FT/SEC	DP	BIT 1 OF HIGH REGISTER =
#			-7
#			2 METERS/CENTI-SEC

42 (Page 21 42)≡

(17 173)

-T-

-2

G

XXX.XX G
(MAX 163.83)

SP

BIT 1 = 10 G

#

#

-FF-

TRIM DEGREES

XXX.XX DEG.
(MAX 388.69)

SP

LOW ORDER BIT = 85.41 SEC
OF ARC

#

-GG-

INERTIA

XXXXXBB. SLUG FT SQ
(MAX 07733BB.)

SP

FRACTIONAL PART OF
20 2
2 KG M

#

#

-II-

THRUST MOMENT

XXXXXBB. FT LBS
(MAX 07733BB.)

SP

FRACTIONAL PART OF 20
2
NEWTON METER

#

-JJ-

POSITION5

XXX.XX NAUT MI

DP

BIT 1 OF LOW REGISTER =
2 METERS

#

#

-KK-

WEIGHT2

XXXXX. LBS

SP

FRACTIONAL PART OF 16
2 KG

#

-LL-

POSITION6

XXXX.X NAUT MI

DP

BIT 1 OF LOW REG =
-28
(6,373,338)(2(PI))x2

1852
NAUT MI.

#

#

#

#

#

#

-MM-

DRAG ACCELERATION

XXX.XX G
MAX (024.99)

DP

BIT 1 OF LOW REGISTER =
-28
25x2 G

#

#

#

-PP-

2 INTEGERS

+XXBYY
(B IS A BLANK
POSITION. DECIMAL
ONLY, DISPLAY, OR
MONITOR ONLY. CANNOT
BE LOADED.)
(MAX 99B99)

DP

BIT 1 OF HIGH REGISTER =
1 UNIT OF XX
BIT 1 OF LOW REGISTER =
1 UNIT OF YY
(EACH REGISTER MUST
LESS THAN 100.)

#

#

#

#

#

#

Apollo-11.nw 43

43

44 (Page 23 44)≡

(17 173)

ALARM CODES FOR 504

REPORT DEFICIENCIES TO JOHN SUTHERLAND: MIT 617-864-6900 X1458

# *9	*18	*60
#		
# CODE	* TYPE	SET BY
#		
# 00110	NO MARK SINCE LAST MARK REJECT	SXTMARK
# 00112	MARK NOT BEING ACCEPTED	SXTMARK
# 00113	NO NBITS	SXTMARK
# 00114	MARK MADE BUT NOT DESIRED	SXTMARK
# 00115	OPTICS TORQUE REQUEST WITH SWITCH NOT AT CGC	EXT VERB OPTICS CDU
#		
# 00116	OPTICS SWITCH ALTERED BEFORE 15 SEC ZERO TIME ELAPSED.	T4RUPT
#		
# 00117	OPTICS TORQUE REQUEST WITH OPTICS NOT AVAILABLE (OPTIND=-0)	EXT VERB OPTICS CDU
#		
# 00120	OPTICS TORQUE REQUEST WITH OPTICS NOT ZEROED.	T4RUPT
#		
# 00121	CDUS NO GOOD AT TIME OF MARK	SXTMARK
# 00122	MARKING NOT CALLED FOR	SXTMARK
# 00124	P17 TPI SEARCH - NO SAFE PERICTR HERE.	TPI SEARCH
# 00205	BAD PIPA READING	SERVICER
# 00206	ZERO ENCODE NOT ALLOWED WITH COARSE ALIGN + GIMBAL LOCK.	IMU MODE SWITCHING
#		
# 00207	ISS TURNON REQUEST NOT PRESENT FOR 90 SEC	T4RUPT
# 00210	IMU NOT OPERATING	IMU MODE SWITCH, IMU-2, R02, P51
#		
# 00211	COARSE ALIGN ERROR - DRIVE > 2 DEGREES	IMU MODE SWITCH
# 00212	PIPA FAIL BUT PIPA IS NOT BEING USED	IMU MODE SWITCH, T4RUPT
# 00213	IMU NOT OPERATING WITH TURN-ON REQUEST	T4RUPT
# 00214	PROGRAM USING IMU WHEN TURNED OFF	T4RUPT
# 00215	PREFERRED ORIENTATION NOT SPECIFIED	P52,P54
# 00217	BAD RETURN FROM STALL ROUTINES	CURTAINS
# 00220	IMU NOT ALIGNED - NO REFSMMAT	R02,P51
# 00401	DESIRED GIMBAL ANGLES YIELD GIMBAL LOCK	IMF ALIGN, IMU-2
# 00404	TARGET OUT OF VIEW - TRUN ANGLE > 90 DEG	R52
# 00405	TWO STARS NOT AVAILABLE	P52,P54
# 00406	REND NAVIGATION NOT OPERATING	P21,R23
# 00407	AUTO OPTICS REQUEST TRUN ANGLE > 50 DEG.	R52
# 00421	W-MATRIX OVERFLOW	INTEGRV
# 00430	* INTEG. ABORT DUE TO SUBSURFACE S. V.	ALL CALLS TO INTEG
# 00600	IMAGINARY ROOTS ON FIRST ITERATION	P32, P72
# 00601	PERIGEE ALTITUDE LT PMIN1	P32,P72

July 12, 2016

Apollo-11.nw

45

# 00602	PERIGEE ALTITUDE LT PMIN2	P32,P72	VARALAR
# 00603	CSI TO CDH TIME LT PMIN22	P32,P72,P33,P73	VARALAR
# 00604	CDH TO TPI TIME LT PMIN23	P32,P72	VARALAR
# 00605	NUMBER OF ITERATIONS EXCEEDS LOOP MAXIMUM	P32,P72,P37	VARALAR
# 00606	DV EXCEEDS MAXIMUM	P32,P72	VARALAR
# 00607	* NO SOLN FROM TIME-THETA OR TIME-RADIUS	TIMETHET,TIMERAD	POODOO

46 (Page 24 46)≡

(17 173)

# 00610	*	LAMBDA LESS THAN UNITY	P37
# 00611		NO TIG FOR GIVEN ELEV ANGLE	P34,P74
# 00612		STATE VECTOR IN WRONG SPHERE OF INFLUENCE	P37
# 00613		REENTRY ANGLE OUT OF LIMITS	P37
# 00777		PIPA FAIL CAUSED ISS WARNING.	T4RUPT
# 01102		CMC SELF TEST ERROR	
# 01103	*	UNUSED CCS BRANCH EXECUTED	ABORT
# 01104	*	DELAY ROUTINE BUSY	EXEC
# 01105		DOWNLINK TOO FAST	T4RUPT
# 01106		UPLINK TOO FAST	T4RUPT
# 01107		PHASE TABLE FAILURE. ASSUME	RESATRT
#		ERASABLE MEMORY IS DESTROYED	
# 01201	*	EXECUTIVE OVERFLOW - NO VAC AREAS	EXEC
# 01202	*	EXECUTIVE OVERFLOW - NO CORE SETS	EXEC
# 01203	*	WAITLIST OVERFLOW - TOO MANY TASKS	WAITLIST
# 01204	*	NEGATIVE OR ZERO WAITLIST CALL	WAITLIST
# 01206	*	SECOND JOB ATTEMPTS TO GO TO SLEEP	PINBALL
#		VIA KEYBOARD AND DISPLAY PROGRAM	
# 01207	*	NO VAC AREA FOR MARKS	SXTMARK
# 01210	*	TWO PROGRAMS USING DEVICE AT SAME TIME	IMU MODE SWITCH
# 01211	*	ILLEGAL INTERRUPT OF EXTENDED VERB	SXTMARK
# 01301		ARCSIN-ARCCOS ARGUMENT TOO LARGE	INTERPRETER
# 01302	*	SQRT CALLED WITH NEGATIVE ARGUMENT. ABORT.	INTERPRETER
# 01407		VG INCREASING	S40.8
# 01426		IMU UNSATISFACTORY	P61,P62
# 01427		IMU REVERSED	P61,P62
# 01501	*	KEYBOARD AND DISPLAY ALARM DURING	PINBALL
#		INTERNAL USE (NVSUB). ABORT.	
# 01502	*	ILLEGAL FLASHING DISPLAY	GOPLAY
# 01520		V37 REQUEST NOT PERMITTED AT THIS TIME	V37
# 01521	*	P01 ILLEGALLY SELECTED	P01, P07
# 01600		OVERFLOW IN DRIFT TEST	OPT PRE ALIGN CALIB
# 01601	*	BAD IMU TORQUE - ABORT.	OPT PRE ALIGN CALIB
# 01602		BAD OPTICS DURING VERIFICATION	OPTALGN CALIB (CSM)
# 01703		INSUF. TIME FOR INTEG., TIG WAS SLIPPED	R41
# 03777		ICDU FAIL CAUSED THE ISS WARNING	T4RUPT
# 04777		ICDU, PIPA FAILS CAUSED THE ISS WARNING	T4RUPT
# 07777		IMU FAIL CAUSED THE ISS WARNING	T4RUPT
# 10777		IMU, PIPA FAILS CAUSED THE ISS WARNING	T4RUPT
# 13777		IMU, ICDU FAILS CAUSED THE ISS WARNING	T4RUPT
# 14777		IMU, ICDU, PIPA FAILS CAUSED THE ISS WARNING	T4RUPT
#			
#	*	INDICATES ABORT TYPE. ALL OTHERS ARE NON-ABORTIVE	

July 12, 2016

Apollo-11.nw 47

47 (Page 25 47)≡

(17 173)

CHECKLIST CODES FOR 504

PLEASE REPORT ANY DEFICIENCIES IN THIS LIST TO JOHN SUTHERLAND.

*9 *17 *26 COLUMN

#

R1 CODE ACTION TO BE EFFECTED

#

00014 KEY IN FINE ALIGNMENT OPTION

00015 PERFORM CELESTIAL BODY ACQUISITION

00016 KEY IN TERMINATE MARK SEQUENCE

00041 SWITCH CM/SM SEPARATION TO UP

00062 SWITCH AGC POWER DOWN

00202 PERFORM GNCS AUTOMATIC MANEUVER

00203 SWITCH TO CMC-AUTO

00204 PERFORM SPS GIMBAL TRIM

00403 SWITCH OPTICS TO MANUAL OR ZERO

#

SWITCH DENOTES CHANGE OF POSITION OF A CONSOLE SWITCH

PERFORM DENOTES START OF END OF A TASK

KEY IN DENOTES KEY IN OF DATA THRU THE DSKY

48a \langle Page 26 48a $\rangle \equiv$

(17 173)

OPTION CODES FOR 504

PLEASE REPORT ANY DEFICIENCIES IN THIS LIST TO JOHN SUTHERLAND.

THE SPECIFIED OPTION CODES WILL BE FLASHED IN COMPONENT R1 IN
 # CONJUNCTION WITH VERBO4NOUN06 TO REQUEST THE ASTRONAUT TO LOAD INTO
 # COMPONENT R2 THE OPTION HE DESIRES.

# *9	*17	#52	#11
#			
# OPTION			
# CODE	PURPOSE	INPUT FOR COMPONENT 2	PROG
#			
# 00001	SPECIFY IMU ORIENTATION	1=PREF 2=NOM 3=REFSMMAT	P50'S
# 00002	SPECIFY VEHICLE	1=THIS 2=OTHER	P21, P
# 00003	SPECIFY TRACKING ATTITUDE	1=PREFERRED 2=OTHER	R63
# 00004	SPECIFY RADAR	1=RR 2=LR	R04
# 00005	SPECIFY SOR PHASE	1=FIRST 2=SECOND	P38
# 00006	SPECIFY RR COARSE ALIGN OPTION	1=LOCKON 2=CONTINUOUS DESIG.	V41N
# 00007	SPECIFY PROPULSION SYSTEM	1=SPS 2=RCS	P37
# 00010	SPECIFY ALIGNEMENT MODE	0=ANY TIME 1=REFSMMAT + G	P57
#		2=TWO BODIES 3=ONE BODY + G	
# 00011	SPEC. SEPARATION MONITOR PHASE	1=DELTAV 2=STATE VECTOR UPDATE	P46
# 00012	SPECIFY CSM ORBIT OPTION	1=NO ORBIT CHANGE 2=CHANGE	P22
#		ORBIT TO PASS OVER LM	

3.2.3 stags for relative setloc

48b \langle stags for relative setloc 48b $\rangle \equiv$

(15b)

3.3 comerase routine

48c \langle comerase routine 48c $\rangle \equiv$

(15a)

 \langle erasable assignments 48d \rangle

3.3.1 erasable assignments

48d \langle erasable assignments 48d $\rangle \equiv$

(48c)

3.4 comaid routines

49a $\langle \text{comaid routines 49a} \rangle \equiv$ (15a)
 $\langle \text{interrupt lead ins 49b} \rangle$
 $\langle \text{st4rupt program 49c} \rangle$
 $\langle \text{downlink lists 49d} \rangle$
 $\langle \text{fresh start and restart 49e} \rangle$
 $\langle \text{restart tables 49f} \rangle$
 $\langle \text{sctmark code 49g} \rangle$
 $\langle \text{extended verbs 50a} \rangle$
 $\langle \text{pinball noun tables 50b} \rangle$
 $\langle \text{csm geometry 50c} \rangle$
 $\langle \text{imu compensation package 50d} \rangle$
 $\langle \text{pinball game buttons and lights 50e} \rangle$
 $\langle \text{r60-62 code 50f} \rangle$
 $\langle \text{anglfind code 50g} \rangle$
 $\langle \text{gimbal lock avoidance 50h} \rangle$
 $\langle \text{kalcmanu steering 50i} \rangle$
 $\langle \text{system test standard lead ins 50j} \rangle$
 $\langle \text{imu calibration and alignment 50k} \rangle$

3.4.1 interrupt lead ins

49b $\langle \text{interrupt lead ins 49b} \rangle \equiv$ (49a)

3.4.2 st4rupt program

49c $\langle \text{st4rupt program 49c} \rangle \equiv$ (49a)

3.4.3 downlink lists

49d $\langle \text{downlink lists 49d} \rangle \equiv$ (49a)

3.4.4 fresh start and restart

49e $\langle \text{fresh start and restart 49e} \rangle \equiv$ (49a)

3.4.5 restart tables

49f $\langle \text{restart tables 49f} \rangle \equiv$ (49a)

3.4.6 sctmark code

49g $\langle \text{sctmark code 49g} \rangle \equiv$ (49a)

3.4.7 extended verbs

50a $\langle \textit{extended verbs 50a} \rangle \equiv$ (49a)

3.4.8 pinball noun tables

50b $\langle \textit{pinball noun tables 50b} \rangle \equiv$ (49a)

3.4.9 csm geometry

50c $\langle \textit{csm geometry 50c} \rangle \equiv$ (49a)

3.4.10 imu compensation package

50d $\langle \textit{imu compensation package 50d} \rangle \equiv$ (49a)

3.4.11 pinball game buttons and lights

50e $\langle \textit{pinball game buttons and lights 50e} \rangle \equiv$ (49a)

3.4.12 r60-62 code

50f $\langle \textit{r60-62 code 50f} \rangle \equiv$ (49a)

3.4.13 anglfind code

50g $\langle \textit{anglfind code 50g} \rangle \equiv$ (49a)

3.4.14 gimbal lock avoidance

50h $\langle \textit{gimbal lock avoidance 50h} \rangle \equiv$ (49a)

3.4.15 kalcmanu steering

50i $\langle \textit{kalcmanu steering 50i} \rangle \equiv$ (49a)

3.4.16 system test standard lead ins

50j $\langle \textit{system test standard lead ins 50j} \rangle \equiv$ (49a)

3.4.17 imu calibration and alignment

50k $\langle \textit{imu calibration and alignment 50k} \rangle \equiv$ (49a)

3.5 comekiss routines

51a $\langle \text{comekiss routines 51a} \rangle \equiv$ (15a)
 $\langle \text{ground tracking determination program 51b} \rangle$
 $\langle \text{p34-35 p74-75 code 51c} \rangle$
 $\langle \text{r31 code 51d} \rangle$
 $\langle \text{p76 code 51e} \rangle$
 $\langle \text{r30 code 51f} \rangle$
 $\langle \text{stable orbit 51g} \rangle$

3.5.1 ground tracking determination program

51b $\langle \text{ground tracking determination program 51b} \rangle \equiv$ (51a)

3.5.2 p34-35 p74-75 code

51c $\langle \text{p34-35 p74-75 code 51c} \rangle \equiv$ (51a)

3.5.3 r31 code

51d $\langle \text{r31 code 51d} \rangle \equiv$ (51a)

3.5.4 p76 code

51e $\langle \text{p76 code 51e} \rangle \equiv$ (51a)

3.5.5 r30 code

51f $\langle \text{r30 code 51f} \rangle \equiv$ (51a)

3.5.6 stable orbit

51g $\langle \text{stable orbit 51g} \rangle \equiv$ (51a)

3.6 trouble routines

52a $\langle \textit{trouble routines 52a} \rangle \equiv$ (15a)

$\langle \textit{p11 code 52b} \rangle$
 $\langle \textit{tpi search 52c} \rangle$
 $\langle \textit{p20 p25 code 52d} \rangle$
 $\langle \textit{p30 p37 code 52e} \rangle$
 $\langle \textit{p32-p33 p72-p73 code (never defined)} \rangle$
 $\langle \textit{p40-p47 code 52f} \rangle$
 $\langle \textit{p51-p53 code 53a} \rangle$
 $\langle \textit{lunar and solar ephemerides subroutines 53b} \rangle$
 $\langle \textit{p61-p67 code 53c} \rangle$
 $\langle \textit{servicer207 code 53d} \rangle$
 $\langle \textit{entry lexicon 53e} \rangle$
 $\langle \textit{reentry control 53f} \rangle$
 $\langle \textit{cm body attitude 53g} \rangle$
 $\langle \textit{p37 p70 code 53h} \rangle$
 $\langle \textit{s-band antenna for cm 53i} \rangle$
 $\langle \textit{lunar landmark selection for cm 53j} \rangle$

3.6.1 p11 code

52b $\langle \textit{p11 code 52b} \rangle \equiv$ (52a)

3.6.2 tpi search

52c $\langle \textit{tpi search 52c} \rangle \equiv$ (52a)

3.6.3 p20 p25 code

52d $\langle \textit{p20 p25 code 52d} \rangle \equiv$ (52a)

3.6.4 p30 p37 code

52e $\langle \textit{p30 p37 code 52e} \rangle \equiv$ (52a)

3.6.5 p32-p33 p72-p73 code

p32-p33 p72-p73 code \equiv

3.6.6 p40-p47 code

52f $\langle \textit{p40-p47 code 52f} \rangle \equiv$ (52a)

3.6.7 p51-p53 code

53a $\langle p51-p53 \text{ code } 53a \rangle \equiv$ (52a)

3.6.8 lunar and solar ephemerides subroutines

53b $\langle lunar \text{ and solar ephemerides subroutines } 53b \rangle \equiv$ (52a)

3.6.9 p61-p67 code

53c $\langle p61-p67 \text{ code } 53c \rangle \equiv$ (52a)

3.6.10 servicer207 code

53d $\langle servicer207 \text{ code } 53d \rangle \equiv$ (52a)

3.6.11 entry lexicon

53e $\langle entry \text{ lexicon } 53e \rangle \equiv$ (52a)

3.6.12 reentry control

53f $\langle reentry \text{ control } 53f \rangle \equiv$ (52a)

3.6.13 cm body attitude

53g $\langle cm \text{ body attitude } 53g \rangle \equiv$ (52a)

3.6.14 p37 p70 code

53h $\langle p37 \text{ p70 code } 53h \rangle \equiv$ (52a)

3.6.15 s-band antenna for cm

53i $\langle s\text{-band antenna for cm } 53i \rangle \equiv$ (52a)

3.6.16 lunar landmark selection for cm

53j $\langle lunar \text{ landmark selection for cm } 53j \rangle \equiv$ (52a)

3.7 tvcdaps routines

54a $\langle tvcdaps \text{ routines } 54a \rangle \equiv$ (15a)
 $\langle tvcinitialize \text{ code } 54b \rangle$
 $\langle tvcexecutive \text{ code } 54c \rangle$
 $\langle tvcmassprop \text{ code } 54d \rangle$
 $\langle tvcrestarts \text{ code } 54e \rangle$
 $\langle tvcdaps \text{ code } 54f \rangle$
 $\langle tvcstroketest \text{ code } 54g \rangle$
 $\langle tvcrolldap \text{ code } 54h \rangle$
 $\langle mysubs \text{ code } 55a \rangle$
 $\langle rcs-csm \text{ digital autopilot } 55b \rangle$
 $\langle automatic \text{ maneuvers } 55c \rangle$
 $\langle rcs-csm \text{ dap executive programs } 55d \rangle$
 $\langle jet \text{ selection logic } 55e \rangle$
 $\langle cm \text{ entry digital autopilot } 55f \rangle$

3.7.1 tvcinitialize code

54b $\langle tvcinitialize \text{ code } 54b \rangle \equiv$ (54a)

3.7.2 tvcexecutive code

54c $\langle tvcexecutive \text{ code } 54c \rangle \equiv$ (54a)

3.7.3 tvcmassprop code

54d $\langle tvcmassprop \text{ code } 54d \rangle \equiv$ (54a)

3.7.4 tvcrestarts code

54e $\langle tvcrestarts \text{ code } 54e \rangle \equiv$ (54a)

3.7.5 tvcdaps code

54f $\langle tvcdaps \text{ code } 54f \rangle \equiv$ (54a)

3.7.6 tvcstroketest code

54g $\langle tvcstroketest \text{ code } 54g \rangle \equiv$ (54a)

3.7.7 tvcrolldap code

54h $\langle tvcrolldap \text{ code } 54h \rangle \equiv$ (54a)

3.7.8 mysubs code

55a $\langle \textit{mysubs code 55a} \rangle \equiv$ (54a)

3.7.9 rcs-csm digital autopilot

55b $\langle \textit{rcs-csm digital autopilot 55b} \rangle \equiv$ (54a)

3.7.10 automatic maneuvers

55c $\langle \textit{automatic maneuvers 55c} \rangle \equiv$ (54a)

3.7.11 rcs-csm dap executive programs

55d $\langle \textit{rcs-csm dap executive programs 55d} \rangle \equiv$ (54a)

3.7.12 jet selection logic

55e $\langle \textit{jet selection logic 55e} \rangle \equiv$ (54a)

3.7.13 cm entry digital autopilot

55f $\langle \textit{cm entry digital autopilot 55f} \rangle \equiv$ (54a)

3.8 chieftan routines

56a \langle chieftan routines 56a $\rangle \equiv$ (15a)

\langle down-telemetry program 56b \rangle
 \langle inter-bank communication 56c \rangle
 \langle interpreter code 56d \rangle
 \langle fixed fixed constant pool 57a \rangle
 \langle interpretive constants 57b \rangle
 \langle single precision subroutines 57c \rangle
 \langle executive code 57d \rangle
 \langle waitlist code 57e \rangle
 \langle latitude longitude subroutines 57f \rangle
 \langle planetary inertial orientation 57g \rangle
 \langle measurement incorporation 57h \rangle
 \langle conic subroutines 57i \rangle
 \langle integration initialization 57j \rangle
 \langle orbital integration 57k \rangle
 \langle inflight alignment routines 58a \rangle
 \langle powered flight subroutines 58b \rangle
 \langle time of free fall 58c \rangle
 \langle star tables 58d \rangle
 \langle agc block two self check 58e \rangle
 \langle phase table maintenance 58f \rangle
 \langle restarts routine 58g \rangle
 \langle imu mode switching routines 58h \rangle
 \langle keyrupt uprupt 58i \rangle
 \langle display interface routines 58j \rangle
 \langle service routines 58k \rangle
 \langle alarm and abort 59a \rangle
 \langle update program 59b \rangle
 \langle rtb op codes 59c \rangle

3.8.1 down-telemetry program

56b \langle down-telemetry program 56b $\rangle \equiv$ (56a)

3.8.2 inter-bank communication

56c \langle inter-bank communication 56c $\rangle \equiv$ (56a)

3.8.3 interpreter code

56d \langle interpreter code 56d $\rangle \equiv$ (56a)

3.8.4 fixed fixed constant pool

57a $\langle \textit{fixed fixed constant pool 57a} \rangle \equiv$ (56a)

3.8.5 interpretive constants

57b $\langle \textit{interpretive constants 57b} \rangle \equiv$ (56a)

3.8.6 single precision subroutines

57c $\langle \textit{single precision subroutines 57c} \rangle \equiv$ (56a)

3.8.7 executive code

57d $\langle \textit{executive code 57d} \rangle \equiv$ (56a)

3.8.8 waitlist code

57e $\langle \textit{waitlist code 57e} \rangle \equiv$ (56a)

3.8.9 latitude longitude subroutines

57f $\langle \textit{latitude longitude subroutines 57f} \rangle \equiv$ (56a)

3.8.10 planetary inertial orientation

57g $\langle \textit{planetary inertial orientation 57g} \rangle \equiv$ (56a)

3.8.11 measurement incorporation

57h $\langle \textit{measurement incorporation 57h} \rangle \equiv$ (56a)

3.8.12 conic subroutines

57i $\langle \textit{conic subroutines 57i} \rangle \equiv$ (56a)

3.8.13 integration initialization

57j $\langle \textit{integration initialization 57j} \rangle \equiv$ (56a)

3.8.14 orbital integration

57k $\langle \textit{orbital integration 57k} \rangle \equiv$ (56a)

3.8.15 inflight alignment routines

58a $\langle \textit{inflight alignment routines 58a} \rangle \equiv$ (56a)

3.8.16 powered flight subroutines

58b $\langle \textit{powered flight subroutines 58b} \rangle \equiv$ (56a)

3.8.17 time of free fall

58c $\langle \textit{time of free fall 58c} \rangle \equiv$ (56a)

3.8.18 star tables

58d $\langle \textit{star tables 58d} \rangle \equiv$ (56a)

3.8.19 agc block two self check

58e $\langle \textit{agc block two self check 58e} \rangle \equiv$ (56a)

3.8.20 phase table maintenance

58f $\langle \textit{phase table maintenance 58f} \rangle \equiv$ (56a)

3.8.21 restarts routine

58g $\langle \textit{restarts routine 58g} \rangle \equiv$ (56a)

3.8.22 imu mode switching routines

58h $\langle \textit{imu mode switching routines 58h} \rangle \equiv$ (56a)

3.8.23 keyrupt uprupt

58i $\langle \textit{keyrupt uprupt 58i} \rangle \equiv$ (56a)

3.8.24 display interface routines

58j $\langle \textit{display interface routines 58j} \rangle \equiv$ (56a)

3.8.25 service routines

58k $\langle \textit{service routines 58k} \rangle \equiv$ (56a)

3.8.26 alarm and abort

59a $\langle alarm\ and\ abort\ 59a \rangle \equiv$ (56a)

3.8.27 update program

59b $\langle update\ program\ 59b \rangle \equiv$ (56a)

3.8.28 rtb op codes

59c $\langle rtb\ op\ codes\ 59c \rangle \equiv$ (56a)

Appendices

Appendix A

labels

Appendix B

Original Files

B.1 README.md

65 `<src/README.md 65>≡`
`Apollo-11`
`=====`

Original Apollo 11 guidance computer (AGC) source code, in assembly, for Command Module (Coman

####Attribution

Copyright: Public domain.
Filename: CONTRACT_AND_APPROVALS.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Mod history: 2009-05-06 RSB Transcribed from page images.

This source code has been transcribed or otherwise adapted from digitized images of a hardcopy from the MIT Museum. The digitization was performed by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many thanks to both. The images (with suitable reduction in storage size and consequent reduction in image quality as well) are available online at www.ibiblio.org/apollo. If for some reason you find that the images are illegible, contact me at info@sandroid.org about getting access to the (much) higher-quality images which Paul actually created.

Notations on the hardcopy document read, in part:

Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969

Page 1

```
#####  
#  
#           THIS AGC PROGRAM SHALL ALSO BE REFERRED TO AS:           *  
#  
#  
#  
#           COLOSSUS 2A                                             *  
#  
#  
#  
# THIS PROGRAM IS INTENDED FOR USE IN THE CM AS SPECIFIED           *  
# IN REPORT R-577.  THIS PROGRAM WAS PREPARED UNDER DSR             *  
# PROJECT 55-23870, SPONSORED BY THE MANNED SPACECRAFT              *  
# CENTER OF THE NATIONAL AERONAUTICS AND SPACE                     *  
# ADMINISTRATION THROUGH CONTRACT NAS 9-4065 WITH THE              *  
# INSTRUMENTATION LABORATORY, MASSACHUSETTS INSTITUTE OF           *  
# TECHNOLOGY, CAMBRIDGE, MASS.                                       *  
#  
#####
```

SUBMITTED: MARGARET H. HAMILTON DATE: 28 MAR 69
M.H.HAMILTON, COLOSSUS PROGRAMMING LEADER
APOLLO GUIDANCE AND NAVIGATION

APPROVED: DANIEL J. LICKLY DATE: 28 MAR 69
D.J.LICKLY, DIRECTOR, MISSION PROGRAM DEVELOPMENT
APOLLO GUIDANCE AND NAVIGATION PROGRAM

APPROVED: FRED H. MARTIN DATE: 28 MAR 69
FRED H. MARTIN, COLOSSUS PROJECT MANAGER
APOLLO GUIDANCE AND NAVIGATION PROGRAM

APPROVED: NORMAN E. SEARS DATE: 28 MAR 69
N.E. SEARS, DIRECTOR, MISSION DEVELOPMENT
APOLLO GUIDANCE AND NAVIGATION PROGRAM

APPROVED: RICHARD H. BATTIN DATE: 28 MAR 69
R.H. BATTIN, DIRECTOR, MISSION DEVELOPMENT
APOLLO GUIDANCE AND NAVIGATION PROGRAM

APPROVED: DAVID G. HOAG DATE: 28 MAR 69

D.G. HOAG, DIRECTOR
 APOLLO GUIDANCE AND NAVIGATION PROGRAM

APPROVED: RALPH R. RAGAN DATE: 28 MAR 69
 R.R. RAGAN, DEPUTY DIRECTOR
 INSTRUMENTATION LABORATORY

This code is written to file `src/README.md`.

B.2 HeaderTemplate

```
67  <src/HeaderTemplate.s 67>≡
    # Copyright:   Public domain.
    # Filename:    XXXXXXXX.agc
    # Purpose:     Part of the source code for Luminary 1A build 099.
    #              It is part of the source code for the Lunar Module's (LM)
    #              Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:  yaYUL
    # Contact:     Ron Burkey <info@sandroid.org>.
    # Website:     www.ibiblio.org/apollo.
    # Pages:       XXXX-XXXX
    # Mod history:  2009-05-XX XXX  Adapted from the corresponding
    #              Luminary131 file, using page
    #              images from Luminary 1A.
    #
    # This source code has been transcribed or otherwise adapted from
    # digitized images of a hardcopy from the MIT Museum. The digitization
    # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
    # the Museum. Many thanks to both. The images (with suitable reduction
    # in storage size and consequent reduction in image quality as well) are
    # available online at www.ibiblio.org/apollo. If for some reason you
    # find that the images are illegible, contact me at info@sandroid.org
    # about getting access to the (much) higher-quality images which Paul
    # actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
    # 16:27 JULY 14, 1969
```

This code is written to file `src/HeaderTemplate.s`.

B.3 AGC BLOCK TWO SELF CHECK

```

68  <src/AGC-BLOCK-TWO-SELF-CHECK.s 68>≡
    # Copyright:      Public domain.
    # Filename:       AGC_BLOCK_TWO_SELF_CHECK.agc
    # Purpose:        Part of the source code for Luminary 1A build 099.
    #                 It is part of the source code for the Lunar Module's (LM)
    #                 Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:     yaYUL
    # Contact:        Onno Hommes <ohommes@cmu.edu>.
    # Website:        www.ibiblio.org/apollo.
    # Pages:          1284-1293
    # Mod history:    2009-05-27 OH   Transcribed from page images.
    #
    # This source code has been transcribed or otherwise adapted from
    # digitized images of a hardcopy from the MIT Museum. The digitization
    # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
    # the Museum. Many thanks to both. The images (with suitable reduction
    # in storage size and consequent reduction in image quality as well) are
    # available online at www.ibiblio.org/apollo. If for some reason you
    # find that the images are illegible, contact me at info@sandroid.org
    # about getting access to the (much) higher-quality images which Paul
    # actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
    #       16:27 JULY 14, 1969
    #
    # Page 1284
    # PROGRAM DESCRIPTION                                DATE: 20 DECEMBER 1967
    # PROGRAM NAME -- SELF-CHECK                        LOG SECTION: AGC BLOCK TWO SELF-CHECK
    # MOD NO -- 1                                       ASSEMBLY SUBROUTINE UTILITM REV 25
    # MOD BY -- GAUNTT
    #
    # FUNCTIONAL DESCRIPTION
    #       PROGRAM HAS TWO MAIN PARTS. THE FIRST IS SELF-CHECK WHICH RUNS AS A ZERO PR
    #       PART OF THE BACK-UP IDLE LOOP. THE SECOND IS SHOW-BANKSUM WHICH RUNS AS A RE
    #       STARTING VERB.
    #
    #       THE PURPOSE OF SELF-CHECK IS TO CHECK OUT VARIOUS PARTS OF THE COMPUTER AS OU
    #
    #       THE PURPOSE OF SHOW-BANKSUM IS TO DISPLAY THE SUM OF EACH BANK, ONE AT A TIME
    #
    #       IN ALL THERE ARE 7 POSSIBLE OPTIONS IN THIS BLOCK II VERSION OF SELF-CHECK.
    #       FOUND IN E-2065 BLOCK II AGC SELF-CHECK AND SHOW BANKSUM BY EDWIN D. SMALLY D

```

```

#
# THE DIFFERENT OPTIONS ARE CONTROLLED BY PUTTING DIFFERENT NUMBERS IN THE SMODE REGISTER
# A DESCRIPTION OF WHAT PARTS OF THE COMPUTER THAT ARE CHECKED BY THE OPTIONS, AND THE CO
# OCTAL, TO LOAD INTO SMODE.
#      +-4          ERASABLE MEMORY
#      +-5          FIXED MEMORY
#      +-1,2,3,6,7,10  EVERYTHING IN OPTIONS 4 AND 5.
#      -0           SAME AS +-10 UNTIL AN ERROR IS DETECTED.
#      +0           NO CHECK, PUTS COMPUTER INTO THE BACKUP IDLE LOOP.
#
# WARNINGS
#      USE OF E MEMORY RESERVED FOR SELF-CHECK (EVEN IN IDLE LOOP) AS TEMP STORAGE BY OTHER PR
#      SMODE SET GREATER THAN OCT 10 PUTS COMPUTER INTO BACKUP IDLE LOOP.
#
# CALLING SEQUENCE
#      TO CALL SELF-CHECK KEY IN
#          V 21 N 27 E      OPTION NUMBER E
#      TO CALL SHOW-BANKSUM KEY IN
#          V 91 E          DISPLAYS FIRST BANK
#          V 33 E          PROCEED, DISPLAYS NEXT BANK
#
# EXIT MODES, NORMAL AND ALARM
#      SELF-CHECK NORMALLY CONTINUES INDEFINITELY UNLESS THERE IS AN ERROR DETECTED.  IF SO +
#      COMPUTER INTO BACKUP IDLE LOOP, - OPTIONS NUMBERS RESTART THE OPTION.
#
#      THE -0 OPTION PROCEEDS FROM THE LINE FOLLOWING THE LINE WHERE THE ERROR WAS DETECTED.
#
#      SHOW-BANKSUM PROCEEDS UNTIL A TERMINATE IS KEYED IN (V 34 E).  THE COMPUTER IS PUT INTO
#
# OUTPUT
# Page 1285
#      SELF-CHECK UPON DETECTING AN ERROR LOADS THE SELF-CHECK ALARM CONSTANT (01102) INTO THE
#      TURNS ON THE ALARM LIGHT.  THE OPERATOR MAY THEN DISPLAY THE THREE FAILREGS BY KEYING I
#      INFORMATION HE MAY KEY IN V 05 N 08 E, THE DSKY DISPLAY IN R1 WILL BE ADDRESS+1 OF WHER
#      IN R2 THE BBCON OF SELF-CHECK, AND IN R3 THE TOTAL NUMBER OF ERRORS DETECTED BY SELF-CH
#      INITIATED FRESH START (SLAP1).
#
#      SHOW-BANKSUM STARTING WITH BANK 0 DISPLAYS IN R1 THE BANK SUM (A +-NUMBER EQUAL TO THE
#      THE BANK NUMBER, AND IN R3 THE BUGGER WORD.
#
# ERASABLE INITIALIZATION REQUIRED
#      ACCOMPLISHED BY FRESH START
#      SMODE SET TO +0
#
# DEBRIS
#      ALL EXITS FROM THE CHECK OF ERASABLE (ERASCHK) RESTORE ORIGINAL CONTENTS TO REGISTERS U

```

```
#      EXCEPTION IS A RESTART.  RESTART THAT OCCURS DURING ERASCHK RESTORES ERASABLE
#      E MEMORY, IN WHICH CASE PROGRAM THEN DOES A FRESH START (DOFSTART).
```

```

                                BANK    25
                                SETLOC  SELFCHEC
                                BANK
                                COUNT*  $$/SELF

SBIT1      EQUALS  BIT1
SBIT2      EQUALS  BIT2
SBIT3      EQUALS  BIT3
SBIT4      EQUALS  BIT4
SBIT5      EQUALS  BIT5
SBIT6      EQUALS  BIT6
SBIT7      EQUALS  BIT7
SBIT8      EQUALS  BIT8
SBIT9      EQUALS  BIT9
SBIT10     EQUALS  BIT10
SBIT11     EQUALS  BIT11
SBIT12     EQUALS  BIT12
SBIT13     EQUALS  BIT13
SBIT14     EQUALS  BIT14
SBIT15     EQUALS  BIT15

S+ZERO     EQUALS  ZERO
S+1        EQUALS  BIT1
S+2        EQUALS  BIT2
S+3        EQUALS  THREE
S+4        EQUALS  FOUR
S+5        EQUALS  FIVE
S+6        EQUALS  SIX
# Page 1286
S+7        EQUALS  SEVEN
S8BITS     EQUALS  LOW8          # 00377
CNTRCON    =      OCT50         # USED IN CNTRCHK
ERASCON1   OCTAL  00061         # USED IN ERASCHK
ERASCON2   OCTAL  01373         # USED IN ERASCHK
ERASCON6   =      OCT1400       # USED IN ERASCHK
ERASCON3   OCTAL  01461         # USED IN ERASCHK
ERASCON4   OCTAL  01773         # USED IN ERASCHK
S10BITS    EQUALS  LOW10        # 01777, USED IN ERASCHK
SBNK03     EQUALS  PRI06        # 06000, USED IN ROPECHK
-MAXADRS   =      HI5          # FOR ROPECHK
SIXTY      OCTAL  00060
SUPRCON    OCTAL  60017         # USED IN ROPECHK
S13BITS    OCTAL  17777
```

CONC+S1	OCTAL	25252	# USED IN CYCLSHFT
CONC+S2	OCTAL	52400	# USED IN CYCLSHFT
ERASCON5	OCTAL	76777	
S-7	=	OCT77770	
S-4	EQUALS	NEG4	
S-3	EQUALS	NEG3	
S-2	EQUALS	NEG2	
S-1	EQUALS	NEGONE	
S-ZERO	EQUALS	NEGO	
	EBANK=	LST1	
ADRS1	ADRES	SKEEP1	
SELFADRS	ADRES	SELFCHK	# SELFCHK RETURN ADDRESS. SHOULD BE PUT # IN SELFRET WHEN GOING FROM SELFCHK TO # SHOWSUM AND PUT IN SKEEP1 WHEN GOING # FROM SHOWSUM TO SELF-CHECK.
PRERRORS	CA	ERESTORE	# IS IT NECESSARY TO RESTORE ERASABLE
	EXTEND		
	BZF	ERRORS	# NO
	EXTEND		
	DCA	SKEEP5	
	INDEX	SKEEP7	
	DXCH	0000	# RESTORE THE TWO ERASABLE REGISTERS
	CA	S+ZERO	
	TS	ERESTORE	
ERRORS	INHINT		
	CA	Q	
	TS	SFAIL	# SAVE Q FOR FAILURE LOCATION
	TS	ALMCADR	# FOR DISPLAY WITH BBANK AND ERCOUNT
	INCR	ERCOUNT	# KEEP TRACK OF NUMBER OF MALFUNCTIONS.
TCALARM2	TC	ALARM2	
	OCT	01102	# SELF-CHECK MALFUNCTION INDICATOR
	CCS	SMODE	
SIDLOOP	CA	S+ZERO	
	TS	SMODE	
# Page 1287			
	TC	SELFCHK	# GO TO IDLE LOOP
	TC	SFAIL	# CONTINUE WITH SELF-CHECK
-1CHK	CCS	A	
	TCF	PRERRORS	
	TCF	PRERRORS	
	CCS	A	
	TCF	PRERRORS	
	TC	Q	

```

SMODECHK      EXTEND
                QXCH    SKEEP1
                TC      CHECKNJ      # CHECK FOR NEW JOB
                CCS      SMODE
                TC      SOPTIONS
                TC      SMODECHK +2  # TO BACKUP IDLE LOOP
                TC      SOPTIONS
                INCR     SCOUNT
                TC      SKEEP1      # CONTINUE WITH SELF-CHECK

SOPTIONS      AD      S-7
                EXTEND
                BZMF     +2          # FOR OPTIONS BELOW NINE.
BNKOPTN       TC      SIDLOOP      # ILLEGAL OPTION. GO TO IDLE LOOP.
                INCR     SCOUNT      # FOR OPTIONS BELOW NINE.
                AD      S+7

                INDEX    A
                TC      SOPTION1
SOPTION1      TC      SKEEP1      # WAS TC+TCF
SOPTION2      TC      SKEEP1      # WAS IN:OUT1
SOPTION3      TC      SKEEP1
SOPTION4      TC      ERASCHK
SOPTION5      TC      ROPECHK
SOPTION6      TC      SKEEP1
SOPTION7      TC      SKEEP1
SOPTION10     TC      SKEEP1      # CONTINUE WITH SELF-CHECK

CHECKNJ       EXTEND
                QXCH     SELFRET     # SAVE RETURN ADDRESS WHILE TESTING NEWJOB
                TC      POSTJUMP     # TO SEE IF ANY JOBS HAVE BECOME ACTIVE.
                CADR     ADVAN

SELFCHK       TC      SMODECHK      # ** CHARLEY, COME IN HERE

# SKEEP7 HOLDS LOWEST OF TWO ADDRESSES BEING CHECKED.
# SKEEP6 HOLDS B(X+1).
# SKEEP5 HOLDS B(X).
# SKEEP4 HOLDS C(EBANK) DURING ERASLOOP AND CHECKNJ
# SKEEP3 HOLDS LAST ADDRESS BEING CHECKED (HIGHEST ADDRESS).
# Page 1288
# SKEEP2 CONTROLS CHECKING OF NON-SWITCHABLE ERASABLE MEMORY WITH BANK NUMBERS IN EB
#
# ERASCHK TAKES APPROXIMATELY 7 SECONDS.

```


ERASCHK	CA	S+1	
	TS	SKEEP2	
OEBANK	CA	S+ZERO	
	TS	EBANK	
	CA	ERASCON3	# 01461
	TS	SKEEP7	# STARTING ADDRESS
	CA	S10BITS	# 01777
	TS	SKEEP3	# LAST ADDRESS CHECKED
	TC	ERASLOOP	
E134567B	CA	ERASCON6	# 01400
	TS	SKEEP7	# STARTING ADDRESS
	CA	S10BITS	# 01777
	TS	SKEEP3	# LAST ADDRESS CHECKED
	TC	ERASLOOP	
2EBANK	CA	ERASCON6	# 01400
	TS	SKEEP7	# STARTING ADDRESS
	CA	ERASCON4	# 01773
	TS	SKEEP3	# LAST ADDRESS CHECKED
	TC	ERASLOOP	
NOEBANK	TS	SKEEP2	# +0
	CA	ERASCON1	# 00061
	TS	SKEEP7	# STARTING ADDRESS
	CA	ERASCON2	# 01373
	TS	SKEEP3	# LAST ADDRESS CHECKED
ERASLOOP	INHINT		
	CA	EBANK	# STORES C(EBANK)
	TS	SKEEP4	
	EXTEND		
	NDX	SKEEP7	
	DCA	0000	
	DXCH	SKEEP5	# STORES C(X) AND C(X+1) IN SKEEP6 AND 5.
	CA	SKEEP7	
	TS	ERESTORE	# IF RESTART, RESTORE C(X) AND C(X+1)
	TS	L	
	INCR	L	
	NDX	A	
	DXCH	0000	# PUTS OWN ADDRESS IN X AND X +1
	NDX	SKEEP7	
	CS	0001	# CS X+1
	NDX	SKEEP7	
	AD	0000	# AD X
	TC	-1CHK	

```

# Page 1289
CA      ERESTORE      # HAS ERASABLE BEEN RESTORED
EXTEND

BZF     ELOOPFIN      # YES, EXIT ERASLOOP.
EXTEND
NDX     SKEEP7
DCS     0000           # COMPLEMENT OF ADDRESS OF X AND X+1
NDX     SKEEP7
DXCH    0000           # PUT COMPLEMENT OF ADDRESS OF X AND X+1
NDX     SKEEP7
CS      0000           # CS X
NDX     SKEEP7
AD      0001           # AD X+1
TC      -1CHK
CA      ERESTORE      # HAS ERASABLE BEEN RESTORED
EXTEND
BZF     ELOOPFIN      # YES, EXIT ERASLOOP.
EXTEND
DCA     SKEEP5
NDX     SKEEP7
DXCH    0000           # PUT B(X) AND B(X+1) BACK INTO X AND X+1
CA      S+ZERO
TS      ERESTORE      # IF RESTART, DO NOT RESTORE C(X), C(X+1)
RELINT
TC      CHECKNJ        # CHECK FOR NEW JOB
CA      SKEEP4          # REPLACES B(EBANK)
TS      EBANK
INCR    SKEEP7
CS      SKEEP7
AD      SKEEP3
EXTEND
BZF     +2
TC      ERASLOOP        # GO TO NEXT ADDRESS IN SAME BANK
CCS     SKEEP2
TC      NOEBANK
INCR    SKEEP2          # PUT +1 IN SKEEP2.
CA      EBANK
AD      SBIT9
TS      EBANK
AD      ERASCON5        # 76777, CHECK FOR BANK E2
EXTEND
BZF     2EBANK
CCS     EBANK
TC      E134567B        # GO TO EBANKS 1,3,4,5,6, AND 7
CA      ERASCON6        # END OF ERASCHK
TS      EBANK

```

CNTRCHK PERFORMS A CS OF ALL REGISTERS FROM OCT. 60 THROUGH OCT. 10.
 # INCLUDED ARE ALL COUNTERS, T6-1, CYCLE AND SHIFT, AND ALL RUPT REGISTERS

CNTRCHK	CA	CNTRCON	# 00050
CNTRLOOP	TS	SKEEP2	
	AD	SBIT4	# +10 OCTAL
	INDEX	A	
	CS	0000	

Page 1290

CCS	SKEEP2
TC	CNTRLOOP

CYCLSHFT CHECKS THE CYCLE AND SHIFT REGISTERS

CYCLSHFT	CA	CONC+S1	# 25252	
	TS	CYR	# C(CYR) = 12525	
	TS	CYL	# C(CYL) = 52524	
	TS	SR	# C(SR) = 12525	
	TS	EDOP	# C(EDOP) = 00125	
	AD	CYR	# 37777	C(CYR) = 45252
	AD	CYL	# 00-12524	C(CYL) = 25251
	AD	SR	# 00-25251	C(SR) = 05252
	AD	EDOP	# 00-25376	C(EDOP) = +0
	AD	CONC+S2	# C(CONC+S2) = 52400	
	TC	-1CHK		
	AD	CYR	# 45252	
	AD	CYL	# 72523	
	AD	SR	# 77775	
	AD	EDOP	# 77775	
	AD	S+1	# 77776	
	TC	-1CHK		
	INCR	SCOUNT +1		
	TC	SMODECHK		

SKEEP1 HOLDS SUM
 # SKEEP2 HOLDS PRESENT CONTENTS OF ADDRESS IN ROPECHK AND SHOWSUM ROUTINES
 # SKEEP2 HOLDS BANK NUMBER IN LOW ORDER BITS DURING SHOWSUM DISPLAY
 # SKEEP3 HOLDS PRESENT ADDRESS (00000 TO 01777 IN COMMON FIXED BANKS)
 # (04000 TO 07777 IN FXFX BANKS)
 # SKEEP3 HOLDS BUGGER WORD DURING SHOWSUM DISPLAY
 # SKEEP4 HOLDS BANK NUMBER AND SUPER BANK NUMBER
 # SKEEP5 COUNTS 2 SUCCESSIVE TC SELF WORDS
 # SKEEP6 CONTROLS ROPECHK OR SHOWSUM OPTION
 # SKEEP7 CONTROLS WHEN ROUTINE IS IN COMMON FIXED OR FIXED FIXED BANKS

ROPECHK	CA	S-ZERO	# *
	TS	SKEEP6	# * -0 FOR ROPECHK
STSHOSUM	CA	S+ZERO	# * SHOULD BE ROPECHK
	TS	SKEEP4	# BANK NUMBER
	CA	S+1	
COMAFX	TS	SKEEP7	
	CA	S+ZERO	
	TS	SKEEP1	
	TS	SKEEP3	
	CA	S+1	
	TS	SKEEP5	# COUNTS DOWN 2 TC SELF WORDS
COMADRS	CA	SKEEP4	
	TS	L	# TO SET SUPER BANK
	MASK	HI5	
# Page 1291			
	AD	SKEEP3	
	TC	SUPDACAL	# SUPER DATA CALL
	TC	ADSUM	
	AD	SBIT11	# 02000
	TC	ADRSCHK	
FXFX	CS	A	
	TS	SKEEP7	
	EXTEND		
	BZF	+3	
	CA	SBIT12	# 04000, STARTING ADDRESS OF BANK 02
	TC	+2	
	CA	SBNK03	# 06000, STARTING ADDRESS OF BANK 03
	TS	SKEEP3	
	CA	S+ZERO	
	TS	SKEEP1	
	CA	S+1	
	TS	SKEEP5	# COUNTS DOWN 2 TC SELF WORDS
FXADRS	INDEX	SKEEP3	
	CA	0000	
	TC	ADSUM	
	TC	ADRSCHK	
ADSUM	TS	SKEEP2	
	AD	SKEEP1	
	TS	SKEEP1	
	CAF	S+ZERO	
	AD	SKEEP1	
	TS	SKEEP1	

	CS	SKEEP2	
	AD	SKEEP3	
	TC	Q	
ADRSCHK	LXCH	A	
	CA	SKEEP3	
	MASK	LOW10	# RELATIVE ADDRESS
	AD	-MAXADRS	# SUBTRACT MAX RELATIVE ADDRESS = 1777.
	EXTEND		
	BZF	SOPTION	# CHECKSUM FINISHED IF LAST ADDRESS.
	CCS	SKEEP5	# IS CHECKSUM FINISHED
	TC	+3	# NO
	TC	+2	# NO
	TC	SOPTION	# GO TO ROPECHK SHOWSUM OPTION
	CCS	L	# -0 MEANS A TC SELF WORD.
	TC	CONTINU	
	TC	CONTINU	
	TC	CONTINU	
	CCS	SKEEP5	
	TC	CONTINU +1	
	CA	S-1	
# Page 1292			
	TC	CONTINU +1	# AD IN THE BUGGER WORD
CONTINU	CA	S+1	# MAKE SURE TWO CONSECUTIVE TC SELF WORDS
	TS	SKEEP5	
	CCS	SKEEP6	# *
	CCS	NEWJOB	# * +1, SHOWSUM
	TC	CHANG1	# *
	TC	+2	# *
	TC	CHECKNJ	# -0 IN SKEEP6 FOR ROPECHK
ADRS+1	INCR	SKEEP3	
	CCS	SKEEP7	
	TC	COMADRS	
	TC	COMADRS	
	TC	FXADRS	
	TC	FXADRS	
NXTBNK	CS	SKEEP4	
	AD	LSTBNKCH	# LAST BANK TO BE CHECKED
	EXTEND		
	BZF	ENDSUMS	# END OF SUMMING OF BANKS.
	CA	SKEEP4	
	AD	SBIT11	
	TS	SKEEP4	# 37 TO 40 INCRMTS SKEEP4 BY END RND CARRY
	TC	CHKSUPR	

17T020	CA	SBIT15	
	ADS	SKEEP4	# SET FOR BANK 20
	TC	GONXTBNK	
CHKSUPR	MASK	HI5	
	EXTEND		
	BZF	NXTSUPR	# INCREMENT SUPER BANK
27T030	AD	S13BITS	
	EXTEND		
	BZF	+2	# BANK SET FOR 30
	TC	GONXTBNK	
	CA	SIXTY	# FIRST SUPER BANK
	ADS	SKEEP4	
	TC	GONXTBNK	
NXTSUPR	AD	SUPRCON	# SET BNK 30 + INCR SUPR BNK AND CANCEL
	ADS	SKEEP4	# ERC BIT OF TEH 37 TO 40 ADVANCE.
GONXTBNK	CCS	SKEEP7	
	TC	COMMFY	
	CA	S+1	
	TC	FXFX	
	CA	SBIT7	# HAS TO BE LARGER THAN NO OF FXSW BANKS.
	TC	COMMFY	
SOPTION	CA	SKEEP4	
	MASK	HI5	# = BANK BITS
	TC	LEFT5	
	TS	L	# BANK NUMBER BEFORE SUPER BANK
# Page 1293			
	CA	SKEEP4	
	MASK	S8BITS	# = SUPER BANK BITS
	EXTEND		
	BZF	SOPT	# BEFORE SUPER BANK
	TS	SR	# SUPER BANK NECESSARY
	CA	L	
	MASK	SEVEN	
	AD	SR	
	TS	L	# BANK NUMBER WITH SUPER BANK
SOPT	CA	SKEEP6	# *
	EXTEND		# *
	BZF	+2	# * ON -0 CONTINUE WITH ROPE CHECK.
	TC	SDISPLAY	# * ON +1 GO TO DISPLAY OF SUM.
	CCS	SKEEP1	# FORCE SUM TO ABSOLUTE VALUE.
	TC	+2	
	TC	+2	
	AD	S+1	
	TS	SKEEP1	
BNKCHK	CS	L	# = - BANK NUMBER

July 12, 2016

79

```
AD      SKEEP1
AD      S-1
TC      -1CHK      # CHECK SUM
TC      NXTBNK

EBANK=   NEWJOB
LSTBNKCH BBCON*      # * CONSTANT, LAST BANK.
```

This code is written to file `src/AGC-BLOCK-TWO-SELF--CHECK.s`.

B.4 AGC BLOCK TWO SELF-CHECK

```

80  <src/AGC-BLOCK-TWO-SELF-CHECK.s 80>≡
    # Copyright:      Public domain.
    # Filename:       AGC_BLOCK_TWO_SELF-CHECK.agc
    # Purpose:        Part of the source code for Colossus 2A, AKA Comanche 055.
    #                  It is part of the source code for the Command Module's (CM)
    #                  Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:     yaYUL
    # Contact:        Ron Burkey <info@sandroid.org>.
    # Website:        www.ibiblio.org/apollo.
    # Pages:          1394-1403
    # Mod history:    2009-05-10 SN      (Sergio Navarro).  Started adapting
    #                  from the Colossus249/ file of the same
    #                  name, using Comanche055 page images.
    #
    # This source code has been transcribed or otherwise adapted from digitized
    # images of a hardcopy from the MIT Museum.  The digitization was performed
    # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum.  Many
    # thanks to both.  The images (with suitable reduction in storage size and
    # consequent reduction in image quality as well) are available online at
    # www.ibiblio.org/apollo.  If for some reason you find that the images are
    # illegible, contact me at info@sandroid.org about getting access to the
    # (much) higher-quality images which Paul actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    #   Assemble revision 055 of AGC program Comanche by NASA
    #   2021113-051.  10:28 APR. 1, 1969
    #
    #   This AGC program shall also be referred to as
    #   Colossus 2A

    # Page 1394
    # PROGRAM DESCRIPTION                                DATE:  20 DECEMBER 1967
    # PROGRAM NAME -- SELF-CHECK                        LOG SECTION:  AGC BLOCK TWO SELF-CHECK
    # MOD NO -- 1                                         ASSEMBLY SUBROUTINE UTILITM REV 25
    # MOD BY -- GAUNTT
    #
    # FUNCTIONAL DESCRIPTION
    #   PROGRAM HAS TWO MAIN PARTS.  THE FIRST IS SELF-CHECK WHICH RUNS AS A ZERO PR
    #   PART OF THE BACK-UP IDLE LOOP.  THE SECOND IS SHOW-BANKSUM WHICH RUNS AS A R
    #   STARTING VERB.
    #
    #   THE PURPOSE OF SELF-CHECK IS TO CHECK OUT VARIOUS PARTS OF THE COMPUTER AS O
    #

```



```

#       THE PURPOSE OF SHOW-BANKSUM IS TO DISPLAY THE SUM OF EACH BANK, ONE AT A TIME.
#
#       IN ALL THERE ARE 7 POSSIBLE OPTIONS IN THIS BLOCK II VERSION OF SELF-CHECK.  MORE DETAILS
#       FOUND IN E-2065 BLOCK II AGC SELF-CHECK AND SHOW BANKSUM BY EDWIN D. SMALLY DECEMBER 1964.
#
#       THE DIFFERENT OPTIONS ARE CONTROLLED BY PUTTING DIFFERENT NUMBERS IN THE SMODE REGISTER.
#       A DESCRIPTION OF WHAT PARTS OF THE COMPUTER THAT ARE CHECKED BY THE OPTIONS, AND THE CODE
#       OCTAL, TO LOAD INTO SMODE.
#           +-4             ERASABLE MEMORY
#           +-5             FIXED MEMORY
#           +-1,2,3,6,7,10  EVERYTHING IN OPTIONS 4 AND 5.
#           -0              SAME AS +-10 UNTIL AN ERROR IS DETECTED.
#           +0              NO CHECK, PUTS COMPUTER INTO THE BACKUP IDLE LOOP.
#
# WARNINGS
#       USE OF E MEMORY RESERVED FOR SELF-CHECK (EVEN IN IDLE LOOP) AS TEMP STORAGE BY OTHER PROGRAMS.
#       SMODE SET GREATER THAN OCT 10 PUTS COMPUTER INTO BACKUP IDLE LOOP.
#
# CALLING SEQUENCE
#       TO CALL SELF-CHECK KEY IN
#           V 21 N 27 E      OPTION NUMBER E
#       TO CALL SHOW-BANKSUM KEY IN
#           V 91 E           DISPLAYS FIRST BANK
#           V 33 E           PROCEED, DISPLAYS NEXT BANK
#
# EXIT MODES, NORMAL AND ALARM
#       SELF-CHECK NORMALLY CONTINUES INDEFINITELY UNLESS THERE IS AN ERROR DETECTED.  IF SO +
#       COMPUTER INTO BACKUP IDLE LOOP, - OPTION NUMBERS RESTART THE OPTION.
#
#       THE -0 OPTION PROCEEDS FROM THE LINE FOLLOWING THE LINE WHERE THE ERROR WAS DETECTED.
#
#       SHOW-BANKSUM PROCEEDS UNTIL A TERMINATE IS KEYED IN (V 34 E).  THE COMPUTER IS PUT INTO
#
# OUTPUT
# Page 1395
#       SELF-CHECK UPON DETECTING AN ERROR LOADS THE SELF-CHECK ALARM CONSTANT (01102) INTO THE
#       TURNS ON THE ALARM LIGHT.  THE OPERATOR MAY THEN DISPLAY THE THREE FAILREGS BY KEYING IN
#       INFORMATION HE MAY KEY IN V 05 N 08 E, THE DSKY DISPLAY IN R1 WILL BE ADDRESS+1 OF WHERE
#       IN R2 THE BBCON OF SELF-CHECK, AND IN R3 THE TOTAL NUMBER OF ERRORS DETECTED BY SELF-CHECK.
#       INITIATED FRESH START (SLAP1).
#
#       SHOW-BANKSUM STARTING WITH BANK 0 DISPLAYS IN R1 THE BANK SUM (A +-NUMBER EQUAL TO THE
#       THE BANK NUMBER, AND IN R3 THE BUGGER WORD.
#
# ERASABLE INITIALIZATION REQUIRED
#       ACCOMPLISHED BY FRESH START

```

```

#           SMODE SET TO +0
#
# DEBRIS
#   ALL EXITS FROM THE CHECK OF ERASABLE (ERASCHK) RESTORE ORIGINAL CONTENTS TO E
#   EXCEPTION IS A RESTART.  RESTART THAT OCCURS DURING ERASCHK RESTORES ERASABLE
#   E MEMORY, IN WHICH CASE PROGRAM THEN DOES A FRESH START (DOFSTART).

          BANK      25
          SETLOC    SELFCHEC
          BANK

          COUNT     43/SELF

SBIT1      EQUALS  BIT1
SBIT2      EQUALS  BIT2
SBIT3      EQUALS  BIT3
SBIT4      EQUALS  BIT4
SBIT5      EQUALS  BIT5
SBIT6      EQUALS  BIT6
SBIT7      EQUALS  BIT7
SBIT8      EQUALS  BIT8
SBIT9      EQUALS  BIT9
SBIT10     EQUALS  BIT10
SBIT11     EQUALS  BIT11
SBIT12     EQUALS  BIT12
SBIT13     EQUALS  BIT13
SBIT14     EQUALS  BIT14
SBIT15     EQUALS  BIT15

S+ZERO     EQUALS  ZERO
S+1         EQUALS  BIT1
S+2         EQUALS  BIT2
S+3         EQUALS  THREE
S+4         EQUALS  FOUR
S+5         EQUALS  FIVE
# Page 1396
S+6         EQUALS  SIX
S+7         EQUALS  SEVEN
S8BITS      EQUALS  LOW8           # 00377
CNTRCON     =       OCT50         # USED IN CNTRCHK
ERASCON1    OCTAL   00061         # USED IN ERASCHK
ERASCON2    OCTAL   01373         # USED IN ERASCHK
ERASCON6    =       OCT1400       # USED IN ERASCHK
ERASCON3    OCTAL   01461         # USED IN ERASCHK
ERASCON4    OCTAL   01773         # USED IN ERASCHK
S10BITS     EQUALS  LOW10         # 01777, USED IN ERASCHK

```

SBNK03	EQUALS	PRI06	# 06000, USED IN ROPECHK
-MAXADRS	=	HI5	# FOR ROPECHK
SIXTY	OCTAL	00060	
SUPRCON	OCTAL	60017	# USED IN ROPECHK
S13BITS	OCTAL	17777	
CONC+S1	OCTAL	25252	# USED IN CYCLSHFT
CONC+S2	OCTAL	52400	# USED IN CYCLSHFT
ERASCON5	OCTAL	76777	
S-7	=	OCT77770	
S-4	EQUALS	NEG4	
S-3	EQUALS	NEG3	
S-2	EQUALS	NEG2	
S-1	EQUALS	NEGONE	
S-ZERO	EQUALS	NEGO	
	EBANK=	LST1	
ADRS1	ADRES	SKEEP1	
SELFADRS	ADRES	SELFCHK	# SELFCHK RETURN ADDRESS. SHOULD BE PUT # IN SELFRET WHEN GOING FROM SELFCHK TO # SHOWSUM AND PUT IN SKEEP1 WHEN GOING # FROM SHOWSUM TO SELF-CHECK.
PRERRORS	CA	ERESTORE	# IS IT NECESSARY TO RESTORE ERASABLE
	EXTEND		
	BZF	ERRORS	# NO
	EXTEND		
	DCA	SKEEP5	
	INDEX	SKEEP7	
	DXCH	0000	# RESTORE THE TWO ERASABLE REGISTERS
	CA	S+ZERO	
	TS	ERESTORE	
ERRORS	INHINT		
	CA	Q	
	TS	SFAIL	# SAVE Q FOR FAILURE LOCATION
	TS	ALMCADR	# FOR DISPLAY WITH BBANK AND ERCOUNT
	INCR	ERCOUNT	# KEEP TRACK OF NUMBER OF MALFUNCTIONS.
TCALARM2	TC	ALARM2	
	OCT	01102	# SELF-CHECK MALFUNCTION INDICATOR
	CCS	SMODE	
SIDLOOP	CA	S+ZERO	
# Page 1397			
	TS	SMODE	
	TC	SELFCHK	# GO TO IDLE LOOP
	TC	SFAIL	# CONTINUE WITH SELF-CHECK
-1CHK	CCS	A	

```

TCF      PRERRORS
TCF      PRERRORS
CCS      A
TCF      PRERRORS
TC       Q

SMODECHK  EXTEND
QXCH     SKEEP1
TC       CHECKNJ      # CHECK FOR NEW JOB
CCS      SMODE
TC       OPTIONS
TC       SMODECHK +2  # TO BACKUP IDLE LOOP
TC       OPTIONS
INCR     SCOUNT
TC       SKEEP1      # CONTINUE WITH SELF-CHECK

OPTIONS  AD       S-7
EXTEND
BZMF     +2          # FOR OPTIONS BELOW NINE.
BNKOPTN  TC       SIDLOOP      # ILLEGAL OPTION. GO TO IDLE LOOP.
INCR     SCOUNT      # FOR OPTIONS BELOW NINE.
AD       S+7

INDEX    A
TC       SOPTION1
SOPTION1 TC       SKEEP1      # WAS TC+TCF
SOPTION2 TC       SKEEP1      # WAS IN:OUT1
SOPTION3 TC       SKEEP1      # WAS COUNTCHK
SOPTION4 TC       ERASCHK
SOPTION5 TC       ROPECHK
SOPTION6 TC       SKEEP1
SOPTION7 TC       SKEEP1
SOPTON10 TC       SKEEP1      # CONTINUE WITH SELF-CHECK

CHECKNJ  EXTEND
QXCH     SELFRET      # SAVE RETURN ADDRESS WHILE TESTING NEWJOB
TC       POSTJUMP     # TO SEE IF ANY JOBS HAVE BECOME ACTIVE.
CADR     ADVAN

SELFCHK  TC       SMODECHK      # ** CHARLEY, COME IN HERE

# SKEEP7 HOLDS LOWEST OF TWO ADDRESSES BEING CHECKED.
# SKEEP6 HOLDS B(X+1).
# SKEEP5 HOLDS B(X).
# SKEEP4 HOLDS C(EBANK) DURING ERASLOOP AND CHECKNJ
# Page 1398

```

```
# SKEEP3 HOLDS LAST ADDRESS BEING CHECKED (HIGHEST ADDRESS).
# SKEEP2 CONTROLS CHECKING OF NON-SWITCHABLE ERASABLE MEMORY WITH BANK NUMBERS IN EB.
#
# ERASCHK TAKES APPROXIMATELY 7 SECONDS.
```

ERASCHK	CA	S+1	
	TS	SKEEP2	
OE BANK	CA	S+ZERO	
	TS	EBANK	
	CA	ERASCON3	# 01461
	TS	SKEEP7	# STARTING ADDRESS
	CA	S10BITS	# 01777
	TS	SKEEP3	# LAST ADDRESS CHECKED
	TC	ERASLOOP	
E134567B	CA	ERASCON6	# 01400
	TS	SKEEP7	# STARTING ADDRESS
	CA	S10BITS	# 01777
	TS	SKEEP3	# LAST ADDRESS CHECKED
	TC	ERASLOOP	
2EBANK	CA	ERASCON6	# 01400
	TS	SKEEP7	# STARTING ADDRESS
	CA	ERASCON4	# 01773
	TS	SKEEP3	# LAST ADDRESS CHECKED
	TC	ERASLOOP	
NOEBANK	TS	SKEEP2	# +0
	CA	ERASCON1	# 00061
	TS	SKEEP7	# STARTING ADDRESS
	CA	ERASCON2	# 01373
	TS	SKEEP3	# LAST ADDRESS CHECKED
ERASLOOP	INHINT		
	CA	EBANK	# STORES C(EBANK)
	TS	SKEEP4	
	EXTEND		
	NDX	SKEEP7	
	DCA	0000	
	DXCH	SKEEP5	# STORES C(X) AND C(X+1) IN SKEEP6 AND 5.
	CA	SKEEP7	
	TS	ERESTORE	# IF RESTART, RESTORE C(X) AND C(X+1)
	TS	L	
	INCR	L	
	NDX	A	
	DXCH	0000	# PUTS OWN ADDRESS IN X AND X +1

```

# Page 1399
NDX      SKEEP7
CS        0001      # CS X+1
NDX      SKEEP7
AD        0000      # AD X
TC        -1CHK
CA        ERESTORE  # HAS ERASABLE BEEN RESTORED

EXTEND
BZF      ELOOPFIN  # YES, EXIT ERASLOOP.
EXTEND
NDX      SKEEP7
DCS       0000      # COMPLEMENT OF ADDRESS OF X AND X+1
NDX      SKEEP7
DXCH      0000      # PUT COMPLEMENT OF ADDRESS OF X AND X+1
NDX      SKEEP7
CS        0000      # CS X
NDX      SKEEP7
AD        0001      # AD X+1
TC        -1CHK
CA        ERESTORE  # HAS ERASABLE BEEN RESTORED

EXTEND
BZF      ELOOPFIN  # YES, EXIT ERASLOOP.
EXTEND
DCA       SKEEP5
NDX      SKEEP7
DXCH      0000      # PUT B(X) AND B(X+1) BACK INTO X AND X+1
CA        S+ZERO
TS        ERESTORE  # IF RESTART, DO NOT RESTORE C(X), C(X+1)

ELOOPFIN RELINT
TC        CHECKNJ  # CHECK FOR NEW JOB
CA        SKEEP4   # REPLACES B(EBANK)
TS        EBANK
INCR      SKEEP7
CS        SKEEP7
AD        SKEEP3
EXTEND
BZF      +2
TC        ERASLOOP  # GO TO NEXT ADDRESS IN SAME BANK
CCS       SKEEP2
TC        NOEBANK
INCR      SKEEP2   # PUT +1 IN SKEEP2.
CA        EBANK
AD        SBIT9
TS        EBANK
AD        ERASCON5 # 76777, CHECK FOR BANK E2
EXTEND

```

BZF	2EBANK	
CCS	EBANK	
TC	E134567B	# GO TO EBANKS 1,3,4,5,6, AND 7
CA	ERASCON6	# END OF ERASCHK
TS	EBANK	

CNTRCHK PERFORMS A CS OF ALL REGISTERS FROM OCT. 60 THROUGH OCT. 10.
 # INCLUDED ARE ALL COUNTERS, T6-1, CYCLE AND SHIFT, AND ALL RUPT REGISTERS

CNTRCHK	CA	CNTRCON	# 00050
CNTRLOOP	TS	SKEEP2	
	AD	SBIT4	# +10 OCTAL
	INDEX	A	

Page 1400

CS	0000
CCS	SKEEP2
TC	CNTRLOOP

CYCLSHFT CHECKS THE CYCLE AND SHIFT REGISTERS

CYCLSHFT	CA	CONC+S1	# 25252
	TS	CYR	# C(CYR) = 12525
	TS	CYL	# C(CYL) = 52524
	TS	SR	# C(SR) = 12525
	TS	EDOP	# C(EDOP) = 00125
	AD	CYR	# 37777 C(CYR) = 45252
	AD	CYL	# 00-12524 C(CYL) = 25251
	AD	SR	# 00-25251 C(SR) = 05252
	AD	EDOP	# 00-25376 C(EDOP) = +0
	AD	CONC+S2	# C(CONC+S2) = 52400
	TC	-1CHK	
	AD	CYR	# 45252
	AD	CYL	# 72523
	AD	SR	# 77775
	AD	EDOP	# 77775
	AD	S+1	# 77776
	TC	-1CHK	
	INCR	SCOUNT +1	
	TC	SMODECHK	

SKEEP1 HOLDS SUM
 # SKEEP2 HOLDS PRESENT CONTENTS OF ADDRESS IN ROPECHK AND SHOWSUM ROUTINES
 # SKEEP2 HOLDS BANK NUMBER IN LOW ORDER BITS DURING SHOWSUM DISPLAY
 # SKEEP3 HOLDS PRESENT ADDRESS (00000 TO 01777 IN COMMON FIXED BANKS)
 # (04000 TO 07777 IN FXFX BANKS)

SKEEP3 HOLDS BUGGER WORD DURING SHOWSUM DISPLAY
 # SKEEP4 HOLDS BANK NUMBER AND SUPER BANK NUMBER
 # SKEEP5 COUNTS 2 SUCCESSIVE TC SELF WORDS
 # SKEEP6 CONTROLS ROPECHK OR SHOWSUM OPTION
 # SKEEP7 CONTROLS WHEN ROUTINE IS IN COMMON FIXED OR FIXED FIXED BANKS

ROPECHK	CA	S-ZERO	# *
	TS	SKEEP6	# * -0 FOR ROPECHK
STSHOSUM	CA	S+ZERO	# * SHOULD BE ROPECHK
	TS	SKEEP4	# BANK NUMBER
	CA	S+1	
COMAFX	TS	SKEEP7	
	CA	S+ZERO	
	TS	SKEEP1	
	TS	SKEEP3	
	CA	S+1	
	TS	SKEEP5	# COUNTS DOWN 2 TC SELF WORDS
COMADRS	CA	SKEEP4	
	TS	L	# TO SET SUPER BANK
# Page 1401			
	MASK	HI5	
	AD	SKEEP3	
	TC	SUPDACAL	# SUPER DATA CALL
	TC	ADSUM	
	AD	SBIT11	# 02000
	TC	ADRSCHK	
FXFX	CS	A	
	TS	SKEEP7	
	EXTEND		
	BZF	+3	
	CA	SBIT12	# 04000, STARTING ADDRESS OF BANK 02
	TC	+2	
	CA	SBNK03	# 06000, STARTING ADDRESS OF BANK 03
	TS	SKEEP3	
	CA	S+ZERO	
	TS	SKEEP1	
	CA	S+1	
	TS	SKEEP5	# COUNTS DOWN 2 TC SELF WORDS
FXADRS	INDEX	SKEEP3	
	CA	0000	
	TC	ADSUM	
	TC	ADRSCHK	
ADSUM	TS	SKEEP2	

	AD	SKEEP1	
	TS	SKEEP1	
	CAF	S+ZERO	
	AD	SKEEP1	
	TS	SKEEP1	
	CS	SKEEP2	
	AD	SKEEP3	
	TC	Q	
ADRSCHK	LXCH	A	
	CA	SKEEP3	
	MASK	LOW10	# RELATIVE ADDRESS
	AD	-MAXADRS	# SUBTRACT MAX RELATIVE ADDRESS = 1777.
	EXTEND		
	BZF	SOPTION	# CHECKSUM FINISHED IF LAST ADDRESS.
	CCS	SKEEP5	# IS CHECKSUM FINISHED
	TC	+3	# NO
	TC	+2	# NO
	TC	SOPTION	# GO TO ROPECHK SHOWSUM OPTION
	CCS	L	# -0 MEANS A TC SELF WORD.
	TC	CONTINU	
	TC	CONTINU	
	TC	CONTINU	
	CCS	SKEEP5	
	TC	CONTINU +1	
# Page 1402			
	CA	S-1	
	TC	CONTINU +1	# AD IN THE BUGGER WORD
CONTINU	CA	S+1	# MAKE SURE TWO CONSECUTIVE TC SELF WORDS
	TS	SKEEP5	
	CCS	SKEEP6	# *
	CCS	NEWJOB	# * +1, SHOWSUM
	TC	CHANG1	# *
	TC	+2	# *
	TC	CHECKNJ	# -0 IN SKEEP6 FOR ROPECHK
ADRS+1	INCR	SKEEP3	
	CCS	SKEEP7	
	TC	COMADRS	
	TC	COMADRS	
	TC	FXADRS	
	TC	FXADRS	
NXTBNK	CS	SKEEP4	
	AD	LSTBNKCH	# LAST BANK TO BE CHECKED
	EXTEND		

	BZF	ENDSUMS	# END OF SUMMING OF BANKS.
	CA	SKEEP4	
	AD	SBIT11	
	TS	SKEEP4	# 37 TO 40 INCRMTS SKEEP4 BY END RND CARRY
	TC	CHKSUPR	
17T020	CA	SBIT15	
	ADS	SKEEP4	# SET FOR BANK 20
	TC	GONXTBNK	
CHKSUPR	MASK	HI5	
	EXTEND		
	BZF	NXTSUPR	# INCREMENT SUPER BANK
27T030	AD	S13BITS	
	EXTEND		
	BZF	+2	# BANK SET FOR 30
	TC	GONXTBNK	
	CA	SIXTY	# FIRST SUPER BANK
	ADS	SKEEP4	
	TC	GONXTBNK	
NXTSUPR	AD	SUPRCON	# SET BNK 30 + INCR SUPR BNK AND CANCEL
	ADS	SKEEP4	# ERC BIT OF THE 37 TO 40 ADVANCE.
GONXTBNK	CCS	SKEEP7	
	TC	COMMFY	
	CA	S+1	
	TC	FXFX	
	CA	SBIT7	# HAS TO BE LARGER THAN NO OF FXSW BANKS.
	TC	COMMFY	
SOPTION	CA	SKEEP4	
	MASK	HI5	# = BANK BITS
	TC	LEFT5	
# Page 1403			
	TS	L	# BANK NUMBER BEFORE SUPER BANK
	CA	SKEEP4	
	MASK	S8BITS	# = SUPER BANK BITS
	EXTEND		
	BZF	SOPT	# BEFORE SUPER BANK
	TS	SR	# SUPER BANK NECESSARY
	CA	L	
	MASK	SEVEN	
	AD	SR	
	TS	L	# BANK NUMBER WITH SUPER BANK
SOPT	CA	SKEEP6	# *
	EXTEND		# *
	BZF	+2	# * ON -0 CONTINUE WITH ROPE CHECK.
	TC	SDISPLAY	# * ON +1 GO TO DISPLAY OF SUM.
	CCS	SKEEP1	# FORCE SUM TO ABSOLUTE VALUE.

	TC	+2	
	TC	+2	
	AD	S+1	
	TS	SKEEP1	
BNKCHK	CS	L	# = - BANK NUMBER
	AD	SKEEP1	
	AD	S-1	
	TC	-1CHK	# CHECK SUM
	TC	NXTBNK	
	EBANK=	NEWJOB	
LSTBNKCH	BBCON*		# * CONSTANT, LAST BANK.
	SBANK=	LOWSUPER	

This code is written to file `src/AGC-BLOCK-TWO-SELF-CHECK.s`.

B.5 AGS INITIALIZATION

```

92  <src/AGS-INITIALIZATION.s 92>≡
    # Copyright:    Public domain.
    # Filename:     AGS_INITIALIZATION.agc
    # Purpose:      Part of the source code for Luminary 1A build 099.
    #               It is part of the source code for the Lunar Module's (LM)
    #               Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:    yaYUL
    # Contact:       Hartmuth Gutsche <hgutsche@explornet.com>.
    # Website:       www.ibiblio.org/apollo.
    # Pages:         206-210
    # Mod history:   2009-05-19 HG    Transcribed from page images.
    #
    # This source code has been transcribed or otherwise adapted from
    # digitized images of a hardcopy from the MIT Museum. The digitization
    # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
    # the Museum. Many thanks to both. The images (with suitable reduction
    # in storage size and consequent reduction in image quality as well) are
    # available online at www.ibiblio.org/apollo. If for some reason you
    # find that the images are illegible, contact me at info@sandroid.org
    # about getting access to the (much) higher-quality images which Paul
    # actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
    #       16:27 JULY 14, 1969
    #
    # Page 206
    #
    # PROGRAM NAME:  AGS INITIALIZATION (R47)
    #
    # WRITTEN BY:    RHODE/KILROY/FOLLETT
    #
    # MOD NO.:       0
    # DATE:           23 MARCH 1967
    # MOD BY:         KILROY
    #
    # MOD NO.:       1
    # DATE:           28 OCTOBER 1967
    # MOD BY:         FOLLETT
    #
    # FUNCT. DESC.:  (1) TO PROVIDE THE AGS ABORT ELECTRONICS ASSEMBLY (AEA) WITH THE LEM
    #                (POSITION,VELOCITY,TIME) IN LEM IMU COORDINATES BY MEANS OF THE LGC I
    #

```

```

#          (2) TO ZERO THE ICDU, LGC, AND AEA GIMBAL ANGLE COUNTER SIMULTANEOUSLY IN ORDER
#          COMMON ZERO REFERENCE FOR THE MEASUREMENT OF GIMBAL (EULER) ANGLES WHICH DEFINE
#
#          (3) TO ESTABLISH THE GROUND ELAPSED TIME OF AEA CLOCK ZERO.  (IF AN AEA CLOCK Z
#          REQUESTED DURING THIS PROGRAM
#
# LOG SECTION:  AGS INITIALIZATION
#
# CALLING SEQ:  PROGRAM IS ENTERED WHEN ASTRONAUT KEYS V47E ON DSKY.
#              R47 MAY BE CALLED AT ANY TIME EXCEPT WHEN ANOTHER EXTENDED VERB IS IN PROGRESS
#
# SUBROUTINES
# CALLED:
#
# NORMAL EXIT:  ENDEXT
#
# ALARM/ABORT:  ALARM -- BAD REFSMMAT -- CODE:220
#              OPERATOR ERROR IF V47 SELECTED DURING ANOTHER EXTENDED VERB.
#
# ERASABLES
# USED:        SAMPTIME      (2)      TIME OF :ENTER: KEYSTROKE
#              AGSK          (2)      GROUND ELAPSED TIME OF THE AEA CLOCK :ZERO:
#              AGSBUFF       (140)    CONTAINS AGS INITIALIZATION DATA (SEE :OUTPUT: BELOW)
#              AGSWORD       (1)      PREVIOUS DOWNLIST SAVED HERE
#
#              EBANK=  AGSBUFF
#
#              BANK      40
#              SETLOC    R47
#              BANK
#
#              COUNT*   $$/R47
#
# AGSINIT      CAF      REFSMBIT
#              MASK     FLAGWRD3      # CHECK REFSMFLG.
#              CCS      A
#
# # Page 207
#              TC       REDSPTEM      # REFSMMAT IS OK
#              TC       ALARM         # REFSMMAT IS BAD
#              OCT      220
#              TC       ENDEXT
#
# NEWAGS       EXTEND
#              DCA      SAMPTIME      # TIME OF THE :ENTER: KEYSTROKE
#              DXCH     AGSK          # BECOMES NEW AEA CLOCK :ZERO:

```

```

REDSPTM      EXTEND
              DCA      AGSK
              DXCH     DSPTEMX
AGSDISPK      CAF      V06N16
              TC       BANKCALL
              CADR     GOMARKF
              TC       ENDEXT
              TC       AGSVCALC
              CS       BIT6
              AD       MPAC
              EXTEND
              BZF      NEWAGS
                      # YES, USE KEYSTROKE TIME FOR
                      # NO, NEW AGSK LOADED VIA V25E
                      # LOADED INTO DSPTEMX BY KEYSTROKE
                      # V25E FOLLOWED BY HRS.,MINS
                      # DISPLAY THE NEW K.

AGSVCALC      TC       INTPRET
              SET
                      NODOFLAG
                      # DON'T ALLOW V37
              SET      EXIT
                      XDSPFLAG

              CAF      V06N16
              TC       BANKCALL
              CADR     EXDSPRET

              TC       INTPRET
              RTB
                      LOADTIME
              STCALL   TDEC1
                      LEMPREC
              CALL
                      SCALEVEC
              STODL    AGSBUFF
                      TAT
              STCALL   TDEC1
                      CSMPREC
              CALL
                      SCALEVEC
              STODL    AGSBUFF +6
                      TAT
              DSU      DDV
                      AGSK
                      # CALCULATE AND STORE THE TIME

```

	STORE	TSCALE	
	EXIT	AGSBUFF +12D	
	CAF	LAGSLIST	
	TS	DNLSTCOD	
	CAF	20SEC	# DELAY FOR 20 SEC WHILE THE AGS
	TC	BANKCALL	# DOWNLIST IS TRANSMITTED
	CADR	DELAYJOB	
	CA	AGSWORD	
	TS	DNLSTCOD	# RETURN TO THE OLD DOWNLIST
	CAF	IMUSEBIT	
	MASK	FLAGWRDO	# CHECK IMUSE FLAG.
	CCS	A	
	TC	AGSEND	# IMU IS BEING USED -- DO NOT ZERO
CKSTALL	CCS	IMUCADR	# CHECK FOR IMU USAGE WHICH AVOIDS THE
	TCF	+3	# IMUSE BIT: I.E., IMU COMPENSATION.
	TCF	+6	# FREE. GO AHEAD WITH THE IMU ZERO.
	TCF	+1	
+3	CAF	TEN	# WAIT .1 SEC AND TRY AGAIN.
	TC	BANKCALL	
	CADR	DELAYJOB	
	TCF	CKSTALL	
+6	TC	BANKCALL	# IMU IS NOT IN USE
	CADR	IMUZERO	# SET IMU ZERO DISCRETE FOR 320 MSECS.
	TC	BANKCALL	# WAIT 3 SEC FOR COUNTERS TO INCREMENT
	CADR	IMUSTALL	
	TC	AGSEND	
AGSEND	TC	DOWNFLAG	# ALLOW V37
	ADRES	NODOFLAG	
	CAF	V50N16	
	TC	BANKCALL	
	CADR	GOMARK3	
	TCF	ENDEXT	
	TCF	ENDEXT	
	TC	ENDEXT	
SCALEVEC	VLOAD	MXV	
		VATT1	
		REFSMMAT	
	VXSC	VSL2	
		VSCALE	

Page 209

VAD	VAD	# THIS SECTION ROUNDS THE VE
	AGSRND1	# CORRECTS FOR THE FACT THAT
	AGSRND2	# IS A 2'S COMPLEMENT MACHIN
RTB		# LGC IS A 1'S COMPLEMENT MA
	VECSGNAG	
STOVL	VATT1	
	RATT1	
MXV	VXSC	
	REFSMMAT	
	RSCALE	
VSL8	VAD	# AGAIN THIS SECTION ROUNDS.
	AGSRND1	# ARE ADDED TO DEFEAT ALSIGNA
VAD	RTB	# CASE OF A HIGH-ORDER ZERO C
	AGSRND2	# A LOW ORDER NEGATIVE PART.
	VECSGNAG	
LXA,1		
	VATT1	
SXA,1	LXA,1	
	MPAC +1	
	VATT1 +2	
SXA,1	LXA,1	
	MPAC +4	
	VATT1 +4	
SXA,1	RVQ	
	MPAC +6	
LAGSLIST	=	ONE
V01N14	VN	0114
V50N00A	VN	5000
V00N25	EQUALS	OCT31
V06N16	VN	0616
V00N34	EQUALS	34DEC
V50N16	VN	5016
TSCALE	2DEC	100 B-10
20SEC	DEC	2000
RSCALE	2DEC	3.280839 B-3
VSCALE	2DEC	3.280839 E2 B-9
AGSRND1	2OCT	0000060000
	2OCT	0000060000
	2OCT	0000060000
AGSRND2	2OCT	0000037777
	2OCT	0000037777
S		
# Page 210		
	2OCT	0000037777

July 12, 2016

97

SBANK= LOWSUPER

FOR SUBSEQUENT LOW 2CADRS.

This code is written to file `src/AGS-INITIALIZATION.s`.

B.6 ALARM AND ABORT

```

98  <src/ALARM-AND-ABORT.s 98>≡
    # Copyright:    Public domain.
    # Filename:     ALARM_AND_ABORT.agc
    # Purpose:      Part of the source code for Comanche, build 055. It
    #               is part of the source code for the Command Module's
    #               (CM) Apollo Guidance Computer (AGC), Apollo 11.
    # Assembler:    yaYUL
    # Reference:     pp. 1493-1496
    # Contact:       Ron Burkey <info@sandroid.org>
    # Website:       http://www.ibiblio.org/apollo.
    # Mod history:   2009-05-07 RSB  Adapted from Colossus249 file of the same
    #               name, and page images. Corrected various
    #               typos in the transcription of program
    #               comments, and these should be back-ported
    #               to Colossus249.
    #
    # The contents of the "Comanche055" files, in general, are transcribed
    # from scanned documents.
    #
    # Assemble revision 055 of AGC program Comanche by NASA
    # 2021113-051. April 1, 1969.
    #
    # This AGC program shall also be referred to as Colossus 2A
    #
    # Prepared by
    #
    #               Massachussets Institute of Technology
    #               75 Cambridge Parkway
    #               Cambridge, Massachusetts
    #
    # under NASA contract NAS 9-4065.
    #
    # Refer directly to the online document mentioned above for further
    # information. Please report any errors to info@sandroid.org.
    #
    # Page 1483
    # THE FOLLOWING SUBROUTINE MAY BE CALLED TO DISPLAY A NON-ABORTIVE ALARM CONDITION.
    # EITHER IN INTERRUPT OR UNDER EXECUTIVE CONTROL.
    #
    # THE CALLING SEQUENCE IS AS FOLLOWS:
    #           TC      ALARM
    #           OCT      NNNNN
    #
    # (RETURNS HERE)

    BLOCK 02

```

```

SETLOC  FFTAG7
BANK

EBANK=  FAILREG

COUNT  02/ALARM

```

ALARM TURNS ON THE PROGRAM ALARM LIGHT, BUT DOES NOT DISPLAY.

```

ALARM          INHINT

ALARM2          CA      Q
                TS      ALMCADR
                INDEX   Q
                CA      0
BORTENT         TS      L

PRIOENT         CA      BBANK
+1             EXTEND
                ROR     SUPERBNK      # ADD SUPER BITS.
                TS      ALMCADR +1

LARMENT         CA      Q              # STORE RETURN FOR ALARM
                TS      ITEMP1

                CA      LOC
                TS      LOCALARM
                CA      BANKSET
                TS      BANKALRM

CHKFAIL1        CCS     FAILREG        # IS ANYTHING IN FAILREG
                TCF     CHKFAIL2      # YES TRY NEXT REG
                LXCH    FAILREG
                TCF     PROGLARM      # TURN ALARM LIGHT ON FOR FIRST ALARM

CHKFAIL2        CCS     FAILREG +1
                TCF     FAIL3
                LXCH    FAILREG +1
                TCF     MULTEXIT

FAIL3           CA      FAILREG +2
# Page 1494
                MASK    POSMAX
                CCS     A
                TCF     MULTFAIL
                LXCH    FAILREG +2

```

```

                                TCF      MULTEXIT

PROGLARM      CS      DSPTAB +11D
              MASK    OCT40400
              ADS     DSPTAB +11D

MULTEXIT      XCH     ITEMP1          # OBTAIN RETURN ADDRESS IN A
              RELINT
              INDEX   A
              TC      1

MULTFAIL      CA      L
              AD      BIT15
              TS      FAILREG +2

                                TCF      MULTEXIT

# PRIOLARM DISPLAYS V05N09 VIA PRIODSPR WITH 3 RETURNS TO THE USER FROM THE ASTRONAUT
# AN IMMEDIATE RETURN TO THE USER AT CALL LOC +4.  EXAMPLE FOLLOWS,
#
#          CAF      OCTXX          # ALARM CODE
#          TC      BANKCALL
#          CADR    PRIOLARM
#          ...     ...
#          ...     ...
#          ...     ...          # ASTRONAUT RETURN
#          TC      PHASCHNG        # IMMEDIATE RETURN TO USER.  RESTART
#          OCT     X.1            # PHASE CHANGE FOR PRIO DISPLAY

                                BANK    10
                                SETLOC  DISPLAYS
                                BANK

PRIOLARM      COUNT   10/DSPLA
              INHINT
              TS      L          # * * * KEEP IN DISPLAY ROUTINE'S BANK
                                # SAVE ALARM CODE

                                CA      BUF2          # 2 CADR OF PRIOLARM USER
                                TS      ALMCADR
                                CA      BUF2 +1
                                TC      PRIOENT +1    # * LEAVE L ALONE
-2SEC        DEC     -200        # *** DON'T MOVE
              CAF     V05N09
              TCF     PRIODSPR

```

July 12, 2016

101

	BLOCK	02	
	SETLOC	FFTAG13	
	BANK		
	COUNT	02/ALARM	
BAILOUT	INHINT		
	CA	Q	
	TS	ALMCADR	
	TC	BANKCALL	
	CADR	VAC5STOR	
	INDEX	ALMCADR	
	CAF	0	
	TC	BORTENT	
OCT40400	OCT	40400	
	INHINT		
WHIMPER	CA	TWO	
	AD	Z	
	TS	BRUPT	
	RESUME		
	TC	POSTJUMP	# RESUME SENDS CONTROL HERE
	CADR	ENEMA	
	SETLOC	FFTAG7	
	BANK		
POOD00	INHINT		
	CA	Q	
	TS	ALMCADR	
	TC	BANKCALL	
	CADR	VAC5STOR	# STORE ERASABLES FOR DEBUGGING PURPOSES.
	INDEX	ALMCADR	
	CAF	0	
	TC	BORTENT	
ABORT2	OCT	77770	# DON'T MOVE
OCT77770	CA	V37FLBIT	# IS AVERAGE G ON
	MASK	FLAGWRD7	
	CCS	A	
	TC	WHIMPER -1	# YES. DON'T DO POOD00. DO BAILOUT.
	TC	DOWNFLAG	

```

      ADRES      STATEFLG
      TC          DOWNFLAG

# Page 1496
      ADRES      REINTFLG
      TC          DOWNFLAG
      ADRES      NODOFLAG

      TC          BANKCALL
      CADR        MR.KLEAN
      TC          WHIMPER

CCSHOLE      INHINT
      CA          Q
      TS          ALMCADR
      TC          BANKCALL
      CADR        VAC5STOR
      CA          OCT1103
      TC          ABORT2
OCT1103      OCT          1103
CURTAINS     INHINT
      CA          Q
      TC          ALARM2
OCT217       OCT          00217
      TC          ALMCADR      # RETURN TO USER

DOALARM      EQUALS      ENDOFJOB

# CALLING SEQUENCE FOR VARALARM
#          CAF          (ALARM)
#          TC          VARALARM
# Page 1486

# VARALARM TURNS ON PROGRAM ALARM LIGHT BUT DOES NOT DISPLAY

VARALARM     INHINT

      TS          L          # SAVE USER'S ALARM CODE

      CA          Q          # SAVE USER'S Q
      TS          ALMCADR

      TC          PRIOENT
OCT14        OCT          14      # DON'T MOVE

```

July 12, 2016

103

```
TC      ALMCADR      # RETURN TO USER

ABORT    EQUALS  BAILOUT    # *** TEMPORARY UNTIL ABORT CALLS OUT
This code is written to file src/ALARM-AND-ABORT.s.
```

B.7 ANGLFIND

```

104  <src/ANGLFIND.s 104>≡
      # Copyright:    Public domain.
      # Filename:     ANGLFIND.agc
      # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
      #               It is part of the source code for the Command Module's (CM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         399-411
      # Mod history:   2009-05-09 RSB   Adapted from the Colossus249/ file
      #               of the same name, using Comanche055 page
      #               images.
      #               2009-05-22 RSB   In NOGOM2, TC ZEROEROR corrected to
      #               CADR ZEROEROR.
      #
      # This source code has been transcribed or otherwise adapted from digitized
      # images of a hardcopy from the MIT Museum. The digitization was performed
      # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
      # thanks to both. The images (with suitable reduction in storage size and
      # consequent reduction in image quality as well) are available online at
      # www.ibiblio.org/apollo. If for some reason you find that the images are
      # illegible, contact me at info@sandroid.org about getting access to the
      # (much) higher-quality images which Paul actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 055 of AGC program Comanche by NASA
      #       2021113-051.  10:28 APR. 1, 1969
      #
      #       This AGC program shall also be referred to as
      #               Colossus 2A

      # Page 399

      BANK      15
      SETLOC    KALCMON1
      BANK

      EBANK=    BCDU

      COUNT     22/KALC

      KALCMAN3   TC      INTPRET
                RTB

```



```

                                READCDUK      # PICK UP CURRENT CDU ANGLES
STORE      BCDU                # STORE THE INITIAL S/C ANGLES
AXC,2      TLOAD               # COMPUTE THE TRANSFORMATION FROM
                                MIS            # INITIAL S/C AXES TO STABLE MEMBER AXES
                                BCDU          # (MIS)

CALL

                                CDUTODCM
AXC,2      TLOAD               # COMPUTE THE TRANSFORMATION FROM
                                MFS            # FINAL S/C AXES TO STABLE MEMBER AXES
                                CPHI          # (MFS)

CALL

                                CDUTODCM
SECAD      AXC,1              CALL          # MIS AND MFS ARRAYS CALCULATED      $2
                                MIS
                                TRANSPOS

VLOAD
STADR
STOVL      TMIS      +12D
STADR
STOVL      TMIS      +6
STADR
STORE      TMIS            # TMIS = TRANSPOSE(MIS) SCALED BY 2
AXC,1      AXC,2
                                TMIS
                                MFS

CALL

                                MXM3
VLOAD      STADR
STOVL      MFI      +12D
STADR
STOVL      MFI      +6
STADR
STORE      MFI            # MFI = TMIS MFS (SCALED BY 4)
SETPD      CALL          # TRANSPOSE MFI IN PD LIST
                                18D
                                TRNSPSPD
VLOAD      STADR
STOVL      TMFI      +12D
STADR
STOVL      TMFI      +6

# Page 400

STADR
STORE      TMFI            # TMFI = TRANSPOSE (MFI) SCALED BY 4

# CALCULATE COFSKEW AND MFISYM

```

```

DLOAD DSU
      TMFI +2
      MFI +2
PDDL DSU # CALCULATE COF SCALED BY 2/SIN(AM)
      MFI +4
      TMFI +4
PDDL DSU
      TMFI +10D
      MFI +10D
VDEF
STORE COFSKEW # EQUALS MFISKEW

# CALCULATE AM AND PROCEED ACCORDING TO ITS MAGNITUDE

DLOAD DAD
      MFI
      MFI +16D
DSU DAD
      DP1/4TH
      MFI +8D
STORE CAM # CAM = (MFIO+MFI4+MFI8-1)/2 HALF-SCALE
ARCCOS
STORE AM # AM=ARCCOS(CAM) (AM SCALED BY 2)
DSU BPL
      MINANG
      CHECKMAX
EXIT # MANEUVER LESS THAN 0.25 DEG
INHINT # GO DIRECTLY INTO ATTITUDE HOLD
CS ONE # ABOUT COMMANDED ANGLES
TS HOLDFLAG # NOGO WILL STOP ANY RATE AND SET UP FOR A
TC LOADCDUD # GOOD RETURN
TCF NOGO

CHECKMAX DLOAD DSU
      AM
      MAXANG
BPL VLOAD
      ALTALC # UNIT
      COFSKEW # COFSKEW
UNIT
STORE COF # COF IS THE MANEUVER AXIS
GOTO # SEE IF MANEUVER GOES THRU GIMBAL LOCK
      LOCSKIRT
ALTALC VLOAD VAD # IF AM GREATER THAN 170 DEGREES
      MFI

```

```

          TMFI
VSR1
STOVL    MFISYM
          MFI      +6
VAD      VSR1
          TMFI     +6
STOVL    MFISYM  +6
          MFI     +12D
VAD      VSR1
          TMFI    +12D
STORE    MFISYM  +12D    # MFISYM=(MFI+TMFI)/2    SCALED BY 4

# CALCULATE COF

DLOAD    SR1
          CAM
PDDL     DSU              # PD0 CAM                      $4
          DPHALF
          CAM
BOVB     PDDL              # PD2 1 - CAM                  $2
          SIGNMPAC
          MFISYM  +16D
DSU      DDV
          0
          2
SQRT     PDDL              # COFZ = SQRT(MFISYM8-CAM)/(1-CAM)
          MFISYM  +8D      #                               $ ROOT 2
DSU      DDV
          0
          2
SQRT     PDDL              # COFY = SQRT(MFISYM4-CAM)/(1-CAM) $ROOT2
          MFISYM
DSU      DDV
          0
          2
SQRT     VDEF              # COFX = SQRT(MFISYM-CAM)/(1-CAM) $ROOT 2
UNIT
STORE    COF

# DETERMINE LARGEST COF AND ADJUST ACCORDINGLY

COFMAXGO  DLOAD    DSU
          COF
          COF      +2
BMN       DLOAD    # COFY G COFX
          COMP12

```

```

# Page 402
DSU      COF
          BMN
          COF      +4

METHOD3   # COFZ G COFX OR COFY
GOTO
METHOD1   # COFX G COFY OR COFZ
COMP12    DLOAD  DSU
          COF      +2
          COF      +4
          BMN
          METHOD3   # COFZ G COFY OR COFX

METHOD2    DLOAD  BPL      # COFY MAX
          COFSKEW +2      # UY
          U2POS
          VLOAD  VCOMP
          COF
          STORE  COF
          U2POS  DLOAD  BPL
          MFISYM +2      # UX UY
          OKU21
          DLOAD  DCOMP      # SIGN OF UX OPPOSITE TO UY
          COF
          STORE  COF
          OKU21  DLOAD  BPL
          MFISYM +10D     # UY UZ
          LOCKSKIRT
          DLOAD  DCOMP      # SIGN OF UZ OPPOSITE TO UY
          COF      +4
          STORE  COF      +4
          GOTO
          LOCKSKIRT
          METHOD1  DLOAD  BPL      # COFX MAX
          COFSKEW      # UX
          U1POS
          VLOAD  VCOMP
          COF
          STORE  COF
          U1POS  DLOAD  BPL
          MFISYM +2      # UX UY
          OKU12
          DLOAD  DCOMP
          COF      +2      # SIGN OF UY OPPOSITE TO UX
          STORE  COF      +2
          OKU12  DLOAD  BPL

```

```

                                MFISYM  +4      # UX UZ
                                LOCKSKIRT
                                DLOAD  DCOMP      # SIGN OF UZ OPPOSITE TO UY
                                COF      +4
                                STORE  COF      +4
                                GOTO
                                LOCKSKIRT
METHOD3      DLOAD  BPL      # COFZ MAX
# Page 403
                                COFSKEW +4      # UZ
                                U3POS
                                VLOAD  VCOMP
                                COF
                                STORE  COF
                                DLOAD  BPL
U3POS
                                MFISYM  +4      # UX UZ
                                OKU31
                                DLOAD  DCOMP
                                COF      # SIGN OF UX OPPOSITE TO UZ
                                STORE  COF
                                DLOAD  BPL
OKU31
                                MFISYM  +10D     # UY UZ
                                LOCKSKIRT
                                DLOAD  DCOMP
                                COF      +2      # SIGN OF UY OPPOSITE TO UZ
                                STORE  COF      +2
                                GOTO
                                LOCKSKIRT

# Page 404
# MATRIX OPERATIONS

MXM3      SETPD      # MXM3 MULTIPLIES 2 3X3 MATRICES
                                0      # AND LEAVES RESULT IN PD LIST
                                DLOAD* PDDL*      # ADDRESS OF 1ST MATRIX IN XR1
                                12D,2      # ADDRESS OF 2ND MATRIX IN XR2
                                6,2
                                PDDL*  VDEF      # DEFINE VECTOR M2(COL 1)
                                0,2
                                MXV*  PDDL*      # M1XM2(COL 1) IN PD
                                0,1
                                14D,2
                                PDDL*  PDDL*
                                8D,2
                                2,2
                                VDEF  MXV*      # DEFINE VECTOR M2(COL 2)

```

```

                                0,1
PDDL*  PDDL*                   # M1XM2(COL2) IN PD
                                16D,2
                                10D,2
PDDL*  VDEF                   # DEFINE VECTOR M2(COL 3)
                                4,2
MXV*   PUSH                   # M1XM2(COL 3) IN PD
                                0,1
GOTO   TRNSPSPD               # REVERSE ROWS AND COLS IN PD AND

# RETURN WITH M1XM2 IN PD LIST
TRANSPOS  SETPD  VLOAD*       # TRANSPOS TRANSPOSES A 3X3 MATRIX
                                0
                                0,1
                                # AND LEAVES RESULT IN PD LIST
PDVL*   PDVL*                # MATRIX ADDRESS IN XR1
                                6,1
                                12D,1
                                # MATRIX IN PD
TRNSPSPD  PUSH               # ENTER WITH MATRIX IN PD LIST
DLOAD    PDDL
                                2
                                6
STODL    2
STADR
STODL    6
                                4
PDDL
                                12D
STODL    4
STADR
STODL    12D
                                10D
PDDL

# Page 405

                                14D
STODL    10D
STADR
STORE    14D
RVQ
                                # RETURN WITH TRANSPOSED MATRIX IN PD LIST
MINANG   DEC    .00069375
MAXANG   DEC    .472222

```

GIMBAL LOCK CONSTANTS

D = MGA CORRESPONDING TO GIMBAL LOCK = 60 DEGREES

NGL = BUFFER ANGLE (TO AVOID DIVISIONS BY ZERO) = 2 DEGREES

```

SD          DEC      .433015      # = SIN(D)                      $2
K3S1        DEC      .86603       # = SIN(D)                      $2
K4          DEC      -.25         # = -COS(D)                     $2
K4SQ        DEC      .125         # = COS(D)COS(D)               $2
SNGLCD      DEC      .008725      # = SIN(NGL)COS(D)             $2
CNGL        DEC      .499695      # = COS(NGL)                   $2
READCDUK    INHINT
CA          CDUZ
TS          MPAC      +2
EXTEND
DCA         CDUX
RELINT
TCF         TLOAD     +6
BANK        16
SETLOC      KALCMON2
BANK

COUNT*    $$/KALC

CDUTODCM    AXT,1     SSP          # SUBROUTINE TO COMPUTE DIRECTION COSINE
OCT         3         # MATRIX RELATING S/C AXES TO STARLE
            S1         # MEMBER AXES FROM 3 CDU ANGLES IN T(MPAC)
OCT         1         # SET XR1, S1, AND PD FOR LOOP
STORE       7
SETPD
            0
LOOP SIN     SLOAD*   RTB
            10D,1
            CDULOGIC
STORE       10D       # LOAD PD WITH 0      SIN(PHI)
SIN         PDDL      #                   2      COS(PHI)
            10D       #                   4      SIN(THETA)
COS         PUSH      #                   6      COS(THETA)
TIX,1       DLOAD     #                   8      SIN(PHI)
            LOOPSIN   #                   10     COS(PHI)
            6
DMP         SL1
            10D
# Page 406
STORE       0,2
DLOAD
            4
DMP         PDDL
            0          # (PD6 SIN(THETA)SIN(PHI))
            6

```

DMP	DMP	
	8D	
	2	
SL1	BDSU	
	12D	
SL1		
STORE	2,2	
DLOAD		
	2	
DMP	PDDL	# (PD7 COS(PHI)SIN(THETA)) SCALED 4
	4	
	6	
DMP	DMP	
	8D	
	0	
SL1		
DAD	SL1	
	14D	
STORE	4,2	
DLOAD		
	8D	
STORE	6,2	
DLOAD		
	10D	
DMP	SL1	
	2	
STORE	8D,2	
DLOAD		
	10D	
DMP	DCOMP	
	0	
SL1		
STORE	10D,2	
DLOAD		
	4	
DMP	DCOMP	
	10D	
SL1		
STORE	12D,2	
DLOAD		
DMP	SL1	# (PUSH UP 7)
	8D	
PDDL	DMP	# (PD7 COS(PHI)SIN(THETA)SIN(PSI)) SCALE 4
	6	
	0	


```

DAD      SL1          # (PUSH UP 7)
STADR
STORE    14D,2
DLOAD
DMP      SL1          # (PUSH UP 6)
          8D
PDDL     DMP          # (PD6 SIN(THETA)SIN(PHI)SIN(PSI)) SCALE 4
          6
          2
DSU      SL1          # (PUSH UP 6)
STADR
STORE    16D,2        # C8=-SIN(THETA)SIN(PHI)SIN(PSI)
RVQ      #           +COS(THETA)COS(PHI)
ENDOCM   EQUALS

BANK     15
SETLOC   KALCMON1
BANK

```

```

# CALCULATION OF THE MATRIX DEL.....

```

```

#
#      *      *      __T      *
#      DEL = (IDMATRIX)COS(A)+UU (1-COS(A))+UX SIN(A)          SCALED 1
#
#
#      -
#      WHERE U IS A UNIT VECTOR (DP SCALED 2) ALONG THE AXIS OF ROTATION.
#      A IS THE ANGLE OF ROTATION (DP SCALED 2).
#
#      -
#      UPON ENTRY THE STARTING ADDRESS OF U IS COF, AND A IS IN MPAC.

```

```

COUNT   22/KALC

DELCOMP   SETPD   PUSH          # MPAC CONTAINS THE ANGLE A
          0
          SIN     PDDL          # PD0 = SIN(A)
          COS     PUSH          # PD2 = COS(A)
          SR2     PDDL          # PD2 = COS(A)          $8
          BDSU    BOVB          # PD4 = 1-COS(A)          $2
          DPHALF
          SIGNMPAC

```

```

# COMPUTE THE DIAGONAL COMPONENTS OF DEL

```

```

PDDL
          COF
DSQ      DMP

```

Page 408

	4			
DAD	SL3			
	2			
BOVB				
	SIGNMPAC			
STODL	DEL		# UX UX(U-COS(A)) +COS(A)	\$1
	COF	+2		
DSQ	DMP			
	4			
DAD	SL3			
	2			
BOVB				
	SIGNMPAC			
STODL	DEL	+8D	# UY UY(1-COS(A)) +COS(A)	\$1
	COF	+4		
DSQ	DMP			
	4			
DAD	SL3			
	2			
BOVB				
	SIGNMPAC			
STORE	DEL	+16D	# UZ UZ(1-COS(A)) +COS(A)	\$1

COMPUTE THE OFF-DIAGONAL TERMS OF DEL

DLOAD	DMP			
	COF			
	COF	+2		
DMP	SL1			
	4			
PDDL	DMP		# D6 UX UY (1-COS A)	\$ 4
	COF	+4		
	0			
PUSH	DAD		# D8 UZ SIN A	\$ 4
	6			
SL2	BOVB			
	SIGNMPAC			
STODL	DEL	+6		
BDSU	SL2			
BOVB				
	SIGNMPAC			
STODL	DEL	+2		
	COF			
DMP	DMP			
	COF	+4		

```

      4
      SL1  PDDL      # D6    UX UZ (1-COS A)          $ 4
            COF      +2
      DMP  PUSH      # D8    UY SIN(A)
            0
      DAD  SL2
            6
# Page 409
      BOVB
            SIGNMPAC
      STODL DEL      +4    # UX UZ (1-COS(A))+UY SIN(A)
      BDSU  SL2
      BOVB
            SIGNMPAC
      STODL DEL      +12D  # UX UZ (U-COS(A))-UY SIGN(A)
            COF      +2
      DMP  DMP
            COF      +4
            4
      SL1  PDDL      # D6    UY UZ (1-COS(A))          $ 4
            COF
      DMP  PUSH      # D6    UX SIN(A)
            0
      DAD  SL2
            6
      BOVB
            SIGNMPAC
      STODL DEL      +14D  # UY UZ(1-COS(A)) +UX SIN(A)
      BDSU  SL2
      BOVB
            SIGNMPAC
      STORE DEL      +10D  # UY UZ(1-COS(A)) -UX SIN(A)
      RVQ

# DIRECTION COSINE MATRIX TO CDU ANGLE ROUTINE
# X1 CONTAINS THE COMPLEMENT OF THE STARTING ADDRESS FOR MATRIX (SCALED 2)
# LEAVES CDU ANGLES SCALED 2PI IN V(MPAC)
# COS(MGA) WILL BE LEFT IN S1 (SCALED 1)
#
# THE DIRECTION COSINE MATRIX RELATING S/C AXES TO STABLE MEMBER AXES CAN BE WRITTEN AS ***
#
#      C =COS(THETA)COS(PHI)
#      0
#
#      C =-COS(THETA)SIN(PHI)COS(PHI)+SIN(THETA)SIN(PHI)
#      1

```

```

#
#      C =COS(THETA)SIN(PSI)SIN(PHI)+SIN(THETA)COS(PHI)
#      2
#
#      C =SIN(PSI)
#      3
#
#      C =COS(PSI)COS(PHI)
#      4
#
#      C =-COS(PSI)SIN(PHI)
#      5
#
#      C =-SIN(THETA)COS(PSI)
#      6
#
#      C =SIN(THETA)SIN(PSI)COS(PHI)+COS(THETA)SIN(PHI)
#      7
# Page 410
#      C =-SIN(THETA)SIN(PSI)SIN(PHI)+COS(THETA)COS(PHI)
#      8
#
#      WHERE   PHI = OGA
#              THETA = IGA
#              PSI = MGA

DCMTOCDU      DLOAD*  ARCSIN
                  6,1
                  PUSH  COS           # PD +0          PSI
                  SL1   BOVB
                  SIGNMPAC
                  STORE  S1
                  DLOAD* DCOMP
                  12D,1
                  DDV    ARCSIN
                  S1
                  PDDL*  BPL           # PD +2          THETA
                  0,1      # MUST CHECK THE SIGN OF COS(THETA)
                  OKTHETA # TO DETERMINE THE PROPER QUADRANT
                  DLOAD  DCOMP
                  BPL    DAD
                  SUHALFA
                  DPHALF
                  GOTO
                  CALCPHI
SUHALFA      DSU

```

```

                                DPHALF
CALCPHI      PUSH
OKTHETA      DLOAD*  DCOMP
                                10D,1
                                DDV   ARCSIN
                                S1
                                PDDL*  BPL          # PUSH DOWN PHI
                                8D,1
                                OKPHI
                                DLOAD   DCOMP        # PUSH UP PHI
                                BPL     DAD
                                SUHALFAP
                                DPHALF
                                GOTO
SUHALFAP     DSU      VECOFANG
                                GOTO
                                DPHALF
                                VECOFANG
OKPHI        DLOAD          # PUSH UP PHI
VECOFANG     VDEF    RVQ

# Page 411
# ROUTINE FOR TERMINATING AUTOMATIC MANEUVERS

NOGOM2       INHINT          # THIS LOCATION ACCESSED BY A BZMF NOGO -2
              TC      BANKCALL
              CADR     ZEROERROR

NOGO         INHINT
              TC      STOPRATE

              CAF      TWO          # TERMINATE MANEUVER
              TC      WAITLIST     # NOTE: ALL RETURNS ARE NOW MADE VIA
              EBANK=   BCDU         # GOODEND
              2CADR    ENDMANU

              TCF      ENDOFJOB

```

This code is written to file `src/ANGLFIND.s`.

B.8 AOSTASK AND AOSJOB

118 `<src/AOSTASK-AND-AOSJOB.s 118>≡`

```
# Copyright:      Public domain.
# Filename:       AOSTASK_AND_AOSJOB.agc
# Purpose:       Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Ron Burkey <info@sandroid.org>.
# Website:       www.ibiblio.org/apollo.
# Pages:        1485-1506
# Mod history:   2009-05-27 RSB   Adapted from the corresponding
#               Luminary131 file, using page
#               images from Luminary 1A.
#               2009-06-05 RSB   Corrected a memory-bank error type.
#               2009-06-07 RSB   Corrected a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum. The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum. Many thanks to both. The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo. If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
#
# Page 1485
# PROGRAM NAME:      1/ACCS
# PROGRAM WRITTEN BY: BOB COVELLI AND MIKE HOUSTON
# LAST MODIFICATION:  FEB. 14, 1969 BY G. KALAN
#
# PROGRAM DESCRIPTION:
#       1/ACCS PROVIDES THE INTERFACE BETWEEN THE GUIDANCE PROGRAMS AND THE DIGITAL A
#       CHANGE IN THE MASS OF THE VEHICLE, IN THE DEADBAND SELECTED, IN THE VEHICLE C
#       DOCKED), AND DURING A FRESH START OR A RESTART, 1/ACCS IS CALLED TO COMMUNICA
#
#       THE INPUTS TO 1/ACCS ARE MASS, ACCELERATION (ABDELV), DEADBAND (DB), OFFSET A
#       STAGE VERIFY BIT (CHAN30, BIT2), DOCKED BIT (DAPBOOLS, BIT13), DRIFT BIT (DAP
#       BIT14), AND SURFACE FLAG (FLAGWRDB, BIT8), AND CH5MASK.
```

```

#
#       1/ACCS COMPUTES THE JET ACCELERATIONS (1JACC, 1JACCQ, 1JACCR) AS FUNCTIONS OF MASS. 1J
#       FORMED BY RESOLVING 1JACCQ AND 1JACCR. IN THE DESCENT CASE, THE DESCENT ENGINE MOMENT
#       COMPUTED AS A FUNCTION OF MASS. THE RATE OF CHANGE OF ACCELERATION DUE TO ROTATION OF
#       ACCDOTR) IS ALSO COMPUTED IN THE DESCENT CASE.
#
#       AFTER THE ABOVE COMPUTATIONS, THE PROGRAM 1/ACCONT COMPUTES THE RECIPROCAL NET ACCELE
#       AND V AXES (2 JETS FOR P-AXIS, BOTH 1 AND 2 JETS FOR U AND V AXES), AND THE RECIPROCAL
#       THE P, U, AND V AXES. THE ACCELERATION FUNCTIONS (ACCFCTZ1 AND ACCFCTZ5) ARE ALSO COMP
#       FIRE AND COAST DEADBANDS AND AXISDIST ARE COMPUTED FOR EACH AXIS. FLAT AND ZONE3LIM, T
#       MINIMUM IMPULSE ZONE, ARE COMPUTED. 1/ACCONT ALSO SETS ACCSWU AND ACCSWV, WHICH INDICA
#       IS NOT SUFFICIENT TO PRODUCE MINIMUM ACCELERATION. AT THE COMPLETION OF 1/ACCS, THE AC
#
# SUBROUTINES CALLED:
#       TIMEGMBL
#       MAKECADR
#       ROT45DEG
#
# CALLING SEQUENCE:
#       TC          BANKCALL          # (1/ACCS MUST BE CALLED BY BANKCALL)
#       CADR        1/ACCS
#
# NORMAL EXIT: VIA BANKJUMP
#
# ALARM AND EXIT MODES: NONE
#
# INPUT/OUTPUT: SEE PROGRAM DESCRIPTION.
#
# DEBRIS:
#       ALL OF THE EXECUTIVE TEMPORARY REGISTERS, EXCEPT FIXLOC AND OVFLND, AND THE CORE SET AR
#
# RESTRICTIONS:
#       1/ACCS MUST BE CALLED BY BANKCALL
#       EBANK IS SET TO 6, BUT NOT RESTORED.
# Page 1486

BANK      20
SETLOC    DAPS3
BANK

COUNT*   $$/DAPAO

EBANK=    AOSQ

# ENTRY IS THROUGH 1/ACCJOB OR 1/ACCSIT WHEN 1/ACCS IS TO BE DONE AS A SEPARATE NOVAC JOB.
# IT IS POSSIBLE FOR MORE THAN ONE OF THESE JOBS TO BE SET UP CONCURRENTLY. HOWEVER, SINCE THE

```

NEWJOB, A SECOND MANIFESTATION CANNOT BE STARTED UNTIL THE FIRST IS COMPLETED.

1/ACCSET	CAF	ZERO	# ENTRY FROM FRESH START/RESTART CODING.
	TS	AOSQ	# NULL THE OFFSET ESTIMATES FOR 1/ACCS
	TS	AOSR	
	TS	ALPHAQ	# NULL THE OFFSET ESTIMATES FOR DOWNLI
	TS	ALPHAR	

1/ACCJOB	TC	BANKCALL	# 1/ACCS ASSUMES ENTRY VIA BANKCALL.
	CADR	1/ACCS +2	# SKIP EBANK SETTING.

TC ENDOFJOB

1/ACCS	CA	EBANK6	# ***** EBANK SET BUT NOT RESTORED *****
	TS	EBANK	
	TC	MAKECADR	# SAVE RETURN SO THAT BUF2 MAY BE USED
	TS	ACCRETRN	

DETERMINE MASS OF THE LEM.

CA	DAPBOOLS	# IS THE CSM DOCKED
MASK	CSMDOCKD	
TS	DOCKTEMP	# STORE RECORD OF STATE IN TEMP (MPAC +3).
CCS	A	
CS	CSMASS	# DOCKED: LEMMAS = MASS - CSMASS
AD	MASS	# LEM ALONE: LEMMASS = MASS
TS	LEMASS	

ON THE BASIS OF APSFLAG:

SET THE P-AXIS RATE COMMAND LIMIT FOR 2-JET/2-JET CONTROL

SET MPAC, WHICH INDICATES THE PROPER SET OF COEFFICIENTS FOR THE LEM-ALONE F

ENSURE THAT THE LEM MASS VALUE IS WITHIN THE ACCEPTABLE RANGE

INHINT

CAE	FLGWRD10	# DETERMINE WHETHER STAGED.
MASK	APSFLBIT	
EXTEND		
BZF	DPSFLITE	

Page 1487

CS	POS MAX	# ASCENT (OR ON LUNAR SURFACE)
TS	-2JET LIM	# ALWAYS 2 JETS FOR P-AXIS RATE COMMAND
CAF	OCT14	# INITIALIZE INDEX AT 12.
TS	MPAC	
CS	LEMASS	# CHECK IF MASS TOO HIGH. CATCH STAGING.
AD	HIASCENT	


```

EXTEND
BZMF  MASSFIX
CS    LEMMASS
AD    LOASCENT
EXTEND
BZMF  F(MASS)

MASSFIX  ADS    LEMMASS
        ZL
        CCS    DOCKTEMP
        CAE    CSMASS
        AD    LEMMASS
        DXCH   MASS
        TCF    F(MASS)

# STORE THE VIOLATED LIMIT AS LEMMASS.
# ALSO CORRECT TOTAL MASS, ZEROING THE
# LOW-ORDER WORD.
# DOCKED:  MASS = LEMMASS + CSMASS
# LEM ALONE:  MASS = LEMMASS

DPSFLITE CS    BIT10
        TS    -2JETLIM
        CAF   SIX
        TS    MPAC
        CS    LEMMASS
        AD    HIDESCNT
EXTEND
BZMF  MASSFIX
CS    LEMMASS
AD    LODESCNT
AD    HIASCENT
EXTEND
BZMF  F(MASS)
TCF   MASSFIX

# FOUR JETS FOR P-AXIS RATE COMMAND ERRORS
# EXCEEDING 1.4 DEG/SEC (SCALED AT 45)
# INITIALIZE INDEX AT 6.

# CHECK IF MASS TOO HIGH.  SHOULD NEVER
# OCCUR EXCEPT PERHAPS BEFORE THE PAD
# LOAD IS DONE.

# CHECK IF MASS TOO LOW.  THIS LIMITS THE
# DECREMENTING BY MASSMON.

# COMPUTATION OF FUNCTIONS OF MASS

F(MASS)  RELINT
        CCS    DOCKTEMP
        TCF    DOCKED
        CA     TWO
        STCTR  TS    MPAC    +1
# DOCKED:  USE SEPARATE COMPUTATION.
# J=2,1,0 FOR 1JACCR,1JACCQ,1JACC

        CS    TWO
        ADS   MPAC
# JX=10,8,6 OR 4,2,0 TO INDEX COEFS.

STCTR1   CAE    LEMMASS
        INDEX  MPAC
        AD     INERCONC
        TS     MPAC    +2
# MASS + C

```

```

EXTEND
INDEX  MPAC
DCA    INERCONA
EXTEND
DV     MPAC      +2
INDEX  MPAC
AD     INERCONB
INDEX  MPAC      +1      # 1JACC(J)=A(JX)/(MASS+C(JX) + B(JX)
TS     1JACC          # 1JACC(-1)=L,PVT-CG   SCALED AT 8 FEET

CCS    MPAC      +1
TCF    STCTR
TCF    COMMEQS
TCF    LRESC

# COEFFQ AND COEFFR ARE COMPUTED IN THIS SECTION.  THEY ARE USED TO RESOLVE Q-R COMPO
# U AND V COMPONENTS (SEE ROT-TOUV SECTION).

COMMEQS      CS      1JACCR
              AD      1JACCQ
              EXTEND
              BZMF    BIGIQ
              EXTEND
              DV      1JACCQ      # EPSILON IS A MEASURE OF COUPLING AND IS
              TS      EPSILON      # DEFINED=1-IQ/IR FOR IR GREATER THAN IQ.
              AD      -EPSMAX      # THE COMPUTED EXPRESSION IS EQUIVALENT
              EXTEND
              BZMF    GOODEPS1
              CS      -EPSMAX
              TS      EPSILON      # EPSILON IS LIMITED TO A MAX. OF .42265
GOODEPS1     CA      EPSILON
              EXTEND
              MP      0.35356
              AD      .7071
              TS      COEFFR      # IN THIS CASE WHERE IR IS GREATER THAN
              CS      POSMAX      # IQ, COEFFQ=-.707(1+.5EPSILON)(1-EPSILON)
              AD      EPSILON      # AND COEFFR=.707(1+.5EPSILON)
              EXTEND
              MP      COEFFR
              TS      COEFFQ
              TCF     JACCUV
BIGIQ        EXTEND
              DV      1JACCR      # EPSILON IS DEFINED AS 1-IR/IQ FOR IQ
              TS      -EPSILON      # GREATER THAN IR.  -EPSILON IS COMPUTED
              CS      -EPSILON      # RATHER THAN EPSILON FOR CONVENIENCE
              AD      -EPSMAX

```

```

# Page 1489
GOODEPS2
    EXTEND
    BZMF    GOODEPS2
    CA      -EPSMAX
    TS      -EPSILON      # EPSILON IS LIMITED TO A MAX. OF .42265
    CA      -EPSILON
    EXTEND
    MP      0.35356
    AD      -.7071
    TS      COEFFQ
    CS      -EPSILON      # IN THIS CASE WHERE IQ IS GREATER THAN
    AD      NEGMAX        # IR, COEFFQ=-.707(1+.5EPSILON) AND
    EXTEND              # COEFFR=.707(1+.5EPSILON)(1-EPSILON)
    MP      COEFFQ
    TS      COEFFR
    CS      COEFFQ
    EXTEND
    MP      1JACCQ        # 1JACCQ IS SCALED AT PI/4
    TS      1JACCU        # 1JACCU USED AS TEMPORARY STORAGE
    CA      COEFFR
    EXTEND
    MP      1JACCR
    AD      1JACCU
    EXTEND
    MP      BIT14        # SCALING CHANGED FROM PI/4 TO PI/2
    TS      1JACCU
    TS      1JACCV        # SCALED AT PI/2 RADIANS/SEC(2)
    CCS     MPAC          # COMPUTE L,PVT-CG IF IN DESCENT
    CAF     ZERO          # ZERO SWITCHES AND GO TO 1/ACCONT IN
    TS      ALLOWGTS      # ASCENT
    TCF     1/ACCONT -1
    CS      TWO
    TS      MPAC
    CS      ONE
    TS      MPAC    +1
    TCF     STCTR1

```

```

# THIS SECTION COMPUTES THE RATE OF CHANGE OF ACCELERATION DUE TO THE ROTATION OF THE GIMBALS.
# IMPLEMENTED IN BOTH THE Y-X PLANE AND THE Z-X PLANE IS -- D(ALPHA)/DT = TL/I*D(DELTA)/DT, WEH
#      T = ENGINE THRUST FORCE
#      L = PIVIT TO CG DISTANCE OF ENGINE
#      I = MOMENT OF INERTIA

```

```

LRESC      CAE      ABDELV      # SCALED AT 2(13) CM/SEC(2)
    EXTEND

```

```

      MP      MASS      # SCALED AT B+16 KGS
      TC      DVOVSUB   # GET QUOTIENT WITH OVERFLOW PROTECTION
      ADRES   GFACTM

# MASS IS DIVIDED BY ACCELERATION OF GRAVITY IN ORDER TO MATCH THE UNITS OF IXX,IYY,Z
# THE RATIO OF ACCELERATION FROM PIPAS TO ACCELERATION OF GRAVITY IS THE SAME IN METERS
# THAT IS UNCONVERTED.  2.20462 CONVERTS KG. TO LB.  NOW T IN IN A SCALED AT 2(14).

      EXTEND
      MP      L,PVT-CG      # SCALED AT 8 FEET.
# Page 1490
      INHINT
      TS      MPAC
      EXTEND
      MP      1JACCR
      TC      DVOVSUB      # GET QUOTIENT WITH OVERFLOW PROTECTION
      ADRES   TORKJET1

      TS      ACCDOTR      # SCALED AT PI/2(7)
      CA      MPAC
      EXTEND
      MP      1JACCQ
      TC      DVOVSUB      # GET QUOTIENT WITH OVERFLOW PROTECTION
      ADRES   TORKJET1

SPSCONT      TS      ACCDOTQ      # SCALED AT PI/2(7)
      EXTEND
      MP      DGBF      # .3ACCDOTQ SCALED AT PI/2(8)
      TS      KQ
      CAE     ACCDOTR      # .3ACCDOTR AT PI/2(8)
      EXTEND
      MP      DGBF
      TS      KRDA
      EXTEND
      READ    CHAN12      # NOW COMPUTE QACCDOT, RACCDOT, THE SIGNED
      TS      MPAC      +1 # JERK TERMS.  STORE CHANNEL 12. WITH GIMBAL
      CAF     BIT2      # DRIVE BITS 9 THROUGH 12 SET LOOP
      TCF     LOOP3      # INDEX TO COMPUTE RACCDOT, THEN QACCDOT.
      CAF     ZERO
      TS      MPAC
      CA      MPAC      +1 # ACCDOTQ AND ACCDOTR ARE NOT NEGATIVE,
      INDEX   MPAC      # BECAUSE THEY ARE MAGNITUDES
      MASK    GIMBLBTS
      EXTEND
      BZF     ZACCDOT      # IF NONE, Q(R)ACCDOT IS ZERO.
      CA      MPAC      +1

```

```

INDEX  MPAC          # GIMBAL IS MOVING.  IS ROTATION POSITIVE.
MASK   GIMBLBTS +1
EXTEND
BZF    FRSTZERO      # IF NOT POSITIVE, BRANCH
INDEX  MPAC          # POSITIVE ROTATION, NEGATIVE Q(R)ACCDOT.
CS     ACCDOTQ
TCF    STACCDOT
FRSTZERO INDEX MPAC  # NEGATIVE ROTATION, POSITIVE Q(R)ACCDOT.
CA     ACCDOTQ
TCF    STACCDOT
ZACCDOT CAF  ZERO
STACCDOT INDEX MPAC
TS     QACCDOT      # STORE Q(R)ACCDOT.
CCS    MPAC
TCF    LOOP3  -1    # NOW DO QACCDOT.

# Page 1491
CS     DAPBOOLS      # IS GIMBAL USABLE?
MASK   USEQRJTS
EXTEND
BZF    DOWNGTS       # NO. BE SURE THE GIMBAL SWITCHES ARE DOWN
CS     T5ADR         # YES.  IS THE DAP RUNNINT?
AD     PAXISADR
EXTEND
BZF    +2
TCF    DOWNGTS       # NO. BE SURE THE GIMBAL SWITCHES ARE DOWN
CCS    INGTS         # YES.  IS GTS IN CONTROL?
TCF    DOCKTEST      # YES.  PROCEED WITH 1/ACCS.
TC     IBNKCALL      # NO. NULL OFFSET AND FIND ALLOWGTS
CADR   TIMEGMBL

DOCKTEST CCS  DOCKTEMP  # BYPASS 1/ACCONT WHEN DOCKED.
TCF    1/ACCRET
TCF    1/ACCONT

```

Page 1492

SUBROUTINE:

DVOVSUB

AUTHOR:

C. WORK, MOD 0, 12 JUNE 68

PURPOSE:

THIS SUBROUTINE PROVIDES A SINGLE-PRECISION MACHINE LANGUAGE DIVISION OPERATION

#

(1) THE QUOTIENT, IF THE DIVISION WAS NORMAL.

#

(2) NEGMAX, IF THE QUOTIENT WAS IMPROPER AND NEGATIVE.

#

(3) POSMAX, IF THE QUOTIENT WAS IMPROPER AND POSITIVE OR IF THERE WAS A ZERO DI

#

THE CALLING PROGRAM IS PRESUMED TO BE A JOB IN THE F BANK WHICH CONTAINS DVOVSU

#

THE DIVISOR FOR THIS ROUTINE MAY BE IN EITHER FIXED OR ERASABLE STORAGE. SIGN

#

ASSUMED BETWEEN THE TWO HALVES OF THE DIVIDEND. (THIS IS CERTAIN IF THE A AND

#

RESULT OF A MULTIPLICATION OPERATION.)

CALL SEQUENCE:

L TC DVOVSUB

```

#           L +1   ADRES   (DIVISOR)
#           L +2   RETURN HERE, WITH RESULT IN A,L
# INPUT:      DIVIDEND IN A,L (SIGN AGREEMENT ASSUMED), DIVISOR IN LOCATION DESIGNATED BY
#             DIVISOR MAY BE IN THE DVOVSUB FBANK, FIXED-FIXED FBANK, EBANK 6, OR UNDESIGNATED FBANK.
# OUTPUT:     QUOTIENT AND REMAINDER, OR POSMAX (NEGMAX), WHICHEVER IS APPROPRIATE.
# DEBRIS:     SCRATCHX, SCRATCHY, SCRATCHZ, A, L (NOTE: SCRATCHX, Y, Z ARE EQUATED TO MEMORY)
# ABORTS OR ALARMS:  NONE
# EXITS:      TO THE CALL POINT +2.
# SUBROUTINES CALLED:  NONE.

```

```

DVOVSUB      TS      SCRATCHY      # SAVE UPPER HALF OF DIVIDEND
              TS      SCRATCHX
              INDEX   Q              # OBTAIN ADDRESS OF DIVISOR.
              CA      0
              INCR    Q              # STEP Q FOR PROPER RETURN SEQUENCE.
              INDEX   A
              CA      0              # PICK UP THE DIVISOR.
              EXTEND
              BZF     MAXPLUS        # RETURN POSMAX FOR A ZERO DIVISOR.

              TS      SCRATCHZ      # STORE DIVISOR.

              CCS     A              # GET ABS(DIVISOR) IN THE A REGISTER.
              AD      BIT1
              TCF     ZEROPLUS
              AD      BIT1

ZEROPLUS     XCH      SCRATCHY      # STORE ABS(DIVISOR).  PICK UP TOP HALF OF
              EXTEND
              BZMF    GOODNEG        # DIVIDENT.
                                      # GET -ABS(DIVIDEND)

# Page 1493
              CS      A

GOODNEG      AD      SCRATCHY      # ABS(DIVISOR) - ABS(DIVIDEND)
              EXTEND
              BZMF    MAKEMAX        # BRANCH IF DIVISION IS NOT PROPER.

              CA      SCRATCHX      # RE-ESTABLISH THE DIVIDEND
              EXTEND
              DV      SCRATCHZ      # QUOTIENT IN THE A, REMAINDER IN L.
              TC      Q              # RETURN TO CALLER.

MAKEMAX      CCS      SCRATCHX      # DETERMINE THE SIGN OF THE QUOTIENT.
              CCS      SCRATCHZ      # SCRATCHX AND SCRATCHZ ARE NON-ZERO.
              TCF     MAXPLUS
              CCS      SCRATCHZ

```

	CAF	NEGMAX	# +,- OR -,+
	TC	Q	
MAXPLUS	CAF	POSMAX	# -,- OR +,+
	TC	Q	

COEFFICIENTS FOR THE JET ACCELERATION CURVE FITS

THE CURVE FITS ARE OF THE FORM --

#

1JACC = A/(MASS + C) + B

#

A IS SCALED AT PI/4 RAD/SEC**2 B+16KG, B IS SCALED AT PI/4 RAD/SEC**2, AND C IS SCALED AT B +

#

THE CURVE FIT FOR L,PVT-CG IS OF THE SAME FORM, EXCEPT THAT A IS SCALED AT 8 FT B+16 KG, B IS

AND C IS SCALED AT B+16 KG.

	2DEC	+.0410511917	# L	A	DESCENT
INERCONA	2DEC	+.0059347674	# 1JACCP	A	DESCENT
	2DEC	+.0014979264	# 1JACCQ	A	DESCENT
	2DEC	+.0010451889	# 1JACCR	A	DESCENT
	2DEC	+.0065443852	# 1JACCP	A	ASCENT
	2DEC	+.0035784354	# 1JACCQ	A	ASCENT
	2DEC	+.0056946631	# 1JACCR	A	ASCENT
	DEC	+.155044	# L	B	DESCENT
	DEC	-.025233	# L	C	DESCENT
# Page 1494					
INERCONB	DEC	+.002989	# 1JACCP	B	DESCENT
INERCONC	DEC	+.008721	# 1JACCP	C	DESCENT
	DEC	+.018791	# 1JACCQ	B	DESCENT
	DEC	-.068163	# 1JACCQ	C	DESCENT
	DEC	+.021345	# 1JACCR	B	DESCENT
	DEC	-.066027	# 1JACCR	C	DESCENT
	DEC	+.000032	# 1JACCP	B	ASCENT
	DEC	-.006923	# 1JACCP	C	ASCENT
	DEC	+.162862	# 1JACCQ	B	ASCENT
	DEC	+.002588	# 1JACCQ	C	ASCENT
	DEC	+.009312	# 1JACCR	B	ASCENT
	DEC	-.023608	# 1JACCR	C	ASCENT

GIMBLBTS	OCTAL	01400	
	OCTAL	01000	
	OCTAL	06000	
	OCTAL	04000	
DGBF	DEC	0.6	# .3 SCALED AT 1/2
0.35356	DEC	0.35356	# .70711 SCALED AT 2
GFACTM	OCT	337	# 979.24/2.20462 AT B+15
.7071	DEC	.70711	
-.7071	DEC	-.70711	
-EPSMAX	DEC	-.42265	

CSM-DOCKED INERTIA COMPUTATIONS

DOCKED	CA	ONE	# COEFTR = 1 FOR INERTIA COEFFICIENTS
SPSLOOP1	TS	COEFCTR	# = 7 FOR CG COEFFICIENTS
	CA	ONE	# MASSCTR = 1 FOR CSM
	TS	MASSCTR	# = 0 FOR LEM

INDEX	COEFCTR
CA	COEFF -1

COEFF -1 = C

EXTEND
MP

EXTEND
MP

LET X = CSMMASS AND Y = LEMMASS

INDEX	COEFCTR
AD	COEFF
TS	MPAC
TCF	+4

COEFF = F

MPAC = C X Y + F

SPSLOOP2	TS	MASSCTR
	EXTEND	
	DIM	COEFCTR
	INDEX	COEFCTR
	CA	COEFF +2
	EXTEND	

LOOP TWICE THROUGH HERE TO OBTAIN
MPAC = MPAC + (A X +D)X + (B Y +E)Y
#
LOOP #1 LOOP #2

COEFF +2 = A OR B

Page 1495

INDEX	MASSCTR
MP	LEMASS
INDEX	COEFCTR
AD	COEFF +4
EXTEND	
INDEX	MASSCTR
MP	LEMASS
ADS	MPAC

COEFF +4 = E OR D


```

      CCS      MASSCTR
      TCF      SPSLOOP2
      CCS      COEFCTR      # IF COEFCTR IS POS, EXIT FROM LOOP WITH
      TCF      +7           # CG X DELDOT = MPAC X 4 PI RAD-CM/SEC
      TORQCONS 2DEC      0.51443 B-14      # CORRESPONDS TO 500 LB-FT

      CA      MPAC
      TS      MPAC      +1      # INERTIA = (MPAC +1) X 2(38) KG-CM(2)
      CA      SEVEN
      TCF      SPSLOOP1

      CA      1JACCCON      # 1JACC=1JACCCON/MASS
      ZL
      TC      DVOVSUB
      ADRES    MASS
      TS      1JACC      # SCALED AT PI/4

      CA      POSMAX      # SET INVERSE JET ACCELERATIONS TO POSMAX,
      TS      1/ANETP      # WHICH CORRESPONDS TO ACCEL. OF 1.4 D/SS.
      TS      1/ANET2 +1
      TS      1/ANET2 +2
      TS      1/ANET2 +17D
      TS      1/ANET2 +18D
      EXTEND
      DCA      TORQCONS
      EXTEND
      DV      MPAC      +1
      INHINT
      TS      1JACCQ      # SCALED AT PI/4
      TS      1JACCR

      CA      -.7071
      TS      COEFFQ      # COEFFQ AND COEFFR ARE CHOSEN TO MAKE U-
      CA      .7071      # AND V-AXES ORTHOGONAL FOR DOCKED CASE
      TS      COEFFR
      CA      MASS      # SCALED AT 2(16) KG
      EXTEND
      MP      MPAC      # SCALED AT 4 PI RAD-CM/SEC
      EXTEND
      MP      ABDELV      # SCALED AT 2(13) CM/SEC(2)
      TC      DVOVSUB      # GET QUOTIENT WITH OVERFLOW PROTECTION

      ADRES    MPAC      +1

      TS      ACCDOTR
      TCF      SPSCONT      # CONTINUE K, KSQ CALCULATIONS

```

1JACCCON OCT 00167 # SCALED AT PI/4X2(16) RAD/SEC(2)-KG

COEFFICIENTS FOR CURVE FIT OF THE FORM $Z = A X^2 + B Y^2 + C X Y + D X + E Y + F$

COEFF	DEC	.19518	# C	COEFFICIENT OF INERTIA
	DEC	-.00529	# F	"
	DEC	-.17670	# B	"
	DEC	-.03709	# A	"
	DEC	.06974	# E	"
	DEC	.02569	# D	"

	DEC	.20096	# C	COEFFICIENT OF CG
	DEC	.13564	# F	"
	DEC	.75704	# B	"
	DEC	-.37142	# A	"
	DEC	-.63117	# E	"
	DEC	.41179	# D	"

ASSIGNMENT OF TEMPORARIES FOR 1/ACCS (EXCLUDING 1/ACCONT)

MPAC, MPAC +1, MPAC +2 USED EXPLICITLY

COEFCTR	EQUALS	MPAC	+4	
MASSCTR	EQUALS	MPAC	+5	
SCRATCHX	EQUALS	MPAC	+4	# SCRATCH AREA FOR DVOVSUB ROUTINE.
SCRATCHY	EQUALS	SCRATCHX	+1	
SCRATCHZ	EQUALS	SCRATCHX	+2	

DOCKTEMP	EQUALS	MPAC	+3	# RECORD OF CSMDOCKED BIT OF DAPBOOLS
EPSILON	EQUALS	MPAC	+1	
-EPSILON	EQUALS	EPSILON		
-.1875	DEC	-.18750		

Page 1497

BANK	20
SETLOC	DAPS3
BANK	
EBANK=	AOSQ
COUNT*	\$\$/DAPAO

-1	TS	INGTS	# ZERO INGTS IN ASCENT
1/ACCONT	CA	DB	# INITIALIZE DBVAL1,2,3
	EXTEND		
	MP	BIT13	

	TS	L	# 0.25 DB
	AD	A	
	TS	DBVAL3	# 0.50 DB
	CS	DBVAL1	
	AD	L	
	TS	DBVAL2	# -.75 DB
GETAOSUV	INHINT		
	CAE	AOSR	# COMPUTE ASOU AND AOSV BY ROTATING
	TS	L	# AOSQ AND AOSR.
	CAE	AOSQ	
	TC	IBNKCALL	
	CADR	ROT-TOUV	
	DXCH	AOSU	
	RELINT		
	CA	DAPBOOLS	
	MASK	DRIFTBIT	# ZERO DURING ULLAGE AND POWERED FLIGHT.
	CCS	A	# IF DRIFTING LIGHT,
	CA	ONE	# SET DRIFTER TO 1
	TS	DRIFTER	# SAVE TO TEST FOR DRIFTING FLIGHT LATER
	AD	ALLOWGTS	# NON-ZERO IF DRIFT OR GTS NEAR
	CCS	A	
	CA	FLATVAL	# DRIFTING FLIGHT, STORE .8 IN FLAT
	TS	FLATEMP	# IN POWERED FLIGHT, STORE ZERO IN FLAT
	EXTEND		
	BZF	DOPAXIS	# IF POWERED AND NO GTS, START P AXIS,
	CCS	DRIFTER	# OTHERWISE SET ZONE3LIM
	CA	ZONE3MAX	# 17.5 MS, SCALED AT 4 SECONDS.
	TS	Z3TEM	
DOPAXIS	CA	1JACC	# 1JACC AT PI/4 = 2JACC AT PI/2 =
	AD	BIT9	# ANET AT PI/2 = ANET/ACOAST AT 2(6).
	TS	FUNTEM	# 1 + ANET/ACOAST AT 2(6)
# Page 1498	CA	1JACC	
	TC	INVERT	
	INHINT		# P AXIS DATA MUST BE CONSISTENT
	TS	1/ANETP	# SCALED AT 2(7)/PI.
	TS	1/ANETP +1	
	CS	BIT9	# -1 AT 2(6)
	EXTEND		
	MP	1/ANETP	# -1/ANET AT 2(13)/PI

```

EXTEND
DV      FUNTEM      # -1/(ANET + ANET**2/ACOAST) AT 2(7)/PI
TS      PACCFUN
TS      PACCFUN +1

CA      1/.03      # NO AOS FOR P AXIS, ACOAST = AMIN
TS      1/ACOSTP
TS      1/ACOSTP +1
RELINT

ZL
CCS     DRIFTER
DXCH    AOSU      # ZERO AOSU,V IF IN DRIFT, JUST TO BE SURE

UAXIS   CA      ZERO      # DO U AXIS COMPUTATIONS
        TS      UV      # ZERO FOR U AXIS, ONE FOR V AXIS.

BOTHAXES TS      SIGNAOS   # CODING COMMON TO U,V AXES
        INDEX   UV
        CCS     AOSU      # PICK UP ABS(AOSU OR AOSV)
        AD      ONE      # RESTORE TO PROPER VALUE
        TCF     +3      # AND LEAVE SIGNAOS AT ZERO
        AD      ONE      # NEGATIVE, RESTORE TO PROPER VALUE
        INCR    SIGNAOS   # AND SET SIGNAOS TO ONE TO SHOW AOS NEG
        TS      ABSAOS    # SAVE ABS(AOS)
        CS      SIGNAOS
        TS      -SIGNAOS  # USED AS AN INDEX

        CA      DBVAL1    # SET DB1, DB2 TO DBVAL1 (= DB)
        TS      DBB1
        TS      DBB2

        CA      ABSAOS    # TEST MAGNITUDE OF ABS(AOS)
        AD      -.03R/S2

        EXTEND
        BZMF    NOTMUCH   # ABS(AOS) LESS THAN AMIN
        CCS     FLATEMP   # AGS(AOS) GREATER THAN AMIN
        TCF     SKIPDB1   # I DRIFT OR GTS, DO NOT COMPUTE DB

        CA      DBVAL1
        INDEX   -SIGNAOS

# Page 1499

        ADS     DBB2      # DB2(1) = 2 DB
        INDEX   SIGNAOS
        TS      DBB4      # DB4(3) = 1 DB
        CA      -.1875    # -.1875 PI/2 RAD/SEC(2) SCALED AT PI/2

```

	AD	ABSAOS	# ABSAOS IS SCALED AT PI/2
	EXTEND		
	BZMF	+3	
	CS	DBVAL3	# -.5 DB
	TCF	DBONE	
	CS	ABSAOS	
	DOUBLE		
	DOUBLE		
	AD	BIT14	
	DOUBLE		# 1-8 ABSAOS. (8 IS 16/PI SCALED AT 2/PI)
	EXTEND		
	MP	DB	
DBONE	INDEX	SIGNAOS	# DB1(2)=(1-8 ABSAOS) DB. IF ABSAOS IS
	TS	DBB1	# GREATER THAN .1875 THEN DB1(2) = -.5 DB
	CA	DBVAL2	
	INDEX	-SIGNAOS	
	TS	DBB3	# DB3(4) = -.75 DB
SKIPDB1	CA	ABSAOS	# ABS(AOS) GREATER THAN AMIN, SO IT IS
	EXTEND		
	MP	BIT12	
	AD	ABSAOS	# (9/8) ABSAOS.
	TC	INVERT	# ALL RIGHT TO DIVIDE
	INDEX	-SIGNAOS	
	TS	1/ACOSTT +1	# 1/ACOSTPOS(NET) = 1/ABS(AOS)
	CA	1/.03	
	INDEX	SIGNAOS	
	TS	1/ACOSTT	# 1/ACOSTNEG(POS) = 1/AIN
	CA	ABSAOS	
	AD	1JACCU	
	AD	1JACCU	# 2 JACC + ABS(AOS)
	AD	BIT9	# MAXIMUM VALUE IN COMPUTATIONS
	TS	A	# TEST FOR OVERFLOW
	TCF	SKIPDB2	# NO OVERFLOW, DO NORMAL COMPUTATION
	CA	ABSAOS	# RESCALE TO PI TO PREVENT OVERFLOW
	EXTEND		
	MP	BIT14	
	AD	1JACCU	# 1 JACC AT PI/2 = 2JACC AT PI
	TS	ANET	# ANETPOS(NEG) MAX SCALED AT PI =
			# ANETPOS(NEG) MAX/ACOSTNEG(POS) AT 2(7)
	AD	BIT8	# 1 + ANETPOS/ACOSTNEG AT 2(7)
	XCH	ANET	# SAVE IN ANET, WHILE PICKING UP ANET
	TC	INVERT	
	EXTEND		

Page 1500

	MP	BIT14	# SCALE 1/ANET AT 2(7)/PI
	TS	1/ANET	
	CA	ACCHERE	# SET UP RETURN FROM COMPUTATION ROUTINE
	TS	ARET	
	CS	BIT8	# -1 AT 2(7)
	TCF	DOACCFUN	# FINISH ACCFUN COMPUTATION
ACCHERE	TCF	ACCTHERE	
NOTMUCH	TS	L	# ABS(AOS) LESS THAN AMIN, SAVE IN L
	CA	1/.03	# ACOASTPOS,NEG = AMIN
	TS	1/ACOSTT	
	TS	1/ACOSTT +1	
	CCS	FLATEMP	
	TCF	SKIPDB2	# DO NOT COMPUTE DB IF DRIFT OR GTS
	CA	.0125RS	# AMIN/2
	AD	L	# L HAS ABS(AOS) - AMIN
	EXTEND		# RESULT IS ABS(AOS)- AMIN/2
	BZMF	NOAOS	# ABS(AOS) LESS THAN AMIN/2
SOMEAOS	CA	DBVAL3	# AMIN/2 LT ABS(AOS) LT AMIN
	INDEX	-SIGNAOS	
	TS	DBB3	# DB3(4) = DB/2
	AD	A	
	INDEX	SIGNAOS	
	TS	DBB4	# DB4(3) = DB
	TCF	SKIPDB2	
NOAOS	CA	DBVAL1	
	TS	DBB3	# DB3,4 = DB
	TS	DBB4	
SKIPDB2	CA	ABSAOS	# ANETPOS(NEG) MAX = 2 JACC + ABS(AOS)
	AD	1JACCU	
	AD	1JACCU	
	TS	ANET	# CANNOT OVERFLOW HERE
CL1/NET+	TC	DO1/NET+	# COMPUTE 1/ANET, ACCFUN
ACCTHERE	INDEX	-SIGNAOS	
	TS	Z5TEM +2	# STORE ACCFUN IN TEMPORARY BUFFER
	CA	1/ANET	
	INDEX	-SIGNAOS	

Page 1501

```

      TS      1/ATEM2 +2      # STORE 1/ANET IN TEMPORARY BUFFER

      CA      ABSAOS          # SEE IF OVERFLOW IN MIN CASE
      AD      1JACCU

      AD      BIT9            # MAXIMUM POSSIBLE VALUE
      TS      A                # OVERFLOW POSSIBLE BUT REMOTE
      TCF     +2
      CA      POSMAX          # IF OVERFLOW, TRUNCATE TO PI/2
      AD      -.03R/S2        # RESTORE TO CORRECT VALUE
      TS      ANET
      TC      DO1/NET+        # COMPUTE 1/ANET, ACCFUN

      INDEX   -SIGNAOS        # STORE MIN VALUES JUST AS MAX VALUES
      TS      Z5TEM
      CA      1/ANET
      INDEX   -SIGNAOS
      TS      1/ATEM2

      CS      ABSAOS          # NOW DO NEG(POS) CASES
      AD      1JACCU
      AD      1JACCU          # ANETNEG(POS) MAX
      TC      1/ANET-        # COMPUTE 1/ANET, ACCFUN, AND ACCSW
      INDEX   SIGNAOS        # STORE NEG(POS) VALUES JUST AS POS(NEG)
      TS      Z1TEM +2
      TS      L                # SAVE IN L FOR POSSIBLE FUTURE USE
      CA      1/ANET
      INDEX   SIGNAOS
      TS      1/ATEM1 +2
      CS      ABSAOS
      AD      1JACCU          # 1/ANETNEG(POS) MIN
      TS      ANET
      AD      -.03R/S2        # TEST FOR AMIN
      EXTEND          # IF ANET LESS THAN AMIN, STORE MAX JET
      BZMF     FIXMIN        # VALUES FOR MIN JETS AND SET ACCSW

      TC      1/NETMIN        # OTHERWISE DO MIN JET COMPUTATIONS
      INDEX   SIGNAOS        # STORE VALUES
      TS      Z1TEM
      CA      1/ANET
      INDEX   SIGNAOS
      TS      1/ATEM1

      INDEX   UV
      CA      +UMASK
      MASK    CH5MASK        # TEST FOR +U (+V) JET FAILURES

```

STMIN-

```

EXTEND
BZF      FAIL-
CA       1/ATEM2      # REPLACE FUNCTION VALUES DEPENDING ON THE
TS       1/ATEM2 +2   # FAILED JET PAIR WITH CORRESPONDING ONE-
CA       Z5TEM        # JET (OR AMIN) FUNCTION VALUES
TS       Z5TEM  +2
INDEX    UV

FAIL-
# Page 1502

CA       -UMASK
MASK     CH5MASK      # TEST FOR -U (-V) JET FAILURES
EXTEND
BZF      DBFUN
CA       1/ATEM1      # REPLACE FUNCTION VALUES DEPENDING ON THE
TS       1/ATEM1 +2   # FAILED JET PAIR WITH CORRESPONDING ONE-
CA       Z1TEM        # JET (OR AMIN) FUNCTION VALUES
TS       Z1TEM  +2

DBFUN    CS          DBB3      # COMPUTE AXISDIST
          AD          DBB1
          AD          FLATEMP
          TS          AXDSTEM
          CS          DBB4
          AD          DBB2
          AD          FLATEMP
          TS          AXDSTEM +1

          INHINT
          CCS         UV       # TEST FOR U OR V AXIS
          TCF         STORV    # V AXIS          STORE V VALUES

          CA          ACCSW    # U AXIS          STORE U VALUES
          TS          ACCSWU

          CA          NINE     # TRANSFER 10 WORDS VIA GENTRAN
          TC          GENTRAN +1
          ADRES       1/ATEM1  # TEMPORARY BUFFER
          ADRES       1/ANET1  # THE REAL PLACE

          RELINT
          DXCH        DBB1     # SAVE U DBS FOR LATER STORING
          DXCH        UDB1
          DXCH        DBB4
          DXCH        UDB4

          DXCH        AXDSTEM
          DXCH        UAXDIST

```



```

CA      ONE      # NOW DO V AXIS
TS      UV
CA      ZERO
TCF     BOTHAXES # AND DO IT AGAIN

STORV   CA      ACCSW      # STORE V AXIS VALUES
        TS      ACCSWV
        CA      NINE
        TC      GENTRAN +1

# Page 1503
        ADRES   1/ATEM1    # TEMPORARY BUFFER
        ADRES   1/ANET1 +16D # THE REAL PLACE

        DXCH    FLATEMP    # NOW STORE DEADBANDS FOR ALL AXES
        DXCH    FLAT      # FLAT AND ZONE3LIM

        CA      DBVAL1    # COMPUTE P AXIS DEADBANDS
        TS      PDB1
        TS      PDB2
        AD      FLAT
        TS      PDB3
        TS      PDB4
        CA      ZERO
        TS      PAXDIST
        TS      PAXDIST +1

        CCS     FLAT
        TCF     DRFDB      # DRIFT OR GTS -- COMPUTE DBS

        DXCH    UDB1      # STORE U DEADBANDS
        DXCH    FIREDDB   # CANNOT USE GENTRAN BECAUSE OF RELINT
        DXCH    UDB4
        DXCH    COASTDB
        DXCH    UAXDIST
        DXCH    AXISDIST
        DXCH    DBB1      # STORE V AXIS DEADBANDS
        DXCH    FIREDDB +16D # COULD USE GENTRAN IF DESIRED
        DXCH    DBB4
        DXCH    COASTDB +16D
        DXCH    AXDSTEM
        DXCH    AXISDIST +16D

        TCF     1/ACCRET +1 # ALL DONE
DRFDB   CA      DBVAL1    # DRIFT DEADBANDS

```

	TS	FIREDB	
	TS	FIREDB +1	
	TS	FIREDB +16D	
	TS	FIREDB +17D	
	AD	FLAT	
	TS	COASTDB	
	TS	COASTDB +1	
	TS	COASTDB +16D	
	TS	COASTDB +17D	
	CA	ZERO	
	TS	AXISDIST	
	TS	AXISDIST +1	
	TS	AXISDIST +16D	
	TS	AXISDIST +17D	
# Page 1504			
1/ACCRET	INHINT		
	CS	DAPBOOLS	# SET BIT TO INDICATE DATA GOOD.
	MASK	ACCSOKAY	
	ADS	DAPBOOLS	
	RELINT		
	CA	ACCRETRN	
	TC	BANKJUMP	# RETURN TO CALLER
INVERT	TS	HOLD	# ROUTINE TO INVERT -INPUT AT PI/2
	CA	BIT9	# 1 AT 2(6)
	ZL		# ZERO L FOR ACCURACY AND TO PREVENT OVFL0
	EXTEND		
	DV	HOLD	
	TC	Q	# RESULT AT 2(7)/PI
DOWNGTS	CAF	ZERO	# ZERO SWITCHES WHEN USEQRJTS BIT IS UP
	TS	ALLOWGTS	# OR DAP IS OFF
	TS	INGTS	
	TCF	DOCKTEST	
1/ANET-	ZL		
	LXCH	ACCSW	# ZERO ACCSW
	TS	ANET	# SAVE ANET
	AD	-.03R/S2	# TEST FOR MIN VALUE
	EXTEND		
	BZMF	NETNEG	# ANET LESS THAN AMIN, SO FAKE IT
1/NETMIN	CA	ANET	
	EXTEND		
	INDEX	-SIGNAOS	
	MP	1/ACOSTT +1	# ANETNEG(POS)/ACOSTPOS(NEG) AT 2(6)

THE FOLLOWING CODING IS VALID FOR BOTH POS OR NEG
VALUES OF AOS

DO1/NET+	AD	BIT9	# 1 + ANET/ACOAST AT 2(6)
	XCH	ANET	# SAVE AND PICK UP ANET
	EXTEND		
	QXCH	ARET	# SAVE RETURN
	TC	INVERT	
	TS	1/ANET	# 1/ANET AT 2(7)/PI
	CS	BIT9	# -1 AT 2(6)
DOACCFUN	EXTEND		
	MP	1/ANET	# -1/ANET AT 2(13)/PI
	EXTEND		
	DV	ANET	# ACCFUN AT 2(7)/PI
	TC	ARET	# RETURN
NETNEG	CS	-.03R/S2	# ANET LESS THAN AMIN -- SET EQUAL TO AMIN
	TS	ANET	
# Page 1505	TCF	1/NETMIN +1	# CONTINUE AS IF NOTHING HAPPENED.
FIXMIN	CCS	SIGNAOS	
	CA	TWO	# IF AOS NEG, ACCSW = +1
	AD	NEGONE	# IF AOS POS, ACCSW = -1
	TS	ACCSW	
	AD	UV	# IF ACCSW = +1, TEST FOR +U (+V) JET FAIL
	INDEX	A	# IF ACCSW = -1, TEST FOR -U (-V) JET FAIL
	CA	-UMASK +1	
	MASK	CH5MASK	
	EXTEND		
	BZF	+4	
	CS	-.03R/S2	# JET FAILURE -- CANNOT USE 2-JET VALUES
	TS	ANET	# ANET = AMIN
	TCF	STMIN- -1	# CALCULATE FUNCTIONS USING AMIN
	CA	L	# L HAS ACCFUN
	TCF	STMIN-	# STORE MAX VALUES FOR MIN JETS

ERASABLE ASSIGNMENTS FOR 1/ACCONT

1/ANETP	EQUALS	BLOCKTOP +2
1/ACOSTP	EQUALS	BLOCKTOP +4
PACCFUN	EQUALS	BLOCKTOP +8D
PDB1	EQUALS	BLOCKTOP +10D
PDB2	EQUALS	BLOCKTOP +11D
PDB4	EQUALS	BLOCKTOP +12D

PDB3	EQUALS	BLOCKTOP +13D	
PAXDIST	EQUALS	BLOCKTOP +14D	
ACCSW	EQUALS	VBUF	# EXECUTIVE TEMPORARIES
			# CANNOT DO CCS NEWJOB DURING 1/ACCS
1/ATEM1	EQUALS	ACCSW +1	# TEMP BUFFER FOR U AND V AXES
1/ATEM2	EQUALS	1/ATEM1 +1	
1/ACOSTT	EQUALS	1/ATEM1 +4	
Z1TEM	EQUALS	1/ATEM1 +6	
Z5TEM	EQUALS	1/ATEM1 +7	
UDB1	EQUALS	1/ATEM1 +10D	# UAXIS DEADBAND BUFFER
UDB2	EQUALS	1/ATEM1 +11D	
UDB4	EQUALS	1/ATEM1 +12D	
UDB3	EQUALS	1/ATEM1 +13D	
UAXDIST	EQUALS	1/ATEM1 +14D	
DBB1	EQUALS	1/ATEM1 +16D	# TEMP DEADBAND BUFFER, ALSO V AXIS
DBB2	EQUALS	1/ATEM1 +17D	
DBB4	EQUALS	1/ATEM1 +18D	
DBB3	EQUALS	1/ATEM1 +19D	
AXDSTEM	EQUALS	1/ATEM1 +20D	
# Page 1506			
FLATEMP	EQUALS	1/ATEM1 +22D	
Z3TEM	EQUALS	1/ATEM1 +23D	# MUST FOLLOW FLATEMP
DBVAL1	EQUALS	DB	
DBVAL2	EQUALS	INTB15+	
DBVAL3	EQUALS	INTB15+ +1	
DRIFTER	EQUALS	INTB15+ +2	
UV	EQUALS	MPAC	
ANET	EQUALS	MPAC +3	
FUNTEM	EQUALS	MPAC +3	
1/ANET	EQUALS	MPAC +4	
ARET	EQUALS	MPAC +5	
ABSAOS	EQUALS	MPAC +6	
SIGNAOS	EQUALS	MPAC +7	
-SIGNAOS	EQUALS	MPAC +8D	
HOLD	EQUALS	MPAC +9D	
ACCRETRN	EQUALS	FIXLOC -1	
ZONE3MAX	DEC	.004375	# 17.5 MS (35 MS FOR 1 JET) AT 4 SECONDS
FLATVAL	DEC	.01778	# .8 AT PI/4 RAD

July 12, 2016

141

-.03R/S2	OCT	77377	# -PI/2(7) AT PI/2
.0125RS	EQUALS	BIT8	# PI/2(+8) AT PI/2
1/.03	EQUALS	POSMAX	# 2(7)/PI AT 2(7)/PI
PAXISADR	GENADR	PAXIS	
			# THE FOLLOWING 4 CONSTANTS ARE JET
			# FAILURE MASKS AND ARE INDEXED
-UMASK	OCT	00110	# -U
	OCT	00022	# -V
+UMASK	OCT	00204	# +U
	OCT	00041	# +V

This code is written to file `src/AOSTASK-AND-AOSJOB.s`.

B.9 AOTMARK

```

142  <src/AOTMARK.s 142>≡
      # Copyright:    Public domain.
      # Filename:     AOTMARK.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         244-261
      # Mod history:   2009-05-10 SN    (Sergio Navarro).  Started adapting
      #               from the Luminary131/ file of the same
      #               name, using Luminary099 page images.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum.  The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum.  Many thanks to both.  The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo.  If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969

      # Page 244

      BANK      12
      SETLOC    AOTMARK1
      BANK

      EBANK=     XYMARK
      COUNT*    $$/MARK

AOTMARK      INHINT
              CCS      MARKSTAT      # SEE IF AOTMARK BUSY
              TC       +2             # MARK SYSTEM BUSY -- DO ALARM
              TC       EXTVBCHK
              TC       POOD00
              OCT      00105

```

EXTVBCHK	CAF	SIX	# SEE IF EXT. VERB WORKING
	MASK	EXTVBACT	
	CCS	A	
	TCF	MKABORT	# YES -- ABORT
	CAF	BIT2	# NO -- DISALLOW SOME EXTENDED VERB ACTION
	ADS	EXTVBACT	# BIT2 RESET IN ENDMARK
MKVAC	CCS	VAC1USE	# LOOK FOR A VAC AREAD -- DO ABORT IF
	TCF	MKVACFND	# NONE AVAILABLE
	CCS	VAC2USE	
	TCF	MKVACFND	
	CCS	VAC3USE	
	TCF	MKVACFND	
	CCS	VAC4USE	
	TCF	MKVACFND	
	CCS	VAC5USE	
	TCF	MKVACFND	
	DXCH	BUF2	
	TC	BAILOUT1	# ALL VAC AREAS OCCUPIED -- ABORT.
	OCT	01207	
MKVACFND	AD	TWO	
	TS	MARKSTAT	# STORE VAC ADR IN LOW 9 OF MARKSTAT
	CAF	ZERO	
	INDEX	MARKSTAT	
	TS	0 -1	# ZERO IN VACUSE REG TO SHOW VAC OCCUPIED
	CAF	PRI015	
	TC	FINDVAC	# SET UP JOB FOR GETDAT
	EBANK=	XYMARK	
	2CADR	GETDAT	
	RELINT		
	TCF	SWRETURN	
# Page 245			
MKABORT	DXCH	BUF2	
	TC	BAILOUT1	# CONFLICT WITH EXTENDED VERB
	OCT	01211	
MKRELEAS	CAF	ZERO	
	XCH	MARKSTAT	# SET MARKSTAT TO ZERO
	MASK	LOW9	# PICK UP VAC AREA AOR
	CCS	A	
	INDEX	A	
	TS	0	# SHOW MKVAC AREA AVAILABLE

	CAF	ONE	
	TC	IBNKCALL	
	CADR	GOODEND	# GO WAKE UP CALLING JOB
# Page 246			
KILLAOT	CAF	ZERO	
	TS	EXTVBACT	# TERMINATE AOTMARK -- ALLOW EXT VERB
	TC	GOTOPOOH	
GETDAT	CS	MARKSTAT	# SET BIT12 TO DISCOURAGE MARKRUPT
	MASK	BIT12	# BIT12 RESET AT GETMARK
	ADS	MARKSTAT	
	CAF	V01N71	# DISPLAY DETENT AND STAR CODE
	TC	BANKCALL	
	CADR	GOMARKF	
	TCF	KILLAOT	# V34 -- DOES GOTOPOOH
ENTERDAT	TCF	DODAT	# V33 -- PROCEED -- USE THIS STAR FOR MARKS
	TCF	GETDAT	# ENTER -- REDISPLAY STAR CODE
DODAT	CAF	HIGH9	# PICK DETENT CODE FROM BITS7-9 OF AOTCODE
	MASK	AOTCODE	# AND SEE IF CODE 1 TO 6
	EXTEND		
	MP	BIT9	
	TS	XYMARK	# STORE DETENT
	EXTEND		
	BZMF	GETDAT	# COAS CALIBRATION CODE - NO GOOD HERE
	AD	NEG7	# SEE IF DETENT 7 FOR COAS
	EXTEND		
	BZF	CODE7	
	TCF	CODE1T06	
CODE7	CAF	V06N87*	# CODE 7, COAS SIGHTING, GET OPTIC AXIS
	TC	BANKCALL	# AZ AND EL OF SIGHTING DEVICE FROM ASTRO
	CADR	GOMARKF	
	TCF	KILLAOT	# V34 -- DOES GOTOPOOH
	TCF	+2	# PROCEED
	TCF	CODE7	# ON ENTER, RECYCLE
	EXTEND		
	DCA	AZ	# PICK UP AZ AND EL IN SP 25 COMP
	INDEX	FIXLOC	
	DXCH	8D	# STORE IN 8D AND 9D OF LOCAL VAC


```

CAF      ZERO      # BACKUP SYSTEM TO BE USED
TCF      COASCODE  # ZERO APPARENT ROTATION

CODE1T06  INDEX    XYMARK      # INDEX AOT POSITION BY DET CODE
          CA       AOTEL -1
          INDEX    FIXLOC
          TS       9D          # STORE ELEVATION IN VAC+9D

# Page 247
          INDEX    XYMARK      # INDEX DET CODE 1,2 OR 3
          CA       AOTAZ -1
          INDEX    FIXLOC
          TS       8D          # STORE AZIMUTH IN VAC +8D

          CA       AOTAZ +1     # COMPENSATION FOR APPARENT ROTATION OF
EXTEND                                # AOT FIELD OF VIEW IN LEFT AND RIGHT
          INDEX    FIXLOC      # DETENTS IS STORED IN VAC +10D IN SP
          MSU      8D          # PRECISION ONE'S COMPLEMENT
COASCODE  INDEX    FIXLOC
          TS       10D         # ROT ANGLE

          TC       INTERPRET    # COMPUTE X AND Y PLANE VECTORS

# Page 248
# THE OPTAXIS SOBRoutine COMPUTES THE X AND Y MARK PLANE VECs AND
# ROTATES THEM THRU THE APPARENT FIELD OF VIEW ROTATION UNIQUE TO AOT
# OPTAXIS USES OANB TO COMPUTE THE OPTIC AXIS
#
#      INPUT --      AZIMUTH ANGLE IN SINGLE PREC AT CDU SCALE IN 8D OF JOB VAC
#                      ELEVATION ANGLE IN SINGLE PREC AT CDU SCALE IN 9D OF JOB VAC
#                      ROTATION ANGLE IN SINGLE PREC IS COMP SCALED BY PI IN 10D OF VAC
#
#      OUTPUT --     OPTIC AXIS VEC IN NG COORDS IN SCAXIS
#                      X-MARK PLANE 1/4VEC IN NB COORDS AT 18D OF JOB VAC
#                      Y-MARK PLANE 1/4VEC IN NB COORDS AT 12D OF JOB VAC

OPTAXIS   CALL      # GO COMPUTE OA AN X AND Y PLANE VECs
          OANB
          SLOAD     SR1      # LOAD APP ROTATION IN ONES COMP
          10D       # RESCALE BY 2PI
          PUSH      SIN      # 1/2SIN(ROT) 0-1
          PDDL      COS
          PUSH      VXSC     # 1/2COS(ROT) 2-3
          18D
          PDDL      VXSC     # 1/4COS(ROT)UYP 4-9
          0

```

```

                24D          # 1/4SIN(ROT)UXP
BVSU   STADR          # UP 4-9
STODL  12D           # YPNB=1/4(COS(ROT)UYP-SIN(ROT)UXP)
VXSC   PDDL          # UP 2-3 UP 0-1 FOR EXCHANGE
                24D          # 1/4COS(ROT)UXP          PUSH 0-5
VXSC   VAD           # 1/4SIN(ROT)UYP
                18D          # UP 0-5
STADR
STOVL  18D           # XPNB=1/4(COS(ROT)UXP+SIN(ROT)UYP)
                L06ZEROS     # INITIALIZE AVE STAR VEC ACCUMULATOR
STORE  STARAD +6
EXIT
TCF    GETMKS

```

Page 249

THE OANB SUBROUTINE COMPUTES THE OPTIC AXIS OF THE SIGHTING INSTRUMENT
FROM AZIMUTH AND ELEVATION INPUT FROM THE ASTRONAUT.

```

#
#      INPUT --      AZIMUTH ANGLE IN SINGLE PREC 2'S COMP IN 8D OF JOB VAC
#                    ELEVATION ANGLE IN SINGLE PREC 2'S COMP IN 9D OF VAC
#
#      OUTPUT --     OPTIC AXIS IN NB COORDS. IN SCAXIS
#                    X-PLANE 1/2VEC IN NB COORDS AT 24D OF VAC
#                    Y-PLANE 1/2VEC IN NB COORDS AT 18D OF VAC

```

```

                BANK    05
                SETLOC  AOTMARK2
                BANK
                COUNT*  $$/MARK

OANB   SETPD   STQ
                0
                GCTR          # STORE RETURN
SLOAD  RTB
                9D           # PICK UP SP ELV
                CDULOGIC
PUSH   COS
PDDL   SIN          # 1/2COS(ELV)   PD 0-1
STADR
STODL  SCAXIS       # OAX=1/2SIN(ELV)
                8D
RTB
                CDULOGIC
PUSH   COS
STORE  20D          # STORE UYP(Y)  20-21

```

July 12, 2016

147

```
PDDL    SIN                # 1/2COS(AZ)    PD 2-3
PUSH    DCOMP              # PUSH 1/2S IN (AZ)    4-5
STODL    22D               # STORE UYP(Z)  22-23
                     LO6ZEROS
STODL    18D               # STORE UYP(X)  18-19
DMP      SL1
          0
STODL    SCAXIS +2         # OAY=1/2COS(ELV)SIN(AZ)
DMP      SL1               # UP    2-3
STADR                      # UP    0-1
STOVL    SCAXIS +4         # OAZ=1/2COS(ELV)COS(AZ)
          18D              # LOAD UYP VEC
VXV      UNIT
          SCAXIS           # UXP VEC=UYP X OA
STORE    24D               # STORE UXP
GOTO
          GCTR
```

Page 250

```
# SURFSTAR COMPUTES A STAR VECTOR IN SM COORDINAGES FOR LUNAR
# SURFACE ALIGNMENT AND EXITS TO AVEIT TO AVERAGE STAR VECTORS.
#
#      GIVEN  X-MARK PLANE 1/4 VEC IN NB AT 18D OF LOCAL VAC
#              Y-MARK PLANE 1/4 VEC IN NB AT 12D OF LOCAL VAC
#              CURSOR SP 2COMP AT POSITION 1 OF INDEXED MARKVAC
#              SPIRAL SP 2COMP AT POSITION 3 OF INDEXED MARKVAC
#              CDUY,Z,X AT POSITIONS 0,2,4 OF INDEXED MARKVAC
```

```
BANK     15
SETLOC   P50S
BANK
COUNT*  $$/R59
```

```
SURFSTAR  VLOAD*
          0,1                # PUT X-MARK CDUS IN CDUSPOT FOR TRG*NBSM
STORE     CDUSPOT
SLOAD*    RTB
          1,1                # PICK UP YROT
          CDULOGIC
STORE     24D                # STORE CURSOR FOR SPIRAL COMP (REVS)
BZE
          YZCHK               # IF YROT ZERO -- SEE IF SROT ZERO
JUSTZY    PUSH    COS
PDDL      SIN                # 1/2COS(YROT)  0-1
VXSC      PDDL          # UP 0-1          1/8SIN(YROT)UXP 0-5
          18D
VXSC      VSU                # UP    0-5
```

```

UNIT      12D      # UYP
UNIT      VXV
          SCAXIS
UNIT      PUSH
SLOAD*    RTB
          3,1      # PICK UP SPIRAL
          CDULOGIC
STORE     26D      # STORE SPIRAL (REVS)
DSU       DAD
          24D
          ABOUTONE
DMP
          DP1/12
STORE     26D      # SEP=(360 + SPIRAL -CURSOR)/12
SIN       VXSC     # UP      0-5
VSL1     PDDL      # 1/2SIN(SEP)(UPP X OA) 0-5
          26D
COS       VXSC
          SCAXIS
VSL1     VAD       # UP      0-5
JUSTOA   UNIT      CALL
          TRG*NBSM
STCALL    24D      # STAR VEC IN SM
          AVEIT     # GO AVERAGE

# Page 251
ABOUTONE 2DEC     .99999999

DP1/12    EQUALS   DEG30      # .08333333
          BANK     7
          SETLOC   AOTMARK1
          BANK
COUNT*   $$/MARK
YZCHK     SLOAD*   BZE        # YROT ZERO AND IF SROT ZERO FORCE STAR
          3,1      # ALONG OPTIC AXIS
          YSZERO
          DLOAD    GOTO
          24D
          JUSTZY   # SROT NOT ZERO -- CONTINUE NORMALLY
YSZERO    VLOAD    GOTO
          SCAXIS
          JUSTOA

# Page 252
# THE GETMKS ROUTINE INITIALIZES THE SIGHTING MARK PROCEDURE

GETMKS    CAF      ZERO      # INITIALIZE MARK ID REGISTER AND MARK CNT

```

	TS	XYMARK	
	TS	MARKCNTR	
	CAF	LOW9	# ZERO BITS10 TO 15 RETAINING MKVAC ADR
	MASK	MARKSTAT	
	TS	MARKSTAT	
PASTIT	CAF	MKVB54*	# DISPLAY VB54 INITIALLY
	TC	BANKCALL	
	CADR	GOMARK4	
	TCF	KILLAOT	# V34 -- DOES GOTOPOOH
	TCF	MARKCHEX	# VB33 -- PROCEED, GOT MARKS, COMPUTE LOS
	TCF	GETDAT	# ENTER -- RECYCLE TO V01N71
MARKCHEX	CS	MARKSTAT	# SET BIT12 TO DISCOURAGE MARKRUPT
	MASK	BIT12	
	ADS	MARKSTAT	
	MASK	LOW9	
	TS	XYMARK	# JAM MARK VAC ADR IN XYMARK FOR AVESTAR
	CAF	ZERO	
	TS	MKDEX	# SET MKDEX ZERO FOR LOS VEC CNTR
	CA	MARKSTAT	
	MASK	PRI03	# SEE IF LAST MK PART COMPLETE
	TS	L	
	CAF	PRI03	# BITS10 AND 11
	EXTEND		
	RXOR	LCHAN	
	EXTEND		
	BZF	AVESTAR	# LAST PAIR COMPLETE -- TO COMPUTE LOS
CNTCHK	CCS	MARKCNTR	# NO PAIR SHOWING -- SEE IF PAIR IN HOLD
	TCF	+2	# PAIR BURIED -- DECREMENT COUNTER
	TCF	MKALARM	# NO PAIR -- ALARM
	TS	MARKCNTR	# STORE DECREMENTED COUNTER
AVESTAR	CAF	BIT12	# INITIALIZE MKDEX FOR STAR LOS COUNTER
	ADS	MKDEX	# MKDEX WAS INITIALIZED ZERO IN MARKCHEX
	CS	MARKCNTR	
	EXTEND		
	MP	SIX	# GET C(L) = -6 MARKCNTR
	CS	XYMARK	
	AD	L	# ADD -- MARK VAC ADR SET IN MARKCHEX
	INDEX	FIXLOC	
	TS	X1	# JAM -- CDU ADR OF X-MARK IN X1
	CA	FIXLOC	# SET PD POINTER TO ZERO
	TS	PUSHLOC	

```

# Page 253
TC      INTERPRET

BON      VLOAD*
          SURFFLAG      # IF ON SURFACE COMPUTE VEC AT SURFSTAR
          SURFSTAR
          1,1            # PUT Y-MARK CDUS IN CDUSPOT FOR TRG*NBSM
STOVL    CDUSPOT
          12D            # LOAD Y-PLANE VECTOR IN NG
CALL
          TRG*NBSM      # CONVERT IT TO STABLE MEMBER
PUSH     VLOAD*
          0,1            # PUT X-MARK CDUS IN CDUSPOT FOR TRG*NBSM
STOVL    CDUSPOT
          18D            # LOAD X-PLANE VECTOR IN NB
CALL
          TRG*NBSM      # CONVERT IT TO STABLE-MEMBER
          UNIT           # UNIT(XPSM * YPSM)
          STADR
          STORE          24D

AVEIT    SLOAD          PDVL      # N(NUMBER OF VECs) IN 0-1
          MKDEX
          24D            # LOAD CURRENT VECTOR
          VSR3           V/SC
          0
          STODL          24D      # VEC/N
          0
          DSU            DDV
          DP1/8          # (N-1)/N
          VXSC           VAD
          STARAD +6      # ADD VEC TO PREVIOUSLY AVERAGED VECTOR
          24D            # (N-1)/N AVESTVEC + VEC/N
          STORE          STARAD +6 # AVERAGE STAR VECTOR
          STORE          STARSAV2
          EXIT
          CCS            MARKCNTR  # SEE IF ANOTHER MARK PAIR IN MKVAC
          TCF            AVESTAR -1 # THERE IS -- GO GET IT -- DECREMENT COUNTER
          CAF            FIVE      # NO MORE MARKS -- TERMINATE AOTMARK
          INHINT
          TC             WAITLIST
          EBANK=         XYMARK
          2CADR          MKRELEAS

          TC             ENDMARK

ENDMARKS

MKALARM  TC             ALARM      # NOT A PAIR TO PROCESS -- DO GETMKS

```

July 12, 2016

151

OCT 111
TCF GETMKS

V01N71 VN 171
V06N87* VN 687

Page 254

MARKRUPT IS ENTERED FROM INTERRUPT LEAD-INS AND PROCESSES CHANNEL 16

CAUSED BY X,Y MARK OR MARK REJECT OR BY THE RATE OF DESCENT SWITCH

```
MARKRUPT      TS      BANKRUPT
               CA      CDUY          # STORE CDUS AND TIME NOW -- THEN SEE IF
               TS      ITEMP3       # WE NEED THEM
               CA      CDUZ
               TS      ITEMP4
               CA      CDUX
               TS      ITEMP5
EXTEND
DCA      TIME2
DXCH     ITEMP1
XCH      Q
TS      QRUPT

CAF      OCT34          # SEE IF X OR Y MARK OR MKREJECT
EXTEND
RAND     NAVKEYIN
CCS      A
TCF      +2            # ITS A LIVE ONE -- SEE IF ITS WANTED
TCF      SOMEKEY       # ITS SOME OTHER KEY

CAF      BIT12          # ARE WE ASKING FOR A MARK
MASK     MARKSTAT
CCS      A
TC       RESUME        # DON'T WANT MARK OR MKREJECT -- DO NOTHING

CCS      MARKSTAT      # ARE MARKS BEING ACCEPTED
TCF      FINDKEY       # THEY ARE -- WHICH ONE IS IT
TC       ALARM         # MARKS NOT BEING ACCEPTED -- DO ALARM
OCT      112
TC       RESUME

FINDKEY     CAF      BIT5          # SEE IF MARK REJECT.
EXTEND
RAND     NAVKEYIN
CCS      A
TCF      MKREJ         # IT'S A MARK REJECT
```

	CAF	BIT4	# SEE IF Y MARK
	EXTEND		
	RAND	NAVKEYIN	
	CCS	A	
	TCF	YMKRUPT	# IT'S A Y MARK
	CAF	BIT3	# SEE IF X MARK
	EXTEND		
	RAND	NAVKEYIN	
# Page 255	CCS	A	
	TCF	XMKRUPT	# IT'S A X MARK
SOMEKEY	CAF	OCT140	# NOT MARK OR MKREJECT -- SEE IF DESCENT BITS
	EXTEND		
	RAND	NAVKEYIN	
	EXTEND		
	BZF	+3	# IF NO BITS
	TC	POSTJUMP	# IF DESCENT BITS
	CADR	DESCBITS	
	TC	ALARM	# NO INBITS IN CHANNEL 16.
	OCT	113	
	TC	RESUME	
XMKRUPT	CAF	ZERO	
	TS	RUPTREG1	# SET X MARK STORE INDEX TO ZERO
	CAF	BIT10	
	TCF	+4	
YMKRUPT	CAF	ONE	
	TS	RUPTREG1	# SET Y MARK STORE INDEX TO ONE
	CAF	BIT11	
	TS	XYMARK	# SET MARK IDENTIFICATION
	TC	MARKTYPE	# SEE IF SURFACE MARK
	TCF	SURFSTOR	# SURFACE MARK -- JUST STORE CDUS
	CAF	BIT14	# GOT A MARK -- SEE IF MARK PARI MADE
	MASK	MARKSTAT	
	EXTEND		
	BZF	VERIFYMK	# NOT A PAIR, NORMAL PROCEDURE

July 12, 2016

153

	CS	MARKCNTR	# GO A PAIR, SEE IF ANOTHER CAN BE MADE
	AD	FOUR	# IF SO, INCREMENT POINTER, CLEAR BITS 10,11
	EXTEND		
	BZMF	5MKALARM	# HAVE FIVE MARK PAIRS -- DON'T ALLOW MARK
	INCR	MARKCNTR	# OK FOR ANOTHER PAIR, INCR POINTER
	CS	PRI023	# CLEAR BITS 10,11,14 FOR NEXT PAIR
	MASK	MARKSTAT	
	TS	MARKSTAT	
VERIFYMK	CA	XYMARK	
	MASK	MARKSTAT	
	CCS	A	
	TCF	+2	# THIS MARK NOT DESIRED
	TCF	VACSTOR	# MARK DESIRED -- STORE CDUS
	TC	ALARM	
	OCT	114	
	TC	RESUME	# RESUME -- DISPLAY UNCHANGED -- WAIT FOR ACTION
# Page 256			
5MKALARM	TC	ALARM	# ATTEMPTING TO MAKE MORE THAN 5 MK PAIRS
	OCT	107	
	TC	MARKTYPE	# SEE IF SURFACE MARK
	TCF	DSPV6N79	# IT IS
	TC	RESUME	# DON'T CHANGE DISPLAY -- DO NOTHING
# Page 257			
MKREJ	TC	MARKTYPE	# SEE IF SURFACE
	TCF	SURFREJ	# SURFACE -- JUST CHECK MARK COUNTER
	CAF	PRI03	# INFLIGHT -- SEE IF MARKS MADE
	MASK	MARKSTAT	
	CCS	A	
	TCF	REJECT	# MARKS MADE -- REJECT ONE
REJALM	TC	ALARM	# NO MARK TO REJECT -- BAD PROCEDURE -- ALARM
	OCT	115	
	TC	RESUME	# DESIRED ACTION DISPLAYED
REJECT	CS	PRI030	# ZERO BIT14, SHOW REJ., SEE IF MARK SINCE
	MASK	MARKSTAT	# LAST REJECT
	AD	BIT13	
	XCH	MARKSTAT	
	MASK	BIT13	
	CCS	A	
	TCF	REJECT2	# ANOTHER REJECT SET BIT 10+11 TO ZERO
	CS	XYMARK	# MARK MADE SINCE REJECT -- REJECT MARK IN 1D

RENEWMK	MASK	MARKSTAT	
	TS	MARKSTAT	
	TCF	REMARK	# GO REQUEST NEW MARK ACTION
REJECT2	CS	PRI03	# ON SECOND REJECT -- DISPLAY VB53 AGAIN
	TCF	RENEWMK	
SURFREJ	CCS	MARKCNTR	# IF MARK DECREMENT COUNTER
	TCF	+2	
	TCF	REJALM	# NO MARKS TO REJECT -- ALARM
	TS	MARKCNTR	
	TC	RESUME	
# Page 258			
# MARKTYPE TESTS TO SEE IF LEM ON LUNAR SURFACE. IF IT IS RETURN TO LOC+1			
MARKTYPE	CS	FLAGWRD8	# SURFFLAG ***** TEMPORARY *****
	MASK	BIT8	
	CCS	A	
	INCR	Q	# IF SURFACE MARK RETURN TO LOC +1
	TC	Q	# IF INFLIGHT MARK RETURN TO LOC +2
SURFSTOR	CAF	ZERO	# FOR SURFACE MARK ZERO MARK KIND INDEX
	TS	RUPTREG1	
	CS	MARKSTAT	# SET BITS10,11 TO SHOW SURFACE MARK
	MASK	PRI03	# FOR MARKCHEX
	ADS	MARKSTAT	
VACSTOR	CAF	LOW9	
	MASK	MARKSTAT	# STORE MARK VAC ADR IN RUPTREG2
	TS	RUPTREG2	
	EXTEND		
	DCA	ITEMP1	# PICK UP MARKTIME
	DXCH	TSIGHT	# STORE LAST MARK TIME
	CA	MARKCNTR	# 6 X MARKCNTR FOR STORE INDEX
	EXTEND		
	MP	SIX	
	XCH	L	# GET INDEX FROM LOW ORDER PART
	AD	RUPTREG2	# SET CDU STORE INDEX TO MARKVAC
	ADS	RUPTREG1	# INCREMENT VAC PICKUP BY MARK FOR FLIGHT
	TS	MKDEX	# STORE HERE IN CASE OF SURFACE MARK
	CA	ITEMP3	
	INDEX	RUPTREG1	
	TS	0	# STORE CDUY
	CA	ITEMP4	

July 12, 2016

155

```
INDEX  RUPTREG1
TS      2          # STORE CDUZ
CA      ITEMP5
INDEX  RUPTREG1
TS      4          # STORE CDUX
TC      MARKTYPE   # IF SURFACE MARK -- JUST DO SURFJOB
TCF     SURFJOB

CAF     BIT13      # CLEAR BIT13 TO SHOW MARK MADE
AD      XYMARK     # SET MARK ID IN MARKSTAT
COM
MASK    MARKSTAT
AD      XYMARK
TS      MARKSTAT
MASK    PRI03      # SEE IF X, Y MARK MADE
TS      L

# Page 259
CA      PRI03
EXTEND
RXOR    LCHAN
CCS     A
TCF     REMARK     # NOT PAIR YET, DISPLAY MARK ACTION
CS      MARKSTAT   # MARK PAIR COMPLETE -- SET BIT14
MASK    BIT14
ADS     MARKSTAT
TCF     REMARK     # GO DISPLAY V54

# Page 260
REMARK  CAF      PRI03      # BITS 10 AND 11
        MASK     MARKSTAT
        EXTEND
        MP      BIT6      # SHIFT MARK IDS TO BE 0 TO 3 FOR INDEX
        TS      MKDEX     # STORE VERB INDEX
SURFJOB CAF      PRI015
        TC      NOVAC     # ENTER JOB TO CHANGE DISPLAY TO
        EBANK=  XYMARK     # REQUEST NEXT ACTION
        2CADR   CHANGEVB

        TC      RESUME

CHANGEVB TC      MARKTYPE
        TCF     DSPV6N79   # SURFACE -- DISPLAY V 06 N 79
        INDEX  MKDEX     # INFLIGHT -- PICK UP MARK VB INDEX
        CAF     MKVB54
        TC      PASTIT    # PASTE UP NEXT MK VERB DISPLAY
```

THE FOUR MKVBS ARE INDEXED -- THEIR ORDER CANNOT BE CHANGED

MKVB54	VN	5471	# MAKE X OR Y MARK
MKVB53	VN	5371	# MAKE Y MARK
MKVB52	VN	5271	# MAKE X MARK
MKVB54*	VN	5471	# MAKE X OR Y MARK
DP1/8	2DEC	.125	

OCT34	OCT	34
V06N71	VN	671
V06N79*	VN	679

Page 261

ROUTINE TO REQUEST CURSOR AND SPIRAL MEASUREMENTS

COUNT* \$\$/R59

DSPV6N79	CAF	V06N79*	# CURSOR -- SPIRAL DISPLAY
	TC	BANKCALL	
	CADR	GOMARKF	
	TCF	KILLAOT	# V34 -- DOES GOTOPOOH
	TCF	SURFEND	# V33 -- PROCEED, END MARKING
	CAF	BIT6	# IF V32(OCT40) IN MPAC DO RECYCLE
	MASK	MPAC	# OTHERWISE IT IS LOAD VB ENTER SO
	CCS	A	# RE-DISPLAY V06N79
	TCF	SURFAGAN	# VB32 -- RECYCLE
	TCF	DSPV6N79	# ENTER
SURFEND	CS	BIT14	# SET BIT14 TO SHOW MARK END
	MASK	MARKSTAT	
	AD	BIT14	
	TS	MARKSTAT	
SURFAGAN	CA	CURSOR	
	INDEX	MKDEX	# HOLDS VAC AREA POINTER FOR SURF MARKING
	TS	1	# STORE CURSOR SP 2COMP
	CA	SPIRAL	
	INDEX	MKDEX	
	TS	3	# STORE SPIRAL
	CS	MARKSTAT	# IF BIT 14 SET -- END MARKING
	MASK	BIT14	
	EXTEND		
	BZF	MARKCHEX	
	CA	MARKCNTR	# THIS IS RECYCLE -- SEE IF 5 MARKS ALREADY

July 12, 2016

157

```
AD      ONE
COM
AD      FIVE
EXTEND
BZMF    5MKALARM      # CAN'T RECYCLE -- TOO MANY MARKS -- ALARM
INCR    MARKCNTR      # OF FOR RECYCLE -- INCR COUNTER
TCF     GETMKS +3      # GO DISPLAY MARK VB
```

This code is written to file `src/AOTMARK.s`.

B.10 ASCENT GUIDANCE

```

158  <src/ASCENT-GUIDANCE.s 158>≡
      # Copyright:    Public domain.
      # Filename:     ASCENT_GUIDNCE.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Hartmuth Gutsche <hgutsche@explornet.com>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         843-856
      # Mod history:   2009-05-23 HG    Transcribed from page images.
      #               2009-06-05 RSB    Fixed a couple of typos.
      #               2009-06-07 RSB    Corrected a typo.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969
      #
      # Page 843
      BANK      34
      SETLOC    ASCFILT
      BANK
      EBANK=    DVCNTR
      COUNT*    $$/ASENT
      ATMAG     TC      PHASCHNG
               OCT      00035
               TC      INTPRET
               BON
               FLRCS
               ASCENT

```

OCT	ASCENT 4
BANK	30
SETLOC	ASENT
BANK	
COUNT*	\$\$/ASENT

ASCENT	VLOAD	ABVAL	
		R	
	STOVL	/R/MAG	
		ZAXIS1	
	DOT	SL1	
		V	# Z.V = ZDOT*2(-8).
	STOVL	ZDOT	# ZDOT*2(-7)
		ZAXIS1	
	VXV	VSL1	
		UNIT/R/	# Z X UR = LAXIS*2(-2)
	STORE	LAXIS	# LAXIS*2(-1)
	DOT	SL1	
		V	# L.V = YDOT*2(-8).
	STCALL	YDOT	# YDOT * 2(-7)
		YCOMP	
	VLOAD		
		GDT1/2	# LOAD GDT1/2*2(-7) M/CS.
	V/SC	DOT	
		2SEC(18)	
		UNIT/R/	# G.UR*2(9) = GR*2(9).
	PDLV	VXV	# STORE IN PDL(0)
		UNIT/R/	# LOAD UNIT/R/ *2(-1)
		V	# UR*2(-1) X V*2(-7) = H/R*2(-8).
	VSQ	DDV	# H(2)/R(2)*2(-16).
		/R/MAG	# H(2)/R(3)*2(9).
	SL1	DAD	
	STADR		
	STODL	GEFF	# GEFF*2(10)m/CS/CS.
		ZDOTD	
	DSU		
		ZDOT	
	STORE	DZDOT	# DZDOT = (ZDOTD - ZDOT) * 2(7) M/CS.
	VXSC	PDDL	
		ZAXIS1	
		YDOTD	
	DSU		
		YDOT	
	STORE	DYDOT	# DYDOT = (YDOTD - YDOT) *2(7) M/CS.
	VXSC	PDDL	
		LAXIS	
		RDOTD	
# Page 846	DSU		
		RDOT	
	STORE	DRDOT	# DRDOT = (RDOTD - RDOT) * 2(7) M/CS.
	VXSC	VAD	


```

UNIT/R/
VAD      VSL1
STADR
STORE    VGVECT      # VG = (DRDOT)R + (DVDOT)L + (DZDOT)Z.
DLOAD    DMP          # LOAD TGO
          TGO          # TGO GEFF
          GEFF
VXSC      VSL1
          UNIT/R/      # TGO GEFF UR
BVSU
          VGVECT      # COMPENSATED FOR GEFF
STORE    VGVECT      # STORE FOR DOWNLINK
MXV      VSL1        # GET VGBODY FOR N85 DISPLAY
          XNBPIP
STOVL     VGBODY
          VGVECT
ABVAL     BOFF        # MAGNITUDE OF VGVECT
          FLRCS        # IF FLRCS=0,DO NORMAL GUIDANCE
          MAINENG
DDV
          AT/RCS
STCALL    TGO        # THIS WILL BE USED ON NEXT CYCLE
          ASCTERM2
MAINENG   DDV         # VG/VE IN PDL(0) (2)
          VE
DMP       BDSU        # 1 - KT VG/VE
          KT1
          NEARONE
DMP       DMP          # TBUP VG(1-KT VG/VE)/VE (0)
          TBUP        # = TGO
DSU
          # COMPENSATE FOR TAILOFF
          TTO
STORE     TGO
SR        DCOMP
          11D
STODL     TTOGO      # TGO *2(-28) CS
          TGO
BON       DSU
          IDLEFLAG
          T2TEST
          4SEC(17)    # ( TGO - 4 ) *2(-17) CS.
BMN
          ENGOFF
T2TEST   DLOAD
          TGO
DSU       BMN        # IF TGO - T2 NEG., GO TO CMPONENT

```

Page 847

```

                                T2A
                                CMPOONENT
                                DSU
DLOAD                          TBUP
                                TGO
DDV                             CALL      # 1- TGO/TBUP
                                TBUP
                                LOGSUB
SL                              PUSH      # -L IN PDL(0)          (2)
                                5
BDDV                           BDSU      # -TGO/L*2(-17)
                                TGO
                                TBUP      # TBUP + TGO/L = D12*2(-17)
PUSH                           BON        # STORE IN PDL(2)          (4)
                                FLPC      # IF FLPC = 1, GO TO CONST
                                NORATES
DLOAD                          DSU
                                TGO
                                T3
BPL                             SET       # FLPC=1
                                RATES
                                FLPC
NORATES                        DLOAD
                                HI6ZEROS
STORE                          PRATE     # B = 0
STORE                          YRATE     # D = 0
GOTO
RATES                          DLOAD
                                CONST     # GO TO CONST
                                DSU
                                TGO
                                O2D      # TGO - D12 = D21*2(-17)
PUSH                           SL1       # IN PDL(4)          (6)
BDSU                           SL3      # (1/2TGO - D21)*2(-13) = E * 2(-13)
                                TGO      #          (8)
PDDL                           DMP       # IN PDL(6)
                                TGO
                                RDOT     # RDOT TGO * 2(-24)
DAD                            DSU       # R + RDOT TGO
                                /R/MAG   # R + RDOT TGO - RCO
                                RCO      # MPAC = -DR *2(-24).
PDDL                           DMP       # -DR IN PDL(8)          (10)
                                DRDOT
                                O4D      # D21 DRDOT*2(-24)
DAD                            SL2      # (D21 DRDOT-DR)*2(-22)   (8)
DDV                            DDV

```

```

                                06D                # (D21 DRDOT-DR)/E*2(-9)
                                TGO
STORE    PRATE                # B * 2(8)
BMN      DLOAD                # B>0 NOT PERMITTED
                                CHKBMAG

#Page 848

                                HI6ZEROS
STCALL   PRATE
                                PROK
CHKBMAG   SR4    DDV            # B*2(4)
                                TBUP        # (B / TAU) * 2(21)
DSU       BPL
                                PRLIMIT      # ( B / TAU ) = 2(21) MAX.
                                PROK
DLOAD     DMP
                                PRLIMIT
                                TBUP        # B MAX. * 2(4)
SL4        # BMAX*2(8)
STORE      PRATE
PROK       DLOAD

                                TGO
DMP        DAD                # YDOT TGO
                                YDOT
Y           # Y + YDOT TGO
DSU        PDDL              # Y + YDOT TGO - YCO
                                YCO          # MPAC = - DY*(-24.) IN PDL(8) (10)
                                DYDOT
DMP        DAD                # D21 DYDOT - DY (8)
                                04D
SL2        DDV                # (D21 DYDOT - DY)/E*2(-9)
DDV        SETPD              # (D21 DYDOT - DY)/E TGO*2(8)
                                TGO          # = D*2(8)
                                04
STORE      YRATE
CONST     DLOAD    DMP        # LOAD B*2(8)
                                PRATE        # B D12*2(-9)
                                02D
PDDL      DDV                # D12 B IN PDL(4) (6)
                                DRDOT        # LOAD DRDOT*2(-7)
                                00D          # -DRDOT/L*2(-7)
SR2        DSU                # (-DRDOT/L-D12 B)=A*2(-9) (4)
STADR
STODL     PCONS
                                YRATE        # D*2(8)
DMP        PDDL              # D12 D,EXCH WITH -L IN PDL(0) (2,2)
BDDV      SR2                # -DYDOT/L*2(-9)

```

Page 849

	DSU	DYDOT	# $(-DYDOT/L-D12\ D)=C*2(-9)$	
		OOD		
CMponent	STORE	YCONS		
	SETPD	DLOAD		
		OOD		
		100CS		
	DMP			
		PRATE	# $B(T-T0)*2(-9)$	
	DAD	DDV	# $(A+B(T-T0))*2(-9)$	
		PCONS	# $(A+B(T-T0))/TBUP*2(8)$	
		TBUP		
	SL1	DSU		
		GEFF	# $ATR*2(9)$	
	STODL	ATR		
		100CS		
	DMP	DAD		
		YRATE		
		YCONS	# $(C+D(T-T0))*2(-9)$	
	DDV	SL1		
		TBUP		
	STORE	ATY	# $ATY*2(9)$	
	VXSC	PDDL	# $ATY\ UY*2(8)$	(6)
		LAXIS		
		ATR		
	VXSC	VAD		
		UNIT/R/		
	VSL1	PUSH	# $AH*2(9)\ IN\ PDL(0)$	(6)
	ABVAL	PDDL	# $AH(2)\ IN\ PDL(34)$	
		AT	# $AHMAG\ IN\ PDL(6)$	(8)
	DSQ	DSU	# $(AT(2)-AH(2))*2(18)$	
		34D	# $=ATP2*2(18)$	
	PDDL	PUSH	#	(12)
		AT		
	DSQ	DSU	# $(AT(2)KR(2)-AH(2))*2(18)$	(10)
		34D	# $=ATP3*2(18)$	
	BMN	DLOAD	# $IF\ ATP3\ NEG,GO\ TO\ NO-ATP$	
		NO-ATP	# $LOAD\ ATP2,\ IF\ ATP3\ POS$	
		8D		
	SQRT	GOTO	# $ATP*2(9)$	
		AIMER		
NO-ATP	DLOAD	BDDV	# $KR\ AT/AH = KH$	(8)
		6D		
	VXSC		# $KH\ AG*2(9)$	
		OOD		

July 12, 2016

165

	STODL	OOD	# STORE NEW AH IN PDL(0)
		HI6ZEROS	
AIMER	SIGN		
		DZDOT	
	STORE	ATP	
	VXSC		
		ZAXIS1	# ATP ZAXIS *2(8).
	VSL1	VAD	# AT*2(0)
		OOD	
	STORE	UNFC/2	# WILL BE OVERWRITTEN IF IN VERT. RISE.
	SETPD	BON	
		OOD	
		FLPI	
		P12RET	
	BON		
# Page 850			
		FLVR	
		CHECKALT	
MAINLINE	VLOAD	VCOMP	
		UNIT/R/	
	STODL	UNWC/2	
		TXO	
	DSU	BPL	
		PIPTIME	
		ASCTERM	
	BON		
		ROTFLAG	
		ANG1CHEK	
CLRFLAG	CLEAR	CLEAR	
		NOR29FLG	# START r29 IN ASCENT PHASE.
		XOVINFLG	# ALLOW X-AXIS OVERRIDE
ASCTERM	EXIT		
	CA	FLAGWRD9	
	MASK	FLRCSBIT	
	CCS	A	
	TCF	ASCTERM3	
	TC	INTPRET	
	CALL		
		FINDCDUW -2	
ASCTERM1	EXIT		
+1	CA	FLAGWRD9	# INSURE THAT THE NOUN 63 DISPLAY IS
	MASK	FLRCSBIT	# BYPASSED IF WE ARE IN THE RCS TRIMMING
	CCS	A	# MODE OF OPERATION
	TCF	ASCTERM3	
	CA	FLAGWRD8	# BYPASS DISPLAYS IF ENGINE FAILURE IS
	MASK	FLUNDBIT	# INDICATED.

	CCS	A	
	TCF	ASCTERM3	
	CAF	V06N63*	
	TC	BANKCALL	
	CADR	GODSPR	
	TCF	ASCTERM3	
ASCTERM2	EXIT		
ASCTERM3	TCF	ENDOFJOB	
ASCTERM4	EXIT		
	INHINT		
	TC	IBNKCALL	# NO GUIDANCE THIS CYCLE -- HENCE ZERO
	CADR	ZATTEROR	# THE DAP COMMANDED ERRORSSs.
	TCF	ASCTERM1 +1	
CHECKALT	DLOAD	DSU	
		/R/MAG	
		/LAND/	
	DSU	BMN	# IF H LT 25K CHECK Z AXIS ORIENTATION
		25KFT	
		CHECKYAW	
# Page 851			
EXITVR	CLEAR	BON	
		FLVR	
		ROTFLAG	
		MAINLINE	
	DLOAD	DAD	
		PIPTIME	
		10SECS	
	STCALL	TXO	
		MAINLINE	
EXITVR1	CLRGO		
		ROTFLAG	
		EXITVR	
	SETLOC	ASENT1	
	BANK		
	COUNT*	\$\$/ASENT	
ANG1CHEK	VLOAD	DOT	
		UNFC/2	
		XNBPIP	
	DSU	BPL	
		COSTHET1	
		OFFROT	
	VLOAD	DOT	
		XNBPIP	

	DSU	UNIT/R/ BMN COSTHET2 KEEPVR1	
OFFROT	CLRGO	ROTFLAG CLRFLAG	
	BANK	7	
	SETLOC	ASENT2	
	BANK		
	COUNT*	\$\$/ASENT	
SETXFLAG	=	CHECKYAW	
CHECKYAW	SET		
		XOVINFLG	# PROHIBIT X-AXIS OVERRRIDE
	DLOAD	VXSC	
		ATY	
		LAXIS	
	PDDL	VXSC	
		ATP	
		ZAXIS1	
	VAD	UNIT	
	PUSH	DOT	
# Page 852			
		YNBPIP	
	ABS	DSU	
		SIN5DEG	
	BPL	DLOAD	
		KEEPVR	
		RDOT	
	DSU	BPL	
		40FPS	
		EXITVR1	
	GOTO		
		KEEPVR	
	BANK	5	
	SETLOC	ASENT3	
	BANK		
	COUNT*	\$\$/ASENT	
SIN5DEG	2DEC	0.08716	B-2
40FPS	2DEC	0.12192	B-7

	BANK	14	
	SETLOC	ASENT4	
	BANK		
	COUNT*	\$\$/ASENT	
KEEPVR	VLOAD	STADR	# RECALL LOSVEC FROM PUSHLIST
	STORE	UNWC/2	
KEEPVR1	VLOAD		
		UNIT/R/	
	STCALL	UNFC/2	
		ASCTERM	
ENGOFF	RTB		
		LOADTIME	
	DSU	DAD	
		PIPTIME	
		TTOGO	
	DCOMP	EXIT	
	TC	TPAGREE	# FORCH SIGN AGREEMENT ON MPAC, MPAC +1.
	CAF	EBANK7	
	TS	EBANK	
	EBANK=	TGO	
	INHINT		
	CCS	MPAC +1	
	TCF	+3	# C(A) = DT - 1 BIT
	TCF	+2	# C(A) = 0
	CAF	ZERO	# C(A) = 0
	AD	BIT1	# C(A) = 1 BIT OR DT.
# Page 853	TS	ENGOFFDT	
	TC	TWIDDLE	
	ADRES	ENGOFF1	
	TC	PHASCHNG	
	OCT	47014	
	-GENADR	ENGOFFDT	
	EBANK=	TGO	
	2CADR	ENGOFF1	
	TC	INTPRET	
	SET	GOTO	
		IDLEFLAG	# DISABLE DELTA-V MONITOR
		T2TEST	
ENGOFF1	TC	IBNKCALL	# SHUT OFF THE ENGINE.
	CADR	ENGNOF2	

July 12, 2016

169

	CAF	PRI017	# SET UP A JOB FOR THE ASCENT GUIDANCE
	TC	FINDVAC	# POSTBURN LOGIC.
	EBANK=	WHICH	
	2CADR	CUTOFF	
	TC	PHASCHNG	
	OCT	07024	
	OCT	17000	
	EBANK=	TGO	
	2CADR	CUTOFF	
	TCF	TASKOVER	
CUTOFF	TC	UPFLAG	# SET FLRCS FLAG.
	ADRES	FLRCS	
-5	CAF	V16N63	
	TC	BANKCALL	
	CADR	GOFLASH	
	TCF	+3	
	TCF	CUTOFF1	
	TCF	-5	
+3	TC	POSTJUMP	
	CADR	TERMASC	
CUTOFF1	INHINT		
	TC	IBNKCALL	# ZERO ATTITUDE ERRORS BEFORE REDUCINT DB.
	CADR	ZATTEROR	
	TC	IBNKCALL	
	CADR	SETMINDB	
	TC	POSTJUMP	
	CADR	CUTOFF2	
# Page 854			
V16N63	VN	1663	
	BANK	30	
	SETLOC	ASENT5	
	BANK		
	COUNT*	\$\$/ASENT	
CUTOFF2	TC	PHASCHNG	
	OCT	04024	
	CAF	V16N85C	
	TC	BANKCALL	

	CADR	GOFLASH	
	TCF	TERMASC	
	TCF	+2	# PROCEED
	TCF	CUTOFF2	
TERMASC	TC	PHASCHNG	
	OCT	04024	
	INHINT		# RESTORE DEADBAND DESIRED BY ASTRONAUT.
	TC	IBNKCALL	
	CADR	RESTORDB	
	TC	DOWNFLAG	# DISALLOW ABORTS AT THIS TIME.
	ADRES	LETABORT	
	TCF	GOTOPOOH	
V16N85C	VN	1685	
	BANK	27	
	SETLOC	ASENT1	
	BANK		
	COUNT*	\$\$/ASENT	
YCOMP	VLOAD	DOT	
		UNIT/R/	
		QAXIS	
	SL2	DMP	
		RCO	
	STORE	Y	
	RVQ		
	BANK	30	
	SETLOC	ASENT	
	BANK		
# Page 855			
100CS	EQUALS	2SEC(18)	
T2A	EQUALS	2SEC(17)	
4SEC(17)	2DEC	400 B-17	
2SEC(17)	2DEC	200 B-17	
T3	2DEC	1000 B-17	
6SEC(18)	2DEC	600 B-18	
BIT4H	OCT	10	
2SEC(9)	2DEC	200 B-9	
V06N63*	VN	0663	
V06N76	VN	0676	
V06N33A	VN	0633	

```

      BANK      33
      SETLOC    ASENT6
      BANK
      COUNT*    $$/ASENT

```

```

KT1          2DEC      0.5000
PRLIMIT      2DEC      -.0639      # (B/TBUP)MIN=-.1FT.SEC(-3)
MINABDV      2DEC      .0356 B-5    # 10 PERCENT BIGGER THAN GRAVITY
1/DVO        =          MASS1

```

```
# Page 856
```

```
# THE LOGARITHM SUBROUTINE
```

```

      BANK      24
      SETLOC    FLOGSUB
      BANK

```

```

# INPUT ..... X IN MPAC
# OUTPUT ..... -LOG(X) IN MPAC

```

```

LOGSUB      NORM      BDSU
                        MPAC +6
                        NEARONE
EXIT
TC          POLY
DEC         6
2DEC        .0000000060
2DEC        -.0312514377
2DEC        -.0155686771
2DEC        -.0112502068
2DEC        -.0018545108
2DEC        -.0286607906
2DEC        .0385598563
2DEC        -.0419361902

CAF         ZERO
TS          MPAC +2
EXTEND
DCA         CLOG2/32
DXCH        MPAC
DXCH        BUF +1
CA          MPAC +6
TC          SHORTMP
DXCH        MPAC +1
DXCH        MPAC
DXCH        BUF +1

```

DAS	MPAC
TC	INTPRET
DCOMP	RVQ

CLOG2/32	2DEC	.0216608494
----------	------	-------------

This code is written to file `src/ASCENT-GUIDANCE.s`.

B.11 ASSEMBLY AND OPERATION INFORMATION

```

173  <src/ASSEMBLY-AND-OPERATION-INFORMATION.s 173>≡
      # Copyright:    Public domain.
      # Filename:     ASSEMBLY_AND_OPERATION_INFORMATION.agc
      # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
      #               It is part of the source code for the Command Module's (CM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Mod history:   2009-05-05 RSB  Adapted from the Colossus249/ file of the
      #               same name, using Comanche055 page images.
      #
      # This source code has been transcribed or otherwise adapted from digitized
      # images of a hardcopy from the MIT Museum. The digitization was performed
      # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
      # thanks to both. The images (with suitable reduction in storage size and
      # consequent reduction in image quality as well) are available online at
      # www.ibiblio.org/apollo. If for some reason you find that the images are
      # illegible, contact me at info@sandroid.org about getting access to the
      # (much) higher-quality images which Paul actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 055 of AGC program Comanche by NASA
      #       2021113-051. 10:28 APR. 1, 1969
      #
      #       This AGC program shall also be referred to as
      #               Colossus 2A

      # Page 2

      <Page 2 18>

      # Page 3

      <Page 3 20>

      # Page 4

      <Page 4 21>

      # Page 5

```

⟨Page 5 22⟩

Page 6

⟨Page 6 23⟩

Page 7

⟨Page 7 24⟩

Page 8

⟨Page 8 25⟩

Page 9

⟨Page 9 26a⟩

Page 10

⟨Page 10 26b⟩

Page 11

⟨Page 11 27⟩

Page 12

⟨Page 12 29⟩

Page 13

⟨Page 13 30⟩

Page 14

⟨Page 14 31⟩

Page 15

⟨Page 15 32⟩

Page 16

⟨Page 16 34⟩

Page 17

⟨Page 17 36⟩

Page 18

⟨Page 18 37⟩

Page 19

⟨Page 19 38⟩

Page 20

⟨Page 20 40⟩

Page 21

⟨Page 21 42⟩

Page 22

⟨Page 22 43⟩

Page 23

⟨Page 23 44⟩

Page 24

⟨Page 24 46⟩

Page 25

⟨Page 25 47⟩

Page 26

⟨Page 26 48a⟩

This code is written to file `src/ASSEMBLY-AND-OPERATION-INFORMATION.s`.

B.12 ATTITUDE MANEUVER ROUTINE

```

176  <src/ATTITUDE-MANEUVER-ROUTINE.s 176>≡
      # Copyright:    Public domain.
      # Filename:     ATTITUDE_MANEUVER_ROUTINE.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         342-363
      # Mod history:   2009-05-16 RSB   Adapted from the corresponding
      #               Luminary131 file, using page
      #               images from Luminary 1A.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969
      #
      # Page 342
      # BLOCK 2 LGC ATTITUDE MANEUVER ROUTINE -- KALCMANU
      #
      # MOD 2          DATE 5/1/67          BY DON KEENE
      #
      # PROGRAM DESCRIPTION
      #
      # KALCMANU IS A ROUTINE WHICH GENERATES COMMANDS FOR THE LM DAP TO CHANGE THE ATTITUDE
      # DURING FREE FALL. IT IS DESIGNED TO MANEUVER THE SPACECRAFT FROM ITS INITIAL ORIENT
      # ORIENTATION SPECIFIED BY THE PROGRAM WHICH CALLS KALCMANU, AVOIDING GIMBAL LOCK IN
      # MOD 2 VERSION, THIS DESIRED ATTITUDE IS SPECIFIED BY A SET OF OF THREE COMMANDED CI
      # SINGLE PRECISION ANGLES IN THE THREE CONSECUTIVE LOCATIONS, CPHI, CTHETA, CPSI, WHI
      #
      #       CPHI = COMMANDED OUTER GIMBAL ANGLE
      #       CTHETA = COMMANDED INNER GIMBAL ANGLE

```



```

#           CPSI = COMMANDED MIDDLE GIMBAL ANGLE
#
# WHEN POINTING A SPACECRAFT AXIS (I.E., X, Y, Z, THE AOT, THRUST AXIS, ETC.) THE SUBROUTINE VEC
# USED TO GENERATE THIS SET OF DESIRED CDU ANGLES (SEE DESCRIPTION IN R60).
#
# WITH THIS INFORMATION KALCMANU DETERMINES THE DIRECTION OF THE SINGLE EQUIVALENCE ROTATION (COF)
# MAGNITUDE OF THE ROTATION (AM) TO BRING THE S/C FROM ITS INITIAL ORIENTATION TO ITS FINAL ORIENTA
# THIS DIRECTION REMAINS FIXED BOTH IN INERTIAL COORDINATES AND IN COMMANDED S/C AXES THROUGHOUT
#
# MANEUVER.  ONCE COF AND AM HAVE BEEN DETERMINED, KALCMANU THEN EXAMINES THE MANEUVER TO SEE IF I
#
# THE S/C THROUGH GIMBAL LOCK.  IF SO, COF AND AM ARE READJUSTED SO THAT THE S/C WILL JUST SKIM
# LOCK ZONE AND ALIGN THE X-AXIS.  IN GENERAL A FINAL YAW ABOUT X WILL BE NECESSARY TO COMPLETE
# NEEDLESS TO SAY, NEITHER THE INITIAL NOR THE FINAL ORIENTATION CAN BE IN GIMBAL LOCK.
#
# FOR PROPER ATTITUDE CONTROL THE DIGITAL AUTOPILOT MUST BE GIVEN AN ATTITUDE REFERENCE WHICH I
# KALCMANU DOES THIS BY GENERATING A REFERENCE OF DESIRED GIMBAL ANGLES (CDUXD, CDUYD, CDUZD) W
# EVERY ONE SECOND DURING THE MANEUVER.  TO ACHIEVE A SMOOTHER SEQUENCE OF COMMANDS BETWEEN SUC
# THE PROGRAM ALSO GENERATES A SET OF INCREMENTAL CDU ANGLES (DELDCDU) TO BE ADDED TO CDU DESIR
# AUTOPILOT.  KALCMANU ALSO CALCULATES THE COMPONENT MANEUVER RATES (OMEGAPD, OMEGAQD, OMEGARD)
#
# BE DETERMINED SIMPLY BY MULTIPLYING COF BY SOME SCALAR (ARATE) CORRESPONDING TO THE DESIRED R
#
# AUTOMATIC MANEUVERS ARE TIMED WITH THE HELP OF WAITLIST SO THAT AFTER A SPECIFIED INTERVAL THE
# DESIRED RATES ARE SET TO ZERO AND THE DESIRED CDU ANGLES (CDUYD, CDUZD) ARE SET EQUAL TO THE
# ANGLES (CTHETA, CPSI).  IF ANY YAW REMAINS DUE TO GIMBAL LOCK AVOIDANCE, THE FINAL YAW MANEUVER
# CALCULATED AND THE DESIRED YAW RATE SET TO SOME FIXED VALUE (ROLLRATE = + OR - 2 DEGREES PER
# IN THIS CASE ONLY AN INCREMENTAL CDUX ANGLE (DELFROLL) IS SUPPLIED TO THE DAP.  AT THE END OF
# MANEUVER OR IN THE EVENT THAT THERE WAS NO FINAL YAW, CDUXD IS SET EQUAL TO CPHI AND THE X-AX
# RATE SET TO ZERO.  THUS, UPON COMPLETION OF THE MANEUVER THE S/C WILL FINISH UP IN A LIMIT CY
# DESIRED GIMBAL ANGLES.
#
# PROGRAM LOGIC FLOW
#
# KALCMANU IS CALLED AS A HIGH PRIORITY JOB WITH ENTRY POINTS AT KALCMAN3 AND VECPOINT.  IT FIR
# UP THE CURRENT CDU ANGLES TO BE USED AS THE BASIS FOR ALL COMPUTATIONS INVOLVING THE INITIAL
# Page 343
# IT THEN DETERMINES THE DIRECTION COSINE MATRICES RELATING BOTH THE INITIAL AND FINAL S/C ORIE
#
# * *
# MEMBER AXES (MIS,MFS).  IT ALSO COMPUTES THE MATRIX RELATING FINAL S/C AXES TO INITIAL S/C AX
# ANGLE OF ROTATION (AM) IS THEN EXTRACTED FROM THIS MATRIX, AND TEST ARE MADE TO DETERMINE IF
#
# A)      AM LESS THAN .25 DEGREES (MINANG)
# B)      AM GREATER THAN 170 DEGREES (MAXANG)
#
# IF AM IS LESS THAN .25 DEGREES, NO COMPLICATED AUTOMATIC MANEUVERING IS NECESSARY.  THEREFORE,

```

```

# SET CDU DESIRED EQUAL TO THE FINAL CDU DESIRED ANGLES AND TERMINATE THE JOB.
#
# IF AM IS GREATER THAN .25 DEGREES BUT LESS THAN 170 DEGREES THE AXES OF THE SINGLE
# - *
# (COF) IS EXTRACTED FROM THE SKEW SYMMETRIC COMPONENTS OF MFI.
#
# IF AM GREATER THAN 170 DEGREES AN ALTERNATE METHOD EMPLOYING THE SYMMETRIC PART OF
# -
# TO DETERMINE COF.
#
# THE PROGRAM THEN CHECKS TO SEE IF THE MANEUVER AS COMPUTED WILL BRING THE S/C THROU
# SO, A NEW MANEUVER IS CALCULATED WHICH WILL JUST SKIM THE GIMBAL LOCK ZONE AND ALIC
# METHOD ASSURES THAT THE ADDITIONAL MANEUVERING TO AVOID GIMBAL LOCK WILL BE KEPT TO
# P AXIS YAW WILL BE NECESSARY, A SWITCH IS RESET (STATE SWITCH 31) TO ALLOW FOR THE
# YAW.
#
# AS STATED PREVIOUSLY, KALCMANU GENERATES A SEQUENCE OF DESIRED GIMBAL ANGLES WHICH
#
# SECOND. THIS IS ACCOMPLISHED BY A SMALL ROTATION OF THE DESIRED S/C FRAME ABOUT TH
# DESIRED REFERENCE MATRIX IS THEN,
#
#      *      *      *
#      MIS      =      MIS      DEL
#      N+1      N
#
#      *
# WHERE DEL IS THE MATRIX CORRESPONDING TO THIS SMALL ROTATION. THE NEW CDU ANGLES C
#
#      *
# FROM MIS.
#
# AT THE BEGINNING OF THE MANEUVER THE AUTOPILOT DESIRED RATES (OMEGAPD, OMEGAQD, OME
# MANEUVER TIMINGS ARE ESTABLISHED. ON THE FIRST PASS AND ON ALL SUBSEQUENT UPDATES
# ANGLES ARE LOADED WITH THE APPROPRIATE VALUES AND THE INCREMENTAL CDU ANGLES ARE C
# (TIME1 AND TIME2) ARE THEN CHECKED TO SEE IF THE MANEUVER WILL TERMINATE BEFORE TH
# NOT, KALCMANU CALLS FOR ANOTHER UPDATE (RUN AS A JOB WITH PRIORITY TBD) IN ONE SECO
# CALLING SEQUENCE ARE AUTOMATICALLY COMPENSATED IN CALLING FOR THE NEXT UPDATE.
#
# IF IT IS FOUND THAT THE MANEUVER IS TO TERMINATE BEFORE THE NEXT UPDATE A ROUTINE
# LIST TASK) TO STOP THE MANEUVER AT THE APPROPRIATE TIME AS EXPLAINED ABOVE.

# Page 344
# CALLING SEQUENCE
#
# IN ORDER TO PERFORM A KALCMANU SUPERVISED MANEUVER, THE COMMANDED GIMBAL ANGLES MUS
# STORED IN LOCATIONS CPHI, CTHETA, CPSI. THE USER'S PROGRAM MUST THEN CLEAR STATE S
# ATTITUDE MANEUVER ROUTINE TO PERFORM ANY FINAL P-AXIS YAW INCURRED BY AVOIDING GIM
# THEN INITIATED BY ESTABLISHING THE FOLLOWING EXECUTIVE JOB
#
#      *
```

179

```
# CAF      Prio XX
#          --
#          INHINT
#          TC      FINDVAC
#          2CADR   KALCMAN3
#          RELINT
#
# THE USER'S PROGRAM MAY EITHER CONTINUE OR WAIT FOR THE TERMINATION OF THE MANEUVER. IF THE U
# WAIT, HE MAY PUT HIS JOB TO SLEEP WTH THE FOLLOWING INSTRUCTIONS:
#
#          L      TC      BANKCALL
#          L+1    CADR     ATTSTALL
#          L+2    (BAD RETURN)
#          L+3    (GOOD RETURN)
#
# UPON COMPLETION OF THE MANEUVER, THE PROGRAM WILL BE AWAKENED AT L+3 IF THE MANEUVER WAS COMP
# SUCCESSFULLY, OR AT L+2 IF THE MANEUVER WAS ABORTED. THIS ABORT WOULD OCCUR IF THE INITIAL C
# WAS IN GIMBAL LOCK.
#
# *** NOTA BENE *** IF IT IS ASSUMED THAT THE DESIRED MANEUVERING RATE (0.5, 2, 5, 10 DEG/SEC)
# KEYBOARD ENTRY PRIOR TO THE EXECUTION OF KALCMANU.
#
# IT IS ALSO ASSUMED THAT THE AUTOPILOT IS IN THE AUTO MODE. IF THE MODE SWITCH IS CHANGED DUR
# MANEUVER, KALCMANU WILL TERMINATE VIA GOODEND WITHIN 1 SECOND SO THAT R60 MAY REQUEST A TRIM
# SUBROUTINES.
#
# KALCMANU USES A NUMBER OF INTERPRETIVE SUBROUTINES WHICH MAY BE OF GENERAL INTEREST. SINCE T
# WERE PROGRAMMED EXCLUSIVELY FOR KALCMANU, THEY ARE NOT, AS YET, GENERALLY AVAILABLE FOR USE E
#
# MXM3
# ----
#
# THIS SUBROUTINE MULTIPLIES TWO 3X3 MATRICES AND LEAVES THE RESULT IN THE FIRST 18 LOCATIONS O
# DOWN LIST, I.E.,
#
#           [ M      M      M ]
#           [  0      1      2 ]
# *         [                ]
# M =       [ M      M      M ]   =   M1      X      M2
#           [  3      4      5 ]
#           [                ]
#           [ M      M      M ]
#           [  6      7      8 ]
#
# Page 345
#
#                                     *
# INDEX REGISTER X1 MUST BE LOADED WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M1, AND X2 M
#
#                                     *
```

```

# LOADED WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M2.  THE ROUTINE USES THE F
# DOWN LIST.  THE FIRST ELEMENT OF THE MATRIX APPEARS IN PDO.  PUSH UP FOR M .
#
# TRANSPOS
# -----
#
# THIS ROUTINE TRANSPOSES A 3X3 MATRIX AND LEAVES THE RESULT IN THE PUSH DOWN LIST,
#
#      *      * T
#      M      =      M1
#
# INDEX REGISTER X1 MUST CONTAIN THE COMPLEMENT OF THE STARTING ADDRESS FOR M1.  PUSH
#
#      *
# SEQUENT COMPONENTS OF M.  THIS SUBROUTINE ALSO USES THE FIRST 20 LOCATIONS OF THE
#
# CDU TO DCM
# -----
#
# THIS SUBROUTINE CONVERTS THREE CDU ANGLES IN T(MPAC) TO A DIRECTION COSINE MATRIX
# THE CORRESPONDING S/C ORIENTATIONS TO THE STABLE MEMBER FRAME.  THE FORMULAS FOR T
#
#      M      =      COSY COSZ
#      0
#
#      M      =      -COSY SINZ COSX + SINY SINX
#      1
#
#      M      =      COSY SINZ SINX + SINY COSX
#      2
#
#      M      =      SINZ
#      3
#
#      M      =      COSZ COSX
#      4
#
#      M      =      -COSZ SINX
#      5
#
#      M      =      -SINY COSZ
#      6
#
#      M      =      SINY SINZ COSX + COSY SINX
#      7
# Page 346
#      M      =      -SINY SINZ SINX + COSY COSX

```

```

#      8
#
# WHERE      X      =      OUTER GIMBAL ANGLE
#            Y      =      INNER GIMBAL ANGLE
#            Z      =      MIDDLE GIMBAL ANGLE
#
# THE INTERPRETATION OF THIS MATRIX IS AS FOLLOWS:
#
# IF A , A , A REPRESENT THE COMPONENTS OF A VECTOR IN S/C AXES THEN THE COMPONENTS OF THE SAME
#   X   Y   Z
# STABLE MEMBER AXES (B , B , B ) ARE
#           X   Y   Z
#
#   [ B ]           [ A ]
#   [ X ]           [ X ]
#   [   ]           [   ]
#   [ B ]           [ A ]
#   [ Y ]   =      *   [ Y ]
#   [   ]           [   ]
#   [ B ]           [ B ]
#   [ Z ]           [ Z ]
#
# THE SUBROUTINE WILL STORE THIS MATRIX IN SEQUENTIAL LOCATIONS OF ERASABLE MEMORY AS SPECIFIED
#
# PROGRAM. TO DO THIS THE CALLING PROGRAM MUST FIRST LOAD X2 WITH THE COMPLEMENT OF THE STARTING
#
# INTERNALLY, THE ROUTINE USES THE FIRST 16 LOCATIONS OF THE PUSH DOWN LIST, ALSO STEP REGISTER
# REGISTER X2.
#
# DCM TO CDU
# -----
#
# THIS ROUTINE EXTRACTS THE CDU ANGLES FROM A DIRECTION COSINE MATRIX (M SCALED BY 2) RELATING
#
# STABLE MEMBER AXES. X1 MUST CONTAIN THE COMPLEMENT OF THE STARTING ADDRESS FOR M. THE SUBROUTINE
# CORRESPONDING GIMBAL ANGLES IN V(MPAC) AS DOUBLE PRECISION 1'S COMPLEMENT ANGLES SCALED BY 2P
# FOR THIS CONVERSION ARE
#
#   Z      =      ARCSIN (M / 3)
#
#   Y      =      ARCSIN (-M / COSZ)
#
# IF M IS NEGATIVE, Y IS REPLACED BY PI SGN Y - Y.
#   0

```

```

# Page 347
#      X      =      ARCSIN (-M /COSZ)
#                               5
#
#
# IF M IS NEGATIVE, X IS REPLACED BY PI SGN X - X.
#      4
#
# THIS ROUTINE DOES NOT SET THE PUSH DOWN POINTER, BUT USES THE NEXT 8 LOCATIONS OF T
# RETURNS THE POINTER TO ITS ORIGINAL SETTING. THIS PROCEDURE ALLOWS THE CALLER TO S
# THE PUSH DOWN LIST.
#
# DELCOMP
# -----
#
#
# THIS ROUTINE COMPUTES THE DIRECTION COSINE MATRIX (DEL) RELATING ON
#
# IS ROTATED WITH RESPECT TO THE FIRST BY AN ANGLE, A, ABOUT A UNIT VECTOR U. THE FO
#
#
#      *
#      DEL      =      I COSA + U U-T (1 - COSA) + V* SINA
#                               X
#
#
# WHERE
#
#      *
#      I      =      [ 1  0  0 ]
#                    [ 0  1  0 ]
#                    [ 0  0  1 ]
#
#
#
#
#      [ 2
#      [ U      U U      U U ]
#      [ X      X Y      X Z ]
#      [
#      [
#      U U-T      =      [ U U      U      U U ]
#                        [ Y X      Y      Y Z ]
#                        [
#                        [
#                        [ U U      U U      U ]
#                        [ Z X      Z Y      Z ]
#
#
#
#
#
#
#      [ 0      -U      U ]
#      [
#      [
#      V*      =      [ U      0      -U ]
#      X      [ Z      X ]
#      [
#      [ -U      U      0 ]

```

```

#           [   Y           X           ]
#
# Page 348
#
#   -
#   U   =   UNIT ROTATION VECTOR RESOLVED INTO S/C AXES.
#   A   =   ROTATION ANGLE
#
#           *
# THE INTERPRETATION OF DEL IS AS FOLLOWS:
#
# IF AX , AY , AZ REPRESENT THE COMPONENTS OF A VECTOR IN THE ROTATED FRAME, THEN THE COMPONENTS
# VECTOR IN THE ORIGINAL S/C AXES (BX , BY , BZ) ARE
#
#           [ BX ]           [ AX ]
#           [ XX ]           [ XX ]
#           [   ]           [   ]
#           [ BY ]           [ AY ]
#           [ YX ]   =   *   DEL   [ YX ]
#           [   ]           [   ]
#           [ BZ ]           [ BZ ]
#           [ ZX ]           [ ZX ]
#
# THE ROUTINE WILL STORE THIS MATRIX (SCALED UNITY) IN SEQUENTIAL LOCATIONS OF ERASABLE MEMORY
#
# THE LOCATION CALLED DEL. IN ORDER TO USE THE ROUTINE, THE CALLING PROGRAM MUST FIRST STORE U
# DOUBLE PRECISION VECTOR) IN THE SET OF ERASABLE LOCATIONS BEGINNING WITH THE ADDRESS CALLED C
# MUST THEN BE LOADED INTO D(MPAC).
#
# INTERNALLY, THE PROGRAM ALSO USES THE FIRST 10 LOCATIONS OF THE PUSH DOWN LIST.
#
# READCDUK
# -----
#
# THIS BASIC LANGUAGE SUBROUTINE LOADS T(MPAC) WITH THE THREE CDU ANGLES.
#
# SIGNMPAC
# -----
#
# THIS IS A BASIC LANGUAGE SUBROUTINE WHICH LIMITS THE MAGNITUDE OF D(MPAC) TO + OR - DPOSMAX C
#
# PROGRAM STORAGE ALLOCATION
#
# 1)      FIXED MEMORY      1059 WORDS
# 2)      ERASABLE MEMORY   98

```

```
#      3)      STATE SWITCHES      3
```

```
# Page 349
```

```
#      4)      FLAGS      1
```

```
#
```

```
# JOB PRIORITIES
```

```
#
```

```
#      1)      KALCMANU      TBD
```

```
#      2)      ONE SECOND UPDATE      TBD
```

```
#
```

```
# SUMMARY OF STATE SWITCHES AND FLAGWORDS USED BY KALCMANU.
```

```
#
```

```
#      STATE      FLAGWRD 2      SETTING      MEANING
#      SWITCH NO.      BIT NO.
```

```
#
```

```
#      *
```

```
#      31      14      0      MANEUVER WENT THROUGH GIMBAL
#      1      MANEUVER DID NOT GO THROUGH C
```

```
#      *
```

```
#      32      13      0      CONTINUE UPDATE PROCESS
#      1      START UPDATE PROCESS
```

```
#      33      12      0      PERFORM FINAL P AXIS YAW IF P
#      1      IGNORE ANY FINAL P-AXIS YAW
```

```
#      34      11      0      SIGNAL END OF KALCMANU
#      1      KALCMANU IN PROCESS.      USER
```

```
#
```

```
#      * INTERNAL TO KALCMANU
```

```
#
```

```
# SUGGESTIONS FOR PROGRAM INTEGRATION
```

```
#
```

```
# THE FOLLOWING VARIABLES SHOULD BE ASSIGNED TO UNSWITCH ERASABLE:
```

```
#
```

```
#      CPHI
#      CTHETA
#      CPSI
#      POINTVSM +5
#      SCAXIS +5
#      DELDCDU
#      DELDCDU1
#      DELDCDU2
#      RATEINDX
```

```
#
```

```
# THE FOLLOWING SUBROUTINES MAY BE PUT IN A DIFFERENT BANK
```

```
#
```

```
#      MXM3
```


Page 350

```
#      TRANSPGS
#      SIGNMPAC
#      READCDUK
#      CDUTODCM
```

Page 351

```
BANK      15
SETLOC    KALCMON1
BANK
```

EBANK= BCDU

```
# THE THREE DESIRED CDU ANGLES MUST BE STORED AS SINGLE PRECISION TWO'S COMPLEMENT ANGLES IN TH
# LOCATIONS, CPHI, CTHETA, CPSI.
```

```

COUNT*  $$/KALC
KALCMAN3  TC      INTPRET      # PICK UP THE CURRENT CDU ANGLES AND
RTB      #          COMPUTE THE MATRIX FROM INITIAL S/C
          READCDUK      #          AXES TO FINAL S/C AXES.
STORE    BCDU      # STORE INITIAL S/C ANGLES
SLOAD    ABS      # CHECK THE MAGNITUDE OF THE DESIRED
          CPSI      # MIDDLE GIMBAL ANGLE
DSU      BPL
          LOCKANGL      # IF GREATER THAN 70 DEG ABORT MANEUVER
          TOOBADF
AXC,2    TLOAD
          MIS
          BCDU
CALL     # COMPUTE THE TRANSFORMATION FROM INITIAL
          CDUTODCM      # S/C AXES TO STABLE MEMBER AXES
AXC,2    TLOAD
          MFS      # PREPARE TO CALCULATE ARRAY MFS
          CPHI
CALL
          CDUTODCM
SECAD    AXC,1    CALL      # MIS AND MFS ARRAYS CALCULATED      $2
          MIS
          TRANSPOS
VLOAD    STADR
STOVL    TMIS +12D
STADR
STOVL    TMIS +6
STADR
STORE    TMIS      # TMIS = TRANSPOSE(MIS) SCALED BY 2
AXC,1    AXC,2
```

```

                                TMIS
                                MFS
CALL
                                MXM3
VLOAD   STADR
STOVL   MFI +12D
STADR
STOVL   MFI +6
STADR
STORE   MFI          # MFI = TMIS MFS (SCALED BY 4)
SETPD   CALL         # TRANSPOSE MFI IN PD LIST

# Page 352

                                18D
                                TRNSPSPD
VLOAD   STADR
STOVL   TMFI      +12D
STADR
STOVL   TMFI      +6
STADR
STORE   TMFI          # TMFI = TRANSPOSE (MFI) SCALED BY 4

# CALCULATE COFSKEW AND MFISYM

DLOAD   DSU
        TMFI      +2
        MFI      +2
PDDL    DSU          # CALCULATE COF SCALED BY 2/SIN(AM)
        MFI      +4
        TMFI      +4
PDDL    DSU
        TMFI      +10D
        MFI      +10D
VDEF
STORE   COFSKEW      # EQUALS MFISKEW

# CALCULATE AM AND PROCEED ACCORDING TO ITS MAGNITUDE

DLOAD   DAD
        MFI
        MFI      +16D
DSU     DAD
        DP1/4TH
        MFI      +8D
STORE   CAM          # CAM = (MFIO+MFI4+MFI8-1)/2 HALF SCALE
ARCCOS
STORE   AM           # AM=ARCCOS(CAM)          (AM SCALED BY 2)

```

```

DSU      BPL
          MINANG
          CHECKMAX

TLOAD    # MANEUVER LESS THAN .25 DEGREES
          # GO DIRECTLY INTO ATTITUDE HOLD
          CPHI
STCALL   # ABOUT COMMANDED ANGLES
          CDUXD
          TOOBADI
          # STOP RATE AND EXIT

CHECKMAX  DLOAD  DSU
          AM
          MAXANG
          BPL    VLOAD
          ALTCALC
          COFSKEW
          # UNIT
          # COFSKEW

          UNIT
STORE    COF
          # COF IS THE MANEUVER AXIS

# Page 353 GOTO
          # SEE IF MANEUVER GOES THRU GIMBAL LOCK
          LOCSKIRT

ALTCALC  VLOAD  VAD
          # IF AM GREATER THAN 170 DEGREES
          MFI
          TMFI

VSR1
STOVL    MFISYM
          MFI      +6
VAD      VSR1
          TMFI     +6
STOVL    MFISYM  +6
          MFI      +12D
VAD      VSR1
          TMFI     +12D
STORE    MFISYM  +12D
          # MFISYM=(MFI+TMFI)/2    SCALED BY 4

# CALCULATE COF

DLOAD    SR1
          CAM
PDDL     DSU
          # PDO CAM
          DPHALF
          CAM
BOVB     PDDL
          # PS2 1 - CAM
          SIGNMPAC
          MFISYM  +16D
DSU      DDV
          0
          2

```

```

      SQR      PDDL      # COFZ = SQR(MFISYM8-CAM)/(1-CAM)
      MFISYM  +8D      #
      DSU      DDV      $ ROOT 2
      0
      2
      SQR      PDDL      # COFY = SQR(MFISYM4-CAM)/(1-CAM)  $ROOT2
      MFISYM
      DSU      DDV
      0
      2
      SQR      VDEF      # COFX = SQR(MFISYM-CAM)/(1-CAM)  $ROOT 2
      UNIT
      STORE    COF

# DETERMINE LARGEST COF AND ADJUST ACCORDINGLY

COFMAXGO      DLOAD    DSU
                  COF
                  COF      +2
      BMN      DLOAD      # COFY G COFX

# Page 354
                  COMP12
                  COF
      DSU      BMN
                  COF      +4
                  METHOD3      # COFZ G COFX OR COFY
      GOTO
                  METHOD1      # COFX G COFY OR COFZ
      COMP12    DLOAD    DSU
                  COF      +2
                  COF      +4
      BMN
                  METHOD3      # COFZ G COFY OR COFX

METHOD2        DLOAD    BPL      # COFY MAX
                  COFSKEW +2      # UY
                  U2POS
      VLOAD      VCOMP
                  COF
      STORE      COF
      U2POS      DLOAD    BPL
                  MFISYM  +2      # UX UY
                  OKU21
      DLOAD      DCOMP      # SIGN OF UX OPPOSITE garbled
                  COF
      STORE      COF

```

OKU21	DLOAD	BPL		
		MFISYM +10D		# UY UZ
		LOCSKIRT		
	DLOAD	DCOMP		# SIGN OF UZ OPPOSITE TO UY
		COF +4		
	STORE	COF +4		
	GOTO			
		LOCSKIRT		
METHOD1	DLOAD	BPL		# COFX MAX
		COFSKEW		# UX
		U1POS		
	VLOAD	VCOMP		
		COF		
	STORE	COF		
U1POS	DLOAD	BPL		
		MFISYM +2		# UX UY
		OKU12		
	DLOAD	DCOMP		
		COF +2		# SIGN OF UY OPPOSITE TO UX
	STORE	COF +2		
OKU12	DLOAD	BPL		
		MFISYM +4		# UX UZ
		LOCSKIRT		
	DLOAD	DCOMP		# SIGN OF UZ OPPOSITE TO UY
		COF +4		
# Page 355				
	STORE	COF +4		
	GOTO			
		LOCSKIRT		
METHOD3	DLOAD	BPL		# COFZ MAX
		COFSKEW +4		# UZ
		U3POS		
	VLOAD	VCOMP		
		COF		
	STORE	COF		
U3POS	DLOAD	BPL		
		MFISYM +4		# UX UZ
		OKU31		
	DLOAD	DCOMP		
		COF		# SIGN OF UX OPPOSITE TO UZ
	STORE	COF		
OKU31	DLOAD	BPL		
		MFISYM +10D		# UY UZ
		LOCSKIRT		
	DLOAD	DCOMP		
		COF +2		# SIGN OF UY OPPOSITE TO UZ

```

                                STORE  COF      +2
                                GOTO
                                LOCSKIRT
# Page 356
# MATRIX OPERATIONS

                                BANK    13
                                SETLOC  KALCMON2
                                BANK
                                EBANK=  BCDU

MXM3                            SETPD  VLOAD*      # MXM3 MULTIPLIES 2 3X3 MATRICES
                                0          # AND LEAVES RESULT IN PD LIST
                                0,1        # AND MPAC
                                VXM*  PDVL*
                                0,2
                                6,1
                                VXM*  PDVL*
                                0,2
                                12D,1
                                VXM*  PUSH
                                0,2
                                RVQ

# RETURN WITH MIXM2 IN PD LIST

TRANSPOS                        SETPD  VLOAD*      # TRANSPOS TRANSPOSES A 3X3 MATRIX
                                0          # AND LEAVES RESULT IN PD LIST
                                0,1        # MATRIX ADDRESS IN XR1
                                PDVL*  PDVL*
                                6,1
                                12D,1

                                PUSH          # MATRIX IN PD
TRNSPSPD                        EXIT          # ENTER WITH MATRIX AT 0 IN PD LIST
                                INDEX  FIXLOC
                                DXCH   12
                                INDEX  FIXLOC
                                DXCH   16
                                INDEX  FIXLOC
                                DXCH   12
                                INDEX  FIXLOC
                                DXCH   14
                                INDEX  FIXLOC
                                DXCH   4
                                INDEX  FIXLOC

```

```

          DXCH      14
          INDEX     FIXLOC
          DXCH      2
          INDEX     FIXLOC
          DXCH      6
          INDEX     FIXLOC
          DXCH      2

# Page 357

          TC        INTPRET
          RVQ

          BANK      15
          SETLOC    KALCMON1
          BANK

          EBANK=    BCDU

MINANG      2DEC      0.00069375

MAXANG      2DEC      0.47222222

# GIMBAL LOCK CONSTANTS

# D = MGA CORRESPONDING TO GIMBAL LOCK = 60 DEGREES
#      NGL = BUFFER ANGLE (TO AVOID DIVISIONS BY ZERO) = 2 DEGREES

SD          2DEC      .433015          # = SIN(D)          $2

K3S1        2DEC      .86603           # = SIN(D)          $1

K4          2DEC      -.25              # = -COS(D)         $2

K4SQ        2DEC      .125              # = COS(D)COS(D)    $2

SNGLCD      2DEC      .008725          # = SIN(NGL)COS(D)  $2

CNGL        2DEC      .499695          # COS(NGL)          $2

LOCKANGL    DEC       .388889          # = 70 DEGREES

# INTERPRETIVE SUBROUTINE TO READ THE CDU ANGLES

READCDUK    CA      CDUZ              # LOAD T(MPAC) WITH CDU ANGLES
            TS      MPAC      +2
            EXTEND
            DCA      CDUX              # AND CHANGE MODE TO TRIPLE PRECISION

```

```

TCF      TLOAD  +6

CDUTODCM  AXT,1  SSP
          OCT    3
          S1
          OCT    1      # SET XR1, S1, AND PD FOR LOOP
          STORE  7
          SETPD
          0
LOOPSIN   SLOAD* RTB
          10D,1
          CDULOGIC

# Page 358

STORE     10D      # LOAD PD WITH 0 SIN(PHI)
SIN        PDDL     #              2 COS(PHI)
          10D      #              4 SIN(THETA)
COS        PUSH     #              6 COS(THETA)
TIX,1     DLOAD     #              8 SIN(PSI)
          LOOPSIN   #              10 COS(PSI)
          6
DMP        SL1
          10D
STORE      0,2      # C0 = COS(THETA)COS(PSI)
DLOAD     DMP
          4
          0
PDDL      DMP      # (PD6 SIN(THETA)SIN(PHI))
          6
          8D
DMP        SL1
          2
BDSU       SL1
          12D
STORE      2,2      # C1=-COS(THETA)SIN(PSI)COS(PHI)
DLOAD     DMP
          2
          4
PDDL      DMP      # (PD7 COS(PHI)SIN(THETA)) SCALED 4
          6
          8D
DMP        SL1
          0
DAD        SL1
          14D
STORE      4,2      # C2=COS(THETA)SIN(PSI)SIN(PHI)
DLOAD

```



```

      8D
STORE 6,2      # C3=SIN(PSI)
DLOAD
      10D
DMP  SL1
      2
STORE 8D,2     # C4=COS(PSI)COS(PHI)
DLOAD DMP
      10D
      0
DCOMP SL1
STORE 10D,2    # C5=-COS(PSI)SIN(PHI)
DLOAD DMP
      4
      10D
DCOMP SL1
STORE 12D,2    # C6=-SIN(THETA)COS(PSI)
DLOAD
DMP  SL1      # (PUSH UP 7)
      8D
PDDL DMP      # (PD7 COS(PHI)SIN(THETA)SIN(PSI)) SCALE 4
      6
      0
DAD  SL1      # (PUSH UP 7)
STADR      # C7=COS(PHI)SIN(THETA)SIN(PSI)
STORE 14D,2   #      +COS(THETA)SIN(PHI)
DLOAD
DMP  SL1      # (PUSH UP 6)
      8D
PDDL DMP      # (PD6 SIN(THETA)SIN(PHI)SIN(PSI)) SCALE 4
      6
      2
DSU  SL1      # (PUSH UP 6)
STADR
STORE 16D,2   # C8=-SIN(THETA)SIN(PHI)SIN(PSI)
RVQ          # +COS(THETA)COS(PHI)

# CALCULATION OF THE MATRIX DEL.....
#
#      *      *      --T      *
#      DEL = (IDMATRIX)COS(A)+UU (1-COS(A))+UX SIN(A)          SCALED 1
#
#      -
#      WHERE U IS A UNIT VECTOR (DP SCALED 2) ALONG THE AXIS OF ROTATION.
#      A IS THE ANGLE OF ROTATION (DP SCALED 2)
#
#      -

```

UPON ENTRY, THE STARTING ADDRESS OF U IS COF, AND A IS IN MPAC

DELCOMP	SETPD	PUSH	# MPAC CONTAINS THE ANGLE A	
		0		
	SIN	PDDL	# PD0 = SIN(A)	
	COS	PUSH	# PD2 = COS(A)	
	SR2	PDDL	# PD2 = COS(A)	\$8
	BDSU	BOVB		
		DPHALF		
		SIGNMPAC		
	PDDL		# PDA = 1-COS(A)	

COMPUTE THE DIAGONAL COMPONENTS OF DEL

		COF		
	DSQ	DMP		
		4		
	DAD	SL3		
		2		
	BOVB			
		SIGNMPAC		
# Page 360	STODL	KEL	# UX UX(1-COS(A)) +COS(A)	\$1
		COF	+2	
	DSQ	DMP		
		4		
	DAD	SL3		
		2		
	BOVB			
		SIGNMPAC		
	STODL	KEL	+8D # UY UY(1-COS(A)) +COS(A)	\$1
		COF	+4	
	DSQ	DMP		
		4		
	DAD	SL3		
		2		
	BOVB			
		SIGNMPAC		
	STORE	KEL	+16D # UZ UZ(1-COS(A)) +COS(A)	\$1

COMPUTE THE OFF DIAGONAL TERMS OF DEL

	DLOAD	DMP	
		COF	
		COF	+2
	DMP	SL1	

PDDL	4 DMP COF +4 0	# D6	UX UY (1-COS A)	\$4
PUSH	DAD 6	# D8	UZ SIN A	\$4
SL2	BOVB SIGNMPAC			
STODL	KEL +6			
BDSU	SL2			
BOVB				
	SIGNMPAC			
STODL	KEL +2 COF			
DMP	DMP COF +4 4			
SL1	PDDL COF +2	# D6	UX UZ (1-COS A)	\$4
DMP	PUSH 0	# D8	UY SIN(A)	
DAD	SL2 6			
BOVB				
	SIGNMPAC			
STODL	KEL +4	# UX UZ (1-COS(A))+UY SIN(A)		
BDSU	SL2			
BOVB				
	SIGNMPAC			
STODL	KEL +12D COF +2	# UX UZ (1-COS(A))-UY SIN(A)		
DMP	DMP COF +4 4			
SL1	PDDL COF	# D6	UY UZ (1-COS(A))	\$ 4
DMP	PUSH 0	# D8	UX SIN(A)	
DAD	SL2 6			
BOVB				
	SIGNMPAC			
STODL	KEL +14D	# UY UZ(1-COS(A)) +UX SIN(A)		
BDSU	SL2			
BOVB				

```

                                SIGNMPAC
                                STORE  KEL      +10D      # UY UZ (1-COS(A)) -UX SIN(A)
                                RVQ

# DIRECTION COSINE MATRIX TO CDU ANGLE ROUTINE
# X1 CONTAINS THE COMPLEMENT OF THE STARTING ADDRESS FOR MATRIX (SCALED 2).
# LEAVE CDU ANGLES SCALED 2PI IN V(MPAC).
# COS(MGA) WILL BE LEFT IN S1 (SCALED 1).
#
# THE DIRECTION COSINE MATRIX RELATING S/C AXES TO STABLE MEMBER AXES CAN BE WRITTEN
#
#      C  = COS(THETA) COS(PHI)
#      0
#
#      C  = -COS(THETA) SIN(PHI) COS(PHI) + SIN(THETA) SIN(PHI)
#      1
#
#      C  = COS(THETA) SIN(PHI) SIN(PHI) + SIN(THETA) COS(PHI)
#      2
#
#      C  = SIN(PHI)
#      3
#
#      C  = COS(PHI) COS(PHI)
#      4
#
#      C  = -COS(PHI) SIN(PHI)
#      5
#
#      C  = -SIN(THETA) COS(PHI)
#      6
#
#      C  = SIN(THETA) SIN(PHI) COS(PHI) + COS (THETA) SIN(PHI)
#      7
#
#      C  = -SIN(THETA) SIN(PHI) SIN(PHI) + COS(THETA)COS(PHI)
#      8
# Page 362
#
# WHERE PHI = OGA
#      THETA = IGA
#      PSI = MGA

DCMTOCDU      DLOAD*  ARCSIN
                6,1
                PUSH   COS      # PD +0      PSI

```

```

SL1      BOVB
          SIGNMPAC
STORE    S1
DLOAD*   DCOMP
          12D,1
DDV      ARCSIN
          S1
PDDL*    BPL          # PD +2          THETA
          0,1          # MUST CHECK THE SIGN OF COS(THETA)
          OKTHETA      # TO DETERMINE THE PROPER QUADRANT.
DLOAD    DCOMP
BPL      DAD
          SUHALFA
          DPHALF
GOTO
          CALCPHI
SUHALFA  DSU
          DPHALF
CALCPHI  PUSH
OKTHETA  DLOAD*   DCOMP
          10D,1
DDV      ARCSIN
          S1
PDDL*    BPL          # PUSH DOWN PHI
          8D,1
          OKPHI
DLOAD    DCOMP          # PUSH UP PHI
BPL      DAD
          SUHALFAP
          DPHALF
GOTO
          VECOFANG
SUHALFAP DSU
          GOTO
          DPHALF
          VECOFANG
OKPHI    DLOAD          # PUSH UP PHI
VECOFANG VDEF      RVQ
# Page 363
# ROUTINES FOR TERMINATING THE AUTOMATIC MANEUVER AND RETURNING TO USER.

TOOBADF  EXIT
          TC      ALARM
          OCT     00401

          TCF     NOGO          # DO NOT ZERO ATTITUDE ERRORS

```

	TC	BANKCALL	
	CADR	ZATTEROR	# ZERO ATTITUDE ERRORS
NOGO	TC	BANKCALL	
	CADR	STOPRATE	# STOP RATES
	CAF	TWO	
	INHINT		# ALL RETURNS ARE NOW MADE VIA GOODEND
	TC	WAITLIST	
	EBANK=	BCDU	
	2CADR	GOODMANU	
	TCF	ENDOFJOB	
TOOBADI	EXIT		
	TCF	NOGO	

This code is written to file `src/ATTITUDE-MANEUVER-ROUTINE.s`.

B.13 AUTOMATIC MANEUVERS

199

<src/AUTOMATIC-MANEUVERS.s 199>≡

```
# Copyright:    Public domain.
# Filename:     AUTOMATIC_MANEUVERS.agc
# Purpose:     Part of the source code for Colossus 2A, AKA Comanche 055.
#              It is part of the source code for the Command Module's (CM)
#              Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:   yaYUL
# Contact:     Ron Burkey <info@sandroid.org>.
# Website:     www.ibiblio.org/apollo.
# Pages:       1025-1036
# Mod history: 2009-05-13 RSB   Adapted from the Colossus249/ file of the
#                               same name, using Comanche055 page images.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#       Colossus 2A

# Page 1025

                BANK      21
                SETLOC    DAPS3
                BANK

                COUNT     21/DAPAM

                EBANK=    KMPAC
AHFNOROT        EXTEND
                READ      CHAN31
                MASK      BIT14
                EXTEND
                BZMF      FREECONT
                CA        RCSFLAGS          # SEE IF RATE FILTER HAS BEEN INITIALIZED
```

```

                                MASK    BIT14
                                CCS      A
                                TCF      REINIT
                                # IF SO, PROCEED WITH ATTITUDE CONTROL
                                # IF NOT, RECYCLE TO INITIALIZE FILTER
                                # AUTOMATIC CONTROL YET

                                EXTEND
                                READ     CHAN31
                                MASK     BIT13
                                EXTEND
                                BZMF     HOLDFUNC

AUTOCONT    CA      HOLDFLAG      # IF HOLDFLAG IS +, GO TO GRABANG.
            EXTEND
            BZMF     ATTHOLD
            TCF      GRABANG      # OTHERWISE, GO TO ATTHOLD.

# MINIMUM IMPULSE CONTROL

FREECONT    CAF     ONE
            TS      HOLDFLAG      # RESET HOLDFLAG
            # INHIBIT AUTOMATIC STEERING

            EXTEND
            READ     CHAN32
            TS      L
            COM
            MASK     MANROT
            MASK     CHANTEMP
            LXCH     CHANTEMP
            TC      STICKCHK
            INDEX    RMANNDX
            CA      MINTAU      # MINTAU      +0
            TS      TAU      #      +1      +14MS MINIMUM IMPULSI
            INDEX    PMANNDX      #      +2      -14MS TIME
            CA      MINTAU      #      +3      +0
            TS      TAU1
            INDEX    YMANNDX
            CA      MINTAU

# Page 1026
            TS      TAU2
            TCF      T6PROGM

MINTAU      DEC     0
            DEC     23      # = 14MS
            DEC     -23     # = -14MS
            DEC     0

# Page 1027

```


CALCULATION OF ATTITUDE ERRORS:

$\bar{AK} = \bar{AMGB} (\bar{CDUX} - \bar{THETADX}) + \bar{BIAS}$

#

I.E., $\bar{AK} = 1 \sin(\bar{PSI}) 0 ** \bar{CDUX} - \bar{THETADX} * \bar{BIAS} *$

$* * *$

$\bar{AK1} = 0 \cos(\bar{PSI})\cos(\bar{PHI}) \sin(\bar{PHI}) ** \bar{CDUY} - \bar{THETADY} * + \bar{BIAS1} *$

$* * *$

$\bar{AK2} = 0 -\cos(\bar{PSI})\sin(\bar{PHI}) \cos(\bar{PHI}) ** \bar{CDUZ} - \bar{THETADZ} * \bar{BIAS2} *$

#

THE BIASES ARE ADDED ONLY WHILE PERFORMING AUTOMATIC MANEUVERS (ESP KALCMANU) TO PROVIDE ADDI
AND PREVENT OVERSHOOT WHEN STARTING AN AUTOMATIC MANEUVER. NORMALLY THE REQUIRED LEAD IS ONL
BUT DURING HIGH RATE MANEUVERS IT CAN BE AS MUCH AS 7 DEGREES. THE BIASES ARE COMPUTED BY KA
FIXED UNTIL THE MANEUVER IS COMPLETED AT WHICH TIME THEY ARE RESET TO ZERO.

ATTHOLD	CA	CDUX
	EXTEND	
	MSU	THETADX
	TS	ERRORX
	CA	CDUY
	EXTEND	
	MSU	THETADY
	TS	T5TEMP
	EXTEND	
	MP	AMGB1
	ADS	ERRORX
	CA	T5TEMP
	EXTEND	
	MP	AMGB4
	TS	ERRORY
	CA	T5TEMP
	EXTEND	
	MP	AMGB7
	TS	ERRORZ
	CA	CDUZ
	EXTEND	
	MSU	THETADZ
	TS	T5TEMP
	EXTEND	
	MP	AMGB5
	ADS	ERRORY
	CA	T5TEMP
	EXTEND	
	MP	AMGB8
	ADS	ERRORZ
	CS	HOLDFLAG

```

# Page 1028
EXTEND
BZMF   JETS
CA     BIAS           # AD BIASES ONLY IF PERFORMING AUTOMATIC
ADS    ERRORX
CA     BIAS1
ADS    ERRORY
CA     BIAS2
ADS    ERRORZ
TCF    JETS

HOLDFUNC CCS    HOLDFLAG
TCF      +3
TCF      ATTHOLD
TCF      +1

GRABANG CAF    ZERO           # ZERO WBODYS AND BIASES
TS       WBODY
TS       WBODY +1
TS       WBODY1
TS       WBODY1 +1
TS       WBODY2
TS       WBODY2 +1
TS       BIAS
TS       BIAS1
TS       BIAS2

CA      RCSFLAGS
MASK    OCT16000
EXTEND
BZF     ENDDAMP        # IS RATE DAMPING COMPLETED
CAF     ZERO           # IF SO, GO TO ENDDAMP
TS       ERRORX        # OTHERWISE, ZERO ERRORS
TS       ERRORY
TS       ERRORZ
TCF     JETS

ENDDAMP TS      HOLDFLAG    # SET HOLDFLAG +0
EXTEND
DCA     CDUX           # PICK UP CDU ANGLES FOR ATTITUDE HOLD
DXCH    THETADX        # REFERENCES
CA      CDUZ
TS      THETADZ
TCF     ATTHOLD

```

Page 1029

JET SWITCHING LOGIC AND CALCULATION OF REQUIRED ROTATION COMMANDS

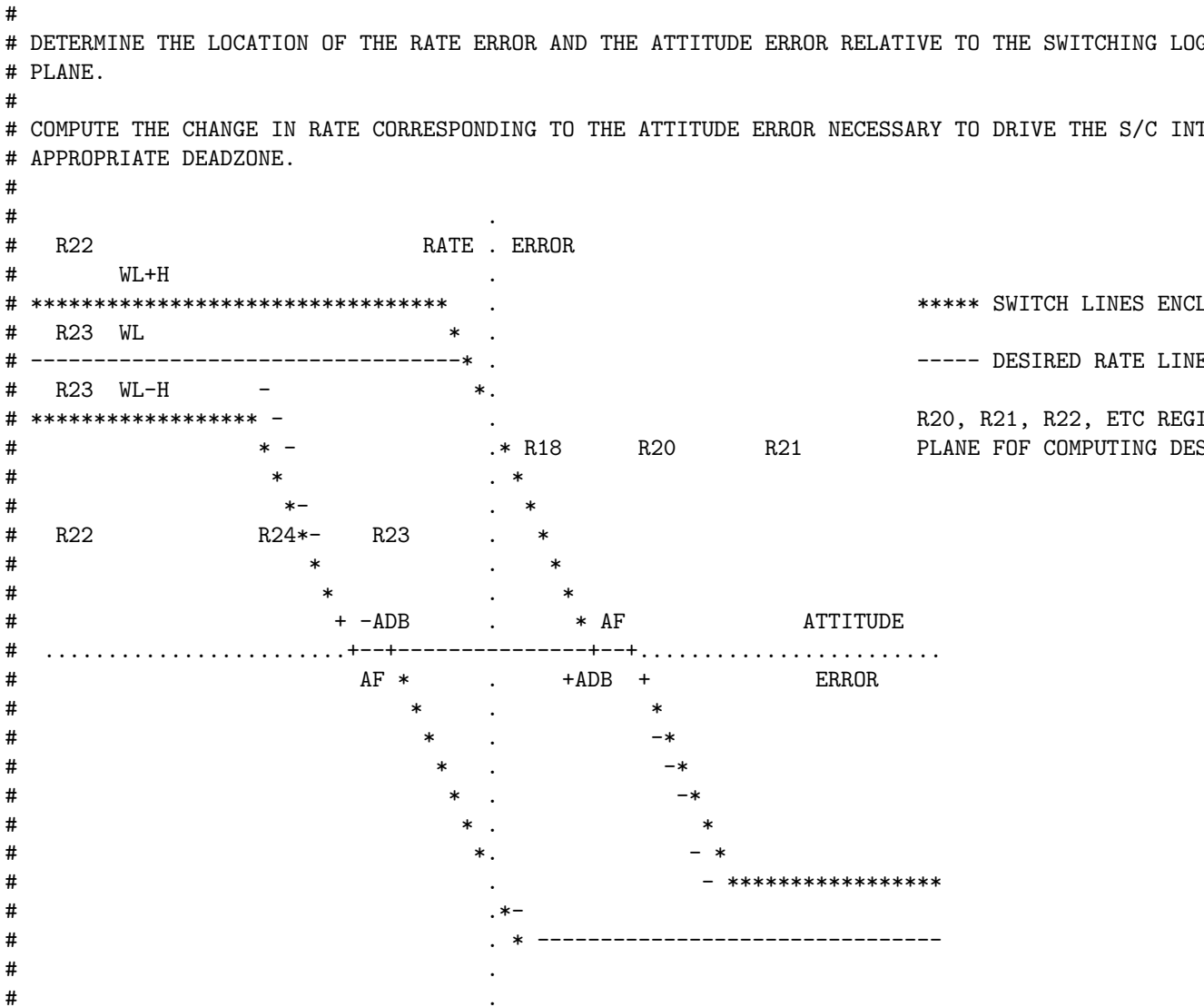


FIG. 1 PHASE PLANE SWITCHING LOGIC

CONSTANTS FOR JET SWITCHING LOGIC

WLH/SLOP	DEC	.00463	# = WL+H/SLOPE = .83333 DEG	\$180
WL-H/SLP	DEC	.00277	# = WL-H/SLOPE = .5 DEG	\$180
WLH	2DEC	.0011111111	# = WL+H = 0.5 DEG/SEC	\$450
WLMH	2DEC	.0006666666	# = WL-H = 0.3 DEG/SEC	\$450
WL	2DEC	.0008888888	# = WL = 0.4 DEG/SEC	\$450

Page 1030

SLOPE2	DEC	.32	# = 0.8 DEG/SEC/DEG	\$450/180
JETS	CA	ADB		
	AD	FOUR	# AF = FLAT REGION = .044 DEG	
	TS	T5TEMP	# ADB+AF	
	CAF	TWO		
JLOOP	TS	SPNDX		
	DOUBLE			
	TS	DPNDX		
	EXTEND			
	INDEX	A		
	DCA	ADOT		
	DXCH	EDOT		
	CA	HOLDFLAG	# HOLDFLAG = +0 MEANS THAT DAP IS IN	
	EXTEND		# ATTITUDE HOLD AND RATE DAMPING IS OVER.	
	BZF	INHOLD	# IF THIS IS THE CASE, BYPASS ADDITION	
			# OF WBODY AND GO TO INHOLD	
	EXTEND			
	INDEX	DPNDX		
	DCS	WBODY		
	DAS	EDOT	# = ADOT-WBODY	
INHOLD	INDEX	SPNDX		
	CA	ERRORX		
	TS	AERR	# AERR = BIAS + AK	
	CCS	EDOT		
	TCF	POSVEL		
	TCF	SIGNCK1		
	TCF	NEGVEL		
SIGNCK1	CCS	EDOT +1		
	TCF	POSVEL		
	TCF	POSVEL		
	TCF	NEGVEL		
	TCF	NEGVEL		
POSVEL	EXTEND			
	DCA	EDOT		
	DXCH	EDOTVEL		
	CA	T5TEMP		
	TS	ADBVEL	# +(ADB+AF)	
	CA	AERR		
	TS	AERRVEL		
	TC	J6.		
NEGVEL	EXTEND			
	DCS	EDOT		
	DXCH	EDOTVEL		

CS	T5TEMP	
TS	ADBVEL	# -(ADB+AF)
CS	AERR	
TS	AERRVEL	

J6.
Page 1031

EXTEND		
SU	ADB	
AD	WLH/SLOP	
EXTEND		
BZMF	J8	
CS	T5TEMP	# (ADB+AF)
AD	AERRVEL	
EXTEND		
BZMF	+2	
TCF	J7	
EXTEND		
DCS	EDOTVEL	
EXTEND		
DV	SLOPE	
EXTEND		
SU	AERRVEL	
AD	ADB	
EXTEND		
BZMF	J18	
TCF	J23	

J7	CS	WL-H/SLP	
	EXTEND		
	SU	T5TEMP	# (ADB+AF)
	AD	AERRVEL	
	EXTEND		
	BZMF	J20	
	TCF	J21	

J8	EXTEND	
	DCS	WLH
	DXCH	WTEMP
	EXTEND	
	DCA	EDOTVEL
	DAS	WTEMP
	CCS	WTEMP
	TCF	J22
	TCF	SIGNCK2
	TCF	NJ22

SIGNCK2	CCS	WTEMP +1	
	TCF	J22	
	TCF	J22	
	TCF	NJ22	
NJ22	EXTEND		
	DCA	EDOTVEL	
	EXTEND		
	DV	SLOPE	
	AD	T5TEMP	# (ADB+AF)
	AD	AERRVEL	
# Page 1032	CCS	A	
	TCF	J23	
	TCF	J23	
	TCF	+2	
	TCF	J23	
	EXTEND		
	DCS	WLMH	# WL - H
	DXCH	WTEMP	
	EXTEND		
	DCA	EDOTVEL	
	DAS	WTEMP	
	CCS	WTEMP	
	TCF	J23	
	TCF	SIGNCK3	
	TCF	NJ23	
SIGNCK3	CCS	WTEMP +1	
	TCF	J23	
	TCF	J23	
	TCF	NJ23	
NJ23	CA	AERRVEL	
	AD	T5TEMP	# (ADB+AF)
	AD	WL-H/SLP	
	CCS	A	
	TCF	J24	
	TCF	J24	
	TCF	J22	
	TCF	J22	
J18	EXTEND		
	DCS	EDOT	
	DXCH	KMPAC	
	TCF	JTIME	

J20	CS	AERR	
	AD	ADBVEL	
	EXTEND		
	MP	SLOPE2	# (HYSTERESIS SLOPE)
	DXCH	KMPAC	
	EXTEND		
	DCS	EDOT	
	DAS	KMPAC	
	TCF	JTIME	
J21	CCS	EDOT	
	TCF	JP	
	TCF	SIGNCK4	
	TCF	JN	
SIGNCK4	CCS	EDOT +1	
# Page 1033			
	TCF	JP	
	TCF	JP	
	TCF	JN	
JN	EXTEND		
	DCS	EDOT	
	DXCH	KMPAC	
	EXTEND		
	DCA	WL	
	DAS	KMPAC	
	TCF	JTIME	
JP	EXTEND		
	DCS	EDOT	
	DXCH	KMPAC	
	EXTEND		
	DCS	WL	
	DAS	KMPAC	
	TCF	JTIME	
J22	CCS	EDOT	
	TCF	JN	
	TCF	SIGNCK5	
	TCF	JP	
SIGNCK5	CCS	EDOT +1	
	TCF	JN	
	TCF	JN	
	TCF	JP	
	TCF	JP	

```

J23      INDEX  SPNDX
        CS      BIT13      # RESET RATE DAMPING FLAG
        MASK    RCSFLAGS   # BIT13 FOR ROLL  (SPNDX = 0)
        TS      RCSFLAGS   # BIT12 FOR PITCH (SPNDX = 1)
                                # BIT11 FOR YAW  (SPNDX = 2)

        INDEX  SPNDX
        CAF    OCT01400    # IS THERE TO BE A FORCED FIRING ON THIS
        MASK    RCSFLAGS   # AXIS
        EXTEND
        BZF     DOJET +2    # NO, GO TO DOJET +2 AND DO NOTHING

        TCF     J18        # YES, GO TO J18 AND FORCE A FIRING

J24      CS      AERR
        EXTEND
        SU      ADBVEL
        EXTEND
        MP      SLOPE2      # (HYSTERESIS SLOPE)
        DXCH    KMPAC
        EXTEND

# Page 1034

        DCS     EDOT
        DAS     KMPAC

# Page 1035
# COMPUTE THE JET ON TIME NECESSARY TO ACCOMPLISH THE DESIRED CHANGE IN RATE, I.E.,
#
#      T = J/M(DELT W)
#      J
#
#      DELTA W = DESIRED CHANGE IN S/C ANGULAR RATE AS DETERMINED BY THE
#                SWITCHING LOGIC, AT THIS POINT STORED IN KMPAC.
#
#      J/M = S/C INERTIA TO TORQUE RATIO SCALED BY
#            (57.3/450)(B24/1600)(1/.8)
#            FOR 1 JET OPERATION (M = 700 FT-LB).
#            I.E., J/M = J(SLUG-FTFT) x 0.00000085601606
#
#            THE CORRESPONDING COMPUTER VARIABLES ESTABLISHED BY
#            KEYBOARD ENTRY ARE
#            J/M (ROLL)
#            J/M1 (PITCH)
#            J/M2 (YAW)
#
#      T = JET-ON TIME      SCALED 16384/1600 SEC

```



```

#           J
#
#           THE COMPUTER VARIABLES ARE
#           TAU  (ROLL)
#           TAU1 (PITCH)
#           TAU2 (YAW)

JTIME      INDEX  SPNDX      # PICK UP S/C INERTIA/TORQUE RATIO
CA          J/M      # SCALED (57.3/450)(B24/1600)
TC          SMALLMP  # FOR 1-JET OPERATION
CA          BIT11
TC          SMALLMP
CCS         KMPAC
TCF         +4
TCF         TAUNORM
TCF         +4
TCF         TAUNORM
CA          POSMAX
TCF         DOJET
CA          NEGMAX
TCF         DOJET

TAUNORM     CA      KMPAC +1
DOJET       INDEX  SPNDX
TS          TAU
CCS         SPNDX
TCF         JLOOP
TCF         T6PROG

# Page 1036
ZEROCMDs    CAF     ZERO
TS          TAU
TS          TAU1
TS          TAU2

T6PROG      EXTEND      # WHEN THE ROTATION COMMANDS (TAUS)
DCA         JETADDR     # HAVE BEEN DETERMINED
DXCH        T5LOC       # RESET T5LOC FOR PHASE3
TCF         RESUME

EBANK=      KMPAC
JETADDR     2CADR      JETSLECT

```

This code is written to file `src/AUTOMATIC-MANEUVERS.s`.

B.14 BURN BABY BURN—MASTER IGNITION ROUTINE

```

210  <src/BURN-BABY-BURN-MASTER-IGNITION-ROUTINE.s 210>≡
      # Copyright:    Public domain.
      # Filename:     BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         731-751
      # Mod history:   2009-05-19 RSB  Adapted from the corresponding
      #               Luminary131 file, using page
      #               images from Luminary 1A.
      #               2009-06-07 RSB  Corrected 3 typos.
      #               2009-07-23 RSB  Added Onno's notes on the naming
      #               of this function, which he got from
      #               Don Eyles.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969
      #
      # Page 731
      ## At the get-together of the AGC developers celebrating the 40th anniversary
      ## of the first moonwalk, Don Eyles (one of the authors of this routine along
      ## with Peter Adler) has related to us a little interesting history behind the
      ## naming of the routine.<br>
      ## <br>
      ## It traces back to 1965 and the Los Angeles riots, and was inspired
      ## by disc jockey extraordinaire and radio station owner Magnificent Montague.
      ## Magnificent Montague used the phrase "Burn, baby! BURN!" when spinning the

```

```
## hottest new records. Magnificent Montague was the charismatic voice of
## soul music in Chicago, New York, and Los Angeles from the mid-1950s to
## the mid-1960s.
# BURN, BABY, BURN -- MASTER IGNITION ROUTINE
```

```
BANK      36
SETLOC    P40S
BANK
EBANK=    WHICH
COUNT*   $$/P40
```

```
# THE MASTER IGNITION ROUTINE IS DESIGNED FOR USE BY THE FOLLOWING LEM PROGRAMS:  P12, P40, P42
# IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION:  IN PARTICULAR, EV
# BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG + 26 SECON
# PROGRAMS THROTTLE UP.
```

```
#
# VARIATIONS AMONG PROGRAMS ARE ACCOMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (FOR AVEGEX
# WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS.  USERS PLACE THE ADRES OF THE HEAD OF THE APPROP
# (OF P61TABLE FOR P61LM, FOR EXAMPLE) IN ERASABLE REGISTER 'WHICH' (E4).  THE IGNITION ROUTINE
# WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY.  THE IGNITION ROUTINE IS INITIATED BY A TC
# THROUGH BANKJUMP IF NECESSARY.  THERE IS NO RETURN.
```

```
#
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLE
```

```
#
#          HONI SOIT QUI MAL Y PENSE
```

```
#
#          *****
#          TABLES FOR THE IGNITION ROUTINE
```

```
#          *****
#
```

```
#          NOLI SE TANGERE
```

```
P12TABLE      VN      0674          # (0)
               TCF      ULLGNOT      # (1)
               TCF      COMFAIL3      # (2)
               TCF      GOCUTOFF      # (3)
               TCF      TASKOVER      # (4)
               TCF      P12SPOT      # (5)
               DEC      0              # (6)      NO ULLAGE
               EBANK=    WHICH
               2CADR     SERVEXIT      # (7)

               TCF      DISPCHNG      # (11)
               TCF      WAITABIT      # (12)
               TCF      P12IGN        # (13)
```

P40TABLE	VN	0640	# (0)
	TCF	ULLGNOT	# (1)
	TCF	COMFAIL4	# (2)
	TCF	GOPOST	# (3)
	TCF	TASKOVER	# (4)
	TCF	P40SPOT	# (5)
# Page 732			
	DEC	2240	# (6)
	EBANK=	OMEGAQ	
	2CADR	STEERING	# (7)
	TCF	P40SJUNK	# (11)
	TCF	WAITABIT	# (12)
	TCF	P40IGN	# (13)
	TCF	REP40ALM	# (14)
P41TABLE	TCF	P41SPOT	# (5)
	DEC	-1	# (6)
	EBANK=	OMEGAQ	
	2CADR	CALCN85	# (7)
	TCF	COMMON	# (11)
	TCF	TIGTASK	# (12)
P42TABLE	VN	0640	# (0)
	TCF	WANTAPS	# (1)
	TCF	COMFAIL4	# (2)
	TCF	GOPOST	# (3)
	TCF	TASKOVER	# (4)
	TCF	P42SPOT	# (5)
	DEC	2640	# (6)
	EBANK=	OMEGAQ	
	2CADR	STEERING	# (7)
	TCF	P40SJUNK	# (11)
	TCF	WAITABIT	# (12)
	TCF	P42IGN	# (13)
	TCF	P42STAGE	# (14)
P63TABLE	VN	0662	# (0)
	TCF	ULLGNOT	# (1)
	TCF	COMFAIL3	# (2)
	TCF	V99RECYC	# (3)
	TCF	TASKOVER	# (4)
	TCF	P63SPOT	# (5)
	DEC	2240	# (6)

```

EBANK=  WHICH
2CADR  SERVEXIT      # (7)

TCF    DISPCHNG      # (11)
TCF    WAITABIT      # (12)

# Page 733

TCF    P63IGN        # (13)

ABRTABLE  VN    0663      # (0)
          TCF    ULLGNOT   # (1)
          TCF    COMFAIL3  # (2)
          TCF    GOCUTOFF  # (3)
          TCF    TASKOVER  # (4)
          NOOP      # (5)
          NOOP      # (6)
          NOOP      # (7)
          NOOP
          TCF    DISPCHNG   # (11)
          TCF    WAITABIT   # (12)
          TCF    ABRTIGN    # (13)

# *****
# GENERAL PURPOSE IGNITION ROUTINES
# *****

BURNBABY  TC    PHASCHNG   # GROUP 4 RESTARTS HERE
          OCT    04024

          CAF    ZERO      # EXTIRPATE JUNK LEFT IN DVTOTAL
          TS     DVTOTAL
          TS     DVTOTAL +1

          TC     BANKCALL   # P40AUTO MUST BE BANKCALLED EVEN FROM ITS
          CADR    P40AUTO   # OWN BANK TO SET UP RETURN PROPERLY

B*RNB*B*  EXTEND
          DCA     TIG       # STORE NOMINAL TIG FOR OBLATENESS COMP.
          DXCH    GOBLTIME  # AND FOR P70 OR P71.

          INHINT
          TC     IBNKCALL
          CADR    ENGINOF3
          RELINT

          INDEX   WHICH
          TCF     5

```

```

P42SPOT      =      P40SPOT      # (5)
P12SPOT      =      P40SPOT      # (5)
P63SPOT      =      P41SPOT      # (5)   IN P63 CLOKTASK ALREADY GOING
P40SPOT      CS      CNTDNDEX      # (5)
# Page 734

P41SPOT      TC      BANKCALL      # MUST BE BANKCALLED FOR GENERALIZED
              CADR      STCLOK2      #          RETURN
              TC      INTPRET      # (5)
              DLOAD      DSU
                      TIG
                      D29.9SEC
              STCALL      TDEC1
                      INITCDUW
              BOFF      CALL
                      MUNFLAG
                      GOMIDAV
                      CSMPREC
              VLOAD      MXV
                      VATT1
                      REFSMMAT
              VSR1
              STOVL      V(CSM)      # CSM VELOCITY -- M/CS*2(7)
                      RATT1
              VSL4      MXV
                      REFSMMAT
              STCALL      R(CSM)      # CSM POSITION -- M*2(24)
                      MUNGRAV
              STODL      G(CSM)      # CSM GRAVITY VEC. -- M/CS*2(7)
                      TAT
              STORE      TDEC1      # RELOAD TDEC1 FOR MIDTOAV.
              CALRB
              MIDTOAV1
              TCF      CALLT-35      # MADE IT IN TIME.

              EXTEND      # TIG WAS SLIPPED, SO RESET TIG TO 29.9
              DCA      PIPTIME1      # SECONDS AFTER THE TIME TO WHICH WE DID
              DXCH      TIG          # INTEGRATE.
              EXTEND
              DCA      D29.9SEC
              DAS      TIG

CALLT-35      DXCH      MPAC
              DXCH      SAVET-30      # DELTA-T UNTIL TIG-30
              EXTEND
              DCS      5SECDP

```

July 12, 2016

215

```
# Page 735

DAS      SAVET-30      # DELTA-T UNTIL TIG-35
EXTEND
DCA      SAVET-30
TC        LONGCALL
EBANK=   TTOGO
2CADR    TIG-35

TC        PHASCHNG
OCT       20254      # 4.25SPOT FOR TIG-35 RESTART.

TC        CHECKMM
DEC       63
TCF       ENDOFJOB   # NOT P63
CS        CNTDNDEX   # P63 CAN START DISPLAYING NOW.
TS        DISPDEX
TC        INTPRET
VLOAD     ABVAL
          VN1
STORE     ABVEL      # INITIALIZE ABVEL FOR P63 DISPLAY
EXIT
TCF       ENDOFJOB

# *****

TIG-35    CAF        5SEC
          TC         TWIDDLE
          ADRES      TIG-30

          TC         PHASCHNG
          OCT        40154      # 4.15SPOT FOR TIG-30 RESTART

          CS         BLANKDEX   # BLANK DSKY FOR 5 SECONDS
          TS         DISPDEX

          INDEX      WHICH
          CS         6          # CHECK ULLAGE TIME.
          EXTEND
          BZMF       TASKOVER
          CAF        4.9SEC      # SET UP TASK TO RESTORE DISPLAY AT TIG-30
          TC         TWIDDLE
          ADRES      TIG-30.1

          CAF        PRI017     # A NEGATIVE ULLAGE TIME INDICATES P41, IN
          TC         NOVAC      # WHICH CASE WE HAVE TO SET UP A JOB TO
          EBANK=     TTOGO      # BLANK THE DSKY FOR FIVE SECONDS, SINCE
          2CADR      P41BLANK   # CLOKJOB IS NOT RUNNING DURING P41.
```

	TCF	TASKOVER	
P41BLANK	TC	BANKCALL	# BLANK DSKY.
	CADR	CLEANDSP	
	TCF	ENDOFJOB	
TIG-30.1	CAF	PRI017	# SET UP JOB TO RESTORE DISPLAY AT TIG-30
	TC	NOVAC	
	EBANK=	TTOGO	
	2CADR	TIG-30A	
	TCF	TASKOVER	
# Page 736			
TIG-30A	CAF	V16N85B	
	TC	BANKCALL	# RESTORE DISPLAY.
	CADR	REGODSP	# REGODSP DOES A TCF ENDOFJOB
#	*****		
TIG-30	CAF	S24.9SEC	
	TC	TWIDDLE	
	ADRES	TIG-5	
	CS	CNTDNDEX	# START UP CLOKTASK AGAIN
	TS	DISPDEX	
	INDEX	WHICH	# PICK UP APPROPRIATE ULLAGE -- ON TIME
	CA	6	# Was CAF --- RSB 2009.
	EXTEND		
	BZMF	ULLGNOT	# DON'T SET UP ULLAGE IF DT IS NEG OR ZERO
	TS	SAVET-30	# SAVE DELTA-T FOR RESTART
	TC	TWIDDLE	
	ADRES	ULLGTASK	
	CA	THREE	# RESTART PROTECT ULLGTASK (1.3SPOT)
	TS	L	
	CS	THREE	
	DXCH	-PHASE1	
	CS	TIME1	
	TS	TBASE1	
	INDEX	WHICH	
	TCF	1	
WANTAPS	CS	FLGWRD10	# (1) FOR P42 ENSURE APSFLAG IS SET. IF IT

July 12, 2016

217

```

                                MASK    APSFLBIT    # WASN'T SET, DAP WILL BE INITIALIZED TO
                                ADS     FLGWRD10     # ASCENT VALUES BY 1/ACCS IN 2 SECONDS.

ULLGNOT                        EXTEND
                                INDEX    WHICH      # (1)
                                DCA      7          # LOAD AVEGEXIT WITH APPROPRIATE 2CADR
                                DXCH     AVEGEXIT

                                CAF      TWO        # 4.2SPOT RESTARTS IMMEDIATELY AT REDO4.2
                                TS       L          #
                                CS       TWO        # AND ALSO AT TIG-5 AT THE CORRECT TIME.
                                DXCH     -PHASE4

                                CS       TIME1
                                TS       TBASE4     # SET TBASE4 FOR TIG-5 RESTART

REDO2.17                       EXTEND
# Page 737                     DCA      NEG0      # CLEAR OUT GROUP 2 SO LAMBERT CAN START
                                DXCH     -PHASE2     # IF NEEDED.

REDO4.2                        CCS      PHASE5     # IF SERVICER GOING?
                                TCF      TASKOVER    # YES, DON'T START IT UP AGAIN.

                                TC       POSTJUMP
                                CADR     PREREAD     # PREREAD END THIS TASK

#                               *****

ULLGTASK                       TC       ONULLAGE   # THIS COMES AT TIG-7.5 OR TIG-3.5
                                TC       PHASCHNG
                                OCT      1
                                TCF      TASKOVER

#                               *****

TIG-5                          EXTEND
                                DCA      NEG0      # INSURE THAT GROUP 3 IS INACTIVE.
                                DXCH     -PHASE3

                                CAF      5SEC
                                TC       TWIDDLE
                                ADRES    TIG-0

                                TC       DOWNFLAG   # RESET IGNFLAG AND ASINFLAG
                                ADRES    IGNFLAG    #         FOR LIGHT-UP LOGIC.
```

	TC	DOWNFLAG	
	ADRES	ASTNFLAG	
	INDEX	WHICH	
	TCF	11	
P40SJUNK	CCS	PHASE3	# (11) P40 AND P42. S40.13 IN PROGRESS?
	TCF	DISPCHNG	# YES
	CAF	PRI020	
	TC	FINDVAC	
	EBANK=	TTOGO	
	2CADR	S40.13	
	TC	PHASCHNG	# 3.5SPOT FOR S40.13
	OCT	00053	
DISPCHNG	CS	VB99DEX	# (11)
	TS	DISPDEX	
# Page 738			
COMMON	TC	PHASCHNG	# RESTART TIG-0 (4.7SPOT)
	OCT	40074	
	TCF	TASKOVER	
#	*****		
TIG-0	CS	FLAGWRD7	# SET IGNFLAG SINCE TIG HAS ARRIVED
	MASK	IGNFLBIT	
	ADS	FLAGWRD7	
	TC	CHECKMM	# IN P63 CASE, THROTTLE-UP IS ZOOMTIME
	DEC	63	# AFTER NOMINAL IGNITION, NOT ACTUAL
	TCF	IGNYET?	
	CA	ZOOMTIME	
	TC	WAITLIST	
	EBANK=	DVCNTR	
	2CADR	P63ZOOM	
	TC	2PHSCHNG	
	OCT	40033	
	OCT	05014	
	OCT	77777	
IGNYET?	CAF	ASTNBIT	# CHECK ASTNFLAG: HAS ASTRONAUT RESPONDED
	MASK	FLAGWRD7	# TO OUR ENGINE ENABLE REQUEST?

July 12, 2016

219

	EXTEND		
	INDEX	WHICH	
	BZF	12	# BRANCH IF HE HAS NOT RESPONDED YET
IGNITION	CS	FLAGWRD5	# INSURE ENGONFLG IS SET.
	MASK	ENGONBIT	
	ADS	FLAGWRD5	
	CS	PRI030	# TURN ON THE ENGINE.
	EXTEND		
	RAND	DSALMOUT	
	AD	BIT13	
	EXTEND		
	WRITE	DSALMOUT	
	EXTEND		# SET TEVENT FOR DOWNLINK
	DCA	TIME2	
	DXCH	TEVENT	
	EXTEND		# UPDATE TIG USING TGO FROM S40.13
	DCA	TGO	
	DXCH	TIG	
	EXTEND		
	DCA	TIME2	
	DAS	TIG	
# Page 739	CS	FLUNDBIT	# PERMIT GUIDANCE LOOP DISPLAYS
	MASK	FLAGWRD8	
	TS	FLAGWRD8	
	INDEX	WHICH	
	TCF	13	
P63IGN	EXTEND		# (13) INITIATE BURN DISPLAYS
	DCA	DSP2CADR	
	DXCH	AVGEXIT	
	CA	Z	# ASSASSINATE CLOKTASK
	TS	DISPDEX	
	CS	FLAGWRD9	# SET FLAG FOR P70-P71
	MASK	LETABBIT	
	ADS	FLAGWRD9	
	CS	FLAGWRD7	# SET SWANDISP TO ENABLE R10.
	MASK	SWANDBIT	
	ADS	FLAGWRD7	

	CS	PULSES	# MAKE SURE DAP IS NOT IN MINIMUM-IMPULSE
	MASK	DAPBOOLS	# MODE, IN CASE OF SWITCH TO P66
	TS	DAPBOOLS	
	EXTEND		# INITIALIZE TIG FOR P70 AND P71.
	DCA	TIME2	
	DXCH	TIG	
	CAF	ZERO	# INITIALIZE WCHPHASE, AND FLPASSO
	TS	WCHPHASE	
	TS	WCHPHOLD	# ALSO WHCPHOLD
	CA	TWO	
	TS	FLPASSO	
P40IGN	TCF	P42IGN	
	CS	FLAGWRD5	# (13)
	MASK	NOTHRBIT	
	EXTEND		
	BZF	P42IGN	
	CA	ZOOMTIME	
	TC	WAITLIST	
	EBANK=	DVCNTR	
	2CADR	P40ZOOM	
P63IGN1	TC	2PHSCHNG	
	OCT	40033	# 3.3SPOT FOR ZOOM RESTART.
	OCT	05014	# TYPE C RESTARTS HERE IMMEDIATELY
	OCT	77777	
# Page 740			
P12IGN	TCF	P42IGN	
	CAF	EBANK6	
	TS	EBANK	
	EBANK=	AOSQ	
	CA	IGNAOSQ	# INITIALIZE DAP BIAS ACCELERATION
	TS	AOSQ	# ESTIMATES AT P12 IGNITION.
	CA	IGNAOSR	
	TS	AOSR	
	CAF	EBANK7	
	TS	EBANK	
	EBANK=	DVCNTR	
ABRTIGN	CA	Z	# (13) KILL CLOKTASK

	TS	DISPDEX	
	EXTEND		# CONNECT ASCENT GUIDANCE TO SERVICER.
	DCA	ATMAGADR	
	DXCH	AVGEXIT	
	CS	FLAGWRD7	# ENABLE R10.
	MASK	SWANDBIT	
	ADS	FLAGWRD7	
P42IGN	CS	DRIFTBIT	# ENSURE THAT POWERED-FLIGHT SWITCHING
	MASK	DAPBOOLS	# CURVES ARE USED.
	TS	DAPBOOLS	
	CAF	IMPULBIT	# EXAMINE IMPULSE SWITCH
	MASK	FLAGWRD2	
	CCS	A	
	TCF	IMPLBURN	
DVMONCON	TC	DOWNFLAG	
	ADRES	IGNFLAG	# CONNECT DVMON
	TC	DOWNFLAG	
	ADRES	ASTNFLAG	
	TC	DOWNFLAG	
	ADRES	IDLEFLAG	
	TC	PHASCHNG	
	OCT	40054	
	TC	FIXDELAY	# TURN ULLAGE OFF HALF A SECOND AFTER
	DEC	50	# LIGHT UP.
ULLAGOFF	TC	NOULLAGE	
WAITABIT	EXTEND		# KILL GROUP 4
	DCA	NEGO	
# Page 741	DXCH	-PHASE4	
	TCF	TASKOVER	
TIGTASK	TC	POSTJUMP	# (12)
	CADR	TIGTASK1	

	SETLOC	P40S3	
	BANK		
	COUNT*	\$\$/P40	
TIGTASK1	CAF	PRI016	
	TC	NOVAC	
	EBANK=	TRKMKCNT	
	2CADR	TIGNOW	
	TC	PHASCHNG	
	OCT	6	# KILL GROUP 6
	TCF	TASKOVER	

P63ZOOM	EXTEND	
	DCA	LUNLANAD
	DXCH	AVEGEXIT
	TC	IBNKCALL
	CADR	FLATOUT
	TCF	P40ZOOMA

P40ZOOM	CAF	BIT13
	TS	THRUST
	CAF	BIT4

EXTEND	
WOR	CHAN14

P40ZOOMA	TC	PHASCHNG
	OCT	3
	TCF	TASKOVER

	EBANK=	DVCNTR
LUNLANAD	2CADR	LUNLAND

Page 742

ZOOM	=	P40ZOOMA
	BANK	36
	SETLOC	P40S
	BANK	
	COUNT*	\$\$/P40

COMFAIL	TC	UPFLAG	# (15)
	ADRES	IDLEFLAG	
	TC	UPFLAG	# SET FLAG TO SUPRESS CONFLICTING DISPLAY
	ADRES	FLUNDISP	
	CAF	FOUR	# RESET DVMON
	TS	DVCNTR	
	CCS	PHASE6	# CLOKTASK ACTIVE?
	TCF	+3	# YES
	TC	BANKCALL	# OTHERWISE, START IT UP
	CADR	STCLOK1	
+3	CS	VB97DEX	
	TS	DISPDEX	
	TC	PHASCHNG	# TURN OFF GROUP 4.
	OCT	00004	
	TCF	ENDOFJOB	
COMFAIL1	INDEX	WHICH	
	TCF	2	
COMFAIL3	CA	Z	# (15) KILL CLOKTASK USING Z
	TCF	+2	
COMFAIL4	CS	CNTDNDEX	
	TS	DISPDEX	
	TC	DOWNFLAG	# RECONNECT DV MONITOR
	ADRES	IDLEFLAG	
	TC	DOWNFLAG	# PERMIT GUIDANCE LOOP DISPLAYS
	ADRES	FLUNDISP	
	TCF	ENDOFJOB	
COMFAIL2	TC	PHASCHNG	# KILL ZOOM RESTART PROTECTION
	OCT	00003	
	INHINT		
	TC	KILLTASK	# KILL ZOOM IN CASE IT'S STILL TO COME
	CADR	ZOOM	
	TC	IBNKCALL	# COMMAND ENGINE OFF
	CADR	ENGINEOF4	
	TC	UPFLAG	# SET THE DRIFT BIT FOR THE DAP.
	ADRES	DRIFTDFL	
# Page 743	TC	INVFLAG	# USE OTHER RCS SYSTEM
	ADRES	AORBTFLG	
	TC	UPFLAG	# TURN ON ULLAGE

```

        ADRES  ULLAGFLG
        CAF    BIT1
        INHINT
        TC     TWIDDLE
        ADRES  TIG-5
        TCF    ENDOFJOB

#          *****
#          SUBROUTINES OF THE IGNITION ROUTINE
#          *****

INVFLAG    CA      Q
           TC      DEBIT
           COM
           EXTEND
           RXOR    LCHAN
           TCF     COMFLAG

#          *****

NOULLAGE   CS      ULLAGER          # MUST BE CALLED IN A TASK OR UNDER INHINT
           MASK    DAPBOOLS
           TS      DAPBOOLS
           TC      Q

#          *****

ONULLAGE   CS      DAPBOOLS         # TURN ON ULLAGE.  MUST BE CALLED IN
           MASK    ULLAGER          # A TASK OR WHILE INHINTED.
           ADS     DAPBOOLS
           TC      Q

#          *****

STCLOK1    CA      ZERO             # THIS ROUTINE STARTS THE COUNT-DOWN
STCLOK2    TS      DISPDEX          # (CLOKTASK AND CLOKJOB).  SETTING
STCLOK3    TC      MAKECADR         # SETTING DISPDEX POSITIVE KILLS IT.
           TS      TBASE4           # RETURN SAVE (NOT FOR RESTARTS).
           EXTEND
           DCA     TIG
           DXCH    MPAC
           EXTEND
           DCS     TIME2

# Page 744

           DAS     MPAC             # HAVE TIG -- TIME2, UNDOUBTEDLY A + NUMBER
           TC      TPAGREE          # POSITIVE, SINCE WE PASSED THE

```


	CAF	1SEC	# 45 SECOND CHECK.
	TS	Q	
	DXCH	MPAC	
	MASK	LOW5	# RESTRICT MAGNITUDE OF NUMBER IN A
	EXTEND		
	DV	Q	
	CA	L	# GET REMAINDER
	AD	TWO	
	INHINT		
	TC	TWIDDLE	
	ADRES	CLOKTASK	
	TC	2PHSCHNG	
	OCT	40036	# 6.3SPOT FOR CLOKTASK
	OCT	05024	
	OCT	13000	
	CA	TBASE4	
	TC	BANKJUMP	
CLOKTASK	CS	TIME1	# SET TBASE6 FOR GROUP 6 RESTART
	TS	TBASE6	
	CCS	DISPDEX	
	TCF	KILLCLOK	
	NOOP		
	CAF	PRI027	
	TC	NOVAC	
	EBANK=	TTOGO	
	2CADR	CLOKJOB	
	TC	FIXDELAY	# WAIT A SECOND BEFORE STARTING OVER
	DEC	100	
	TCF	CLOKTASK	
KILLCLOK	EXTEND		# KILL RESTART
	DCA	NEGO	
	DXCH	-PHASE6	
	TCF	TASKOVER	
CLOKJOB	EXTEND		
	DCS	TIG	
	DXCH	TTOGO	
	EXTEND		
# Page 745	DCA	TIME2	
	DAS	TTOGO	

		INHINT		
		CCS	DISPDEX	# IF DISPDEX HAS BEEN SET POSITIVE BY A
		TCF	ENDOFJOB	# TASK OR A HIGHER PRIORITY JOB SINCE THE
		TCF	ENDOFJOB	# LAST CLOKTASK, AVOID USING IT AS AN
		COM		# INDEX.
		RELINT		# ***** DISPDEX MUST NEVER B -0 *****
		INDEX	A	
		TCF	DISPNOT -1	# (-1 DUE TO EFFECT OF CCS)
VB97DEX	=	OCT35		# NEGATIVE OF THIS IS PROPER FOR DISPDEX
-35		CS	ZERO	# INDICATE VERB 97 PASTE
		TS	NVWORD1	
		CA	NVWORD +2	# NVWORD+2 CONTAINS V06 & APPROPRIATE NOUN
		TC	BANKCALL	
		CADR	CLOCPLAY	
		TCF	STOPCLOK	# TERMINATE CLOKTASK ON THE WAY TO POOH
		TCF	COMFAIL1	
		TCF	COMFAIL2	
				# THIS DISPLAY IS CALLED VIA ASTNCLOK
-25		CAF	V06N61	# IT IS PRIMARILY USED BY THE CREW IN P63
		TC	BANKCALL	# TO RESET HIS EVENT TIMER TO AGREE WITH
		CADR	REFLASH	# TIG.
		TCF	STOPCLOK	
		TCF	ASTNRETN	
		TCF	-6	
CNTDNDEX	=	LOW4		# OCT17: NEGATIVE PROPER FOR DISPDEX
-17		INDEX	WHICH	# THIS DISPLAY COMES UP AT ONE SECOND
			# Was CAF --- RSB 2009	
		CA	0	# INTERVALS. IT IS NORMALLY OPERATED
		TC	BANKCALL	# BETWEEN TIG-30 SECONDS AND TIG-5 SECONDS
		CADR	REGODSP	# REGODSP DOES ITS OWN TCF ENDOFJOB
VB99DEX	=	ELEVEN		# OCT13: NEGATIVE PROPER FOR DISPDEX
V99RECYC		EQUALS		
-13		CS	BIT9	# INDICATE VERB 99 PASTE
		TS	NVWORD1	
		INDEX	WHICH	# THIS IS THE "PLEASE ENABLE ENGINE"
			# Was CAF --- RSB 2004	
		CA	0	# DISPLAY; IT IS INITIATED AT TIG-5 SEC.
		TC	BANKCALL	# THE DISPLAY IS A V99NXX, WHERE XX IS

	CADR	CLOCPLAY	# NOUN THAT HAD PREVIOUSLY BEEN DISPLAYED
	TCF	STOPCLOK	# TERMINATE GOTOPPOH TURNS OFF ULLAGE.
	TCF	*PROCEED	
	TCF	*ENTER	
# Page 746			
BLANKDEX	=	TWO	# NEGATIVE OF THIS IS PROPER FOR DISPDEX
-2	TC	BANKCALL	# BLANK DSKY. THE DSKY IS BLANKED FOR
	CADR	CLEANDSP	# 5 SECONDS AT TIG-35 TO INDICATE THAT
DISPNOT	TCF	ENDOFJOB	# AVERAGE G IS STARTING.
STOPCLOK	TC	NULLCLOK	# STOP CLOKTASK & TURN OFF ULLAGE ON THE
	TCF	GOTOPPOH	# WAY TO P00 (GOTOPPOH RELINTS)
NULLCLOK	INHINT		
	EXTEND		
	QXCH	P40/RET	
	TC	NOULLAGE	# TURN OFF ULLAGE ...
	TC	KILLTASK	# DON'T LET IT COME ON, EITHER ...
	CADR	ULLGTASK	
	TC	PHASCHNG	# NOT EVEN IF THERE'S A RESTART.
	OCT	1	
	CA	Z	# KILL CLOKTASK
	TS	DISPDEX	
	TC	P40/RET	
ASTNRETN	TC	PHASCHNG	
	OCT	04024	
	CAF	ZERO	# STOP DISPLAYING BUT KEEP RUNNING
	TS	DISPDEX	
	CAF	PRI013	
	TC	FINDVAC	
	EBANK=	STARIND	
	2CADR	ASTNRET	
	TCF	ENDOFJOB	
*PROCEED	TC	UPFLAG	
	ADRES	ASTNFLAG	
	TCF	IGNITE	
*ENTER	INHINT		
	INDEX	WHICH	
	TCF	3	

GOPOST	CAF	PRI012	# (3) MUST BE LOWER PRIORITY THAN CLOKJOB
	TC	FINDVAC	
	EBANK=	TTOGO	
	2CADR	POSTBURN	

Page 747

INHINT		# SET UP THE DAP FOR COASTING FLIGHT.
TC	IBNKCALL	
CADR	ALLCOAST	
TC	NULLCLOK	
TC	PHASCHNG	# 4.13 RESTART FOR POSTBURN
OCT	00134	
TCF	ENDOFJOB	

GOCUTOFF	CAF	PRI017	# (3)
	TC	FINDVAC	
	EBANK=	TGO	
	2CADR	CUTOFF	
	TC	DOWNFLAG	
	ADRES	FLUNDISP	

INHINT		# SET UP THE DAP FOR COASTING FLIGHT.
TC	IBNKCALL	
CADR	ALLCOAST	
TC	NULLCLOK	
TC	PHASCHNG	
OCT	07024	
OCT	17000	
EBANK=	TGO	
2CADR	CUTOFF	
TCF	ENDOFJOB	

IGNITE	CS	FLAGWRD7	# (2)
	MASK	IGNFLBIT	
	CCS	A	
	TCF	IGNITE1	
	CAF	BIT1	
	INHINT		
	TC	TWIDDLE	
	ADRES	IGNITION	

CAF	OCT23	# IMMEDIATE RESTART AT IGNITION
-----	-------	---------------------------------

July 12, 2016

229

```

      TS      L
      COM
      DXCH    -PHASE4

IGNITE1      CS      CNTDNDEX      # RESTORE OLD DISPLAY.
              TS      DISPDEX

              TCF      ENDOFJOB

# Page 748
# *****

P40ALM        TC      ALARM          # PROGRAM SELECTION NOT CONSISTENT WITH
              OCT      1706          # VEHICLE CONFIGURATION

REP40ALM      CAF      V05N09        # (14)
              TC      BANKCALL
              CADR      GOFLASH

              TCF      GOTOP00H      # V34E          TERMINATE
              TCF      +2            # PROCEED        CHECK FOR P42
              TCF      REP40ALM      # V32E          REDISPLAY ALARM

              INDEX    WHICH        # FOR P42, ALLOW CREW TO PRECEED EVEN
              TCF      14            # THOUGH VEHICLE IS UNSTAGED.

# *****

      BANK      31
      SETLOC    P40S2
      BANK

      COUNT*    $$/P40

P40AUTO      TC      MAKECADR        # HELLO THERE.
              TS      TEMPR60        # FOR GENERALIZED RETURN TO OTHER BANKS.
P40A/P      TC      BANKCALL        # SUBROUTINE TO CHECK PGNC S CONTROL
              CADR      G+N,AUTO      # AND AUTO STABILIZATION MODES
              CCS      A              # +0 INDICATES IN PGNC S, IN AUTO
              TCF      TURNITON      # + INDICATES NOT IN PGNC S AND/OR AUTO
              CAF      APSFLBIT      # ARE WE ON THE DESCENT STAGE?
              MASK      FLGWRD10
              CCS      A
              TCF      GOBACK        # RETURN
              CAF      BIT5          # YES, CHECK FOR AUTO-THROTTLE MODE
              EXTEND
```

	RAND	CHAN30	
	EXTEND		
	BZF	GOBACK	# IN AUTO-THROTTLE MODE -- RETURN
TURNITON	CAF	P40A/PMD	# DISPLAYS V50N25 R1=203 PLEASE PERFORM
	TC	BANKCALL	# CHECKLIST 203 TURN ON PGNC S ETC.
	CADR	GOPERF1	
	TCF	GOTOP00H	# V34E TERMINATE
	TCF	P40A/P	# RECYCLE
GOBACK	CA	TEMPR60	
	TC	BANKJUMP	# GOODBYE. COME AGAIN SOON.
P40A/PMD	OCT	00203	

Page 749

BANK	36
SETLOC	P40S
BANK	
COUNT*	\$\$/P40

```
# *****
#   CONSTANTS FOR THE IGNITION ROUTINE
#   *****
```

SERVCADR	=	P63TABLE +7
P40ADRES	ADRES	P40TABLE
P41ADRES	ADRES	P41TABLE -5
P42ADRES	ADRES	P42TABLE
DSP2CADR	EBANK=	DVCNTR
	2CADR	P63DISPS -2
ATMAGADR	EBANK=	DVCNTR
	2CADR	ATMAG
?	=	GOTOP00H
D29.9SEC	2DEC	2990
S24.9SEC	DEC	2490
4.9SEC	DEC	490

OCT20 = BIT5

V06N61 VN 0661

Page 750

KILLTASK

MOD NO: NEW PROGRAM

MOD BY: COVELLI

#

FUNCTIONAL DESCRIPTION:

#

KILLTASK IS USED TO REMOVE A TASK FROM THE WAITLIST BY SUBSTITUTING A NULL TASK CALLED
WHICH MERELY DOES A TC TASKOVER. IF THE SAME TASK IS SCHEDULED MORE THAN ONCE, ONLY THE
FIRST IS REMOVED. IF THE TASK IS NOT SCHEDULED, KILLTASK TAKES NO ACTION AND RETURNS W
LEAVES INTERRUPTS INHIBITED SO CALLER MUST RELINT

#

CALLING SEQUENCE

L TC KILLTASK # IN FIXED-FIXED

L+1 CADR ???????? # CADR (NOT 2CADR) OF TASK TO BE REMOVED.

L+2 (RELINT) # RETURN

#

EXIT MODE: AT L+2 OF CALLING SEQUENCE.

#

ERASABLE INITIALIZATION: NONE.

#

OUTPUT: 2CADR OF NULLTASK IN LST2

#

DEBRIS: ITEMP1 - ITEMP4, A, L, Q.

	EBANK=	LST2	
	BLOCK	3	# KILLTASK MUST BE IN FIXED-FIXED.
	SETLOC	FFTAG6	
	BANK		
	COUNT*	\$\$/KILL	
KILLTASK	CA	KILLBB	
	INHINT		
	LXCH	A	
	INDEX	Q	
	CA	0	# GET CADR.
	LXCH	BBANK	
	TCF	KILLTSK2	# CONTINUE IN SWITCHED FIXED.
	EBANK=	LST2	
KILLBB	BBCON	KILLTSK2	
	BANK	27	

```

SETLOC P40S1
BANK
COUNT* $$/KILL

KILLTSK2      LXCH  ITEMP2      # SAVE CALLER'S BBANK
# Page 751

            INCR  Q
            EXTEND
            QXCH  ITEMP1      # RETURN 2ADR IN ITEMP1,ITEMP2

            TS    ITEMP3      # CADR IS IN A
            MASK  LOW10
            AD    BIT11
            TS    ITEMP4      # GENADR OF TASK

            CS    LOW10
            MASK  ITEMP3
            TS    ITEMP3      # FBANK OF TASK

ADRSCAN      ZL
            INDEX L
            CS    LST2
            AD    ITEMP4      # COMPARE GENADRS
            EXTEND

LETITLIV     BZF  TSTFBANK      # IF THEY MATCH, COMPARE FBANKS
            CS    LSTLIM
            AD    L
            EXTEND      # ARE WE DONE?
            BZF  DEAD          # YES -- DONE, SO RETURN
            INCR L
            INCR L
            TCF  ADRSCAN      # CONTINUE LOOP.

DEAD         DXCH  ITEMP1
            DTCB

TSTFBANK     CS    LOW10
            INDEX L
            MASK  LST2      +1  # COMPARE FBANKS ONLY.
            EXTEND
            SU    ITEMP3
            EXTEND
            BZF  KILLDEAD      # MATCH -- KILL IT.
            TCF  LETITLIV      # NO MATCH -- CONTINUE.

```



```
KILLDEAD      CA      TCTSKOVR
               INDEX   L
               TS      LST2      # REMOVE TASK BY INSERTING TASKOVER
               TCF      DEAD

LSTLIM        EQUALS   BIT5      # DEC 16
This code is written to file src/BURN-BABY-BURN--MASTER-IGNITION-ROUTINE.s.
```

B.15 CM BODY ATTITUDE

```

234  <src/CM-BODY-ATTITUDE.s 234>≡
      # Copyright:    Public domain.
      # Filename:     CM_BODY_ATTITUDE.agc
      # Purpose:      Part of the source code for Comanche, build 055. It
      #               is part of the source code for the Command Module's
      #               (CM) Apollo Guidance Computer (AGC), Apollo 11.
      # Assembler:    yaYUL
      # Reference:     pp. 883-889
      # Contact:       Ron Burkey <info@sandroid.org>
      # Website:       http://www.ibiblio.org/apollo.
      # Mod history:   2009-05-12 RSB  Adapted from Colossus249 file of the same
      #               name and Comanche 055 page images.
      #
      # The contents of the "Comanche055" files, in general, are transcribed
      # from scanned documents.
      #
      # Assemble revision 055 of AGC program Comanche by NASA
      # 2021113-051. April 1, 1969.
      #
      # This AGC program shall also be referred to as Colossus 2A
      #
      # Prepared by
      #               Massachusetts Institute of Technology
      #               75 Cambridge Parkway
      #               Cambridge, Massachusetts
      #
      # under NASA contract NAS 9-4065.
      #
      # Refer directly to the online document mentioned above for further
      # information. Please report any errors to info@sandroid.org.

      # Page 883

      BANK      35

      SETLOC    BODYATT
      BANK

      COUNT     37/CMBAT

      # PDL 12D - 15D SAFE.

      # VALUE OF GIMBAL AND BODY ANGLES VALID AT PIP TIME ARE SAVED DURING READACCS.

      EBANK=    RTINIT          # LET INTERPRETER SET EB

```

July 12, 2016

235

CM/POSE	TC	INTPRET	# COME HERE VIA AVEGEXIT.	
	SETPD	VLOAD		
		0		
		VN	# KVSACLE = (12800/ .3048) /2VS	
	VXSC	PDVL		
		-KVSACLE	# KVSACLE = .81491944	
		UNITW	# FULL UNIT VECTOR	
	VXV	VXSC	# VREL = V - WE*R	
		UNITR		
		KWE		
	VAD	STADR		
	STORE	-VREL	# SAVE FOR ENTRY GUIDANCE.	REF COORDS
	UNIT	LXA,1		
		36D	# ABVAL(-VREL) TO X1	
	STORE	UXA/2	# -UVREL	REF COORDS
	VXV	VCOMP		
		UNITR	# .5 UNIT	REF COORDS
	UNIT	SSP	# THE FOLLOWING IS TO PROVIDE A STABLE	
		S1	# UN FOR THE END OF THE TERMINAL PHASE.	
SPVQUIT	DEC	.019405	# 1000/ 2 VS	
	TIX,1	VLOAD	# IF V-VQUIT POS, BRANCH.	
		CM/POSE2	# SAME UYA IN OLDUYA	
		OLDUYA	# OTHERWISE CONTINUE TO USE OLDUYA	
CM/POSE2	STORE	UYA/2	#	REF COORDS
	STORE	OLDUYA	# RESTORE, OR SAVE AS CASE MAY BE.	
	VXV	VCOMP		
		UXA/2	# FINISH OBTAINING TRAJECTORY TRIAD.	
	VSL1			
	STORE	UZA/2	#	REF COORDS
# Page 884	TLOAD		# PICK UP CDUX, CDUY, CDUZ CORRESPONDING	
		AOG/PIP	# TO PIPUP TIME IN 2'S C AND SAVE.	
CM/TRIO	STODL	24D		
		25D	# AIG/PIP	
	RTB	PUSH	# TO PDL0	
		CDULOGIC		
	COS			
	STODL	UBX/2	# CI /2	
			# AIG/PIP FROM PDL 0	

SIN	DCOMP		
STODL	UBX/2 +4	# -SI /2	
	26D	# AMG/PIP	
RTB	PUSH	# TO PDL 0	
	CDULOGIC		
SIN	PDDL	# XCH PDL 0. SAVE SM /2	
COS	PDDL	# CM /2 TO PDL 2	
	0	# SM /2	
DCOMP	VXSC		
	UBX/2		
VSL1		# NOISE WON'T OVFL	
STODL	UBY/2	# =(-SMCI, NOISE, SMSI)/2	
	2	# CM /2 REPLACES NOISE	
STODL	UBY/2 +2	# UBY/2=(-SMCI, CM, SMSI)/2	
	24D	# AOG/PIP	
RTB	PUSH	# TO PDL 4	
	CDULOGIC		
SIN	PDDL	# XCH PDL 4. SAVE SO /2	
COS	VXSC	# CO /2	
	UBY/2		
STODL	UBY/2	# UBY/2=(-COSMCI, COCM, COSMSI)/4	
	4D	# SO /2	
DMP	DCOMP		
	UBX/2 +4	# -SI /2	
DAD			
	UBY/2	# INCREMENT BY (SOSI /4)	
STODL	UBY/2		
		# SO /2 FROM PDL 4	
DMP	DAD		
	UBX/2	# CI /2	
	UBY/2 +4		
STOVL	UBY/2 +4	# YB/4	PLATFORM COORD
		# YB = (-COSMCI + SOSI , COCM , COSMSI + SOC	
	UBY/2		
VXM	VSL2		
	REFSMMAT	# .5 UNIT	
STODL	UBY/2	# YB/2 DONE	REF COORDS
		# CM /2 FROM PDL 2	
VXSC	VSL1		
	UBX/2		
STODL	UBX/2	# =(CMCI, NOISE, -CMSI)/2	
STADR		# SM /2 FROM PDL 0	
STOVL	UBX/2 +2	# SM /2 REPLACES NOISE	

	UBX/2	# XB/2	PLATFORM COORDS
		# XB = (CMCI , SM , -CMSI)	
VXM	VSL1		
	REFSMMAT	# .5 UNIT	
STORE	UBX/2	# XB/2 DONE	REF COORDS
VXV	VSL1		
	UBY/2		
STOVL	UBZ/2	# ZB/2 DONE	REF COORDS
		# EQUIVALENT TO	
		# ZB = (SOSMCI + COSI , -SOCM , -SOSMSI + COCI)	
	UXA/2	# -UVREL/2 = -UVA/2	
VXV	UNIT	# GET UNIT(-UVREL*UBY)/2 = UL/2	
	UBY/2	# YB/2	
PUSH	DOT	# UL/2 TO PDL 0,5	
	UZA/2	# UNA/2	
STOVL	COSTH	# COS(ROLL)/4	
	0	# UL/2	
DOT			
	UYA/2		
STCALL	SINTH	# -SIN(ROLL)/4	
	ARCTRIG		
STOVL	6D	# -(ROLL/180) /2	
	UBY/2		
DOT	SL1	# -UVA.UBY = -SIN(BETA)	
	UXA/2	# -UVREL/2	
ARCSIN			
STOVL	7D	# -(BETA/180) /2	
	UBX/2	# XB/2	
DOT		# UL.UBX = -SIN(ALFA)	
	0	# UL/2	
STOVL	SINTH	# -SIN(ALFA)/4	
DOT		# UL/2 FROM PDL 0	
	UBZ/2		
STCALL	COSTH	# COS(ALFA)/2	
	ARCTRIG		
STOVL	8D	# -(ALFA/180) /2	
	UNITR	# UR/2	REF COORDS
DOT	SL1		
	UZA/2	# MORE ACCURATE AT LARGE ARG.	

```

      ARCCOS
STORE  10D          # (-GAMA/180)/2

      TLOAD  EXIT    # ANGLES IN MPAC IN THE ORDER
                     # -( (ROLL, BETA, ALFA) /180)/2
                     # THESE VALUES CORRECT AT PIPUP TIME.
                     6D

# Page 887
# BASIC SUBROUTINE TO UPDATE ATTITUDE ANGLES

      EBANK=  AOG

CM/ATUP  CA      EBAOG
        TS      EBANK
CMTR1    INDEX   FIXLOC
        CS      10D          # (GAMA/180)/2
        XCH     GAMA
        TS      L

      INHINT

                                     # MUST REMAIN INHINTED UNTIL UPDATE OF BODY
                                     # ANGLES, SO THAT GAMDIFSW IS VALID FIRST PAS
                                     # INDICATOR.

      CS      CM/FLAGS
      MASK    BIT11          # GAMDIFSW=94D BIT11      INITLY=0
      EXTEND                                     # DON'T CALC GAMA DOT UNTIL HAVE FORMD
                                     # ONE DIFFERENCE.
      BZF     DOGAMDOT        # IS OK, GO ON.
      ADS     CM/FLAGS        # KNOW BIT IS 0
      TC      NOGAMDOT        # SET GAMDOT = 0

DOGAMDOT CS      L
        AD      GAMA          # DEL GAMA/360= T GAMDOT/360
        EXTEND
        MP      TCDU          # TCDU = .1 SEC, T = 2 SEC.
        TS      GAMDOT        # GAMA DOT TCDU / 180

      EXTEND                                     # IGNORE GAMDOT IF LEQ .5 DEG/SEC
      BZMF     +2
      COM
      AD      FIVE
      EXTEND
      BZMF     +3          # SET GAMDOT=+0 AS TAG IF TOO SMALL.

NOGAMDOT CA      ZERO          # COME HERE INHINTED

```

```

# Page 888
TS      GAMDOT
        # FOR NOW LEAVE IN 2'S C
        # UPDATE ANGLES BY CORRECTING EULER ANG
        # FOR ACCRUED INCREMENT SINCE PIPUP
        # R = R EUL + R(NOW) - R(PIPUP)
CS      MPAC
DOUBLE
TC      CORANGOV
EXTEND
SU      ROLL/PIP
AD      ROLL/180
TC      CORANGOV
        # GET INCR SINCE PIPUP
        # ONLY SINGLE OVFL POSSIBLE.
        # CORRECT FOR OVFL IF ANY

TS      TEMPROLL

CS      MPAC +2
DOUBLE
TC      CORANGOV
EXTEND
SU      ALFA/PIP
AD      ALFA/180
TC      CORANGOV
        # CORRECT FOR OVFL IF ANY
TS      TEMPALFA

CMTR2
CS      MPAC +1
DOUBLE
EXTEND
SU      BETA/PIP
AD      BETA/180
XCH     TEMPBETA
        # OVFL NOT EXPECTED.

CA      EBANK3
TS      EBANK

EBANK=  PHSNAME5
EXTEND
DCA     REPOSADR
DXCH    PHSNAME5
        # THIS ASSUMES THAT THE      TC PHASCHNG
        # IS NOT CHANGED IN          OCT 10035
        # SERVICER.

CA      EBAOG
TS      EBANK

EBANK=  AOG
EXTEND
DCA     TEMPROLL
        # RE-STARTS COME HERE

```

	DXCH	ROLL/180	
	CA	TEMPBETA	
	TS	BETA/180	
	RELINT		
CM/POSE3	TC	INTPRET	# CAN'T TC DANZIG AFTER PHASCHNG.
	VLOAD	ABVAL	# RETURN FROM CM/ATUP. (RESTART)
		VN	# 2(-7) M/CS
	STORE	VMAGI	# FOR DISPLAY ON CALL.
	GOTO		
		POSEXIT	# ENDEXIT, STARTENT, OR SCALEPOP.
CORANGOV	TS	L	
	TC	Q	
	INDEX	A	
# Page 889			
	CA	LIMITS	
	ADS	L	
	TC	Q	# COSTS 2 MCT TO USE. SEE ANGOVCOR.
-KVSCALE	2DEC	-.81491944	# -12800/(2 VS .3048)
TCDU	DEC	.1	# TCDU = .1 SEC.
	EBANK=	AOG	
REPOSADR	2CADR	REDOPOSE	

This code is written to file `src/CM-BODY-ATTITUDE.s`.

B.16 CM ENTRY DIGITAL AUTOPILOT

241

<src/CM-ENTRY-DIGITAL-AUTOPILOT.s 241>≡

```
# Copyright:      Public domain.
# Filename:       CM_ENTRY_DIGITAL_AUTOPILOT.agc
# Purpose:       Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1063-1092
# Mod history:   2009-05-13 RSB   Adapted from the Colossus249/ file of the
#               same name, using Comanche055 page images.
#               2009-05-20 RSB   Corrections: Removed an extraneous label
#               EXDAPIN, added a missing instruction in
#               COMPAT.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#               Colossus 2A
#
# Page 1063
# SUBROUTINE TO READ GIMBAL ANGLES AND FORM DIFFERENCES.  GIMBAL ANGLES ARE SAVED IN 2'S COMPLE
# DIFFERENECES ARE IN 1'S COMP.  ENTER AND READ ANGLES EACH .1 SEC.
#
#       CM/DSTBY = 1 FOR DAP OPERATION
#       CM/DSTBY = 0 TO TERMINATE DAP OPERATION
#
# BANK      15
#
# SETLOC  ETRYDAP
# BANK
```

```

COUNT    15/DAPEN

EBANK=    AOG

READGYMB   CA      TEN      # KEEP RESTART DT GOING RELATIVE TO
ADS        CM/GYMDT      # PIPTIME.  (GROUP 6)

# IF A RESTART OCCURS, SKIP PRESENT CYCLE.
# PHASCHNG PROTECTION IS IN CM/DAPIC.

CA      BIT6      # CHECK FOR FINE ALIGN MODE OF CDU.
MASK    IMODES33  # (PROTECT AOG/PIP ETC AS WELL AS
EXTEND   # GIMBAL DIFFERENCES)
BZF      READGYM1 # OK

CS      BIT1      # NOT IN FINE ALIGN, SO IDLE
MASK    CM/FLAGS  # SET GYMDIFSW = 0
TS      CM/FLAGS
TC      FLUSHJET
TC      CM/GYMIC +2 # QUENCH JETS, SINCE MAY BE A WHILE.

READGYM1   CA      CDUX
XCH      AOG
EXTEND
MSU      AOG      # -DELAOG=AOG(N-1) - AOG(N)
TS      -DELAOG

CA      CDUY
XCH      AIG
EXTEND
MSU      AIG
TS      -DELAIG

CA      CDUZ
XCH      AMG
EXTEND
MSU      AMG
TS      -DELAMG

# Page 1064
DOBRATE?   CS      CM/FLAGS      # CM/DSTBY=103D BIT2  GYMDIFSW=104D BIT1
MASK      THREE
INDEX     A
TC        +1
TC        DOBRATE      # OK, GO ON

```

July 12, 2016

243

	TC	CM/GYMIC	# DON'T CALC BODYRATE ON FIRST PASS.
	NOOP		
	TC	FLUSHJET	# TURN OFF ALL JETS
	TC	PHASCHNG	
	OCT	00006	# DEACTIVATE DAP GROUP 6.
	TC	TASKOVER	
DOBRATE	CA	ONE	# DO BODYRATE
DOBRATE1	TS	JETEM	# SKIP BODYRATE
	CA	TEN	# KEEP CDU READ GOING.
	TC	WAITLIST	
	EBANK=	AOG	
	2CADR	READGYMB	
			# DOES NOT PROTECT TEMK, SQ IN SPSIN/COS
	CCS	JETEM	
	TC	BODYRATE	
	TC	TASKOVER	# SKIP CALC ON INITIAL PASS. (PASSES)
CM/GYMIC	ADS	CM/FLAGS	# GYMDIFSW: C(A)=1, KNOW BIT IS 0
	CAF	ZERO	
	TS	JETAG	
	TS	OLDELP	
	TS	OLDELQ	
	TS	OLDELR	
	TS	GAMDOT	# NO GYM DIF, PROB NO GAM DIF.
	TC	DOBRATE1	
# Page 1065			
# COME HERE TO CORRECT FOR OVERFLOW IN ANGULAR CALCULATIONS			
ANGOVCOR	TS	L	# THIS COSTS 2 MCT TO USE.
	TC	Q	# NO OVFL
	INDEX	A	
	CAF	LIMITS	
	ADS	L	
	TC	Q	
	BLOCK	3	
	COUNT	03/DAPEN	

```

FLUSHJET      CA      7      # COME HERE TO TURN OFF ALL JETS.
              EXTEND
              WRITE    ROLLJETS  # ZERO CHANNEL 6
              EXTEND
              WRITE    PYJETS    # ZERO CHANNEL 5
              TC      Q

              BANK      15

              COUNT    15/DAPEN

              SETLOC   ETRYDAP
              BANK

RATEAVG       COM      # SUBROUTINE TO ESTIMATE RATES IN PRESENCE
              AD      JETEM  # OF CONSTANT ACCELERATION.
              EXTEND
              MP      HALF   # DELV (EST) = DELV +(DELV-OLDELV)/2
              AD      JETEM
              TC      Q

```

Page 1066

THESE ARE CALLED FOR THE VARIOUS INITIALIZATIONS NEEDED.

```

              BANK      20
              SETLOC   DAPS1
              BANK

              COUNT    20/DAPEN
              EBANK=   AOG

CM/DAPON      CA      EBAOG
              TS      EBANK

              TC      DOWNFLAG  # RESET DAPBIT1.  T5 RESTART IDENTIFIER.
              ADRES   DAPBIT1   # BIT 15 FLAG 6          CMFLAGS.
              TC      DOWNFLAG  # RESET DAPBIT2
              ADRES   DAPBIT2   # BIT 14 FLAG 6
              EXTEND
              DCA      T5IDLER1  # DISABLE RCS CALCULATION
              DXCH     T5LOC
              EXTEND
              DCA      T5IDLER1  # DISABLE RCS JET CALLS
              DXCH     T6LOC

              TC      FLUSHJET   # JETS DEPARTED ON SM. ZERO JET BITS.

```

```

CS      13,14,15
MASK    DAPDATR1      # SET CONFIG BITS=0 FOR ENTRY
TS      DAPDATR1
TC      +4

NOTYET  CA      .5SEC
        TC      BANKCALL
        CADR    DELAYJOB      # (DELAYJOB DOES INHINT)
        +4     CA      BIT11    # GAMDIFSW = 94D BIT11, INITLY=0
        MASK    CM/FLAGS      # IF ZERO, WAIT UNTIL CM/POSE UPDATE.
        EXTEND
        BZF     NOTYET

CS      ONE           # ACTIVATE CM/DAP
TS      RCSFLAGS      # USE BIT3 TO INITIALIZE NEEDLER ON
                        # NEXT PASS.
TS      P63FLAG       # SO WAKEP62 WILL NOT BE INITIATED UNTIL
                        # HEADSUP IS SET IN P62.

                        # FLAG TO PREVENT MULTIPLE CALLS TO WAKEP62

CA      7
TS      JETAG
TS      PAXERR1       # KEEP NEEDLES ZERO UNTIL DAP UPDATE
                        # IN CASE CMDAPMOD IS NOT +1.

# Page 1067

INHINT
EXTEND
DCA     ALFA/180      # DO ATTITUDE HOLD UNTIL KEYBOARD
DXCH    ALFACOM        # ESTABLISHES HEADSUP.
CA      ROLL/180
TS      ROLLHOLD      # FOR ATTITUDE HOLD IN MODE +1.
EXTEND
MP      HALF
TS      ROLLC         # NOT INTERESTED IN LO WORD.

CS      CM/FLAGS
MASK    BIT12         # CMDAPARM =93D BIT12  INITLY=0
ADS     CM/FLAGS      # SET BIT TO 1.

CS      FLAGWRD2      # SET NODOFLAG TO PREVENT FURTHER
MASK    BIT1          # V 37 ENTRIES.
ADS     FLAGWRD2

RELINT

```

TC POSTJUMP
 CADR P62.1

Page 1068

INITIALIZE CM/DAP. WAITLIST CALL FOR READGYMB. SET SWITCH CM/DSTBY =1
 # SO READACCS WILL ENTER A WTLST CALL FOR SETJTAG .
 # CMDAPARM = 0, SO ONLY BODY RATE AND ATTITUDE CALCULATIONS ARE DONE.
 # SET AVEGEXIT TO CONTINUE AT CM/POSE

CM/DAPIC CA EBAOG
 TS EBANK

CM/DAP2C INHINT
 CS PIPTIME +1

PRIO OF P62 L PRIO AVG.:PIPTM=PIPTM1.

TS JETEM

CA POS1/2

AD POS1/2

AD TIME1

OVFL GUARANTEED

ADS JETEM

C(A) = DELTA TIME SINCE PIPUP

CS FIVE

AD JETEM

CCS A

AD -CDUT+1

TCF -2

NOOP

AD ONE

SEND NO ZERO TO WTLST

TS CM/GYMDT

FOR RESTART

TC WAITLIST

EBANK= AOG

2CADR READGYMB

CS CM/SWIC1

GAMDIFSW, GYMDIFSW, CM/DSTBY

MASK CM/FLAGS

DAPARM, .05GSW, LATSW, ENTRYDSP

AD CM/SWIC2

SET CM/DSTBY, LATSW

DISABLE ENTRY DISPLAY, SINCE DES. GIMB.

CALC. (P62.3) GOES TO ENDEXIT

TS CM/FLAGS

CA 7

TS BETA/180

NECESSARY: NO OVFL CORRECTIO

CA ONE

INITIALIZE THE TM OF BODY RATES VIA

TS SW/NDX

UPBUFF.

```

TC      2PHSCHNG      # DOES INHINT/RELINT
OCT     40116         # SAVE TBASE6
OCT     05024
OCT     13000

TC      POSTJUMP

# Page 1069

CADR    P62.2

CM/SWIC1    OCT     16017
CM/SWIC2    =       TEN      # 00012: CM/DSTBY, LATSW
-CDUT+1     OCT     77766
EBANK=     T5LOC
T5IDLER1    2CADR   T5IDLOC

```

Page 1070

```

# THIS SECTION CALCULATES THE ANGULAR BODY RATES EACH .1 SEC.  THE ANGULAR RATES ARE THOSE ALON
# XB, YB, ZB, AND ARE NORMALLY DESIGNATED P, Q, R.      REQUIREMENT:  TEMPORARILY ERASE.  JETEM,
#
# SINCE RESTARTS ZERO THE JET OUTPUT CHANNELS, NO ATTEMPT IS MADE TO RESTART THE ENTRY DAPS.  T
# THE 0.1 SEC DAPS WILL MISS A CYCLE, AND WILL PICK UP AT THE NEXT 0.1 SEC UPDATE.  MOST OF THE
# ROLL SYSTEM WILL MISS ONLY 0.1 SEC OF CONTROL.  HOWEVER, IF THE RESTART OCCURS AFTER THE SECT
# STARTED, THEN THE ROLL SYSTEM WILL MISS ONE CYCLE.
# THIS IS NECESSARY UNDER THE GROUND-RULE THAT NO JET COMMANDS SHALL BE LESS THAN 14 MS.

```

```

EBANK=  AOG
BANK    15
SETLOC  ETRYDAP
BANK

COUNT  15/DAPEN

BODYRATE CA      AMG      # THESE ARE 2'S COMPL NOS, BUT USE ANYWAY.
         TC      SPCOS
         TS      COSM

         CA      AOG      # C(AOG) = AOG/180
         TC      SPSIN    # SINO
         TS      SINO     # SINO = SIN(AOG)

EXTEND
MP      COSM
TS      SINOCOSM      # SO CM

CA      AOG

```

TC SPCOS # COSO
 TS COSO

EXTEND
 MP COSM
 TS COSOCOSM # CO CM

PITCHDOT: $Q \text{ TCDU}/180 = \text{IDOT TCDU}/180 \text{ COSO COSM} + \text{MDOT TCDU}/180 \text{ SINO}$

CS -DELAMG
 EXTEND
 MP SINO
 DXCH JETEM # 2 LOCS
 CS -DELAIG
 EXTEND
 MP COSOCOSM
 DAS JETEM
 CA JETEM
 XCH OLDELQ
 TC RATEAVG
 TS QREL # PITCHDOT = $Q \text{ TCDU}/180$

Page 1071

YAWDOT: $R \text{ TCDU}/180 = -\text{IDOT TCDU}/180 \text{ COSM SINO} + \text{MDOT TCDU}/180 \text{ COSO}$

CS -DELAMG
 EXTEND
 MP COSO
 DXCH JETEM
 CA -DELAIG
 EXTEND
 MP SINOCOSM
 DAS JETEM
 CA JETEM
 XCH OLDELQ
 TC RATEAVG
 TS RREL # YAWDOT = $R \text{ TCDU}/180$

ROLLDOT: $P \text{ TCDU}/180 = \text{ODOT TCDU}/180 + \text{IDOT TCDU}/180 \text{ SINM}$

CA AMG
 TC SPSIN
 TS SINM
 EXTEND
 MP -DELAIG


```

TS      JETEM
CA      ZERO
DDOUBL                                     # ROUND L INTO A
AD      -DELAOG
AD      JETEM
CS      A
TS      JETEM
XCH     OLDELP
TC      RATEAVG
TS      PREL                             # ROLLDOT = P TCDU/180

                                           # IF GAMDOT < 0.5 DEG/SEC, THEN GAMDOT =0

CCS     GAMDOT
TC      +2
TC      NOGAMDUT
CS      ROLL/180
TC      SPSIN
EXTEND
MP      GAMDOT
TS      JETEM +1                         # -SR GAMDOT
EXTEND
MP      SINTRIM                         # SIN(-20)      (FOR NOMINAL L/D = .3)
ADS     PREL                             # PREL TCDU/180=(P-SALF SR GAMDOT)TCDU/180

CA      ROLL/180
TC      SPCOS

# Page 1072
COM
EXTEND
MP      GAMDOT
ADS     QREL                             # QREL TCDU/180=(Q-CR GAMDOT) TCDU/180

CS      JETEM +1                         # B( ) = -SR GAMDOT
EXTEND
MP      COSTRIM                         # COS(-20)      (FOR NOMINAL L/D = .3)
ADS     RREL                             # RREL TCDU/180=(R+CALF SR GAMDOT)TCDU/180

NOGAMDUT CA      BIT12                     # CMDAPARM = 93D BIT 12
MASK     CM/FLAGS
EXTEND
STBYDUMP BZF     TASKOVER                 # DAP NOT ARMED.

CA      POSMAX                         # PICK UP AT ATTRATES IN 10 MS OR SO.
TS      TIME5

```

```

EXTEND
DCA      ATDOTCAD
DXCH     T5LOC
# DOES NOT PROTECT TEMK, SQ IN SPSIN/COS

TC        TASKOVER

EBANK=    AOG
ATDOTCAD  2CADR  ATTRATES

# Page 1073
# CALCULATE BODY ATTITUDE RATES AND INTEGRATE TO OBTAIN ATTITUDE ANGLES.
#
#      CB PHIDOT TCDU/180 = (CA PREL + SA RREL) TCDU/180
#      BETADOT TCDU/180 = (-SA PREL + CA RREL) TCDU/180
#      ALFADOT TCDU = (QREL + SB PHIDOT) TCDU/180

ATTRATES  LXCH    BANKRUPT      # CONTINUE HERE VIA T5
EXTEND
QXCH      QRUPT
CA         SR
DOUBLE
TS         CM/SAVE
# DOES NOT PROTECT TEMK, SQ IN SPSIN/COS

CA         QREL
AD         ALFA/180
TC         ANGOVCOR
TS         ALFA/180
TC         SPCOS
TS         CALFA      # CALFA
TS         PHIDOT

EXTEND
MP         PREL
XCH        PHIDOT      # CA PREL
EXTEND
MP         RREL        # CA RREL
TS         BETADOT

CA         ALFA/180
TC         SPSIN
TS         SALFA      # SIN(ALFA)

EXTEND
MP         RREL        # SA RREL

```

July 12, 2016

251

```
ADS      PHIDOT      # CB PHIDOT, SAVED.

CS      SALFA
EXTEND
MP      PREL
ADS      BETADOT      # SAVE BETADOT TCDU/180
ADS      BETA/180      # BETA DONE.

TC      SPSIN
EXTEND
MP      PHIDOT      # NEGLECT CB IN CB PHIDOT
AD      ALFA/180
TC      ANGOVCOR
TS      ALFA/180      # ALFA DONE.

# Page 1074
COM
AD      ALFACOM
TC      ANGOVCOR      # JUST IN CASE ...
TS      AK1
TS      QAXERR      # FOR PITCH FDAI AND EDIT.

CA      PHIDOT      # PHIDOT TCDU/180, NEGLECTING CB
AD      ROLL/180
TC      ANGOVCOR
TS      ROLLTM      # ROLL/180 FOR TM.
TS      ROLL/180      # ROLL DONE.

# START YAW AUTOPILOT HERE.  RATE DAMPING WITH ENFORCED COORDINATED ROLL MANEUVER.

CS      BETA/180      # IF IN ATM, SAVE 'RAXERR' FOR TM DNLST.
AD      BETACOM
TS      RAXERR      # IF OUTSIDE ATM, USE TM REGISTER 'RAXERR'
                        # AS A TEMPORARY. (DAP OPERATION IS IN INTERRUPT, SO
                        # IS OK.) FINAL C(RAXERR) AT END OF DAP CYCLE WILL
                        # BE R-AXIS ERROR.

CA      BIT3      # .05GSW = 102D BIT3      SW=0, LESS .05G
MASK    CM/FLAGS      # SWITCH =1, GREATER THAN .05 G
EXTEND
BZF     EXDAP      # IF G LESS THAN .05
CS      ONE      # IF G GEQ THAN .05
TS      CMDAPMOD      # SAVE -1 FOR USE IN CM/RCS

TS      AK1      # TO ZERO PITCH AND YAW FDAI NEEDLES
TS      AK2      # IN ATM. (MODE ==-1)
```

```

CS      PREL      # YAW ERROR = RREL - PREL TAN(ALFA)
EXTEND
MP      SINTRIM   # LET SIN(-20) BE APPROX FOR TAN(-20)
AD      RREL
TC      2D/SDZ    # GO TEST DZ.  GET TAG: +0 IF IN DZ
INDEX   A         # +/- 1 IF NOT
CAF     YJETCODE

TS      JETEM

# START PITCH AUTOPILOT HERE.  RATE DAMPING ONLY.

CA      QREL
TC      2D/SDZ
EXDAPIN INDEX A   # COME HERE FROM EX ATM DAP
CAF     P/RJCODE
ADS     JETEM     # COMBINE ALL NEW BITS.

# Page 1075
EXTEND      # DOES NOT REQUIRE SAVING OLD CODES.
WRITE PYJETS # SET PYCHAN TO DESIRED BIT CONFIG.

CCS      JETAG
TC      CM/RCS
TC      CM/FDAI
TC      CM/FDAIR -1 # (JETAG=-1 EQUIVALENT TO CMDAPMOD=+1)

# Page 1076
# DEAD ZONE LOGIC USED BY ENTRY DIGITAL AUTOPILOTS.

3DDZ     CCS      A      # YAWLIM=1.0-3/180=16384-273=16111
          AD      YAWLIM
          TCF     DZCOM
          AD      YAWLIM
          TCF     DZNOCOM

BIASEDZ  TS      JETEM2   # BIASED DZ FOR EXT ATM DAP.
          CCS     A        # SAVE RATE/180.  ERROR/180 IS IN L.
          CS      CM/BIAS  # START ERROR DZ.
          TCF     +2       # = .6/180
          CA      CM/BIAS
          AD      L        # BIAS THE ERROR.
          LXCH    Q        # SAVE CALLER'S RETURN ADDRES.
          TC      3DDZ     # GO GENERATE THE ERROR BIT.
          DXCH    L        # BIT TO L, RESTORE CALLER'S Q.

```

July 12, 2016

253

4D/SDZ	CCS	JETEM2	# CAME HERE IN EXT ATM. C(L) = ERROR BIT
	AD	4D/SLIM	# IF RATE GEQ 4D/S, SET L=0 AND TAKE
	TCF	+2	# JET BITS ACCORDING TO SGN OF RATE.
	AD	4D/SLIM	
	TS	A	
	TCF	+2	# RATE OK. CONTINUE
	ZL		# RATE GEQ 4 D/S. OVER RIDE ERROR BIT
	XCH	JETEM2	# AND CONTINUE TO GET SIGN.
2D/SDZ	CCS	A	# COME HERE TO TEST IF A WITHIN 2DEG/S DZ
	AD	YDOTLIM	# 1.0 - YDOT DZ (OR PDOT)
	TCF	+3	
	AD	YDOTLIM	# YDOT DZ = 2 DEG/SEC
DZCOM	COM		
DZNOCOM	TS	JETEM +1	# GENERATE TAG, SET C(A)= +-1 OUTSIDE DZ
	CA	ZERO	# SET C(A) = +0 INSIDE
	TC	Q	

Page 1077

EXTRA ATMOSPHERIC DIGITAL AUTOPILOT

```
#
# 1. IF ABS(CALF) -C(45) POS, USE IF CALFA POS, CMDAPMOD= +0
# BETA: YAW ERROR = SGN(CALF) (BETACOM -BETA) IF CALFA NEG, CMDAPMOD= -0
# RATE = BETADOT IF CMDAPMOD = -0, RATE = RREL
# R-AXIS = CONTROL
#
# ROLL: ROLL ERROR = SGN(CALF) (ROLLC - ROLL) IF CMDAPMOD = -0, RATE DAMP ONLY.
# RATE = PREL
# P-AXIS = CONTROL
#
# 2. IF C(45) GEQ CALFA GEQ -C(45), USE CMDAPMOD = +1
# BETA: ROLL ERROR = SGN(-SALF) (BETACOM -BETA)
# RATE = BETADOT
# P-AXIS = CONTROL
#
# ROLL: YAW ERROR = SGN(SALF) (ROLLC - ROLL) RATE DAMP ONLY.
# RATE = RREL
# R-AXIS = CONTROL
#
# 3. FOR ALL CASES, USE
# ALFA: PITCH ERROR = (ALFACOM - ALFA)
# RATE = QREL
# Q-AXIS = CONTROL
```

EXDAP	TS	CMDAPMOD	# +0 FOR NOW
	CCS	CALFA	

July 12, 2016

```

AD      C45LIM      # =1.0-COS(45)
TCF     +2
AD      C45LIM
TS      A
TCF     EXDAP2      # HERE IF ABS(CALFA) L COS(45)

CCS     CALFA      # |CALFA| > 0.707
TCF     +1          # CONTINUE IF POS; GO TO EXDAP4 IF NEG.

CCS     P63FLAG      # VALID VALUES ARE:  -1, +1, +0.
TC      EXDAP4
TC      +2
TC      EXDAP4
TC      PHASCHNG      # SINGLE PASS THROUGH HERE.
OCT     40334
CS      ONE
TS      P63FLAG      # SET FLAG TO ASSURE SINGLE PASS.
CA      NSEC
TC      WAITLIST
EBANK=  AOG
2CADR   WAKEP62      # CALL TO TERMINATE P62 IN N SEC.

# Page 1078

# 65 DEG/ 3DEG/SEC = 21 SEC NOMINAL
# TRANSIT TIME FROM ALFA=45 TO ALFA TRIM.

EXDAP4  CCS     JETAG      # ROLLJET INTERFACE TEST BETWEEN .1 SEC
TCF     EXDAP3      # DAP AND THE 2 SEC CM/RCS DAP
TCF     EXDAP3
CA      ZERO
EXTEND
WRITE   ROLLJETS
TS      JETAG

# TURN OFF ROLL JETS IF ON AND WAIT
# UNTIL START OF 2 SEC CM/RCS CYCLE
# RESTORE PROPER VALUE +0

EXDAP3  CCS     CALFA      # ROLL FDAI WILL BE IN ERROR UNTIL NEXT CM/R
CA      RAXERR      # HERE IF ABS(CALFA) GEQ COS(45)
TCF     EXDAP1      # C()= BETACOM - BETA/180
CS      ZERO
TS      CMDAPMOD      # FOR CM/RCS
CS      RAXERR      # COMPLEMENT OF YAW ERROR.
EXDAP1  TS      RAXERR      # FOR YAW FDAI
TS      AK2          # WANT RAXERR FOR TM.
TS      L
CCS     CMDAPMOD      # COORDINATE BETA CONTROL.
TC      +3          # C(CMDAPMOD) CAN BE +1, +0, OR -0.
CA      ONE          # USE BETADOT TO COORD IN MODE +0

```

```

INDEX  A                # OTHERWISE USE RREL.
CA      RREL
TC      BIASEDZ         # GO TEST DZ  +0 IF IN DX, +-1 OTHERWISE
                        # IF GEQ 4D/S, SET ERROR BIT IN L=0

EXTEND
ROR     LCHAN           # L HAS BETA BIT
INDEX   A
CAF     YJETCODE
TS      JETEM

CA      QAXERR          # ALFA ERROR.
TS      L
CA      QREL            # FOR ALPHADOT USE QREL
TC      BIASEDZ
EXTEND
ROR     LCHAN
TCF     EXDAPIN         # CONTINUE ON IN DAP

EXDAP2  INCR            CMDAPMOD      # SET CMDAPMOD TO +1

CS      ONE             # INDICATE CHANGE FROM .1 SEC UPDATE TO
TS      JETAG           # TO 2 SEC FOR ROLL JETS. (IF CMDAPMOD
                        # =0 AND JETAG =-1, QUENCHES JETS IF ON)

CCS     P63FLAG         # IF FLAG WAS +1, SET =0.
TS      P63FLAG

# Page 1079

NOOP

CCS     SALFA           # BETA CONTROL WITH P JETS
CS      RAXERR          # B()= BETACOM - BETA/180
TCF     +2
CA      RAXERR
TS      PAXERR1         # TEMP SAVE.  ERROR/180
EXTEND
MP      HALF            # CM/FDAI EXPECTS ERROR/360.
XCH     PAXERR1         # ERROR/360 FOR FDAI, GET ERROR/180.
TS      L
CCS     SALFA
CS      BETADOT         # USE BETADOT TO COORD IN MODE +1
TC      +2
CA      BETADOT
TC      BIASEDZ
EXTEND
ROR     LCHAN
INDEX   A

```

```

CAF      P/RJCODE      # GET ROLL CODE
EXTEND
WRITE    ROLLJETS      # ROLL CONTROL WITH YAW JETS.
                                # WE'LL SKIP REGULAR ROLL SYST

CA      ROLLHOLD      # ROLL/180 AT CM/DAPON TIME.
EXTEND
MSU      ROLL/180      # 1'S COMPL, BUT SO WHAT'S A BIT?
TS      L              # FORCE A LIMIT CYCLE IN YAW RATE.
CCS      SALFA
CA      L              # TO REMOVE ITS BIASING EFFECT ON M DOT.
TC      EXDAP1
CS      L
TC      EXDAP1

NSEC      DEC      2100      # 65 DEG/ 3 DEG/SEC
                                # IF NSEC IS CHANGED, REMEMBER TO CHANGE 4.3
4D/SLIM    DEC      16348      # 1.0 -4/180 D/S = 4/1800 EXP 14
YDOTLIM    DEC      16366      # =1.0 - YDOT DZ= 16384 -18
                                # YDOT DZ = YDOT TCDU/180 = 2/1800 EXP 14

CM/BIAS    DEC      55        # =.6/180 B14 = 55
YAWLIM     DEC      16055      # YAWLIM=1.0-3.6/180=16384-329=16055
C45LIM     DEC      .29289     # =1.0-COS(45)

SINTRIM    DEC      -.34202    # SIN(-20)      (FOR NOMINAL L/D = .3)
COSTRIM     DEC      .93969    # COS(-20)      (FOR NOMINAL L/D = .3)

# TO MAKE DAP INSENSITIVE TO PITCH ERRORS DUE TO ACCUMULATED NAV ERRORS, USE NOMINAL
# USED DURING ATMOSPHERIC COORDINATION.  OUTSIDE ATMOSPHERE, NAV ERRORS WILL BE SLIGH
# FROM TRIM, SO USE ON-BOARD ESTIMATES.

# Page 1080
# JET CODE TABLES FOLLOW

YJETCODE    OCTAL      00120      # POS Y
OCTAL      00000      # RCS JET BITS
OCTAL      00240      # NEG Y
OCTAL      00005      # POS R JET BITS      ALSO POS P JET BITS
P/RJCODE    OCTAL      00000
OCTAL      00012      # NEG R      ALSO NEG P

# Page 1081
# RCS      THIS SECTION IS ENTERED EACH 2 SEC BY WAITLIST CALL FOLLOWING A DELAY
# THE TASK SETJTAG SETS A FLAG IN JETAG TO SIGNIFY THAT ROLL UPDATE IS DUE.  IN P
# EXECUTED AND JETAG WILL CAUSE CM/RCS TO ACT ON ROLL IMMEDIATELY THEREAFTER.  TH
# TASK SAVES THE CALL TIME SO THAT CM/RCS CAN DETERMINE HOW MUCH OF THE 2 SEC INTERVA

```


July 12, 2016

257

NEXT UPDATE.

```
SETJTAG      CS      TIME1      # SAVE NOMINAL UPDATE TIME FOR SYNCH
              TS      TUSED
              # THE 5 CS APPEARS IN TIMETST.
              CA      ONE        # RATHER THAN INCR, FOR SAFETY
              TS      JETAG      # SET JETAG=1 TO CAUSE CM/RCS TO BE
              TC      PHASCHNG
              OCT     00001
              TC      TASKOVER   # EXECUTED AFTER NEXT BODYRATE UPDATE
```

PREDICTIVE ROLL SYSTEM ENTRY STEERING PROVIDES ROLL COMMAND IN LOC ROLLC. THE FOLLOWING
TRAJECTORY TO THE ORIGIN IN PHASE PLANE (X,V). PROGRAM ENTERS JET ON AND OFF CALLS INTO WTLST
THE DESIRED TRAJECTORY. ONLY THOSE CALLS WHICH CAN BE EXECUTED WITHIN THE INTERVAL T (2 SECS)
WTLST, THE REMAINDER ARE RECONSIDERED AT NEXT UPDATE.

HALFPR EQUALS NEG1/2 +1

```
# CLEAR JETAG BEFORE TIMETST. SET TO +0 TO SHOW
# ROLL DAP CALLED. IN EVENT OF RESTART, BODYRATE
# MAY MISS A CYCLE. CM/RCS WILL MISS A CYCLE ONLY
# IF A RESTART OCCURS AFTER TIMETST COMMENCES.
```

```
CM/RCS      CS      ONE
              TS      JNDX      # SET NDX FOR POS ROLL, AND CHANGE LATER

              CS      2T/TCDU    # ROLLDOT = DELAOG + DELAIG SINM =DELR
EXTEND
MP          PREL      # DELR/180 = RDOT TCDU/180 = RDOT/1800
AD          L         # -2 RDOT T/180 IN L
TS          -VT/180   # SAVE -2VT/180 HERE

              CS      ROLL/180
              TS      SR        # SAVE (-R/180) /2

              CS      CM/FLAGS
MASK        BIT4      # LATSW = 101D BIT4
EXTEND
BZF         GETLCX    # NO, TAKE SHORTEST PATH
ADS         CM/FLAGS  # YES, ENFORCE ROLL OVER TOP.. (BIT =0)
CA          ROLLC     # (ROLLC/180) /2
AD          SR        # -(R/180) /2
XCH         LCX/360   # DIFFERENT X REQD HERE. DISCONT AT 180.
TCF        COMPAT    # POSSIBLE OVFL ABOVE.
```

GETLCX

FORM RCOM/360

AD ROLLC

```
# IGNORE POSSIBLE OVFL.
```

FORM -R/360

```
# IGNORE OVFL
```

-R/360

$$\# \text{ LCX}/360 = \text{RCOM}/360 - \text{R}/360 \quad \text{RANGE } (-1,1)$$

```

      CCS      -VT/180      # TAKE SHORTEST ANGULAR PATH
      AD       ONE         # (BASED ON SINGLE JET ACCELERATION)

```

```
# TAKE SHORTEST ANGULAR PATH
```

(BASED ON SINGLE JET ACCELERATION)

AD ONE

MP -VT/180

11. 100, 100, 100, 100, 100, 100, 100, 100, 100, 100

$$\# = 180 / (16 \text{ A1 TT})$$

DV CALFA

CCS	LCX/360
-----	---------

CH	+3
TCF	+2

CS	POS1/2
----	--------

AD LCX/360

IS LCX/360 LESS THAN 180 DEGS \$

AD	E
AD	L

AD	E
TS	L

IS	E
TCE	COMPAT

YES GO ON

TRTAGXPT

INDEX A

```
# NO SHIFT X BY = SGN(X) 2 PT
```

INDEX	A
CS	HALFPR

```
# +A YIELDS =1/2
```

DOUBT F

DOUBLE
ADS LCY/360

COMPAT

CA LCX/360

```
# CORRECT FOR ASSUMED COORD TURN.
```

EXTEND

MP CALFA

COS ALFA

TS	LCX/360
----	---------

```
# SCALED LCX OK HERE.
```

CCS CMDAPMOD

```
# FOUR POSSIBILITIES HERE
```

TC DZCALL1

```
# EXIT, SETTING JETAG=0. (C(A)=0)
```

```
# ALL 3 AXES ALREADY DONE.
```

```

TC      +1      # G LESS THAN .05.  CA POS. CONTINUE
CA      LCX/360  # G GEQ .05.  CONTINUE IN CM/RCS
TS      LCX/360  # CMDAPMOD=-0.  DAMPING ONLY. SET LCX=0
TS      ERRORZ   # INITIAL ROLL ERROR (UNREFLECTED) FOR TM.
TS      PAXERR1  # SAVE LCX FOR FDAI AND EDIT.  (/360)

# Page 1083

CA      -VT/180  # GET - 2 VT/180
TS      SR
CA      SR      # GET -VT/180, LEAVE -VT/360 IN SR FOR DZ
TS      -VT/180E #
XCH     -VT/180  # NOW CONTENTS OF -VT/180 AS LABELED
EXTEND
MP      -VT/180  # B(A) = -ZVT/180
EXTEND
MP      180/8ATT
TS      VSQ/4API

# IS SGN(VT) ( (180/4A1 TT) VT/180 VT/180 - .5 BUFLIM/360 ) -X/360 - .5 BUFLIM/360 POS?

WHICHALF DOUBLE      # FOR SECOND BURN, A1
COM
AD      BUFLIM      # =BUFLIM/(2 360)
TS      L
CCS     -VT/180
CS      L
TCF     +2
CA      L
AD      LCX/360
AD      BUFLIM
EXTEND
BZMF    REFLECT      # POINT (X,V) IN LHP.

# IS SGN(VT) ( (180/4A1 TT) VT/180 VT/180 - .5 BUFLIM/360 ) -X/360 + .5 BUFLIM/360 NEG?

COM
AD      BUFLIM
AD      BUFLIM
EXTEND
BZMF    DZ1          # POINT (X,V) IN RHP

# IS POINT WITHIN VELOCITY DZ?

CS      VSQMIN      # IS VSQ/4API - (VSQ/4API) MIN NEG?
AD      VSQ/4API
EXTEND
BZMF    DZCALL      # YES.

```

POINT IS IN BUFFER ZONE. THRUST TO X AXIS.

CS JNDX
TS JNDX1
TC OVRLINE1

REFLECT CS -VT/180 # RELFECT LHP INTO RHP REL TO TERM CONTR
TS -VT/180
TS SR # -VT/360 SAVED FOR DZ.

Page 1084

CS LCX/360
TS LCX/360
CS JNDX
TS JNDX

IS VSQ/4API - (VSQ/4API) MIN NEG?

DZ1 CS VSQMIN # IS VSQ/4API - (VSQ/4API) MIN NEG \$
AD VSQ/4API
EXTEND
BZMF DZ2 # YES, GO TEST FURTHER.
TCF MAXVTEST # NO

IS X/360 - XMIN/360 -VT/360 NEG?

DZ2 CS XMIN/360 # XMIN/360 = 4/360
AD LCX/360
AD SR # C(SR) = -VT/360
EXTEND # IS X/360 - XMIN/360 -VT/360 NEG \$
BZMF DZCALL # YES, IN DZ. EXIT SETTING JETAG=0.

IS XD/360 - VM/360K - XS/360 POS?

MAXVTEST CS JNDX
TS JNDX1 # NOW CAN SET JNDX1 FOR TON2 JETS.
CS XS/360 # XS/360 = (XMIN -YMIN/K) /360
AD VSQ/4API
AD LCX/360
TS XD/360 # XD/360= X/360 +VSQ/4API X INTERCEPT
BUT C(XD/360) = (XD - XS) /360
AD -VM/360K # X INTERCEPT FOR MAX V (VM)
COM
EXTEND
BZMF MAXVTIM1 # YES, THRUST TO VM
CA XD/360

```

EXTEND
MP      KTRCS
DDOUBL                                     # GO SAVE PREDICTED DRIFTING VELOCITY.

TC      GETON1                           # INSURE THAT Q IS POS AS TAG.
EXTEND
ZQ                                             # SET +Q AS TAG
CS      -VMT/180
TS      VDT/180                           # VDT/180 OR VMT/180
AD      -VT/180
DOUBLE
EXTEND
MP      180/8ATT
TS      TON1                             # TON1 / 4T
# Page 1085
EXTEND
BZMF    OVRLINE
TC      GETON2                           # RESET Q POS IF CAME FROM MAXVTIM1

OVRLINE  CCS      Q
TCF      OVRLINE1
MAXVTIM2 CA      JNDX1                     # ABOVE VM, SO THRUST DOWN
TS      JNDX
CS      TON1
TCF      OVRLINE2 +1

OVRLINE1 CS      -VT/180                   # DRIFT AT V
TS      VDT/180
OVRLINE2 CA      ZERO
TS      TON1
GETON2   CA      VDT/180                   # VDT/180, OR VMT/180 OR VT/180
DOUBLE
EXTEND
MP      180/8ATT
DOUBLE
TS      TON2                             # FOR SECOND BURN, A1
                                           # = TON2 / 4T

COM
EXTEND
BZMF    GETOFF
TS      TON2
CA      JNDX
TS      JNDX1

GETOFF   CS      TON2                     # TON2 / 4T
EXTEND

```

```

MP      VDT/180      # VDT/180, OR VT/180, OR VMT/180.
TS      XD/360      # USE AS TEMP
CS      VDT/180
EXTEND
BZF     TOFFOVFL     # OMIT THE DIVIDE IF DEN = 0.
AD      -VT/180
EXTEND
MP      TON1         # TON1 /4T
AD      XD/360      # TEMP = -VDT/180 / 2 TON2
AD      LCX/360
ZL
XCH     L           # TEST THE DIVIDE
EXTEND
DV      VDT/180
EXTEND
BZF     GETOFF2      # DIVIDE OK

TOFFOVFL  CA      2JETT      # OVFL, USE 2T FOR CONVENIENCE.
          TCF     TIMSCAL

# Page 1086
GETOFF2   XCH     L           # GET NUMERATOR.
          EXTEND
          DV      VDT/180     # C(A) = TOFF / 2T
          EXTEND
          MP      2JETT
          TS      TOFF        # IN CS

TIMSCAL   CAF     4JETT
          EXTEND
          MP      TON1        # C(TON1) = TON1 / 4T
          TS      TON1        # IN CS

          CAF     4JETT
          EXTEND
          MP      TON2        # C(TON2) = TON2 / 4T
          TS      TON2        # IN CS

          CA      ZERO        # CANNOT REDO AFTER TIMETST. TUSED GONE
          TS      JETAG       # SET +0 TO SHOW ROLL DAP CALLED.

          # CAUSE THE TM OF BODY RATES VIA UPBUFF TO BE
          # INITIALIZED. ALSO CAUSE NEEDLES TO BE DONE
          # AND ON ALTERNATE PASSES THROUGH CM/DUMPR.

          CA      ONE

```

TS SW/NDX

Page 1087

TIMETEST SECTION FOR RCS

#

ENTER WITH THREE TIME INTERVALS AND THE CORRESPONDING JET CODE INDEXES IN ERASABLE LOCS TON1,

JNDX1. SECTION PROCESSES TIME INTERVALS FOR WTLST CALLS AND ASSURES THAT WTLST CALLS ARE MADE

(1) FOR POS INTERVALS GREATER THAN A SPECIFIED MINIMUM (HERE CHOSEN AS 2 CS) AND

(2) FOR THE INTERVALS THAT WILL BE EXECUTED WITHIN THE TIME REMAINING IN THE SAMPLE INTERVAL

TIMETST ESTABLISHES 6 LOCS CONTAINING JET CODES AND CORRESPONDING TIME INTERVALS. THUS: TON1

TOFF, TBITS, TON2, T2BITS. OF THESE THE FIRST 2 LOCS ARE TEMPORARY, FOR IMMEDIATE ACTION, IN

SECTION JETCALL BELOW PROCESSES THIS LIST.

TIMETST	CA	TIME1	# CORRECT FOR POSSIBLE TIME1 OVFL.
	AD	POS1/2	
	AD	POS1/2	# OVFL GUARANTEED.
	ADS	TUSED	# B(TUSED) ==TUSED ==OLTIME1
	CA	-T-3	# ==T +2 -5 (SEE SETJTAG)
			# THE +2 REQUIRED FOR PROPER BRANCH.
	ADS	TUSED	# TUSED = TIME(K)-TIME(K-1)-T+2
	CS	TWO	# USE 2 SINCE TIME3 UNCERTAIN TO 1
	AD	TON1	
	EXTEND		
	BZMF	TIMETST1	
	INDEX	JNDX	
	CAF	P/RJCODE	
	TS	T1BITS	
	CA	TON1	
	ADS	TUSED	
	EXTEND		
	BZMF	TOFFTEST	
	CA	ZERO	
	TCF	TIMETST3	
TIMETST1	CS	ONE	
	TS	TON1	
TOFFTEST	CS	TWO	
	AD	TOFF	
	EXTEND		
	BZMF	TIMETST2	
	CA	TOFF	
	ADS	TUSED	
	EXTEND		
	BZMF	TON2TEST	

	CA	ZERO
	TCF	TIMETST4
TIMETST2	CS	ONE
	TS	TOFF
TON2TEST	CS	TWO
	AD	TON2
	EXTEND	
	BZMF	TIMETST5
# Page 1088		
	INDEX	JNDX1
	CAF	P/RJCODE
	TS	T2BITS
	CA	TON2
	ADS	TUSED
	EXTEND	
	BZMF	JETCALL1
	CA	ZERO
	TCF	TIMETST5 +1
TIMETST3	TS	TON1
	CS	ONE
TIMETST4	TS	TOFF
TIMETST5	CS	ONE
	TS	TON2

SECTION JETCALL EXAMINES CONTENTS OF JET TIMES IN LIST, ESTABLISHES WTLST ENTRIES
 # JET CODES. A POSITIVE NZ NUMBER IN A TIME REGISTER INDICATES THAT A WTLST CALL IS
 # EXECUTED. A +0 INDICATES THAT THE TIME INTERVAL DOES NOT APPLY, BUT THE CORRESPONDING
 # EXECUTED. A NEG NUMBER INDICATES THAT THE TIME INTERVAL HAS BEEN PROCESSED. IN EVENT
 # SUBSEQUENT TIME REGISTER IS EXAMINED FOR POSSIBLE ACTION. THUS JET BITS TO BE EXECUTED
 # THAN ONE REGISTER.

JETCALL1	CA	ZERO	
	TS	OUTTAG	
	TS	NUJET	
	TS	TBITS	
	DXCH	TON1	
	CCS	A	
	TCF	JETCALL2	# CALL WTLST
JETCALL3	LXCH	NUJET	# WTLST ENTRIES COME HERE FROM JETCALL
	CS	ONE	
	DXCH	TOFF	
	CCS	A	
	TCF	JETCALL2	# CALL WTLST
	LXCH	NUJET	
	CS	ONE	
	DXCH	TON2	

265

```

CCS      A
TCF      JETCALL2      # CALL WTLST
LXCH     NUJET
TC       JETACTN      # C(A) = +0
JETCALL2 XCH          L      # SAVE JET BITS FOR AFTER WTLST CALL
ADS      NUJET
XCH      L
AD       ONE          # RESTORE FOR CCS
TC       WAITLIST
EBANK=   AOG
2CADR    JETCALL

JETACTN   CA          NUJET      # COME HERE WHEN DESIRED JET CODE IS KNOWN
# Page 1089

EXTEND    # NO NEED TO SAVE OLD CODES
WRITE     ROLLJETS      # SET RCHAN TO NEW BIT CONFIG.

CCS      OUTTAG
TC       TASKOVER
ROLLDUMP  TC          CM/FDAIR

# EDIT DUMP AT ABOVE LOCATION.

# WAITLIST ENTRIES COME HERE.

JETCALL   CAF          BIT2      # CM/DSTBY =103D BIT2
TS        OUTTAG      # SIGNIFY WTLST ENTRY
MASK      CM/FLAGS     # IS SYSTEM DISABLED $
EXTEND
BZF       JETACTN +1    # YES, QUENCH ROLL JETS, IF ON AND EXIT.
ZL        # NO, CONTINUE.
TCF       JETCALL3     # C(A) POS, C(L) = +0

# DEAD ZONE ENTRIES COME HERE.

DZCALL    CS          CMDAPMOD    # POSSIBLE VALUES OF CMDAPMOD: -1, +0, -0.
MASK      BIT1
TS        L          # C(L)=0 FOR -0: C(L)=1 FOR -1 OR +0.
INDEX     A          # ERASABLE ORDER: ROLLTM, ROLLC, ROLLC +1.
CA        ROLLTM      # GET ROLL/180 OR ROLLC (/360).
INDEX     L
TS        A          # IF C(L)=1, STORE 'ROLLC' IN 'L'.
AD        L          # (BOTH MUST BE SCALED DEG/180)
TC        ANGOVCOR    # C(A)=ROLL/180 OR 2 ROLLC.
TS        ROLLHOLD    # IF CMDAPMOD =-0, SAVE ROLL ANGLE.
#          OTHERWISE, SAVE ROLL COMMAND.

```

```

CA          ZERO          # COME HERE IF IN DZ, AND CANCEL JETS.
EXTEND      # INHINT NOT NEEDED HERE.
WRITE      ROLLJETS      # TURN OFF ALL ROLL JETS.
TS          VDT/180      # SET =0 TO SHOW IN DEAD ZONE.
DZCALL1     TS          JETAG      # COME HERE WITH C(A)=0.
TC          ROLLDUMP

# Page 1090
# CM ENTRY FDAI DISPLAY
#
# CALCULATE BY INTEGRATION THE ROLL ERROR BETWEEN THE 2 SEC CM/RCS UPDATES.  DISPLAY
#      ATM DAP:          DISPLAY ONLY ROLL ATTITUDE ERROR.
#      EXT ATM DAP:      PRESENT 3 ATTITUDE ERRORS RELATIVE TO THE APPROPRIATE BODY A
#                          ROLL      ROLLC-ROLL
#                          PITCH     ALFAC-ALFA
#                          YAW       BETAC-BETA
#
# DURING ENTRY, THE FDAI NEEDLES HAVE FULL SCALE OF 67.5 DEG IN ROLL AND 16.875 DEG
# THE SUBROUTINE NEEDLER EXPECTS (ANGLE/180) AND SCALES TO 16.875 DEG FULL SCALE.

CM/FDAI     CS          PHIDOT      # COME HERE EACH .1 SEC.  (CMDAPMOD=+1 COMES
EXTEND      # INTEGRATE ROLL ERROR 'TWEEN 2SEC UPDATES
MP          CALFA          # FOR ASSUMED COORDINATION.
EXTEND
MP          HALF
ADS        PAXERR1        # ROLL ERROR/360.  OVFL OK.

CM/FDAIR    CA          HALF        # EDIT DUMP AT ABOVE LOCATION.
EXTEND
MP          PAXERR1        # FULL SCALE FOR FDAI (ROLL) IS 67.5 D
TS          PAXERR         # .25 (ROLL ERROR/180) FOR FDAI NEEDLE.

# PROGRAM TO FILE BODY RATES FOR TM ON ONE P
# TO UPDATE THE NEEDLE DISPLAY ON THE NEXT.
# SYNCHRONIZATION WITH CM/RCS IS USED SO THA
# IS DONE WITH THE ROLL SYSTEM AND NEEDLES S
# THE SUBSEQUENT PASS.

CM/DUMPR    CS          SW/NDX      # COMBINED ALTERNATION SWITCH AND FILE
TS          SW/NDX
EXTEND      # INDEX
BZMF       CMTMFILE        # FILE STARTS WITH SW/NDX +1 AND GOES TO
# ENDBUF.

```

July 12, 2016

267

INDEX IS POS FOR NEEDLES.

TC IBNKCALL
CADR NEEDLER

TC CM/END

INDEX IS NEG FOR TM FILE

CMTMFILE AD THREE
EXTEND
BZMF SAVENDX

Page 1091

SAVENDX CA TIME1
TS CMTMTIME
CS THIRTEEN
TS SW/NDX
EXTEND
DCA PREL
INDEX SW/NDX
DXCH ENDBUF -1
CA RREL
INDEX SW/NDX
TS ENDBUF +1

INITIALIZE THE TM LIST IN UPBUFF.

INITIALIZE COUNTER

A NEGATIVE NUMBER.

CM/END CA CM/SAVE
TS SR

DOES NOT PROTECT TEMK, SQ IN SPSIN/COS

EXTEND
DCA T5IDLER2
DXCH T5LOC
TC RESUME

T5IDLER2 EBANK= T5LOC
2CADR T5IDLOC

DEFINE THE FOLLOWING 17D REGISTERS IN UPBUFF TO BE
USED TO TELEMETER CM VEHICLE BODY RATE INFORMATION.
THE INFORMATION IS FILED EACH 0.2 SEC, GIVING 15D
DATA POINTS EACH 1 SEC. TM LIST IS READ TWICE
EACH 2 SECONDS.

#

THE SEQUENCE IS: SP TIME INITIAL TIME
SWITCH ALSO INDEX.

```

#
#
#
#
# P
# Q
# R
# ETC.
#
#
#
#
# CMTMTIME = UPBUFF
# SW/NDX = UPBUFF +1
# ENDBUF = UPBUFF +16D

# Page 1092
# SPACER
#
# CONSTANTS USED IN THE ROLL CONTROL SYSTEM:
# CONSTANTS ARE THE FOLLOWING: A = 9.1 DEG/SECSQ, VM = 20 DEG/SEC, T = 2 SEC, TCDU =
# XMIN = 4 DEG, VMIN = 2 DEG/SEC, K = .25, A1 = 4.55 DEG/SECSQ, VI = 1 DEG/SEC, INTER
# XBUF = 4DEG

-T-3 DEC -203 # CS
VSQMIN DEC .61050061 E-3 # VSQ MIN/4 A PI = 4/(4 (9.1) 180)
2T/TCDU = OCT50 # T/TCDU EXP-14 TCDU = .1SEC
180/8ATT DEC .61813187 # 180/(8 (9.1) 4)=(180/ATT) EXP -3
-VM/180 = -VM/360K # = 20 (2) / 180
2JETT = 4SECS # CS 2 (2) 100 INTEGER
4JETT DEC 800 # CS 4 (2) 100 INTEGER
XMIN/360 DEC 182 # XMIN/360 = 4/ 360 EXP 14 = 182 INTEGER
-VM/360K DEC -.22222222 # =-20/( 360 (.25))
1/16A1 = 180/8ATT # 1/16A1 = 180/(16 A1 TT)
# = 180/(16 4.55 4)
XS/360 DEC 91 # = (XMIN +VI (T-1/K))/360 = 2/360 EXP 14
BUFLIM = XS/360 # 4/(2 360)

KTRCS = HALF # KT = (.25) 2 = .5

# *** END OF TVCDAPS .011 ***

```

This code is written to file `src/CM-ENTRY-DIGITAL-AUTOPILOT.s`.

B.17 CONIC SUBROUTINES

269

(src/CONIC-SUBROUTINES.s 269)≡

```
# Copyright:      Public domain.
# Filename:       CONIC_SUBROUTINES.agc
# Purpose:       Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Ron Burkey <info@sandroid.org>.
# Website:       www.ibiblio.org/apollo.
# Pages:        1262-1308
# Mod history:   2009-05-08 RSB   Adapted from the Colossus249/ file of the
#               same name, using Comanche055 page images.
#               2009-05-20 RSB   Corrected: Fixed four interpreter
#               instructions.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051. 10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#       Colossus 2A
#
# Page 1262
# PROGRAM DESCRIPTION -- ENTIRE CONIC SUBROUTINE LOG SECTION      DATE -- 1 SEPTEMBER 1967
# MOD NO. -- 0                                                    LOG SECTION -- CONIC SUBROUTINE
# MOD BY KRAUSE                                                    ASSEMBLY -- COLOSSUS REVISION 8
#
# FUNCTIONAL DESCRIPTION --
#       THE FOLLOWING SET OF SUBROUTINES SOLVE VARIOUS PROBLEMS INVOLVING THE TRAJECTORY PRODU
#       INVERSE-SQUARE FORCE ACTING ON A POINT MASS, AS OUTLINED IN THE CMC AND LGC LUNAR LANDI
#       5.5.1.2. A GENERAL USAGE POINT-OF-VIEW WAS TAKEN IN FORMULATING, MECHANIZING, AND SCAL
#       RATHER THAN OPTIMIZING EACH FOR A PARTICULAR USE. THEREFORE, MULTIPLE USAGE CAN BE MAD
#       INVOLVING ANY REALISTIC SET OF CONSTRAINTS. IT SHOULD BE NOTED THAT ONLY ONE SET OF CO
#       EARTH, MOON, OR ANY OTHER CELESTIAL BODY IS SPECIFIED AS THE CENTRAL BODY OF THE PROBLE
```

```

# THE INHERENT SCALE CHANGE REQUIRED IN POSITION, VELOCITY, MU, AND TIME, AS OF
# DEFINITION MEMO NO. 10. THIS CAN BE ACCOMPLISHED BY SIMPLY ADDING TO THE MU
# APPROPRIATELY.
#
# DUE TO THE UNIFORMITY OF THE EQUATIONS INVOLVED, CODING WAS MINIMIZED BY TREATING
# BLOCKS OF EQUATIONS AS SUBROUTINES OF LOWER RANK WHENEVER POSSIBLE. AS A RESULT,
# DIRECTLY USABLE AS INDEPENDENT SUBROUTINES, WERE GENERATED.
#
# RESTRICTIONS --
# THE ONLY LIMITATION IN THE SCOPE OF THE PROBLEM WHICH CAN BE SOLVED BY A PARTICULAR
# LIMIT OF EACH PARAMETER AS SPECIFIED IN THE GSOP. THESE SCALING LIMITS WERE
# COULD BE HANDLED.
#
# SINCE THE SUBROUTINES (EXCEPT KEPLER) USE COMMON SUBROUTINES OF LOWER RANK WITHIN
# THE PUSHLIST (DUE TO ITS LIMITED SIZE) AND COMMON INTERPRETIVE SWITCHES, THERE IS
# TO INTERRUPT EACH OTHER. IT IS UP TO THE USER TO GUARANTEE THIS CONDITION.

# Page 1263
# PROGRAM DESCRIPTION -- KEPLER SUBROUTINE DATE -- 11 OCTOBER 1968
# MOD NO. -- 1 LOG SECTION -- CONIC
# MOD BY KRAUSE ASSEMBLY -- COLOSSUS
# MOD NO. -- 2 (AUGUST 1968) BY ROBERTSON: TO PERMIT BACKDATING BY MORE THAN ONE ORBIT
# MOD NO. -- 3 (DEC 1968) BY ROBERTSON: SUPPRESSION OF X-MODULO-ING
# MOD NO. -- 4 (JAN 1969) BY ROBERTSON: CLEAR OVFLND AT KEPLER ENTRY
#
# FUNCTIONAL DESCRIPTION --
# THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR AND THE DESIRED TRANSFER TIME, WILL
# BE UPDATED ALONG A CONIC TRAJECTORY, COMPUTES THE NEW, UPDATED STATE VECTOR.
# SECTION -- CIRCULAR, ELLIPTIC, PARABOLIC, HYPERBOLIC, OR RECTILINEAR WITH RESPECT TO
# USE OF THE SUBROUTINE CAN BE EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADJUSTMENT
# INTRODUCING ANY CODING CHANGES, ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN THE
# TECHNIQUE IS UTILIZED IN THE COMPUTATION.
#
# IF A NEGATIVE TIME-OF-FLIGHT IS INPUT, THE PROGRAM WILL SOLVE FOR THE STATE VECTOR
# EXTRAPOLATING THE POSITION BACKWARD IN TIME.
#
# IF THE ABSOLUTE VALUE DESIRED TRANSFER TIME EXCEEDS THE ORBITAL PERIOD, THE SUBROUTINE
# MODULAR TECHNIQUE, WILL COMPUTE THE STATE CORRESPONDING TO THE DESIRED TIME AND
#
# THE RESTRICTIONS ARE --
# 1. (PREVIOUS RESTRICTION ON THE NEGATIVE DESIRED TRANSFER TIME IS NOW DELETED)
# 2. THE PARAMETERS IN THE PROBLEM CANNOT EXCEED THEIR SCALING LIMITS AS SPECIFIED
# ANY OF THESE LIMITS ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
# THE NUMBER OF ITERATIONS AND, THEREFORE, THE COMPUTATION SPEED IS DEPENDENT ON THE
# GUESS, XKFPNEW. THE AGC COMPUTATION TIME IS APPROXIMATELY .061 SECONDS FOR 1

```

```

#       FINAL COMPUTATIONS, PLUS .083 SECONDS FOR EACH ITERATION.
#
# REFERENCES --
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP, SECTION
#       MEMO 67-4.
#
# INPUT -- ERASABLE INITIALIZATION REQUIRED
#           SCALE FACTOR
#           VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#           -----
#           RRECT         +29 FOR EARTH        DP INITIAL POSITION VECTOR IN METERS
#                           +27 FOR MOON
#
# Page 1264
#           VRECT         +7 FOR EARTH        DP INITIAL VELOCITY VECTOR IN METERS/CENTISECON
#                           +5 FOR MOON
#           X1 (38D)       NONE                INDEX REGISTER SET TO -2D OR -10D ACCORDING TO
#                                           RESPECTIVELY, IS THE CENTRAL BODY
#           TAU            +28                DESIRED TRANSFER TIME IN CENTISECONDS (DP)
#                                           MAY BE POS OR NEG AND ABSOLUTE VALUE MA
#           XKEPNEW        +17 FOR EARTH        DP GUESS OF ROOT X OF KEPLERS EQN IN SQRT(METER
#                           +16 FOR MOON        AND ABS VALUE SHOULD BE LESS THAN THAT
#                                           MAJOR AXIS), FOR SPEED OF CONVERGENCE,
#                                           BY KEPLER TO A POOR BUT VALID GUESS.
#           TC             +28                DP PREV. VALUE OF TIME IN CENTISECS. MUST BE LE
#           XPREV          +17 FOR EARTH        PRVIOUS VALUE OF X IN SQRT(METERS). MUST BE LE
#                           +16 FOR MOON        ORBITAL PERIOD, VIZ, 2PI SQRT(SEMI-MAJ
#
# SUBROUTINES CALLED --
#       DELTIME
#
# CALLING SEQUENCE AND NORMAL EXIT MODES --
#       KEPRTN-2          GOTO                # MUST BE IN INTERPRETIVE MODE BUT OVFIN
#       KEPRTN-1          KEPLER              # RETURNS WITH XPREV IN MPAC. PL IS AT 0.
#       KEPRTN            ...                # CONTINUE
#
#       KEPLER MUST NOT BE CALLED DIRECTLY SINCE AN INTERRUPTION OF IT WOULD DESTROY THE ERASAB
#       THE INTERRUPTED JOB. THEREFORE THE USER MUST CALL CSMCONIC OR LEMCONIC WHICH GUARANTEE
#       ALSO CALLS KEPPREP TO COMPUTE A GUESS OF XKEPNEW.
#
# ABORT EXIT MODE --
#       NONE
#
# OUTPUT --
#           SCALE FACTOR
#           VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#           -----

```

```

#      RCV      +29 FOR EARTH      DP TERMINAL POSITION VECTOR IN METERS
#      +27 FOR MOON
#      VCV      +7 FOR EARTH      DP TERMINAL VELOCITY VECTOR IN METERS
#      +5 FOR MOON
#      TC       +28              DP TRANSFER TIME IN CENTISECS TO WHICH
#      XPREV    +17 FOR EARTH      DP VALUE OF X IN SQRT(METERS) TO WHICH
#      +16 FOR MOON              CORRESPONDING TO ONE PERIOD.

```

```

# Page 1265

```

```

#      FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#

```

```

# DEBRIS --

```

```

#      PARAMETERS WHICH MAY BE OF USE --

```

	SCALE FACTOR	
VARIABLE	IN POWERS OF 2	DESCRIPTION AND REMARKS
-----	-----	-----
URRECT	+1	DP UNIT VECTOR OF INITIAL POSITION
R1	+29 FOR EARTH	DP MAGNITUDE OF INITIAL POSITION IN M
	+27 FOR MOON	
ALPHA	-22 FOR EARTH	DP INVERSE OF SEMI-MAJOR AXIS IN 1/M
	-20 FOR MOON	
TMODULO	+28	DP INTEGRAL NUMBER OF PERIODS IN CENT
		TAU. OF LESS THAN ONE PERIOD

```

#
# PARAMETERS OF NO USE --

```

```

#      DP PARAMETERS -- FPSILENT, DELX, DELT, RCNORM, XMODULO, PLUS PUSHLIST REGISTER

```

```

# Page 1266

```

```

# PROGRAM DESCRIPTION -- LAMBERT SUBROUTINE

```

```

DATE -- 1 SEPTEMBER 1

```

```

# MOD NO. -- 0

```

```

LOG SECTION -- CONIC

```

```

# MOD BY KRAUSE

```

```

ASSEMBLY -- COLOSSUS
#

```

```

# FUNCTIONAL DESCRIPTION --

```

```

#      THIS SUBROUTINE CALCULATES THE INITIAL VELOCITY REQUIRED TO TRANSFER A POINT-
#      FROM AN INITIAL POSITION TO A TERMINAL POSITION IN A PRESCRIBED TIME INTERVAL.
#      A SECTION OF A CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EA
#      SUBROUTINE CAN BE EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO
#      CODING CHANGES, ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND V
#      UTILIZED IN THE COMPUTATION.
#

```

```

# THE RESTRICTIONS ARE: --

```

1. RECTILINEAR TRAJECTORIES CANNOT BE COMPUTED.
2. AN ACCURACY DEGRADATION OCCURS AS THE COSINE OF THE TRUE ANOMALY DIFFERENCE
3. THE ANGLE BETWEEN ANY POSITION VECTOR AND ITS VELOCITY VECTOR MUST BE GREATER
AND LESS THAN 178 DEGREES 12.5 MINUTES.
4. NEGATIVE TRANSFER TIME IS AMBIGUOUS AND WILL RESULT IN NO SOLUTION.
5. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED


```

#           LIMITS ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
#           THE NUMBER OF ITERATIONS AND, THEREFORE, THE COMPUTATION'S SPEED IS DEPENDENT ON THE AGC
#           GUESS OF THE INDEPENDENT VARIABLE, COGA.  THE AGC COMPUTATION TIME IS APPROXIMATELY
#           .105 SECONDS FOR INITIALIZATION, .069 SECONDS FOR FINAL COMPUTATIONS, PLUS .205 SECONDS
#
# REFERENCES --
#           R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP -- SECTION
#           SGA MEMO 67-4.
#
# INPUT -- ERASABLE INITIALIZATION REQUIRED
#           SCALE FACTOR
#           VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#           -----
#           R1VEC          +29 FOR EARTH        DP INITIAL POSITION VECTOR IN METERS
#                           +27 FOR MOON
#           R2VEC          +29 FOR EARTH        DP TARGET OR TERMINAL POSITION VECTOR IN METERS
#                           +27 FOR MOON
#           TDESIRED       +28                  DP DESIRED TRANSFER TIME IN CENTISECONDS
#           X1 (38D)       NONE                  INDEX REGISTER SET TO -2D OR -10D ACCORDING TO
#                                           RESPECTIVELY, IS THE CENTRAL BODY
#           GEOMSGN        NONE                  SP +.5 IF DESIRED TRANSFER ANGLE IS LESS THAN 1
#           GUESSW         NONE                  AN INTERPRETER SWITCH TO BE SET IF NO GUESS OF
#
# Page 1267
#
#           COGA IS TO BE USED BY LAMBERT
#           COGA           +5                  DP GUESS OF COTANGENT OF FLIGHT PATH ANGLE (MEAN
#                                           IGNORED IF GUESSW IS SET.
#           NORMSW         NONE                  AN INTERPRETER SWITCH TO BE SET IF UN IS TO BE
#                                           LAMBERT IS TO COMPUTE ITS OWN NORMAL (UN)
#           UN              +1                  DP UNIT NORMAL TO THE DESIRED ORBIT PLANE IN THE
#                                           MOMENTUM VECTOR.  THIS WILL BE IGNORED
#           VTARGETAG      NONE                  A S.P. TAG TO BE SET TO ZERO IF LAMBERT IS TO C
#                                           AT R1VEC.
#
# SUBROUTINES CALLED --
#           GEOM, GETX, DELTIME, ITERATOR, LAMENTER (PART OF NEWSTATE)
#
# CALLING SEQUENCE AND NORMAL EXIT MODES --
#           L      CALL      # MUST BE IN INTERPRETIVE MODE BUT OVFLND ARBITRARY.
#           L+1          LAMBERT # RETURNS WITH PL AT 0 AND WITH VVEC IN MPAC IF VTARGETAG
#                               #           IN MPAC IF VTARGETAG WAS ZERO
#           L+2      BON      # CONTINUE IF SOLNSW CLEAR SINCE SOLUTION IS ACCEPTABLE
#           L+3          SOLNSW
#           L+4          LAMABORT
#
#           IF A LAMBER RESULT IS TO BE A FIRST GUESS FOR THE NEXT LAMBERT CALCULATION, COGA MUST BE

```

```

#          GUESSW MUST BE CLEAR FOR EACH SUCCEEDING LAMBERT CALL.
#
# ABORT EXIT MODES --
# IF SOLNSW WAS SET UPON EXITING, EITHER LAMBERT WAS ASKED TO COMPUTE A TRANSFER
# WAS TOO SMALL TO PRODUCE A REALISTIC TRANSFER BETWEEN R1VEC AND R2VEC. IN E
# ACCORDING TO THE NEEDS OF THE PARTICULAR USER. THE ABORT EXIT MODE MAY BE CO
# LAMBERT          DLOAD    ABS          # A MEASURE OF THE PROXIMITY TO 0 OR
#                  1-CHTH    # 360 DEGREES.
#                  DSU      BWM
#                  ONEBIT
#                  CHANGER2    # CHANGE R2VEC DIRECTION SLIGHTLY.
#                  DLOAD    DAD
#                  TDESIRED
#                  SOMETIME
#                  STCALL    TDESIRED    # INCREASE TDESIRED
#                  LAMBERT
#
# OUTPUT --
#          SCALE FACTOR
#          VARIABLE    IN POWERS OF 2    DESCRIPTION AND REMARKS
# Page 1268
# -----
#          VVEC          +7 FOR EARTH    DP INITIAL VELOCITY VECTOR IN METERS,
#          +5 FOR MOON    PROBLEM.
#          VTARGET       +7 FOR EARTH    DP RESULTANT VELOCITY VECTOR AT R2VEC
#          +5 FOR MOON
#          SOLNSW        NONE            INTERPRETER SWITCH WHICH IS SET IF TH
#          SOLUTION EXISTS.
#
#          FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS --
#          PARAMETERS WHICH MAY BE OF USE --
#          SCALE FACTOR
#          VARIABLE    IN POWERS OF 2    DESCRIPTION AND REMARKS
#          -----
#          SNTH          +1              DP SIN OF ANGLE BETWEEN R1VEC AND R2V
#          CSTD          +1              DP COSINE OF ANGLE
#          1-CSTD        +2              DP 1-CSTD
#          COGA          +5              DP COTAN OF INITIAL REQUIRED FLIGHT P
#          P             +4              DP RATIO OF SEMILATUS RECTUM TO INIT
#          R1A           +6              DP RATIO OF INITIAL RADIUS TO SEMI-M
#          R1 (32D)      +29 FOR EARTH    DP INITIAL RADIUS IN METERS
#          +27 FOR MOON
#          UR1           +1              DP UNIT VECTOR OF R1VEC
#          U2            +1              DP UNIT VECTOR OF R2VEC

```

```

#
#       PARAMETERS OF NO USE --
#       DP PARAMETERS -- EPSILONL, CSTH-RHO, TPREV, TERRLAMB, R2, RTNLAMB (SP), PLUS PU
#       ADDITIONAL INTERPRETIVE SWITCHES USED -- INFINFLG, 360SW, SLOPSW, ORDERSW

# Page 1269
# PROGRAM DESCRIPTION -- TIME-THETA SUBROUTINE                      DATE -- 1 SEPTEMBER 1967
# MOD NO. -- 0                                                    LOG SECTION -- CONIC SUBROUTINE
# MOD BY KRAUSE                                                    ASSEMBLY -- COLOSSUS REVISION 8
#
# FUNCTIONAL DESCRIPTION --
#       THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR AND A DESIRED TRUE-ANOMALY-DIFFERENCE TH
#       STATE IS TO BE UPDATED ALONG A CONIC TRAJECTORY, CALCULATES THE CORRESPONDING TIME-OF-F
#       PROVIDES THE OPTION OF COMPUTING THE NEW UPDATED STATE VECTOR.  THE RESULTING TRAJECTORY
#       CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EARTH OR THE MOON.  THE USE
#       EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO THE MUTABLE WTIHOUT INTRODUC
#       ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.
#
# THE RESTRICTIONS ARE --
#       1. THE ANGLE BETWEEN ANY POSITION VECTOR AND ITS VELOCITY VECTOR MUST BE GREATER THAN 1
#          AND LESS THAN 178 DEGREES 12.5 MINUTES.
#       2. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED IN THE
#          ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
#       THE AGC COMPUTATION TIME IS APPROXIMATELY .292 SECONDS.
#
# REFERENCES --
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP-SECTION 5
#
# INPUT -- ERASABLE INITIALIZATION REQUIRED
#
#       SCALE FACTOR
#       VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#       -----      -
#       RVEC          +29 FOR EARTH        DP INITIAL POSITION VECTOR IN METERS
#                   +27 FOR MOON
#       VVEC          +7 FOR EARTH         DP INITIAL VELOCITY VECTOR IN METERS/CENTISECON
#                   +5 FOR MOON
#       SNTH          +1                   DP SINE OF THE TRUE-ANOMALY-DIFFERENCE THROUGH
#       CSTH          +1                   DP COSINE OF THE ANGLE
#       RVSW          NONE                 AN INTERPRETIVE SWITCH TO BE SET IF ONLY TIME I
#                                     IS TO BE COMPUTED ALSO.
#       X1 (38D)      NONE                 INDEX REGISTER TO BE SET TO -2D OR -10D ACCORDI
#                                     RESPECTIVELY, IS THE CENTRAL BODY.
#
# SUBROUTINES CALLED --
# Page 1270

```

```

#      PARAM, GEOM, GETX, DELTIME, NEWSTATE
#
# CALLING SEQUENCE AND NORMAL EXIT MODES --
#      IF ONLY TIME IS DESIRED AS OUTPUT --
#      L      SET      CALL      # MUST BE IN INTERPRETIVE MODE BUT OVFINDE AR
#      L+1      RVS      W      #
#      L+2      TIMETHET      # RETURN WITH PL AT 0 AND T IN MPAC
#      L+3      ...      # CONTINUE
#
#      IF THE UPDATE STATE VECTOR IS DESIRED AS WELL --
#      L      CLEAR      CALL      # MUST BE IN INTERPRETIVE MODE BUT OVFINDE AR
#      L+1      RVS      W      #
#      L+2      TIMETHET      # RETURNS WITH PL AT 6.  THE INITIAL POSITION
#      # THE INITIAL VELOCITY VECTOR IN MPAC.
#      L+3      STOV      L      NEWVVEC
#      L+4      STADR
#      L+5      STORE      NEWRVEC      # NEWVVEC AND NEWRVEC ARE SYMBOLIC REPRESENTA
#      L+6      ...      # CONTINUE.
#
# ABORT EXIT MODES --
#      IF COGAFLAG AND/OR INFINFLG IS SET AT THE EXIT TO TIME-THETA, TIME-THETA WILL
#      AN ALARM CODE (ORIGINALLY 00607), AND NOT RETURN TO THE CALLING PROGRAM.  (P
#
# OUTPUT --
#
#      SCALE FACTOR
#      VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#      -----
#      T(30D)      +28      DP TRANSFER TIME IN CENTISECONDS
#      INFINFLG      NONE      AN INTERPRETIVE SWITCH WHICH IS SET I
#      # INFINITY (NO SOLUTION), CLEAR
#      COGAFLAG      NONE      AN INTERPRETIVE SWITCH WHICH IS SET I
#      # CLEAR IF NOT.
#
#      IN ADDITION, IF RVS      W      IS CLEAR, THE FOLLOWING ARE OUTPUT --
#      MPAC -      +7 FOR EARTH      DP TERMINAL VELOCITY VECTOR IN METERS
#      MPAC+5      +5 FOR MOON
#      OD - 5D      +29 FOR EARTH      DP TERMINAL POSITION VECTOR IN METERS
#      +27 FOR MOON
#
#      FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
# Page 1271
#
# DEBRIS --
#      PARAMETERS WHICH MAY BE OF USE --
#      SCALE FACTOR
#      VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS

```

```

#          -----
#          R1 (32D)          +29 FOR EARTH          DP MAGNITUDE OF INITIAL POSITION VECTOR, RVEC,
#          +27 FOR MOON
#          R1A              +6                    DP RATIO OF R1 TO SEMI-MAJOR AXIS (NEG. FOR HYP
#          P                 +4                    DP RATIO OF SEMILATUS RECTUM TO R1
#          COGA              +5                    DP COTAN OF ANGLE BETWEEN RVEC AND VVEC
#          UR1               +1                    DP UNIT VECTOR OF RVEC
#          U2                +1                    DP UNIT VECTOR OF VVEC
#          UN                +1                    DP UNIT VECTOR OF UR1*U2
#
# PARAMETERS OF NO USE --
#       SP PARAMETERS -- RTNTT, GEOMSGN, RTNPRM, MAGVEC2=R2 (DP), PLUS PUSHLIST LOCATIONS 0-11D
#       ADDITIONAL INTERPRETIVE SWITCHES USED -- NORMSW, 360SW

# Page 1272
# PROGRAM DESCRIPTION -- TIME-RADIUS SUBROUTINE          DATE -- 11 OCTOBER 1967
# MOD NO. -1                      LOG SECTION -- CONIC SUBROUTINES
# MOD BY KRAUSE                      ASSEMBLY -- COLOSSUS REVISION 88
#
# FUNCTIONAL DESCRIPTION --
#       THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR AND A DESIRED RADIUS TO WHICH THE
#       STATE IS TO BE UPDATED ALONG A CONIC TRAJECTORY, CALCULATES THE CORRESPONDING TIME-OF-F
#       PROVIDES THE OPTION OF COMPUTING THE NEW UPDATED STATE VECTOR.  THE RESULTING TRAJECTOR
#       CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EARTH OR THE MOON.  THE USE
#       EXTENDED USING OTHER PRIMARY BODIES BY SIMMPE ADDITIONS TO THE MUTABLE WITHOUT INTRODUC
#       ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.
#
#       IF THE DESIRED RADIUS IS BEYOND THE RADIUS OF APOCENTER OF THE CONIC OR BELOW THE RADIU
#       APSW WILL BE SET AND THE SUBROUTINE WILL RETURN THE APOCENTER OR PERICENTER SOLUTION,
#
# THE RESTRICTIONS ARE --
#       1. THE ANGLE BETWEEN ANY POSITION VECTOR AND ITS VELOCITY VECTOR MUST BE GREATER THAN 1
#          AND LESS THAN 178 DEGREES 12.5 MINUTES.
#       2. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED IN THE
#          EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#       3. AN ACCURACY DEGRADATION OCCURS AS THE SENSITIVITIES OF TIME AND UPDATED STATE VECTOR
#          RDESIRED INCREASE.  THIS WILL OCCUR NEAR EITHER APSIS OF THE CONIC AND WHEN THE CONI
#          PARTICULAR, IF THE CONIC IS AN EXACT CIRCLE, THE PROBLEM IS UNDEFINED AND THE SUBROU
#
#       THE AGC COMPUTATION TIME IS APPROXIMATELY .363 SECONDS.
#
# REFERENCES --
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP-SECTION 5
#
# INPUT -- ERASABLE INITIALIZATION REQUIRED.
#          SCALE FACTOR

```

#	VARIABLE	IN POWERS OF 2	DESCRIPTION AND REMARKS
#	-----	-----	-----
#	RVEC	+29 FOR EARTH	DP INITIAL POSITION VECTOR IN METERS
#		+27 FOR MOON	
#	VVEC	+7 FOR EARTH	DP INITIAL VELOCITY VECTOR IN METERS
#		+5 FOR MOON	
#	RDESIRED	+29 FOR EARTH	DP TERMINAL RADIAL DISTANCE ON CONIC
#		+27 FOR MOON	COMPUTED
#	SGNRDOT	NONE	SP TAG SET TO +.5 OR -.5 ACCORDING TO
#			POSITIVE OR NEGATIVE, RESPECTIVELY
#	# Page 1273		
#			SINGLE-VALUED PROBLEM.
#	X1 (38D)	NONE	INDEX REGISTER TO BE SET TO -2D OR -3D
#			RESPECTIVELY, IS THE CENTRAL
#	RVSW	NONE	AN INTERPRETIVE SWITCH TO BE SET IF C
#			IS TO BE COMPUTED ALSO.
#	# SUBROUTINES CALLED --		
#	PARAM, GEOM, GETX, DELTIME, NEWSTATE		
#	#		
#	# CALLING SEQUENCE AND NORMAL EXIT MODES --		
#	# IF ONLY TIME IS DESIRED AS OUTPUT --		
#	L	SET CALL	# MUST BE IN INTERPRETIVE MODE BUT OVFLND ARE
#	L+1	RVSW	
#	L+2	TIMERAD	# RETURN WITH PL AT 0 AND T IN MPAC
#	L+3	...	# CONTINUE
#	#		
#	# IF THE UPDATE STATE VECTOR IS DESIRED AS WELL --		
#	L	CLEAR CALL	# MUST BE IN INTERPRETIVE MODE BUT OVFLND ARE
#	L+1	RVSW	
#	L+2	TIMERAD	# RETURNS WITH PL AT 6. THE INITIAL POSITION
#			# THE INITIAL VELOCITY VECTOR IN MPAC.
#	L+3	STOVL NEWVVEC	
#	L+4	STADR	
#	L+5	STORE NEWRVEC	# NEWVVEC AND NEWRVEC ARE SYMBOLIC REPRESENTA
#	L+6	...	# CONTINUE
#	#		
#	# ABORT EXIT MODES --		
#	# IF SOLNSW AND/OR COGAFLAG AND/OR INFINFLG IS SET AT THE EXIT TO TIME-RADIUS,		
#	# TO POODOO WITH AN ALARM CODE (ORIGINALLY 00607), AND NOT RETURN TO THE CALLIN		
#	#		
#	# OUTPUT --		
#		SCALE FACTOR	
#	VARIABLE	IN POWERS OF 2	DESCRIPTION AND REMARKS
#	-----	-----	-----
#	T (30D)	+28	DP TRANSFER TIME IN CENTISECONDS.

```

#      INFINFLG      NONE      AN INTERPRETIVE SWITCH WHICH IS SET IF RDESIRED
#                                     INFINITY (NO SOLUTION), CLEAR IF A PHYS
#      COGAFLAG      NONE      AN INTERPRETIVE SWITCH WHICH IS SET IF RESTRICT
#                                     CLEAR IF NOT.
#      APSESW        NONE      AN INTERPRETIVE SWITCH WHICH IS SET IF RDESIRED
# Page 1274
#                                     LESS THAN RADIUS OF PERICENTER.  THE AP
#                                     WILL THEN BE RETURNED.  THE SWITCH IS C
#                                     APOCENTER.
#      SOLNSW        NONE      AN INTERPRETIVE SWITCH WHICH IS SET IF THE CONI
#                                     POINT IS AMBIGUOUS, VIOLATING RESTRICTI
#                                     MINUS-18, THE SWITCH IS CLEAR.
#
#      IN ADDITION, IF RVSW IS CLEAR, THE FOLLOWING ARE OUTPUT --
#      MPAC -          +7 FOR EARTH      DP TERMINAL VELOCITY VECTOR IN METERS/CENTISEC.
#      MPAC+5          +5 FOR MOON
#      OD - 5D         +29 FOR EARTH      DP TERMINAL POSITION VECTOR IN METERS (PL AT 6D
#      OD - 5D         +27 FOR MOON
#
#      FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS --
#      PARAMETERS WHICH MAY BE OF USE --
#                                     SCALE FACTOR
#      VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#      -----      -
#      R1 (32D)      +29 FOR EARTH      DP MAGNITUDE OF INITIAL POSITION VECTOR, RVEC,
#      R1 (32D)      +27 FOR MOON
#      R1A           +6                DP RATIO OF R1 TO SEMI-MAJOR AXIS (NEG. FOR HYP
#      P             +4                DP RATIO OF SEMILATUS RECTUM TO R1
#      COGA          +5                DP COTAN OF ANGLE BETWEEN RVEC AND VVEC
#      UR1           +1                DP UNIT VECTOR OF RVEC
#      U2            +1                DP UNIT VECTOR OF VVEC
#      UN            +1                DP UNIT VECTOR OF UR1*U2
#      CSTH          +1                DP COSINE OF TRUE ANOMALY DIFFERENCE BETWEEN RV
#      SNTH          +1                DP SINE OF TRUE ANOMALY DIFFERENCE.
#
#      PARAMETERS OF NO USE --
#      SP PARAMETERS -- RTNTT, GEOMSGN, RTNPRM, MAGVEC2*R2 (DP), PLUS PUSHLIST LOCATIO
#      ADDITIONAL INTERPRETIVE SWITCHES USED -- NORMSW, 360SW
#
# Page 1275
# PROGRAM DESCRIPTION -- APSIDES SUBROUTINE      DATE -- 1 SEPTEMBER 1967
# MOD NO. -- 0      LOG SECTION -- CONIC SUBROUTINES
# MOD BY KRAUSE      ASSEMBLY -- COLOSSUS REVISION 88

```

```

#
# FUNCTIONAL DESCRIPTION --
#   THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR, CALCULATES THE RADIUS OF PERIAPSE,
#   ECCENTRICITY OF THE RESULTING CONIC TRAJECTORY, WHICH MAY BE A STRAIGHT LINE,
#   CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EARTH OR THE MOON.
#   BE EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO THE MUTABLE WEIGHTS.
#   ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.
#
# THE RESTRICTIONS ARE --
#   1. IF APOCENTER IS BEYOND THE SCALING OF POSITION, THE SCALE FACTOR LIMIT (53)
#   TO THE EARTH OR 134,217,727.5 METERS WITH RESPECT TO THE MOON) WILL BE REACHED.
#   2. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED.
#   ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
#   THE AGC COMPUTATION TIME IS APPROXIMATELY .103 SECONDS.
#
# REFERENCES --
#   MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP-SECTION 10.
#
# INPUT -- ERASABLE INITIALIZATION REQUIRED
#
#   SCALE FACTOR
#   VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#   -----
#   RVEC          +29 FOR EARTH        DP INITIAL POSITION VECTOR IN METERS
#               +27 FOR MOON
#   VVEC          +7 FOR EARTH         DP INITIAL VELOCITY VECTOR IN METERS,
#               +5 FOR MOON
#   X1 (38D)      NONE                 INDEX REGISTER TO BE SET TO -2D OR -1D
#                                   RESPECTIVELY, IS THE CENTRAL BODY
#
# SUBROUTINES CALLED --
#   PARAM, GEOM
#
# CALLING SEQUENCE AND NORMAL EXIT MODES --
# Page 1276
#   IF ONLY TIME IS DESIRED AS OUTPUT --
#   L      CALL      # MUST BE IN INTERPRETIVE MODE BUT OVFINDD ARE
#   L+1    APSIDES   # RETURNS WITH PL AT 0, RADIUS OF APOCENTER 1
#   L+2    STODL     APOAPSE
#   L+3    OD
#   L+4    STORE     PERIAPSE      # APOAPSE AND PERIAPSE ARE SYMBOLIC REPRESENTATIONS
#   L+5    ...       # CONTINUE
#
# OUTPUT --
#
#   SCALE FACTOR
#   VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS

```



```

#      -----
#      MPAC          +29 FOR EARTH      DP RADIUS OF APOCENTER IN METERS
#                  +27 FOR MOON
#      OD-1D        +29 FOR EARTH      DP RADIUS OF PERICENTER IN METERS
#                  +27 FOR MOON
#      ECC           +3                 DP ECCENTRICITY OF CONIC TRAJECTORY
#
#      FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS --
#      PARAMETERS WHICH MAY BE OF USE --
#                  SCALE FACTOR
#      VARIABLE      IN POWERS OF 2      DESCRIPTION AND REMARKS
#      -----
#      R1 (32D)      +29 FOR EARTH      DP MAGNITUDE OF INITIAL POSITION VECTOR, RVEC,
#                  +27 FOR MOON
#      R1A           +6                 DP RATIO OF R1 TO SEMI-MAJOR AXIS (NEG. FOR HYP
#      P             +4                 DP RATIO OF SEMILATUS RECTUM TO R1
#      COGA          +5                 DP COTAN OF ANGLE BETWEEN RVEC AND VVEC
#      UR1           +1                 DP UNIT VECTOR OF RVEC
#      U2            +1                 DP UNIT VECTOR OF VVEC
#      UN            +1                 DP UNIT VECTOR OF UR1*U2
#      MAGVEC2       +7 FOR EARTH      DP MAGNITUDE OF VVEC
#                  +5 FOR MOON
#
#      PARAMETERS OF NO USE --
#      SP PARAMETERS -- RTNAPSE, GOMSGN, RTNPRM, PLUS PUSHLIST LOCATIONS 0-5, 10D-11D,
#      ADDITIONAL INTERPRETIVE SWITCHES USED -- NORMSW
#
#      SETLOC  CONICS
#
# Page 1277
#      BANK
#
#      COUNT  12/CONIC
#
#      EBANK=  UR1
#      SETPD   BOV
#              0
#              +1
#      VLOAD*
#              MUTABLE,1
#      STOVL   14D
#              RRECT
#      UNIT    SSP
#              ITERCTR
#              20D

```

	STODL	URRECT	
		36D	
	STOVL	R1	
		RRECT	
	DOT	SL1R	
		VRECT	
	DMP	SL1R	
		1/ROOTMU	# 1/ROOTMU (-17 OR -14)
	STOVL	KEPC1	# C1=R.V/ROOTMU (+17 OR +16)
		VRECT	
	VSQ	DMPR	
		1/MU	# 1/MU (-34 OR -28)
	DMP	SL3	
		R1	
	DSU	ROUND	
		D1/64	
	STORE	KEPC2	# C2=RV.V/MU -1 (+6)
	BDSU	SR1R	
		D1/64	
	DDV		
		R1	
	STORE	ALPHA	# ALPHA=(1-C2)/R1 (-22 OR -20)
	BPL	DLOAD	# MAXIMUM X DEPENDS ON TYPE OF CONIC
		1REV	
		-50SC	# -50SC (+12)
	DDV	BOV	
		ALPHA	
		STOREMAX	
	SQRT	GOTO	
		STOREMAX	
# Page 1278			
1REV	SQRT	BDDV	
		2PISC	# 2PISC (+6)
	BOV		
		STOREMAX	
STOREMAX	STORE	XMAX	
	DMP	PDDL	
		1/ROOTMU	
		ALPHA	
	NORM	PDDL	
		X1	
	SL*	DDV	
		0	-6,1

	BOV	BMN	
		MODDONE	
		MODDONE	# MPAC=PERIOD
PERIODCH	PDDL	ABS	# OD=PERIOD
		TAU.	
	DSU	BMN	
		OD	
		MODDONE	
	SIGN		
		TAU.	
	STODL	TAU.	
	GOTO		
		PERIODCH	
MODDONE	SETPD	DLOAD	
		0	
		XKEPNEW	
	STORE	X	
	SIGN	BZE	
		TAU.	
		BADX	
	BMN	ABS	
		BADX	
	DSU	BPL	
		XMAX	
		BADX	
STORBND	DLOAD	BPL	
		TAU.	
		STOREMIN	
	DLOAD	DCOMP	
		XMAX	
	STODL	XMIN	
		KEPZERO	
	STCALL	XMAX	
		DXCOMP	
STOREMIN	DLOAD		
		KEPZERO	
	STORE	XMIN	
DXCOMP	DLOAD	DMPR	
# Page 1279			
		TAU.	
		BEE22	
	ABS		
	STODL	EPSILONT	
		XPREV	
XDIFF	BDSU		
		X	

```

STORE    DELX

KEPLOOP  DLOAD  DSQ
          X      # X=XKEP
          NORM  PUSH      # OD=XSQ (+34 OR +32 -N1)      PL AT 2
          X1
          DMP    SRR*
          ALPHA
          0      -6,1
          STCALL XI      # XI=ALPHA XSQ (+6)
          DELTIME
          BOV    BDSU
          TIMEOVFL      # UNLIKELY
          TAU.
          STORE  DELT      # DELT=DELINDEP
          ABS    BDSU
          EPSILONT
          BPL    DLOAD
          KEPCONVG
          T
          DSU    NORM
          TC
          X1
          PDDL  NORM
          DELX
          X2
          XSU,1 DMP
          X2
          DELT
          SLR*   DDV
          1,1
          SR1    PUSH      # OD=TRIAL DELX      PL AT 2
          BPL    DLOAD
          POSDELX
          X
          STORE  XMAX      # MOVE MAX BOUND IN
          BDSU   DSU      #
          XMIN      PL AT 0
          BOV    BPL
          NDXCHNGE
          NDXCHNGE
          DLOAD  GOTO
          OD
          NEWDELX

```

July 12, 2016

285

NDXCHNGE	DLOAD	DSU XMIN X	
	DMPR	GOTO DP9/10 NEWDELX	# TO FORCE MPAC +2 TO ZERO
POSDELX	DLOAD	X	
	STORE	XMIN	# MOVE MIN BOUND IN
	BDSU	DSU	#
		XMAX	PL AT 0
	BOV	BMN PDXCHNGE PDXCHNGE	
	DLOAD		
NEWDELX		OD	
	STORE	DELX	
	BZE	DAD KEPCONVG X	
	STODL	X T	
BRNCHCTR	STORE	TC	
	RTB	BHIZ CHECKCTR KEPCONVG	
	GOTO	KEPLOOP	# ITERATE
PDXCHNGE	DLOAD	DSU XMAX X	
	DMPR	GOTO DP9/10 NEWDELX	# TO FORCE MPAC +2 TO ZERO
BADX	DLOAD	SR1 XMAX	
	SIGN	TAU.	
	STCALL	X STORBND	
# Page 1281 TIMEOVFL	DLOAD	BMN X	# X WAS TOO BIG

		NEGTOVFL	
	STORE	XMAX	
CMNTOVFL	DLOAD	SR1	
		DELX	
	STORE	DELX	
	BZE	BDSU	
		KEPRTN	
		X	
	STODL	X	
		TC	
	STCALL	T	
		BRNCHCTR	
NEGTOVFL	STCALL	XMIN	
		CMNTOVFL	
KEPCONVG	DLOAD	SR4R	
		R1	
	DSU	VXSC	
		XSQC(XI)	
		URRECT	
	VSL1	PDDL	# OD=(R1-XSQC(XI))URRECT (+33 OR +31)
		X	
	DSQ	NORM	
		X1	
	DMPR	DMPR	
		1/ROOTMU	
		X	
	DMP	SRR*	
		S(XI)	
		0	-7,1
	BDSU		
		T	
	SL1	VXSC	
		VRECT	
	VSL1	VAD	
	VSL4		
	STORE	RCV	# RCV (+29 OR +27)
	ABVAL	NORM	
		X2	
	STODL	RCNORM	
		XI	
	DMPR	DSU	
		S(XI)	
		D1/128	
	DMP	SL1R	
		ROOTMU	

July 12, 2016

287

Page 1282

```
DMP      SLR*
          X
          0      -3,2
DDV      VXSC
          RCNORM
          URRECT
VSL1     PDDL      # OD=URRECT(XI S(XI)-1)X ROOTMU/RCV (+15
          XSQC(XI)  # OR +13)                      PL AT 6
SLR*     DDV
          0      -4,2
          RCNORM
BDSU     VXSC
          D1/256
          VRECT
VAD      VSL8
STADR    #
STODL    VCV      # VCV (+7 OR +5)
          T
STODL    TC
          X
STCALL   XPREV
          KEPRTN
```

Page 1283

DELTIME

```
EXIT      # MPAC=XI (+6), OD=XSQ (+34 OR +32 -N1)
TC        POLY
DEC       8
2DEC      .083333334
2DEC      -.266666684
2DEC      .406349155
2DEC      -.361198675
2DEC      .210153242
2DEC      -.086221951
2DEC      .026268812
2DEC      -.006163316
2DEC      .001177342
```

```
RTB      PDDL      # 3D=C2 XSQ S(XI) (+35 OR +33) PL AT 6
          TPMODE
          R1
```


July 12, 2016

289

	SR	TAD	#	PL AT 3
		6		
	NORM	DMP	# TO PRESERVE SIGNIF.	
		X1		
		X		
	SR*	TAD	# X(C2 XSQ S(XI) +R1) (+49 OR +46) PL AT 0	
		0	-3,1	
	SL4R	DMPR		
		1/ROOTMU		
	STORE	T		
	RVQ			
# Page 1285				
ITERATOR	BONCLR	DLOAD		
		SLOPESW		
		FIRSTIME		
		DEP		
	DSU	NORM		
		DEPREV		
		X1		
	PDDL	NORM		
		DELINDEP		
		X2		
	XSU,1	DMP		
		X2		
		DELDEP		
	SLR*	DDV	#	PL UP 2
		1,1		
	SR1	BOFF		
		ORDERSW		
		SGNCHECK		
	ABS	SIGN	# IN CASE 2ND DERIV. CHANGED SIGN, MUST	
		DELDEP	# DISREGARD IT TO FIND MIN.	
SGNCHECK	PUSH	BPL	# TRIAL DELINDEP	PL DOWN 2
		POSDEL		
	DLOAD	BON		
		INDEP		
		ORDERSW		
		MINCHECK		
	STORE	MAX	# IF NOT 2ND ORDER, CAN MOVE MAX BOUND IN.	
MINCHECK	BDSU	DSU		
		MIN		
	BOV	BPL		
		MODNGDEL		

		MODNGDEL	
	GOTO	DELOK	
MODNGDEL	DLOAD	DSU	# TRIAL DELINDEP WOULD EXCEED MIN BOUND
		MIN	
		INDEP	
	DMP	GOTO	
		DP9/10	
		NEWDEL	
FIRSTIME	DLOAD	DMP	
		MIN	
		TWEEKIT	# DLOAD TWEEKIT(40D) SENSITIVE TO CHANGE.
	PDDL	DMP	# S2(41D) SHOULDN'T CONTAIN HI ORDER ONES
# Page 1286		MAX	
		TWEEKIT	
	DSU		
	SIGN	GOTO	
		DELDEP	
		SGNCHECK	
POSDEL	DLOAD	BON	
		INDEP	
		ORDERSW	
		MAXCHECK	
	STORE	MIN	# IF NOT 2ND ORDER, CAN MOVE MIN BOUND IN.
MAXCHECK	BDSU	DSU	
		MAX	
	BOV	BMN	
		MODPSDEL	
		MODPSDEL	
DELOK	DLOAD		
		OD	
NEWDEL	STORE	DELINDEP	
	RVQ		
MODPSDEL	DLOAD	DSU	
		MAX	
		INDEP	
	DMP	GOTO	
		DP9/10	
		NEWDEL	

July 12, 2016

291

CHECKCTR	CS	ONE
	INDEX	FIXLOC
	AD	ITERCTR
	INDEX	FIXLOC
	TS	ITERCTR
	TS	MPAC
	TC	DANZIG

Page 1287

NEWSTATE	DLOAD	SR4R	
		R1	
	DSU	VXSC	
		XSQC(XI)	
		UR1	
	VSL1	PDDL	# OD=(R1-XSQC(XI))UR1 (+33 OR 31) PL AT 6
		X	
	DSQ	NORM	
		X1	
	DMPR	DMPR	
		1/ROOTMU	
		X	
	DMP	SRR*	
		S(XI)	
		0	-7,1
	BDSU		
		T	
	SL1	VXSC	
		VVEC	
	VSL1	VAD	#
	VSL4	PUSH	PL AT 0
	ABVAL		
LAMENTER	NORM		
		X1	
	STODL	R2	
		XI	
	DMP	DSU	
		S(XI)	
		D1/128	
	DMP	SL1R	
		ROOTMU	
	DMP	SLR*	
		X	
		0 -3,1	
	DDV	VXSC	
		R2	
		UR1	

VSL1	PDDL	# 6D=V2VEC PART (+15 OR 13)	PL AT 12
	XSQC(XI)		
SLR*	DDV		
	0 -4,1		
	R2		
BDSU			
	D1/256		
VXSC	VAD		
	VVEC	#	PL AT 6
VSL8	RVQ		

Page 1288

SETLOC CONICS1
BANK

COUNT 04/CONIC

DO NOT DISTURB THE ORDER OF THESE CDS, OVERLAYS HAVE BEEN MADE.

BEE17 DEC 0 # KEEP WITH D1/8 2DEC 1.0B-17 (0000004000)

D1/8 2DEC 1.0 B-3

D1/128 2DEC 1.0 B-7

D1/64 2DEC 1.0 B-6

D1/4 2DEC 1.0 B-2

D1/16 2DEC 1.0 B-4

D1/32 2DEC 1.0 B-5

D1/1024 2DEC 1.0 B-10

D1/256 2DEC 1.0 B-8

DP9/10 2DEC .9

KEPZERO EQUALS L06ZEROS
-50SC 2DEC -50.0 B-12

2PISC 2DEC 6.28318530 B-6

BEE19 EQUALS D1/32 -1 # 2DEC 1.0 B-19 (00000 01000)

BEE22 EQUALS D1/256 -1 # 2DEC 1.0 B-22 (00000 00100)

ONEBIT 2DEC 1.0 B-28

COGUPLIM 2DEC .999511597

293

PARAM	STQ	CLEAR	# MPAC=V1VEC, OD=R1VEC	PL AT 6
		RTNPRM		
		NORMSW		

```

# Page 1290
CLEAR
COGAFLAG
SSP CALL
GEOMSGN
37777 # GAMMA ALWAYS LESS THAN 180DEG
GEOM # MPAC=SNGA (+1), OD=CSGA (+1) PL AT 2
STODL 36D # 36D=SIN GAMMA (+1) PL AT 0

SR DDV
5

36D
STOVL* COGA
MUTABLE,1
STODL 1/MU
MAGVEC2
DSQ NORM
X1
DMPR DMP
1/MU
R1

SRR*
0 -3,1
PUSH BDSU # OD=R1 V1SQ/MU (+6) PL AT 2
D1/32
STODL R1A # R1A (+6) PL AT 0

DMP NORM
36D
X1
DMP SR*
36D
0 -4,1
STCALL P # P (+4)
RTNPRM

# Page 1291
GEOM UNIT # MPAC=V2VEC, OD=R1VEC PL AT 6
STODL U2 # U2 (+1)
36D
STOVL MAGVEC2 # PL AT 0
UNIT
STORE UR1 # UR1 (+1)
DOT SL1
U2
PDDL # OD=CSTH (+1) PL AT 2

```

		36D		
	STOVL	R1	# R1 (+29 OR +27)	
		UR1		
	VXV	VSL1		
		U2		
	BON	SIGN		
		NORMSW		
		HAVENORM		
		GEOMSGN		
	UNIT	BOV		
		COLINEAR		
UNITNORM	STODL	UN	# UN (+1)	
		36D		
	SIGN	RVQ	# MPAC=SNTH (+1), 34D=SNTH.SNTH (+2)	
		GEOMSGN		
COLINEAR	VSR1	GOTO		
		UNITNORM		
HAVENORM	ABVAL	SIGN		
		GEOMSGN		
	RVQ		# MPAC=SNTH (+1), 34D=SNTH.SNTH (+2)	
# Page 1292				
	BANK	12		
	SETLOC	CONICS		
	BANK			
	COUNT	12/CONIC		
GETX	AXT,2	SSP	# ASSUMES P (+4) IN MPAC	
		3		
		S2		
		1		
	CLEAR			
		360SW		
	SQRT	PDDL	# OD=SQRT(P)	PL AT 2
		CSTH		
	SR1	BDSU		
		D1/4		
	PDDL	SRR	#	PL AT 4D
		SNTH		
		6		
	DDV		#	PL AT 2
	BOV			
		360CHECK		

	DSU	DMP		
		COGA	#	PL AT 0
	SL2R	BOV		
		360CHECK		
WLOOP	PUSH	DSQ	# OD=W (+5)	PL AT 2
	TLOAD	PDDL	# 2D=WSQ (+10)	PL AT 5
		MPAC		
		R1A		
	SR4	TAD	#	PL AT 2
	BMN	SQRT		
		INFINITY		
	ROUND	DAD	#	PL AT OD
	BOV	TIX,2		
		RESETX2		
		WLOOP		
	BDDV	BOV		
		D1/128		
		INFINITY		
POLYCOEF	BMN	PUSH	# OD=1/W (+2) OR 16/W (+6)	PL AT 2
		INFINITY		
	DSQ			
	NORM	DMP		
		X1		
		R1A		
	SRR*	EXIT		
		0	-10D,1	
# Page 1293				
	TC	POLY		
	DEC	5		
	2DEC	.5		
	2DEC	-.166666770		
	2DEC	.100000392		
	2DEC	-.071401086		
	2DEC	.055503292		
	2DEC	-.047264098		
	2DEC	.040694204		
	TC	INTPRET		
	DMP	SL1R	#	PL AT OD

	PUSH	BON		
		360SW		
		TRUE360X		
XCOMMON	DSQ	NORM		
		X1		
	DMP	SRR*		
		R1A		
		0	-12D,1	
	STODL	XI	# XI (+6)	
		R1		
	SR1	SQRT		
	ROUND	DMP		
	SL4R		#	PL AT 0
	STORE	X	# X (+17 OR +16)	
	DSQ	NORM		
		X1		
	PDDL	DMP	# OD=XSQ (+34 OR +32 -N1)	PL AT 2
		P		
		R1		
	SL3	SQRT		
	DMP	SL3R		
		COGA		
	STODL	KEPC1		
		R1A		
	BDSU	CLEAR		
		D1/64		
		INFINFLG		
	STORE	KEPC2		
# Page 1294	RVQ			
RESETX2	AXT,2			
		3		
360CHECK	SETPD	BPL		
		OD		
		INVRSEQN		
	SET			
		360SW		
INVRSEQN	DLOAD	SQRT		
		P		
	PDDL	DMP	# OD=SQRT(P) (+2)	PL AT 2
		SNTH		

	COGA			
SL1	PDDL	# 2D=SNTH COGA (+5)		PL AT 4
	CSTH			
SR4	DAD			
	D1/32			
DSU	DMP	#		PL AT 2,0
NORM	BDDV			
	X1			
	SNTH			
SLR*	ABS	# NOTE: NEAR 360 CASE TREATED DIFFERENTLY		
	0	-5,1		
PUSH	DSQ	# OD=1/W (-1)		PL AT 2
STODL	34D			
	D1/16			
1/WLOOP	PUSH	# 2D=G (+4)		PL AT 4
	RTB	#		PL AT 7
	TPMODE			
	R1A			
DMP	SR4			
	34D			
TAD				
BMN	SQRT	#		PL AT 4
	INFINITY			
DAD		#		PL AT 2
TIX,2	NORM			
	1/WLOOP			
	X1			
BDDV				
SLR*	GOTO	#		PL AT 0
	0	-7,1		
	POLYCOEF			
# Page 1295				
TRUE360X				
DLOAD	BMN			
	R1A			
	INFINITY			
SQRT	NORM			
	X1			
BDDV	SL*			
	2PISC			
	0	-3,1		
DSU	PUSH	# OD=2PI/SQRT(R1A) -X		PL AT 0,2
GOTO				
	XCOMMON			
INFINITY	SETPD			
	BOV	# NO SOLUTION EXISTS SINCE CLOSURE THROUGH		
	0	# INFINITY IS REQUIRED		

OVFLCLR	SET	OVFLCLR RVQ INFINFLG		
# Page 1296 LAMBERT	STQ	SETPD RTNLAMB OD		
	BOV	+1		
	SSP	VLOAD* ITERCTR 20D MUTABLE,1		
	STODL	1/MU TDESIRED		
	DMPR	BEE19		
	STORE	EPSILONL		
	SET	VLOAD SLOPESW R1VEC		
	PDVL	CALL	# OD=R1VEC (+29 OR +27)	PL AT 6
		R2VEC	# MPAC=R2VEC (+29 OR +27)	
		GEOM		
	STODL	SNTH	# OD=CSTH (+1)	PL AT 2
		MAGVEC2		
	NORM	PDDL	#	PL AT 4
		X1		
		R1		
	SR1	DDV	#	PL AT 2
	SL*	PDDL	# DXCH WITH OD, OD=R1/R2 (+7)	PL AT 0,2
		0	-6,1	
	STADR			
	STORE	CSTH	# CSTH (+1)	
	SR1	BDSU		
		D1/4		
	STORE	1-CSTH	# 1-CSTH (+2)	
	ROUND	BZE		
		360LAMB		
	NORM	PDDL	#	PL AT 4
		X1		
		OD		
	SR1	DDV	#	PL AT 2
	SL*	SQRT		

```

                                0      -3,1
                                PDDL  SR      # 2D=SQRT(2R1/R2(1-CSTH)) (+5) PL AT 4
                                SNTH
                                6
                                DDV  DAD      #                               PL AT 2
                                1-CSTH
                                STADR
                                STORE COGAMAX
                                BOV  BMN      # IF OVFL, COGAMAX=COGUPLIM
# Page 1297
                                UPLIM
                                MAXCOGA      # IF NEG, USE EVEN IF LT COGLOLIM, SINCE
                                BMN          # THIS WOULD BE RESET IN LAMBLOOP
                                COGUPLIM     # IF COGAMAX GT COGUPLIM, COGAMAX=COGUPLIM
                                MAXCOGA      # OTHERWISE OK, SO GO TO MAXCOGA
UPLIM                          DLOAD
                                COGUPLIM     # COGUPLIM=.999511597 = MAX VALUE OF COGA
                                STORE COGAMAX # NOT CAUSING OVFL IN R1A CALCULATION
MAXCOGA                        DLOAD
                                CSTH
                                SR  DSU      #                               PL AT 0
                                6
                                STADR
                                STODL CSTH-RHO
                                GEOMSGN
                                BMN  DLOAD
                                LOLIM
                                CSTH-RHO
                                SL1  DDV
                                SNTH
                                BOV
                                LOLIM
MINCOGA                        STORE COGAMIN # COGAMIN (+5)
                                BON  SSP
                                GUESSW
                                NOGUESS
                                TWEKIT
                                00001
                                DLOAD
                                COGA
LAMBLOOP                       DMP
                                SNTH
                                SR1  DSU
                                CSTH-RHO
                                NORM  PDDL   # OD=SNTH COGA-(CSTH-RHO) (+7+C(XI)) PL=2

```

Page 1298

```

X1
1-CSTH
SL*   DDV          # 1-CSTH (+2)          PL AT 0
      0 -9D,1
BMN   BZE
      NEGP
      NEGP
STODL P            # P=(1-CSTH)/(SNTH COGA-(CSTH-RHO)) (+4)
      COGA
DSQ   DAD
      D1/1024
NORM  DMP
      X1

P
SR*   BDSU
      0            -8D,1
      D1/32
STODL R1A          # R1A=2-P(1+COGA COGA) (+6)

P
BOV   CALL
      HIENERGY
      GETX
DLOAD
T
STODL TPREV
      XI
BON   CALL
      INFINFLG
      NEGP          # HAVE EXCEEDED THEORETICAL BOUNDS
      DELTIME
BOV   BDSU
      BIGTIME
      TDESIRE
STORE TERRLAMB
ABS   BDSU
      EPSILONL
BPL   RTB
      INITV
      CHECKCTR
BHIZ  CALL
      SUFFCHEK
      ITERATOR
DLOAD BZE
      MPAC

```

		SUFFCHEK	
	DAD		
		COGA	
	STCALL	COGA	
		LAMBLOOP	
NEGP	DLOAD	BPL	# IMPOSSIBLE TRAJECTORY DUE TO INACCURATE
		DCOGA	# BOUND CALCULATION. TRY NEW COGA.
		LOENERGY	
HIENERGY	SETPD	DLOAD	# HIGH ENERGY TRAJECTORY RESULTED.
		0	
		COGA	# IN OVFL OF P OR R1A, OR XI EXCEEDING 50.
	STORE	COGAMIN	# THIS IS THE NEW BOUND.
COMMONLM	DLOAD	SR1	
		DCOGA	
# Page 1299			
	STORE	DCOGA	# USE DCOGA/2 AS DECREMENT
	BZE	BDSU	
		SUFFCHEK	
		COGA	
	STCALL	COGA	
		LAMBLOOP	
BIGTIME	DLOAD		
		TPREV	
	STORE	T	
LOENERGY	SETPD	DLOAD	# LOW ENERGY TRAJECTORY RESULTED
		0	
		COGA	# IN OVERFLOW OF TIME.
	STCALL	COGAMAX	# THIS IS THE NEW BOUND.
		COMMONLM	
SUFFCHEK	DLOAD	ABS	
		TERRLAMB	
	PDDL	DMP	
		TDESIRED	#
		D1/4	
	DAD	DSU	#
		ONEBIT	
	BPL	SETGO	
		INITV	
		SOLNSW	
		RTNLAMB	
			PL AT 2D
			PL AT 0D

July 12, 2016

303

360LAMB	SETPD	SETGO 0 SOLNSW RTNLAMB	# LAMBERT CANNOT HANDLE Csth=1	
NOGUESS	SSP	DLOAD TWEKIT 20000 COGAMIN		
	SR1	PDDL COGAMAX	#	PL AT 2
	SR1	DAD		
	STADR		#	PL AT 0
	STORE	COGA		
	STCALL	DCOGA LAMBLOOP		
# Page 1300 LOLIM	DLOAD	GOTO COGLOLIM MINCOGA	# COGLOLIM=-.999511597	
INITV	DLOAD	NORM R1 X1		
	PDDL	SR1 P	#	PL AT 2
	DDV		#	PL AT 0
	SL*	SQRT 0 -4,1		
	DMP	SL1 ROOTMU		
	PUSH	DMP COGA	# OD=VTAN (+7)	PL AT 2
	SL	VXSC 5 UR1		
	PDDL		# XCH WITH OD	PL AT 0,6
	VXSC	VSL1 UN		
	VXV	VAD UR1	#	PL AT 0
	VSL1	CLEAR SOLNSW		
	STORE	VVEC		
	SLOAD	BZE VTARGETAG		

		TARGETV		
	GOTO	RTNLAMB		
TARGETV	DLOAD	CALL		
		MAGVEC2		
		LAMENTER		
	STCALL	VTARGET		
		RTNLAMB		
# Page 1301				
TIMERAD	STQ	SETPD	#	PL AT 0
		RTNTR		
		0		
	BOV			
		+1		
	VLOAD	PDVL	#	PL AT 6
		RVEC		
		VVEC		
	CALL			
		PARAM		
	BOV	DLOAD		
		COGAOVFL	#	PL AT 0
		D1/32		
	DSU	DMP		
		R1A		
		P		
	SQRT	DMP		
		COGA		
	SL4	VXSC		
		U2		
	PDDL	DSU	#	PL AT 6
		D1/64		
		R1A		
	VXSC	VSU	#	PL AT 0
		UR1		
	VSL4	UNIT		
	BOV			
		CIRCULAR		
	PDDL	NORM	# 0D=UNIT(ECC) (+3)	PL AT 6
		RDESIRED	# 35D=ECC (+3)	
		X1		
	PDDL	DMP	#	PL AT 8
		R1		
		P		
	SL*	DDV	#	PL AT 6

	0,1	
DSU	DDV	
	D1/16	
	36D	# 36D=ECC (+3)
STORE	COSF	
BOV	DSQ	
	BADR2	
BDSU	BMN	
	D1/4	
	BADR2	
SQRT	SIGN	
	SGNRDOT	
CLEAR		
	APSESW	

Page 1302
TERMNVEC

VXSC	VSL1		
	UN		
VXV	PDVL	# VXCH WITH OD	PL AT 0,6
	OD		
VXSC	VAD	#	PL AT 0
	COSF		
VSL1	PUSH	# OD=U2	PL AT 6
DOT	DDV	# LIMITS RESULT TO POSMAX OR NEGMAX	
	UR1		
	DP1/4		
SR1	BOV	# SCALE BACK DOWN TO NORMAL	
	+1	# CLEAR OVFINF IF SET	
STOVL	CSTH	# CSTH (+1)	
	UR1		
VXV	VSL1		
DOT	SL1		
	UN		
STODL	SNTH	# SNTH (+1)	
	P		
CALL			
	GETX		
CLRGO			
	SOLNSW		
	COMMNOUT		

CIRCULAR	SETPD	SETGO
		0
		SOLNSW
		ABTCONIC

BADR2	DLOAD	SIGN		
		LODPHALF		
		COSF		
	STODL	COSF		
		KEPZERO		
	SETGO			
		APSESW		
		TERMNVEC		
# Page 1303				
APSIDES	STQ	SETPD	#	PL AT 0
		RTNAPSE		
		OD		
	BOV			
		+1		
	VLOAD	PDVL	#	PL AT 6
		RVEC		
		VVEC		
	CALL			
		PARAM		
	BOV		#	PL AT 0
		GETECC		
GETECC	DMP	SL4		
		R1A		
	BDSU	SQRT		
		D1/64		
	STORE	ECC		
	DAD	PDDL	#	PL AT 2
		D1/8		
		R1		
	DMP	SL1		
		P		
	DDV		#	PL AT 0
	PDDL	NORM	# OD=RP (+29 OR +27)	PL AT 2
		R1A		
		X1		
	PDDL	SL*	#	PL AT 4
		R1		
		0	-5,1	
	DDV	DSU	#	PL AT 2,0
	BOV	BMN		
		INFINAPO		
		INFINAPO		
	GOTO			
		RTNAPSE		

July 12, 2016

307

```
INFINAPO      DLOAD  GOTO          # RETURNS WITH APOAPSIS IN MPAC, PERIAPSIS
                LDPOSMAX
                RTNAPSE          # THAT PL IS AT 0.
```

Page 1304

```
ABTCONIC      EXIT
                TC          P00D00
                OCT         00607
```

Page 1305

```
SETLOC  CONICS1
BANK
```

```
COUNT  04/CONIC
```

```
MUTABLE      2DEC*  3.986032 E10 B-36*    # MUE
                2DEC*  .25087606 E-10 B+34*  # 1/MUE
                2DEC*  1.99650495 E5 B-18*    # SQRT(MUE)
                2DEC*  .50087529 E-5 B+17*    # 1/SQRT(MUE)
                2DEC    4.902778 E8 B-30      # MUM
                2DEC    .203966 E-8 B+28      # 1/MUM
                2DEC*  2.21422176 E4 B-15*    # SQRT(MUM)
                2DEC*  .45162595 E-4 B+14*    # 1/SQRT(MUM)
```

```
LDPOSMAX      EQUALS  LODPMAX          # DPPOSMAX IN LOW MEMORY.
```

ERASABLE ASSIGNMENTS

KEPLER SUBROUTINE

INPUT --

```
# RRECT      ERASE  +5
# VRECT      ERASE  +5
# TAU.       ERASE  +1
# XKEP       ERASE  +1
# TC         ERASE  +1
# XPREV      ERASE  +1
1/MU         EQUALS 14D
ROOTMU      EQUALS 16D
```

```

1/ROOTMU      EQUALS  18D

# OUTPUT --
# RCV          ERASE   +5
# VCV          ERASE   +5
# RC           ERASE   +1
# XPREV        ERASE   +1

# DEBRIS --
ALPHA         EQUALS  8D
XMAX          EQUALS  10D
# Page 1306
XMIN          EQUALS  12D
X             EQUALS  20D
XI            EQUALS  24D
S(XI)         EQUALS  26D
XSQC(XI)      EQUALS  28D
T             EQUALS  30D
R1            EQUALS  32D
KEPC1         EQUALS  34D
KEPC2         EQUALS  36D

# DELX         ERASE   +1
# DELT         ERASE   +1
# URRECT       ERASE   +5
# RCNORM       ERASE   +1
# XPREV        EQUALS  XKEP

# LAMBERT SUBROUTINE
#
# INPUT --
# R1VEC        ERASE   +5
# R2VEC        ERASE   +5
# TDESIRED     ERASE   +1
# GEOMSGN      ERASE   +0
# GUESSW              # 0 IF COGA GUESS AVIABLE, 1 IF NOT
# COGA          ERASE   +1 # INPUT ONLY IF GUESS IS ZERO.
# NORMSW              # 0 IF UN TO BE COMPUTED, 1 IF UN INPUT
# UN            ERASE   +5 # ONLY USED IF NORMSW IS 1
# VTARGETAG     ERASE   +0
# TWEKIT        EQUALS  40D # ONLY USED IF GUESSW IS 0

# OUTPUT --
# VTARGET       ERASE   +5 # AVAILABLE ONLY IF VTARGETAG IS ZERO.
# V1VEC        EQUALS  MPAC

```

```

# DEBRIS --
# RTNLAMB      ERASE    +0
# U2           ERASE    +5
# MAGVEC2      ERASE    +1
# UR1          ERASE    +5
# R1           EQUALS   31D
# UN           ERASE    +5
# SNTH         ERASE    +1
# CSTH         ERASE    +1
# 1-CSTH       ERASE    +1
# CSTH-RHO     ERASE    +1

COGAMAX        EQUALS   14D    # CLOBBERS 1/MU
COGAMIN        EQUALS    8D
DCOGA          EQUALS   12D

# TWEKIT       EQUALS   40D
# P            ERASE    +1
# Page 1307
# COGA         ERASE    +1
# R1A          ERASE    +1
# X            EQUALS   20D
# XSQ          EQUALS   22D
# XI           EQUALS   24D
# S(XI)        EQUALS   26D
# XSQC(XI)     EQUALS   28D
# T            EQUALS   30D
# KEPC1        EQUALS   34D
# KEPC2        EQUALS   36D
# SLOPSW
# SOLNSW

# OTHERS --
# RVEC         EQUALS   R1VEC
# VVEC         ERASE    +5
# COGAFLAG
# RVSW
# INFINFLG
# APSESW
# 360SW
# RTNTT        EQUALS   RTNLAMB
# ECC          ERASE    +1
# RTNTR        EQUALS   RTNLAMB
# RTNAPSE      EQUALS   RTNLAMB
# R2           EQUALS   MAGVEC2

```

```

COSF          EQUALS  24D

# RTNPRM      ERASE   +0
# SCNRDOT     ERASE   +0
# RDESIRED    ERASE   +1

# ITERATOR SUBROUTINE

# ORDERSW
MAX           EQUALS  14D          # CLOBBERS 1/MU
MIN           EQUALS   8D

# INDEP       ERASE   +1

DELINDEP      EQUALS  12D
ITERCTR       EQUALS  22D
DEP           EQUALS  30D

# DELDEP      ERASE   +1
# DEPREV      ERASE   +1

TWEKIT        EQUALS  40D

# MORE KEPLER

# EPSILONT    ERASE   +1

# Page 1308
# MORE LAMBERT

# TERRLAMB    EQUALS  DELDEP
# TPREV       EQUALS  DEPREV

# EPSILONL    EQUALS  EPSILONT +2  # DOUBLE PRECISION WORD

```

This code is written to file `src/CONIC-SUBROUTINES.s`.

B.18 CONTRACT AND APPROVALS

```

311  <src/CONTRACT-AND-APPROVALS.s 311>≡
    # Copyright:    Public domain.
    # Filename:     CONTRACT_AND_APPROVALS.agc
    # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
    #               It is part of the source code for the Command Module's (CM)
    #               Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:    yaYUL
    # Contact:       Ron Burkey <info@sandroid.org>.
    # Website:       www.ibiblio.org/apollo.
    # Mod history:   2009-05-06 RSB  Transcribed from page images.
    #
    # This source code has been transcribed or otherwise adapted from digitized
    # images of a hardcopy from the MIT Museum.  The digitization was performed
    # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum.  Many
    # thanks to both.  The images (with suitable reduction in storage size and
    # consequent reduction in image quality as well) are available online at
    # www.ibiblio.org/apollo.  If for some reason you find that the images are
    # illegible, contact me at info@sandroid.org about getting access to the
    # (much) higher-quality images which Paul actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    #       Assemble revision 055 of AGC program Comanche by NASA
    #       2021113-051.  10:28 APR. 1, 1969
    #
    # Page 1

```

<Page 1 16>

This code is written to file `src/CONTRACT-AND-APPROVALS.s`.

B.19 CONTROLLED CONSTANTS

```

312  <src/CONTROLLED-CONSTANTS.s 312>≡
      # Copyright:    Public domain.
      # Filename:     CONTROLLED_CONSTANTS.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      #
      # Assembler:    yaYUL
      # Contact:       Jim Lawton <jim.lawton@gmail.com>
      # Website:       www.ibiblio.org/apollo.
      # Pages:         038-053
      # Mod history:   2009-05-16      JVL      Transcribed from page images.
      #
      # This source code has been transcribed or otherwise adapted from digitized
      # images of a hardcopy from the MIT Museum. The digitization was performed
      # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
      # thanks to both. The images (with suitable reduction in storage size and
      # consequent reduction in image quality as well) are available online at
      # www.ibiblio.org/apollo. If for some reason you find that the images are
      # illegible, contact me at info@sandroid.org about getting access to the
      # (much) higher-quality images which Paul actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #   Assemble revision 001 of AGC program LMY99 by NASA 2021112-061
      #   16:27 JULY 14, 1969

      # Page 38
      # DPS AND APS ENGINE PARAMETERS

                                SETLOC  P40S
                                BANK
                                COUNT*  $$/P40

      # *** THE ORDER OF THE FOLLOWING SIX CONSTANTS MUST NOT BE CHANGED ***

      FDPS          2DEC      4.3670 B-7          # 9817.5 LBS FORCE IN NEWTONS
      MDOTDPS       2DEC      0.1480 B-3          # 32.62 LBS/SEC IN KGS/CS
      DTDECAY       2DEC      -38
      FAPS          2DEC      1.5569 B-7          # 3500 LBS FORCE IN NEWTONS
      MDOTAPS       2DEC      0.05135 B-3         # 11.32 LBS/SEC IN KGS/CS
      ATDECAY       2DEC      -10

      # *****

```


July 12, 2016

313

FRCS4	2DEC	0.17792 B-7	# 400 LBS FORCE IN NEWTONS
FRCS2	2DEC	0.08896 B-7	# 200 LBS FORCE IN NEWTONS

SETLOC P40S1
BANK
COUNT* \$\$/P40

*** APS IMPULSE DATA FOR P42 *****

K1VAL	2DEC	124.55 B-23	# 2800 LB-SEC
K2VAL	2DEC	31.138 B-24	# 700 LB-SEC
K3VAL	2DEC	1.5569 B-10	# FAPS (3500 LBS THRUST)

S40.136	2DEC	.4671 B-9	# .4671 M NEWTONS (DPS)
S40.136_	2DEC	.4671 B+1	# S40.136 SHIFTED LEFT 10.

SETLOC ASENT1
BANK
COUNT* \$\$/P70

(1/DV)A	2DEC	15.20 B-7	# 2 SECONDS WORTH OF INITIAL ASCENT
---------	------	-----------	-------------------------------------

Page 39

STAGE ACCELERATION -- INVERTED (M/CS)
1) PREDICATED ON A LIFTOFF MASS OF
4869.9 KG (SNA-8-D-027 7/11/68)
2) PREDICATED ON A CONTRIBUTION TO VEH-
ICLE ACCELERATION FROM RCS THRUSTERS
EQUIV. TO 1 JET ON CONTINUOUSLY.

K(1/DV)	2DEC	436.70 B-9	# DPS ENGINE THRUST IN NEWTONS / 100 CS.
---------	------	------------	--

(AT)A	2DEC	3.2883 E-4 B9	# INITIAL ASC. STG. ACCELERATION ** M/CS.
(TBUP)A	2DEC	91902 B-17	# ASSUMPTIONS SAME AS FOR (1/DV)A.

ESTIMATED BURN-UP TIME OF THE ASCENT STG.
ASSUMPTIONS SAME AS FOR (1/DV)A WITH THE
ADDITIONAL ASSUMPTION THAT NET MASS-FLOW
RATE = 5.299 KG/SEC = 5.135 (APS) +
.164 (1 RCS JET).

SETLOC ASENT
BANK
COUNT* \$\$/ASENT

AT/RCS	2DEC	.0000785 B+10	# 4 JETS IN A DRY LEM
--------	------	---------------	-----------------------

SETLOC SERVICES
BANK
COUNT* \$\$/SERV

*** THE ORDER OF THE FOLLOWING TWO CONSTANTS MUST NOT BE CHANGED *****

APSVEX DEC -3030 E-2 B-5 # 9942 FT/SEC IN M/CS.
DPSVEX DEC* -2.95588868 E+1 B-05* # VE (DPS) +2.95588868E+ 3

SETLOC F2DPS*31
BANK
COUNT* \$\$/F2DPS

TRIMACCL 2DEC* +3.50132708 E-5 B+08* # A (T) +3.50132708E- 1

Page 40

THROTTLING AND THRUST DETECTION PARAMETERS

SETLOC P40S
BANK
COUNT* \$\$/P40

THRESH1 DEC 24
THRESH3 DEC 12
HIRTHROT = BIT13

SETLOC FFTAG5
BANK
COUNT* \$\$/P40

THRESH2 DEC 308

SETLOC FTHROT
BANK
COUNT* \$\$/THROT

FMAXODD	DEC	+3841	# FSAT	+4.81454413 E+4
FMAXPOS	DEC	+3467	# FMAX	+4.34546769 E+4
THROTLAG	DEC	+20	# TAU (TH)	+1.99999999 E-1
SCALEFAC	2DEC*	+7.97959872 E+2 B-16*	# BITPERF	+7.97959872 E-2

SETLOC F2DPS*32
BANK
COUNT* \$\$/F2DPS

July 12, 2016

315

DPSTHRSH DEC 36 # (THRESH1 + THRESH3 FOR P63)

Page 41

LM HARDWARE-RELATED PARAMETERS

SETLOC RADARUPT
BANK
COUNT* \$\$/RRUPT

LVELBIAS DEC -12288 # LANDING RADAR BIAS FOR 153.6 KC.
RDOTBIAS 2DEC 17000 # BIAS COUNT FOR RR RANGE RATE.

SETLOC LRS22
BANK
COUNT* \$\$/LRS22

RDOTCONV 2DEC -.0019135344 B7 # CONVERTS RR RDOT READING TO M/CS AT 2(7)
RANGCONV 2DEC 2.859024 B-3 # CONVERTS RR RANGE READING TO M. AT 2(-29)

SETLOC SERVICES
BANK
COUNT* \$\$/SERV

HBEAMANT 2DEC -.4687018041 # RANGE BEAM IN LR ANTENNA COORDINATES.
 2DEC 0
 2DEC -.1741224271

HSCAL 2DEC -.3288792 # SCALES 1.079 FT/BIT TO 2(22)M.

***** THE SEQUENCE OF THE FOLLOWING CONSTANTS MUST BE PRESERVED *****

VZSCAL 2DEC +.5410829105 # SCALES .8668 FT/SEC/BIT TO 2(18) M/CS.
VYSCAL 2DEC +.7565672446 # SCALES 1.212 FT/SEC/BIT TO 2(18) M/CS.
VXSCAL 2DEC -.4020043770 # SCALES -.644 FT/SEC/BIT TO 2(18) M/CS.

KPIP DEC .0512 # SCALES DELV TO UNITS OF 2(5) M/CS.
KPIP1 2DEC .0128 # SCALES DELV TO UNITS OF 2(7) M/CS.
KPIP2 2DEC .0064 # SCALES DELV TO UNITS OF 2(8) M/CS.

Page 42

ALTCONV 2DEC 1.399078846 B-4 # CONVERTS M*2(-24) TO BIT UNITS *2(-28).
ARCONV1 2DEC 656.167979 B-10 # CONV. ALTRATE COMP. TO BIT UNITS<

	SETLOC	R10	
	BANK		
	COUNT*	\$\$/R10	
ARCONV	OCT	24402	# 656.1679798B-10 CONV ALTRATE TO BIT
ARTOA	DEC	.1066098 B-1	# .25/2.345 B-1 4X/SEC CYCLE RATE.
ARTOA2	DEC	.0021322 B8	# (.5)/(2.345)(100)
VELCONV	OCT	22316	# 588.914 B-10 CONV VEL. TO BIT UNITS
KPIP1(5)	DEC	.0512	# SCALES DELV TO M/CS*2(-5).
MAXVBITS	OCT	00547	# MAX. DISPLAYED VELOCITY 199.9989 F

SETLOC DAPS3
BANK
COUNT* \$\$/DAPAO

TORKJET1	DEC	.03757	# 550 / .2 SCALED AT (+16) 64 / 180
----------	-----	--------	-------------------------------------

Page 43

PARAMETERS RELATING TO MASS, INERTIA, AND VEHICLE DIMENSIONS

	SETLOC	FRANDRES	
	BANK		
	COUNT*	\$\$/START	
FULLAPS	DEC	5050 B-16	# NOMINAL FULL ASCENT MASS -- 2(16) K
	SETLOC	LOADDAP1	
	BANK		
	COUNT*	\$\$/R03	
MINLMD	DEC	-2850 B-16	# MIN. DESCENT STAGE MASS -- 2(16) K
MINMINLM	DEC	-2200 B-16	# MIN ASCENT STAGE MASS -- 2(16) KG.
MINCSM	=	BIT11	# MIN CSM MASS (OK FOR 1/ACCS) = 9050

SETLOC DAPS3
BANK
COUNT* \$\$/DAPAD

LOASCENT	DEC	2200 B-16	# MIN ASCENT LEM MASS -- 2(16) KG.
HIDESCNT	DEC	15300 B-16	# MAX DESCENT LEM MASS -- 2(16) KG.
LODESCNT	DEC	1750 B-16	# MIN DESCENT STAGE (ALONE) -- 2(16)

Page 44

PHYSICAL CONSTANTS (TIME - INVARIANT)

SETLOC IMU2

July 12, 2016

317

```
BANK
COUNT*  $$/P07

OMEG/MS      2DEC      .24339048

SETLOC  R30LOC
BANK
COUNT*  $$/R30

# *** THE ORDER OF THE FOLLOWING TWO CONSTANTS MUST BE PRESERVED *****

1/RTMUM      2DEC*     .45162595 E-4 B14*
1/RTMUE      2DEC*     .50087529 E-5 B17*

# *****

SETLOC  P40S1
BANK
COUNT*  $$/S40.9

EARTHMU      2DEC*     -3.986032 E10 B-36*      # M(3)/CS(2)

SETLOC  ASENT1
BANK
COUNT*  $$/P12

MUM(-37)     2DEC*     4.9027780 E8 B-37*
MOONRATE     2DEC*     .26616994890062991 E-7 B+19*      # RAD/CS.

SETLOC  SERVICES
BANK
COUNT*  $$/SERV

# *** THE ORDER OF THE FOLLOWING TWO CONSTANTS MUST BE PRESERVED *****

-MUDT        2DEC*     -7.9720645 E+12 B-44*
-MUDT1       2DEC*     -9.8055560 E+10 B-44*

# *****

-MUDTMUN     2DEC*     -9.8055560 E+10 B-38*
RESQ         2DEC*     40.6809913 E12 B-58*

# Page 45
20J          2DEC      3.24692010 E-2
2J           2DEC      3.24692010 E-3
```

```

SETLOC  P50S1
BANK
COUNT*  $$/LOSAM

RSUBEM   2DEC  384402000 B-29
RSUBM    2DEC  1738090 B-29
RSUBE    2DEC  6378166 B-29
ROE      2DEC  .00257125

SETLOC  CONICS1
BANK
COUNT*  $$/LT-LG

ERAD      2DEC  6373338 B-29      # PAD RADIUS
504RM     2DEC  1738090 B-29      # METERS B-29 (EQUATORIAL MOON RADIUS)

SETLOC  CONICS1
BANK
COUNT*  $$/CONIC

# *** THE ORDER OF THE FOLLOWING CONSTANTS MUST BE PRESERVED *****

MUTABLE   2DEC*  3.986032 E10 B-36*      # MUE
          2DEC*  .25087606 E-10 B+34*    # 1/MUE
          2DEC*  1.99650495 E5 B-18*     # SQRT(MUE)
          2DEC*  .50087529 E-5 B+17*     # 1/SQRT(MUE)
          2DEC*  4.902778 E8 B-30*       # MUM
          2DEC*  .203966 E-8 B+28*       # 1/MUM
          2DEC*  2.21422176 E4 B-15*     # SQRT(MUM)
          2DEC*  .45162595 E-4 B+14*     # 1/SQRT(MUM)

# *****

# Page 46

SETLOC  INTINIT
BANK
COUNT*  $$/INTIN

OMEGMOON  2DEC*  2.66169947 E-8 B+23*

SETLOC  ORBITAL2
BANK
COUNT*  $$/ORBIT

# *** THE ORDER OF THE FOLLOWING CONSTANTS MUST NOT BE CHANGED *****

```

	2DEC*	1.32715445 E16 B-54*
MUM	2DEC*	4.9027780 E8 B-30*
MUEARTH	2DEC*	3.986032 E10 B-36*
	2DEC	0
J4REQ/J3	2DEC*	.4991607391 E7 B-26*
	2DEC	-176236.02 B-25
2J3RE/J2	2DEC*	-.1355426363 E5 B-27*
	2DEC*	.3067493316 E18 B-60*
J2REQSQ	2DEC*	1.75501139 E21 B-72*
3J22R2MU	2DEC*	9.20479048 E16 B-58*

SETLOC TOF-FF1
BANK
COUNT* \$\$/TFF

1/RTMU 2DEC* .5005750271 E-5 B17* # MODIFIED EARTH MU

SETLOC SBAND
BANK
COUNT* \$\$/R05

REMDIST 2DEC 384402000 B-29 # MEAN DISTANCE BETWEEN EARTH AND MOON.

Page 47

PHYSICAL CONSTANTS (TIME - VARIANT)

SETLOC STARTAB
BANK
COUNT* \$\$/STARS

2DEC	+.8342971408 B-1	# STAR 37	X
2DEC	-.2392481515 B-1	# STAR 37	Y
2DEC	-.4966976975 B-1	# STAR 37	Z
2DEC	+.8139832631 B-1	# STAR 36	X
2DEC	-.5557243189 B-1	# STAR 36	Y
2DEC	+.1691204557 B-1	# STAR 36	Z
2DEC	+.4541086270 B-1	# STAR 35	X
2DEC	-.5392368197 B-1	# STAR 35	Y
2DEC	+.7092312789 B-1	# STAR 35	Z
2DEC	+.3201817378 B-1	# STAR 34	X

Page 48

2DEC	-.4436021946	B-1	# STAR 34	Y
2DEC	-.8370786986	B-1	# STAR 34	Z
2DEC	+.5520184464	B-1	# STAR 33	X
2DEC	-.7933187400	B-1	# STAR 33	Y
2DEC	-.2567508745	B-1	# STAR 33	Z
2DEC	+.4537196908	B-1	# STAR 32	X
2DEC	-.8779508801	B-1	# STAR 32	Y
2DEC	+.1527766153	B-1	# STAR 32	Z
2DEC	+.2069525789	B-1	# STAR 31	X
2DEC	-.8719885748	B-1	# STAR 31	Y
2DEC	-.4436288486	B-1	# STAR 31	Z
2DEC	+.1217293692	B-1	# STAR 30	X
2DEC	-.7702732847	B-1	# STAR 30	Y
2DEC	+.6259880410	B-1	# STAR 30	Z
2DEC	-.1124304773	B-1	# STAR 29	X
2DEC	-.9694934200	B-1	# STAR 29	Y
2DEC	+.2178116072	B-1	# STAR 29	Z
2DEC	-.1146237858	B-1	# STAR 28	X
2DEC	-.3399692557	B-1	# STAR 28	Y
2DEC	-.9334250333	B-1	# STAR 28	Z
2DEC	-.3516499609	B-1	# STAR 27	X
2DEC	-.8240752703	B-1	# STAR 27	Y
2DEC	-.4441196390	B-1	# STAR 27	Z
2DEC	-.5326876930	B-1	# STAR 26	X
2DEC	-.7160644554	B-1	# STAR 26	Y
2DEC	+.4511047742	B-1	# STAR 26	Z
2DEC	-.7861763936	B-1	# STAR 25	X
2DEC	-.5217996305	B-1	# STAR 25	Y
2DEC	+.3311371675	B-1	# STAR 25	Z
2DEC	-.6898393233	B-1	# STAR 24	X
2DEC	-.4182330640	B-1	# STAR 24	Y
2DEC	-.5909338474	B-1	# STAR 24	Z
2DEC	-.5812035376	B-1	# STAR 23	X
2DEC	-.2909171294	B-1	# STAR 23	Y

July 12, 2016

321

2DEC	+.7599800468 B-1	# STAR 23	Z
2DEC	-.9170097662 B-1	# STAR 22	X
2DEC	-.3502146628 B-1	# STAR 22	Y
2DEC	-.1908999176 B-1	# STAR 22	Z

Page 49

2DEC	-.4523440203 B-1	# STAR 21	X
2DEC	-.0493710140 B-1	# STAR 21	Y
2DEC	-.8904759346 B-1	# STAR 21	Z
2DEC	-.9525211695 B-1	# STAR 20	X
2DEC	-.0593434796 B-1	# STAR 20	Y
2DEC	-.2986331746 B-1	# STAR 20	Z
2DEC	-.9656605484 B-1	# STAR 19	X
2DEC	+.0525933156 B-1	# STAR 19	Y
2DEC	+.2544280809 B-1	# STAR 19	Z
2DEC	-.8608205219 B-1	# STAR 18	X
2DEC	+.4636213989 B-1	# STAR 18	Y
2DEC	+.2098647835 B-1	# STAR 18	Z
2DEC	-.7742591356 B-1	# STAR 17	X
2DEC	+.6152504197 B-1	# STAR 17	Y
2DEC	-.1482892839 B-1	# STAR 17	Z
2DEC	-.4657947941 B-1	# STAR 16	X
2DEC	+.4774785033 B-1	# STAR 16	Y
2DEC	+.7450164351 B-1	# STAR 16	Z
2DEC	-.3612508532 B-1	# STAR 15	X
2DEC	+.5747270840 B-1	# STAR 15	Y
2DEC	-.7342932655 B-1	# STAR 15	Z
2DEC	-.4118589524 B-1	# STAR 14	X
2DEC	+.9065485360 B-1	# STAR 14	Y
2DEC	+.0924226975 B-1	# STAR 14	Z

Page 50

2DEC	-.1820751783 B-1	# STAR 13	X
2DEC	+.9404899869 B-1	# STAR 13	Y
2DEC	-.2869271926 B-1	# STAR 13	Z
2DEC	-.0614937230 B-1	# STAR 12	X
2DEC	+.6031563286 B-1	# STAR 12	Y

2DEC	-.7952489957 B-1	# STAR 12	Z
2DEC	+.1371725575 B-1	# STAR 11	X
2DEC	+.6813721061 B-1	# STAR 11	Y
2DEC	+.7189685267 B-1	# STAR 11	Z
2DEC	+.2011399589 B-1	# STAR 10	X
2DEC	+.9690337941 B-1	# STAR 10	Y
2DEC	-.1432348512 B-1	# STAR 10	Z
2DEC	+.3507315038 B-1	# STAR 9	X
2DEC	+.8926333307 B-1	# STAR 9	Y
2DEC	+.2831839492 B-1	# STAR 9	Z
2DEC	+.4105636020 B-1	# STAR 8	X
2DEC	+.4988110001 B-1	# STAR 8	Y
2DEC	+.7632988371 B-1	# STAR 8	Z
2DEC	+.7032235469 B-1	# STAR 7	X
2DEC	+.7075846047 B-1	# STAR 7	Y
2DEC	+.0692868685 B-1	# STAR 7	Z
2DEC	+.5450107404 B-1	# STAR 6	X
2DEC	+.5314955466 B-1	# STAR 6	Y
2DEC	-.6484410356 B-1	# STAR 6	Z
2DEC	+.0130968840 B-1	# STAR 5	X
2DEC	+.0078062795 B-1	# STAR 5	Y
2DEC	+.9998837600 B-1	# STAR 5	Z
2DEC	+.4917678276 B-1	# STAR 4	X
2DEC	+.2204887125 B-1	# STAR 4	Y
2DEC	-.8423473935 B-1	# STAR 4	Z
2DEC	+.4775639450 B-1	# STAR 3	X
2DEC	+.1166004340 B-1	# STAR 3	Y
2DEC	+.8708254803 B-1	# STAR 3	Z
2DEC	+.9342640400 B-1	# STAR 2	X
2DEC	+.1735073142 B-1	# STAR 2	Y
2DEC	-.3115219339 B-1	# STAR 2	Z
2DEC	+.8748658918 B-1	# STAR 1	X
2DEC	+.0260879174 B-1	# STAR 1	Y
2DEC	+.4836621670 B-1	# STAR 1	Z

CATALOG DEC 6970

SETLOC EPHEM1
BANK
COUNT* \$\$/EPHEM

KONMAT	2DEC	1.0 B-1	# *****
	2DEC	0	# *
	2DEC	0	# *
	2DEC	0	# *
	2DEC	.91745 B-1	# K1 COS(OBL) *
	2DEC	-.03571 B-1	# K2 SIN(OBL)SIN(IM) *
	2DEC	0	# *
	2DEC	.39784 B-1	# K3 SIN(OBL) *

Page 52

	2DEC	.082354 B-1	# K4 COS(OBL)SIN(IM) *
CSTODAY	2DEC	8640000 B-33	# * NOTE: *
RCB-13	OCT	00002	# * TABLES CONTAIN *
	OCT	00000	# * CONSTANTS FOR *
RATESP	2DEC	.03660098 B+4	# LOMR * 1969 - 1970 *
	2DEC	.00273779 B+4	# LOSR
	2DEC	-.00014719 B+4	# LONR
	2DEC	.815282336	# LOMO
	2DEC	.274674910	# LOSO
	2DEC	.986209499	# LONO
VAL67	2DEC*	.01726666666 B+1*	# AMOD
	2DEC	.530784445	# AARG
	2DEC	.036291712 B+1	# 1/27
	2DEC	.003505277 B+1	# BMOD
	2DEC	.585365625	# BARG
	2DEC	.03125 B+1	# 1/32
	2DEC	.005325277 B+1	# CMOD
	2DEC	-.01106341036	# CARG
	2DEC	.002737925 B+1	# 1/365

SETLOC PLANTIN2
BANK
COUNT* \$\$/LUROT

COSI	2DEC	.99964173 B-1	# COS (5521.5 SEC.) B-1
SINI	2DEC	.02676579 B-1	# SIN (5521.5 SEC.) B-1

```

NODDOT      2DEC      -.457335121 E-2
FDDOT       2DEC      .570863327
# Page 53
BDDOT       2DEC      -3.07500686 E-8
NODIO       2DEC      .986209434
FSUB0       2DEC      .829090536
BSUB0       2DEC      .0651201393
WEARTH      2DEC      .973561595

```

```

# REV/CSEC B+28 = -1.07047011 E-8 RAD
# REV/CSEC B+27 =  2.67240410 E-6 RAD
# REV/CSEC B+28 = -7.19757301 E-14 RAD
# REVS B-D      =  6.19653663041 RAD
# REVS B-D      =  5.20932947829 RAD
# REVS B-D      =  0.40916190299 RAD
# REV/CSEC B+23 =  7.29211494 E-5 RAD

```

This code is written to file `src/CONTROLLED-CONSTANTS.s`.

B.20 CSM GEOMETRY

325

<src/CSM-GEOMETRY.s 325>≡

```
# Copyright:      Public domain.
# Filename:       CSM_GEOMETRY.agc
# Purpose:       Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Ron Burkey <info@sandroid.org>.
# Website:       www.ibiblio.org/apollo.
# Pages:        285-296
# Mod history:   2009-05-08 RSB   Adapted from the Colossus249/ file of the
#                               same name, using Comanche055 page images.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#       Colossus 2A
#
# Page 285
#
#       BANK      22
#       SETLOC    COMGEOM1
#       BANK
#
# THIS ROUTINE TAKES THE SHAFT AND TRUNNION ANGLES AS READ BY THE CM OPTICAL SYSTEM AND CONVERT
# VECTOR REFERENCED TO THE NAVIGATION BASE COORDINATE SYSTEM AND COINCIDENT WITH THE SEXTANT LI
#
# THE INPUTS ARE:  1) THE SEXTAND SHAFT AND TRUNNION ANGLES ARE STORED SP IN LOCATIONS 3 AND 5
# MARK VAC AREA.  2) THE COMPLEMENT OF THE BASE ADDRESS OF THE MARK VAC AREA IS STORED SP AT LO
# JOB VAC AREA.
#
# THE OUTPUT IS A HALF-UNIT VECTOR IN NAVIGATION BASE COORDINATES AND STORED AT LOCATION 32D OF
# OUTPUT IS ALSO AVAILABLE AT MPAC.
```

```

COUNT    23/GEOM

SXTNB      SLOAD*  RTB          # PUSHDOWN 00,02,04,(17D-19D),32D-36D
              5,1          # TRUNNION = TA
              CDULOGIC
              RTB          PUSH
              SXTLOGIC
              SIN          SL1
              PUSH        SLOAD*      # PD2 = SIN(TA)
              3,1          # SHAFT = SA
              RTB          PUSH        # PD4 = SA
              CDULOGIC

              COS          DMP
              2
              STODL        STARM        # COS(SA)SIN(TA)

              SIN          DMP
              STADR
              STODL        STARM    +2    # SIN(SA)SIN(TA)

              COS
              STOVL        STARM    +4
              STARM        # STARM = 32D
              MXV          VSL1
              NB1NB2
              STORE        32D
              RVQ

SXTLOGIC    CAF          10DEGS-      # CORRECT FOR 19.775 DEGREE OFFSET
              ADS          MPAC
              CAF          QUARTER
              TC          SHORTMP
              TC          DANZIG

```

Page 286

CALCSXA COMPUTES THE SEXTANT SHAFT AND TRUNNION ANGLES REQUIRED TO POSITION THE OPTIC
 # OF-SIGHT LIES ALONG THE STAR VECTOR. THE ROUTINE TAKES THE GIVEN STAR VECTOR AND POSITION
 # RECORDED TO THE OPTICS COORDINATE SYSTEM. IN ADDITION IT SETS UP THREE UNIT VECTORS
 # REFERENCED TO THE OPTICS COORDINATE SYSTEM.

#

THE INPUTS ARE: 1) THE STAR VECTOR REFERRED TO THE PRESENT STABLE MEMBER COORDINATE
 # INPUT AS *SMNB*, I.E., SINES AND COSINES OF THE CDU ANGLES, IN THE ORDER Y Z X, AT
 # TO CDUTRIG WILL PROVIDE THIS INPUT.

#

THE OUTPUT ARE THE SEXTANT SHAFT AND TRUNNION ANGLES STORED DP AT SAC AND PAC RESPECTIVELY.
EQUAL TO ZERO).

CALCSXA	ITA	VLOAD	# PUSHDOWN 00-26D, 28D, 30D, 32D-36D
		28D	
		STAR	
	CALL		
		SMNB	
	MXV	VSL1	
		NB2NB1	
	STOVL	STAR	
		HIUNITX	
	STOVL	XNB1	
		HIUNITY	
	STOVL	YNB1	
		HIUNITZ	
	STCALL	ZNB1	
		SXTANG1	

Page 287

SXTANG COMPUTES THE SEXTANT SHAFT AND TRUNNION ANGLES REQUIRED TO POSITION THE OPTICS SUCH TH
SIGHT LIES ALONG THE STAR VECTOR.

#

THE INPUTS ARE: 1) THE STAR VECTOR REFERRED TO ANY COORDINATE SYSTEM STORED AT STAR. 2) THE
COORDINATES REFERRED TO THE SAME COORDINATE SYSTEM. THESE THREE HALF-UNIT VECTORS ARE STROED
ZNB.

#

THE OUTPUTS ARE THE SEXTANT SHAFT AND TRUNNION ANGLES STORED DP AT SAC AND PAC RESPECTIVELY.
EQUAL TO ZERO).

SXTANG	ITA	RTB	# PUSHDOWN 16D,18D,22D-26D,28D
		28D	
		TRANSP1	# EREF WRT NB2
	VLOAD	MXV	
		XNB	
		NB2NB1	
	VSL1		
	STOVL	XNB1	
		YNB	
	MXV	VSL1	
		NB2NB1	
	STOVL	YNB1	
		ZNB	
	MXV	VSL1	
		NB2NB1	
	STORE	ZNB1	

```

RTB      RTB
          TRANSP1
          TRANSP2

SXTANG1  VLOAD  VXV
          ZNB1
          STAR
          BOV
          +1
          UNIT  BOV
          ZNB=S1
          STORE PDA          # PDA = UNIT(ZNB X S)

          DOT      DCOMP
          XNB1
          STOVL    SINTH      # SIN(SA) = PDA . -XNB
          PDA

          DOT
          YNB1
          STCALL   COSTH      # COS(SA) = PDA . YNB
          ARCTRIG

# Page 288

RTB
          1STO2S
          STOVL    SAC
          STAR
          BOV
          +1
          DOT      SL1
          ZNB1
          ACOS
          BMN      SL2
          SXTALARM      # TRUNNION ANGLE NEGATIVE
          BOV      DSU
          SXTALARM      # TRUNNION ANGLE GREATER THAN 90 DEGREES
          20DEG-

          RTB
          1STO2S
          STORE    PAC        # FOR FLIGHT USE, CULTFLAG IS ON IF
          CLRG0      # TRUNION IS GREATER THAN 90 DEG
          CULTFLAG
          28D
          SXTALARM  SETGO      # ALARM HAS BEEN REMOVED FROM THIS
          CULTFLAG

```



```

                28D                # SUBROUTINE, ALARM WILL BE SET BY MPI
ZNB=S1          DLOAD
                270DEG
                STODL          SAC
                20DEGS-
                STORE          PAC
                CLRG0
                CULTFLAG
                28D

```

Page 289

THESE TWO ROUTINES COMPUTE THE ACTUAL STATE VECTOR FOR LM, CSM BY ADDING
THE CONIC R,V AND THE DEVIATIONS R,V. THE STATE VECTORS ARE CONVERTED TO
METERS B-29 AND METERS/CSEC B-7 AND STORED APPROPRIATELY IN RN,VN OR
R-OTHER, V-OTHER FOR DOWNLINK. THE ROUTINES NAMES ARE SWITCHED IN THE
OTHER VEHICLES COMPUTER.

#

INPUT

STATE VECTOR IN TEMPORARY STORAGE AREA
IF STATE VECTOR IS SCALED POS B27 AND VEL B5
SET X2 TO +2
IF STATE VECTOR IS SCALED POS B29 AND VEL B7
SET X2 TO 0

#

OUTPUT

R(T) IN RN, V(T) IN VN, T IN PIPTIME
OR
R(T) IN R-OTHER, V(T) IN V-OTHER (T IS DEFINED BY T-OTHER)

```

                BANK    23
                SETLOC  COMGEOM2
                BANK
                COUNT   10/GEOM
SVDWN1         BOF     RVQ                # SW=1=AVETOMID DOING W-MATRIX INTEG
                AVEMIDSW
                +1
                VLOAD   VSL*
                TDELTA V
                0        -7,2
                VAD     VSL*
                RCV
                0,2
                STOVL   RN
                TNUV
                VSL*    VAD
                0        -4,2

```

```

                                VCV
                                VSL*
                                0,2
                                STODL VN
                                TET
                                STORE PIPTIME
                                RVQ
SVDWN2 VLOAD VSL*
                                TDELTA
                                0 -7,2
                                VAD VSL*
                                RCV

```

Page 290

```

                                0,2
                                STOVL R-OTHER
                                TNUV
                                VSL* VAD
                                0 -4,2
                                VCV
                                VSL*
                                0,2
                                STORE V-OTHER
                                RVQ

```

Page 291

SUBROUTINE TO COMPUTE THE NATURAL LOG OF C(MPAC, MPAC +1).

#

ENTRY: CALL

LOG

#

SUBROUTINE RETURNS WITH -LOG IN DP MPAC.

#

EBANK IS ARBITRARY.

```

BANK 14
SETLOC POWFLIT2
BANK
COUNT 23/GEOM

```

```

LOG NORM BDSU # GENERATES LOG BY SHIFTING ARG
      MPAC +3 # UNTIL IT LIES BETWEEN .5 AND 1.
      NEARLY1 # THE LOG OF THIS PART IS FOUND AND THE
      EXIT # LOG OF THE SHIFTED PART IS COMPUTED
      TC POLY # AND ADDED IN. SHIFT COUNT STORED
      DEC 2 # (N-1, SUPPLIED BY SMERZH)

```

July 12, 2016

331

```
2DEC      0                # IN MPAC +3.
2DEC      .031335467
2DEC      .0130145859
2DEC      .0215738898

CAF      ZERO
TS       MPAC      +2
EXTEND
DCA      CLOG2/32
DXCH     MPAC
DXCH     MPAC      +3
COM
TC       SHORTMP          # LOAD POSITIVE SHIFT COUNT IN A.
                          # MULTIPLY BY SHIFT COUNT.

DXCH     MPAC      +1
DXCH     MPAC
DXCH     MPAC      +3
DAS      MPAC
TC       INTERPRET        # RESULT IN MPAC, MPAC +1

RVQ
```

```
NEARLY1    2DEC      .999999999
```

Page 292

```
CLOG2/32    2DEC      .0216608494
```

Page 293

SUBROUTINE NAME: EARTH ROTATOR (EARROT1 OR EARROT2)

DATE: 15 FEB 67

MOD NO: N +1

LOG SECTION: POWERED P

MOD BY: ENTRY GROUP (BAIRNSFATHER)

#

FUNCTIONAL DESCRIPTION: THIS ROUTINE PROJECTS THE INITIAL EARTH TARGET VECTOR RTINIT AHEAD T
THE ESTIMATED TIME OF FLIGHT. INITIAL CALL RESOLVES THE INITIAL TARGET VECTOR RTINIT I
AND NORMAL COMPONENTS RTEAST AND RTNORM. INITIAL AND SUBSEQUENT CALLS ROTATE THIS VECT
ABOUT THE (FULL) UNIT POLAR AXIS UNITW THROUGH THE ANGLE WIE DTEAROT TO OBTAIN THE RO
TARGET VECTOR RT. ALL VECTORS EXCEPT UNITW ARE HALF UNIT.
THE EQUATIONS ARE:

```
#      -      -      -
#      RT = RTINIT + RTNORM (COS(WT) - 1) + RTEAST SIN(WT)
```

```
#
# WHERE WT = WIE DTEAROT
```

```
#      RTINIT = INITIAL TARGET VECTOR
```

```
#      -      -      -
#      RTEAST = UNITW * RTINIT
```

```

#
#      -      -      -
#      RTNORM = RTEAST * UNITW
#
#      FOR CONTINUOUS UPDATING, ONLY ONE ENTRY TO EARROT1 IS REQUIRED, WITH SUBSEQUENT
#
# CALLING SEQUENCE:      FIRST CALL                      SUBSEQUENT CALL
#                       STCALL  DTEAROT                  STCALL  DTEAROT
#                       EARROT1                          EARROT2
#                       C(MPAC) UNSPECIFIED              C(MPAC) = DTEAROT
#      PUSHLOC = PDL+0, ARBITRARY.  6 LOCATIONS USED.
#
# SUBROUTINES USED:  NONE
#
# NORMAL EXIT MODES:  RVQ
#
# ALARMS:  NONE
#
# OUTPUT:      RTEAST  (-1)          .5 UNIT VECTOR EAST, COMPNT OF RTINIT  LEFT
#              RTNORM  (-1)          .5 UNIT VECTOR NORML, COMPNT OF RTINIT  LEFT
#              RT      (-1)          .5 UNIT TARGET VECTOR, ROTATED          LEFT
#              DTEAROT (-28) CS      MAY BE CHANGED BY EARROT2, IF OVER 1 DAY
#
# ERASABLE INITIALIZATION REQUIRED:
#              UNITW  (0)            UNIT POLAR VECTOR                      PAD 1
#              RTINIT (-1)          .5 UNIT INITIAL TARGET VECTOR          LEFT
#              DTEAROT (-28) CS      TIME OF FLIGHT                        LEFT
#
# DEBRIS:  QPRET, PDL+0 ... PDL+5
# Page 294
#
#      EBANK=  RTINIT
#
# EARROT1      VLOAD  VXV
#              UNITW          # FULL UNIT VECTOR
#              RTINIT        # .5 UNIT
#              STORE  RTEAST  # .5 UNIT
#
#              VXV
#              UNITW          # FULL UNIT
#              STODL  RTNORM  # .5 UNIT
#              DTEAROT      # (-28) CS
#
# EARROT2      BOVB  DDV
#              TCDANZIG      # RESET OVFLND, IF ON
#              1/WIE
#              BOV  PUSH
#              OVERADAY

```

July 12, 2016

333

	COS	DSU	
		HIDPHALF	
	VXSC	PDDL	# XCH W PUSH LIST
		RTNORM	# .5 UNIT
	SIN	VXSC	
		RTEAST	# .5 UNIT
	VAD	VSL1	
	VAD	UNIT	# INSURE THAT RT IS "UNIT".
		RTINIT	# .5 UNIT
	STORE	RT	# .5 UNIT TARGET VECTOR
	RVQ		
OVERADAY	DLOAD	SIGN	
		1/WIE	
		DTEAROT	
	BDSU		
		DTEAROT	
	STORE	DTEAROT	
	GOTO		
		EARROT2	
#WIE	2DEC	.1901487997	
1/WIE	2DEC	8616410	
NB2NB1	2DEC	+.8431756920 B-1	
	2DEC	0	
	2DEC	-.5376381241 B-1	
# Page 295			
ZERINFLT	2DEC	0	
HALFNFLT	2DEC	.5	
	2DEC	0	
	2DEC	+.5376381241 B-1	
	2DEC	0	
	2DEC	+.8431756920 B-1	
NB1NB2	2DEC	+.8431756920 B-1	
	2DEC	0	
	2DEC	+.5376381241 B-1	
	2DEC	0	
	2DEC	.5	
	2DEC	0	
	2DEC	-.5376381241 B-1	
	2DEC	0	
	2DEC	+.8431756920 B-1	
# Page 296			
10DEGS-	DEC	3600	

270DEG	OCT	60000	# SHAFT 270 DEGREES	25 COMP.
	OCT	00000		
20DEGS-	DEC	-07199		
	DEC	-00000		
20DEG-	DEC	03600		
	DEC	00000		

This code is written to file `src/CSM-GEOMETRY.s`.

B.21 DAPIDLER PROGRAM

335

<src/DAPIDLER-PROGRAM.s 335>≡

```
# Copyright:    Public domain.
# Filename:     DAPIDLER_PROGRAM.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:   yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1410-1420
# Mod history:  2009-05-10 SN    (Sergio Navarro).  Started adapting
#               from the Luminary131/ file of the same
#               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
```

Page 1410

```
# THE DAPIDLER PROGRAM IS STARTED BY FRESH START AND RESTART.  THE DAPIDLER PROGRAM IS DONE 10
# PER SECOND UNTIL THE ASTRONAUT DESIRES THE DAP TO WAKE UP, AND THE IMU AND CDUS ARE READY FOR
# THE NECESSARY INITIALIZATION OF THE DAP IS DONE BY THE DAPIDLER PROGRAM.
```

```
BANK      16
SETLOC    DAPS1
BANK

EBANK=    AOSQ

COUNT*   $$/DAPID
```

```
CHEKBITS      EXTEND
READ          CHAN31          # IF BOTH BIT13 AND BIT14 ARE ONE, THEN
```



```

                                BZF      MOREIDLE

STARTDAP      TC      IBNKCALL      # ZERO ATTITUDE ERROR AND DESIRED RATES.
              FCADR    ZATTEROR
              CAF      ZERO          # ***** INITIALIZE: *****
              TS      TJP
              TS      TJU
              TS      TJV
              TS      OMEGAP        # RATES IN BODY (PILOT) COORDINATES.
              TS      OMEGAQ
              TS      OMEGAR
              TS      TRAPEDP
              TS      TRAPEDQ
              TS      TRAPEDR
              TS      AOSQ          # OFFSET ACCELERATION ESTIMATES.
              TS      AOSQ +1
              TS      AOSR
              TS      AOSR +1
              TS      ALPHAQ        # COPIES OF OFFSET ESTIMATES FOR DOWNLIST.
              TS      ALPHAR
              TS      NEGUQ
              TS      NEGUR
              TS      AOSQTERM      # QRAXIS RATE DERIVATION TERMS AND KALMAN
              TS      AOSRTERM      # FILTER INITIALIZATION TERMS.
              TS      QACCDOT       # DESCENT ACCELERATION DERIVATIVE EST.
              TS      RACCDOT

# Page 1412
              TS      ALLOWGTS      # AOSTASK FLAG FOR QRAXIS RCS CONTROL USE.
              TS      COTROLER      # DO TRYGTS ON FIRST PASS (WILL GO TO RCS)
              TS      INGTS         # RECOGNIZE FIRST GTS PASS AS SUCH.
              TS      QGIMTIMR      # STOP GIMBAL DRIVES. (PROBABLY WOULD BE
              TS      RGIMTIMR      #      GOOD ENOUGH JUST TO INACTIVATE TIMERS)
              TS      OLDPMIN       # MINIMUM IMPULSE MODE ERASABLES
              TS      OLDQRMIN
              TS      PJETCTR        # INITIALIZE DOCKED JET INHIBITION
              TS      UJETCTR        # COUNTERS
              TS      VJETCTR

CALLGMBL      EQUALS    BIT5        # RCSFLAGS INITIALIZATION.
              CS      MANFLAG
              MASK     RCSFLAGS      # NEGUQ(R) HAVE BEEN GENERATED.
              TS      RCSFLAGS

# SET UP "OLD" MEASURED CDU ANGLES:

              EXTEND
              DCA      CDUX          # OLDXFORP AND OLDYFORP

```

	DXCH	OLDXFORP	
	CA	CDUZ	
	TS	OLDZFORQ	
	CS	RCSFLAGS	
	MASK	BIT12	
	ADS	RCSFLAGS	# BIT 12 SET TO 1.
	CA	FOUR	
	TS	SKIPU	
	TS	SKIPV	
	CA	POSMAX	
	TS	TIME6	
	TS	T6NEXT	
	TS	T6FURTHA	
	CA	ZERO	
	TS	T6NEXT +1	
	TS	T6FURTHA +1	
	TS	NXT6ADR	
	TS	NEXTP	
	TS	NEXTU	
	TS	NEXTV	
	CS	TEN	
	TS	DAPZRUP	# JASK NOT IN PROGRESS, INITIALIZE NEG.
	CA	TWO	
	TS	NPTRAPS	
	TS	NQTRAPS	
	TS	NRTRAPS	
	EXTEND		
	DCA	PAXADIDL	
SETTIME5	DXCH	T5ADR	
	CAF	MS100	
	TS	TIME5	
# Page 1413			
	TCF	RESUME	
	EBANK=	AOSQ	
IDLERADR	2CADR	DAPIDLER	
MOREIDLE	TC	IBNKCALL	# CALCULATE Q,R-AXES ATTITUDE ERRORS.
	CADR	QERRCALC	
	TC	IBNKCALL	
	CADR	CALCPERR	# CALCULATE P AXIS ATTITUDE ERRORS.
SHUTDOWN	EXTEND		
	DCA	IDLERADR	
	DXCH	T5ADR	

July 12, 2016

339

```
CAF      ZERO      # KILL ANY POSSIBLE JET REQUESTS
TS       NEXTP
TS       NEXTU
TS       NEXTV
EXTEND
WRITE    CHAN5
EXTEND
WRITE    CHAN6
CS       BGIM23    # TURN TRIM GIMBAL OFF
EXTEND
WAND     CHAN12
TCF      SETTIME5  # RETURN IN 100 MSEC.
```

```
MANFLAG  OCT      03021
BGIM23   OCTAL    07400
EBANK=   OMEGAP
PAXADIDL 2CADR    PAXIS
```

```
MS100    =        OCT37766
COSMG     =        ITEMP1
JUMPDSP   EXTEND   # TRANSFER TO BANK 20
DCA       DSPCADR  # FOR ATTITUDE ERROR DISPLAYS
DTCB
```

```
EBANK=   AK
DSPCADR  2CADR    ALTDSPY
```

Page 1414

```
BANK     20
SETLOC   DAPS3
BANK
COUNT*  $$/NEEDL
```

```
# PROGRAM:      ALTDSPY
# MOD 0.        6 DEC 1967
# AUTHOR:       CRAIG WORK, DON KEENE, MIT IL
# MOD 3 BY DON KEENE AUG 1, 1968 -- MOVED PROGRAM TO BANK 20
#
```

PROGRAM DESCRIPTION:

```
# ALTDSPY REVERSES THE DSPLYALT BIT OF RCSFLAGS EACH TIME IT IS CALLED, WHICH IS PRESUMABLY
# IF THE REVERSED BIT IS ONE, NEEDLER IS CALLED TO DISPLAY ATTITUDE ERRORS. IF THE BIT IS
# ORS ARE CALCULATED AS 1) DAP FOLLOWING ERRORS, IF NEEDLFLG = 0, AND 2) TOTAL ATTITUDE ERROR
#
```

```
# WARNING:      ALTDSPY MAY ONLY BE CALLED WITH INTERRUPT INHIBITED
#
```

```

# WARNING:      EBANK MUST BE SET TO 6 WHEN USING THIS ROUTINE.
#
# INPUT:        RCSFLAGS AND      1) IF NEEDLFLG = 0, INPUT PERROR, QERROR, RERROR.
#                                     2) IF NEEDLFLG = 1, INPUT CPHI, CTHETA, CPSI, CDUX, CDUY.
#
#
# OUTPUTS:      RCSFLAGS WITH DSPLYALT REVERSED, AK, AK1, AK2, + NEEDLER OUTPUTS.
#
# ENTRY:        TCF      ALTDSPY
#
# EXIT:         TCF      CHEKMORE
#
# ALARM OR ABORT EXITS: NONE
#
# SUBPROGRAMS CALLED:  NEEDLER, OVERSUB2
#
# DEBRIS:       A, L, AND NEEDLER DEBRIS.

ALTDSPY      CA      RCSFLAGS      # INVERT THE DISPLAY ALTERNATION BIT.
             TS      L
             CA      DSPLYALT
             EXTEND
             RXOR     LCHAN
             TS      RCSFLAGS

             MASK     DSPLYALT
             CCS      A              # IS ALTERNATION FLAG ZERO?
             TCF      NEEDLER

             CAE      FLAGWRDO      # NEEDLFLG WILL INDICATE TOTAL OR DAP AT-
# Page 1415
             MASK     NEEDLBIT      # TITUDE ERROR DISPLAY REQUEST.
             CCS      A
             TCF      DSPLYTOT      # TOTAL ERROR IS NEEDED IN AK, AK +1, AK +2

             CS      QERROR          # YES.  DISPLAY ATT ERRORS ON THE -BALL.
             TS      AK +1          # ERROR COMPLEMENTS ARE INPUT TO NEEDLER.
             CS      RERROR
             TS      AK +2
             CS      PERROR
             XCH      AK

             TCF      RETNMORE      # DISPLAY THESE THE NEXT TIME THROUGH

# CALCULATE GIMBAL ANGLE TOTAL ERRORS, RESOLVE INTO PILOT AXES, STORE TOTAL ERRORS FOR

```

```

DSPLYTOT      EXTEND
               QXCH   ITEMP1      # SAVE Q FOR CHEKBITS RETURN.

               CA     CTHETA      # DESIRED ATTITUDE, Y-AXIS, 2'S COMP.
EXTEND         # SUBTRACT CURRENT ATTITUDE.
               MSU    CDUY      # DIFFERENCE SCALED AT PI, 1'S COMP.
               TS     AK        # SAVE FOR R-ERROR CALCULATION.
EXTEND
               MP     M21      # (CTHETA-CDUY)*M21 SCALED AT PI RADIANS.
               XCH    AK +1     # STORE FIRST TERM OF Q ERROR.
               CA     CPSI      # DESIRED ATTITUDE, Z-AXIS, 2'S COMP.
EXTEND         # SUBTRACT CURRENT ATTITUDE.
               MSU    CDUZ      # DIFFERENCE SCALED AT PI, 1'S COMP.
               TS     AK +2     # SAVE Z-AXIS TERM FOR R ERROR CALCULATION
EXTEND
               MP     M22      # (CPSI-CDUZ)*M22, SCALED AT PI RADIANS.
               AD     AK +1     # Q ERROR COMPLETE           , AT PI RAD.
               TC     OVERSUB2  # PIN NEEDLES IN CASE OF OVERFLOW
               TS     AK +1

# R ERROR CALCULATION NEXT.

               CA     AK        # Y-AXIS DIFFERENCE STORED BY Q-AXIS CALC.
EXTEND
               MP     M31      # (CTHETA-CDUY)*M31, SCALED AT PI RADIANS.
               XCH    AK +2     # FIRST TERM OF R ERROR.
                               # Z-AXIS DIFFERENCE, STORED BY A CALC. IS
EXTEND         # RECOVERED BY THE EXCHANGE.
               MP     M32      # (CPSI-CDUZ)*M32, SCALED AT PI RADIANS.
               AD     AK +2     # R ERROR COMPLETE           , AT PI RAD.
               TC     OVERSUB2  # PIN NEEDLES IN CASE OF OVERFLOW.
               TS     AK +2

# NOW CALCULATE P ERROR.  (NOTE THAT M13 = 1, SCALED AT 1, SO THE MULTIPLICATION IS BY-PASSED.)
# Page 1416
               CA     AK        # Y-AXIS DIFFERENCE STORED BY Q AXIS CALC.
EXTEND
               MP     M11      # (CTHETA-CDUY)*M11 SCALED AT PI RADIANS.
               XCH    AK        # FIRST TERM OF P ERROR IN AK, AT PI RAD.
               CAE    CPHI      # DESIRED ATTITUDE, X-AXIS, 2'S COMP.
EXTEND         # SUBTRACT CURRENT X ATTITUDE.
               MSU    CDUX      # X-AXIS DIFFERENCE, 1'S COMP, AT PI RAD.

# M13 = 1, SO BYPASS THE MULTIPLICATION.
#           EXTEND
#           MP     M13      # (CPHI-CDUX)*M13 SCALED AT PI RADIANS.

```

```

AD      AK      # P ERROR COMPLETE      , SCALED AT PI RAD
TC      OVERSUB2  # PIN NEEDLES IN CASE OF OVERFLOW.
TS      AK

EXTEND
QXCH    ITEMP1    # RESTORE Q FOR CHEKBITS RETURN.

TCF     RETNMORE   # DISPLAY THESE THE NEXT TIME THROUGH

# Page 1417
# FDAI ATTITUDE ERROR DISPLAY SUBROUTINE
#
# PROGRAM DESCRIPTION:          D. KEENE          5/24/67
# MOD 1 BY CRAIG WORK, 12 DEC 67
# MOD 2 BY CRAIG WORK, 6 APRIL 68, CONVERTS ATTITUDE ERROR DISPLAY SCALING FROM 16 7/
#
# THIS SUBROUTINE IS USED TO DISPLAY ATTITUDE ERRORS ON THE FDAI VIA THE DIGITAL TO A
# IN THE CDUS. CARE IS TAKEN TO METER OUT THE APPROPRIATE NUMBER OF PULSES TO THE IN
# OVERFLOW, TO CONTROL THE RELAY SEQUENCING, AND TO AVOID INTERFERENCE WITH THE COARS
# THE DACS.
#
# CALLING SEQUENCE:
# DURING THE INITIALIZATION SECTION OF THE USER'S PROGRAM, BIT3 OF RCSFLAGS SH
# TURN-ON SEQUENCE WITHIN THE NEEDLES PROGRAM:
# CS      RCSFLAGS      # IN EBANK6
# MASK    BIT3
# ADS     RCSFLAGS
# THEREAFTER, THE ATTITUDE ERRORS GENERATED BY THE USER SHOULD BE TRANSFERRED T
# AK      SCALED 180 DEGREES      NOTE:  THESE LOCATIONS ARE SUBJECT
# AK1     SCALED 180 DEGREES      TO CHANGE
# AK2     SCALED 180 DEGREES
# FULL SCALED DEFLECTION OF THE NEEDLES CORRESPONDS TO 5 1/16 DEGREES, WHILE 38
# CORRESPONDS TO 42 3/16 DEGREES. (DAC MAXIMUM CAPACITY IS 384 BITS.) 46 BITS
#
# A CALL TO NEEDLER WILL THE UPDATE THE DISPLAY:
# INHINT
# TC      IBNKCALL      # NOTE:  EBANK SHOULD BE SET TO E6
# CADR    NEEDLER
# RELINT
# THIS PROCESS SHOULD BE REPEATED EACH TIME THE ERRORS ARE UPDATED. AT LEAST 3
# REQUIRED BEFORE ANYTHING IS ACTUALLY DISPLAYED ON THE ERROR METERS.
# NOTE:  EACH CALL TO NEEDLER MUST BE SEPARATED BY AT LEAST 50 MS. TO ASSURE PR
#
# ERASABLES USED:
# AK      CDUXCMD

```

```

#      AK1      CDUYCMD
#      AK2      CDUZCMD
#      EDRIX    A,L,Q
#      EDRIY    T5TEMP
#      EDRIEZ   DINDX
# Page 1418
#
# SWITCHES:      RCSFLAGS  BITS 3,2
#
# I/O CHANNELS:  CHAN12  BIT 4 (COARSE ALIGN -- READ ONLY)
#                  CHAN12  BIT 6 (IMU ERROR COUNTER ENABLE)
#                  CHAN14  BIT 13,14,15 (DAC ACTIVITY)
#
# SIGN CONVENTION:  AK = THETAC - THETA
#                   WHERE  THETAC = COMMAND ANGLE
#                   THETA = PRESENT ANGLE

NEEDLER          CA      RCSFLAGS
                  MASK    SIX
                  EXTEND
                  BZF     NEEDLES3
                  MASK    BIT3
                  EXTEND
                  BZF     NEEDLER2          # BIT3 = 0, BIT2 = 1

                  CS      BIT6              # FIRST PASS BIT3 = 1
                  EXTEND                    # DISABLE IMU ERROR COUNTER TO ZERO DACS
                  WAND     CHAN12            # MUST WAIT AT LEAST 60 MS BEFORE
NEEDLE11          CS      ZERO              # ENABLING COUNTERS.
                  TS       AK                # ZERO THE INPUTS ON FIRST PASS
                  TS       AK1
                  TS       AK2
                  TS       EDRIX             # ZERO THE DISPLAY REGISTERS
                  TS       EDRIY
                  TS       EDRIEZ
                  TS       CDUXCMD           # ZERO THE OUT COUNTERS
                  TS       CDUYCMD
                  TS       CDUZCMD
                  CS       SIX              # RESET RCSFLAGS FOR PASS2
                  MASK     RCSFLAGS
                  AD        BIT2
                  TS       RCSFLAGS
                  TCF       RETNMORE

NEEDLER2          CAF      BIT6              # ENABLE IMU ERROR COUNTERS
                  EXTEND

```

```

WOR      CHAN12
CS        SIX      # RESET RCSFLAGS TO DISPLAY ATTITUDE
MASK     RCSFLAGS  # ERRORS.  WAIT AT LEAST 4 MS FOR
TS        RCSFLAGS  # RELAY CLOSURE.
TCF      RETNMORE

NEEDLES3  CAF      BIT6      # CHECK TO SEE IF IMU ERROR COUNTER
          EXTEND    # IS ENABLED
          RAND      CHAN12

# Page 1419

          CCS      A          # IF NOT, RE-INITIALIZE NEEDLER.
          TCF      NEEDLES

          CS        RCSFLAGS  # SET UP INITIALIZATION FLAG IN RCSFLAGS.
          MASK     BIT3
          ADS       RCSFLAGS
          TCF      RETNMORE

NEEDLES   CAF      TWO
DACLOOP   TS        DINDX
          CS        ONETENTH  # RESCALE INPUTS TO + OR - 1800 DEGREES.
          EXTEND
          INDEX     DINDX
          MP        AK
          TS        L
          CCS       A
          CA        DACLIMIT
          TCF       +2
          CS        DACLIMIT
          AD        L
          TS        T5TEMP    # OVFL0 CHK
          TCF       +4
          INDEX     A          # ON OVERFLOW LIMIT OUTPUT TO +-384
          CAF       DACLIMIT
          TS        L
          INDEX     DINDX
          CS        EDRIXEX    # CURRENT VALUE OF DAC
          AD        L
          INDEX     DINDX
          ADS       CDUXCMD
          INDEX     DINDX
          LXCH      EDRIXEX
          CCS       DINDX
          TCF       DACLOOP
          CAF       13,14,15
          EXTEND

```


July 12, 2016

345

	WOR	CHAN14	# SET DAC ACTIVITY BITS
	TCF	RETNMORE	
	DEC	-384	
DACLIMIT	DEC	16000	
	DEC	384	
ONETENTH	OCT	03146	# DECIMAL +0.1, SCALED AT 1.
DSPLYALT	EQUALS	BIT4	# 100 MS ALTERNATION BIT IN RCSFLAGS
OVERSUB2	TS	7	# RETURNS A UNCHANGED OR LIMITED TO
	TC	Q	# POSMAX OR NEGMAX IF A HAS OVERFLOW
	INDEX	A	
# Page 1420	CS	LIMITS	# DUPLICATE CODING IN BANK 16
	TC	Q	
RETNMORE	EXTEND		# RETURN TO CHEKMORE
	DCA	MORECADR	
	DTCB		
	EBANK=	AOSQ	
MORECADR	2CADR	CHEKMORE	

This code is written to file `src/DAPIDLER-PROGRAM.s`.

B.22 DAP INTERFACE SUBROUTINES

```

346  <src/DAP-INTERFACE-SUBROUTINES.s 346>≡
# Copyright:    Public domain.
# Filename:     DAP_INTERFACE_SUBROUTINES.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:   yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1406-1409
# Mod history:  2009-05-10 SN    (Sergio Navarro).  Started adapting
#               from the Luminary131/ file of the same
#               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
#
# Page 1406
#
#       BANK      20
#       SETLOC    DAPS3
#       BANK
#
#       EBANK=    CDUXD
#       COUNT*    $$/DAPIF
#
# MOD 0          DATE    11/15/66          BY GEORGE W. CHERRY
# MOD 1          1/23/67          MODIFICATION BY PETER ADLER
#
# FUNCTIONAL DESCRIPTION
#       HEREIN IS A COLLECTION OF SUBROUTINES WHICH ALLOW MISSION CONTROL PROGRAMS TO
#       AND INTERFACE WITH THE DAP.
#

```

```

# CALLING SEQUENCES
#       IN INTERRUPT OR WITH INTERRUPT INHIBITED
#           TC      IBNKCALL
#           FCADR   ROUTINE
#       IN A JOB WITHOUT INTERRUPT INHIBITED
#           INHINT
#           TC      IBNKCALL
#           FCADR   ROUTINE
#           RELINT
#
# OUTPUT
#       SEE INDIVIDUAL ROUTINES BELOW
#
# DEBRIS
#       A, L, AND SOMETIMES MDUETEMP                ODE      NOT IN PULSES MODE

# Page 1407
# SUBROUTINE NAMES:
#       SETMAXDB, SETMINDB, RESTORDB, PFLITEDB
# MODIFIED:      30 JANUARY 1968 BY P. S. WEISSMAN TO CREATE RESTORDB.
# MODIFIED:      1 MARCH 1968 BY P. S. WEISSMAN TO SAVE EBANK AND CREATE PFLITEDB
#
# FUNCTIONAL DESCRIPTION:
#       SETMAXDB -- SET DEADBAND TO 5.0 DEGREES
#       SETMINDB -- SET DEADBAND TO 0.3 DEGREE
#       RESTORDB -- SET DEADBAND TO MAX OR MIN ACCORDING TO SETTINGS OF DBSELECT BIT OF DAPBOOL
#       PFLITEDB -- SET DEADBAND TO 1.0 DEGREE AND ZERO THE COMMANDED ATTITUDE CHANGE AND COMMA
#
#       ALL ENTRIES SET UP A NOVAC JOB TO DO 1/ACCS SO THAT THE TJETLAW SWITCH CURVES ARE POSIT
#       REFLECT TEH NEW DEADBAND.  IT SHOULD BE NOTED THAT THE DEADBAND REFERS TO THE ATTITUDE
#
# SUBROUTINE CALLED:      NOVAC
#
# CALLING SEQUENCE:      SAME AS ABOVE
#                         OR      TC RESTORDB +1      FROM ALLCOAST
#
# DEBRIS:                A, L, Q, RUPTREG1, (ITEMPS IN NOVAC)

RESTORDB      CAE      DAPBOOLS      # DETERMINE CREW-SELECTED DEADBAND.
              MASK     DBSELECT
              EXTEND
              BZF      SETMINDB

SETMAXDB      CAF      WIDEDB      # SET 5 DEGREE DEADBAND.
              +1      TS      DB

```

	EXTEND		# SET UP JOB TO RE-POSITION SWITCH CURVES.
	QXCH	RUPTREG1	
CALLACCS	CAF	PRI027	
	TC	NOVAC	
	EBANK=	AOSQ	
	2CADR	1/ACCJOB	
	TC	RUPTREG1	# RETURN TO CALLER.
SETMINDB	CAF	NARROWDB	# SET 0.3 DEGREE DEADBAND.
	TCF	SETMAXDB +1	
PFLITEDB	EXTEND		# THE RETURN FROM CALLACCS IS TO RUPTREG1.
	QXCH	RUPTREG1	
	TC	ZATTEROR	# ZERO THE ERRORS AND COMMANDED RATES.
	CAF	POWERDB	# SET DB TO 1.0 DEG.
	TS	DB	
	TCF	CALLACCS	# SET UP 1/ACCS AND RETURN TO CALLER.
NARROWDB	OCTAL	00155	# 0.3 DEGREE SCALED AT 45.
# Page 1408			
WIDEDB	OCTAL	03434	# 5.0 DEGREES SCALED AT 45.
POWERDB	DEC	.02222	# 1.0 DEGREE SCALED AT 45.
ZATTEROR	CAF	EBANK6	
	XCH	EBANK	
	TS	L	# SAVE CALLERS EBANK IN L.
	CAE	CDUX	
	TS	CDUXD	
	CAE	CDUY	
	TS	CDUYD	
	CAE	CDUZ	
	TS	CDUZD	
	TCF	STOPRATE +3	
STOPRATE	CAF	EBANK6	
	XCH	EBANK	
	TS	L	# SAVE CALLERS EBANK IN L.
+3	CAF	ZERO	
	TS	OMEGAPD	
	TS	OMEGAQD	
	TS	OMEGARD	
	TS	DELCDUX	
	TS	DELCDUY	
	TS	DELCDUZ	
	TS	DELPEROR	
	TS	DELQEROR	

```

        TS      DELREROR
        LXCH     EBANK      # RESTORE CALLERS EBANK.
        TC       Q

# SUBROUTINE NAME:      ALLCOAST
# WILL BE CALLED BY FRESH STARTS AND ENGINE OFF ROUTINES.
#
# CALLING SEQUENCE:      (SAME AS ABOVE)
#
# EXIT:                  RETURN TO Q.
#
# SUBROUTINES CALLED:    STOPRATE, RESTORDB, NOVAC
#
# ZERO:                  (FOR ALL AXES) AOS, ALPHA, AOSTERM, OMEGAD, DELCDU, DELEROR
#
# OUTPUT:                DRIFTBIT/DAPBOOLS, OE, JOB TO DO 1/ACCS
#
# DEBRIS:                A, L, Q, RUPTREG1, RUPTREG2, (ITEMPS IN NOVAC)

ALLCOAST      EXTEND      # SAVE Q FOR RETURN
               QXCH      RUPTREG2

# Page 1409

        TC      STOPRATE  # CLEAR RATE INTERFACE. RETURN WITH A=0
        LXCH     EBANK    # AND L=EBANK6. SAVE CALLER'S EBANK.
        TS       AOSQ
        TS       AOSQ +1
        TS       AOSR
        TS       AOSR +1
        TS       ALPHAQ   # FOR DOWNLIST.
        TS       ALPHAR
        TS       AOSQTERM
        TS       AOSRTERM
        LXCH     EBANK    # RESTORE EBANK (EBANK6 NO LONGER NEEDED)

        CS       DAPBOOLS  # SET UP DRIFTBIT
        MASK     DRIFTBIT
        ADS      DAPBOOLS
        TC       RESTORDB +1 # RESTORE DEADBANK TO CREW-SELECTED VALUE.

        TC       RUPTREG2  # RETURN.

```

This code is written to file `src/DAP-INTERFACE-SUBROUTINES.s`.

B.23 DISPLAY INTERFACE ROUTINES

```

350  <src/DISPLAY-INTERFACE-ROUTINES.s 350>≡
# Copyright:    Public domain.
# Filename:     DISPLAY_INTERFACE_ROUTINES.agc
# Purpose:      Part of the source code for Comanche, build 055. It
#               is part of the source code for the Command Module's
#               (CM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:   yaYUL
# Reference:    pp. 1452-1484
# Contact:      Ron Burkey <info@sandroid.org>
# Website:      http://www.ibiblio.org/apollo.
# Mod history:  2009-05-07 RSB   Adapted from Colossus249 file of the same
#                               name, and page images. Corrected various
#                               typos in the transcription of program
#                               comments, and these should be back-ported
#                               to Colossus249.
#
# The contents of the "Comanche055" files, in general, are transcribed
# from scanned documents.
#
# Assemble revision 055 of AGC program Comanche by NASA
# 2021113-051. April 1, 1969.
#
# This AGC program shall also be referred to as Colossus 2A
#
# Prepared by
#
#           Massachusetts Institute of Technology
#           75 Cambridge Parkway
#           Cambridge, Massachusetts
#
# under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information. Please report any errors to info@sandroid.org.
#
# Page 1452
# DISPLAYS CAN BE CLASSIFIED INTO THE FOLLOWING CATEGORIES --
# 1. PRIORITY DISPLAYS -- DISPLAYS WHICH TAKE PRIORITY OVER ALL OTHER DISPLAYS
#    OUT UNDER CRITICAL ALARM CONDITIONS.
# 2. EXTENDED VERB DISPLAYS -- ALL EXTENDED VERBS AND MARK ROUTINES SHOULD USE
# 3. NORMAL DISPLAYS -- ALL MISSION PROGRAM DISPLAYS WHICH INTERFACE WITH THE
#    SEQUENCE OF EVENTS.
# 4. MISC. DISPLAYS -- ALL DISPLAYS NOT HANDLED BY THE DISPLAY INTERFACE ROUTINE
#    MM DISPLAYS AND SPECIAL PURPOSE DISPLAYS HANDLED BY PINBALL.
# 5. ASTRONAUT INITIATED DISPLAYS -- ALL DISPLAYS INITIATED EXTERNALLY.

```

```

#
# THE FOLLOWING TERMS ARE USED TO DESCRIBE THE STATUS OF DISPLAYS --
#
#   1. ACTIVE -- THE DISPLAY WHICH IS (1) BEING DISPLAYED TO THE ASTRONAUT AND WAITING FOR
#   (2) WAITING FIRST IN LINE FOR THE ASTRONAUT TO FINISH USING THE DSKY OR (3) BEING D
#   BUT NOT WAITING FOR A RESPONSE.
#
#   2. INACTIVE -- A DISPLAY WHICH HAS (1) BEEN ACTIVE BUT WAS INTERRUPTED BY A DISPLAY OF
#   (2) BEEN PUT INTO THE WAITING LIST AT TIME IT WAS REQUESTED DUE TO THE FACT A HIGHE
#   WAS ALREADY DOING, (3) BEEN INTERRUPTED BY THE ASTRONAUT (CALLED A PINBRANCH CONDIT
#   OF INACTIVE DISPLAY IS USUALLY REACTIVATED ONLY BY PINBALL) OR (4) A DISPLAY WHICH
#   HAS INFO SAVED FOR RESTART PURPOSES.
#
# DISPLAY PRIORITIES WORK AS FOLLOWS --
#   INTERRUPTS --
#
#   1. THE ASTRONAUT CAN INTERRUPT ANY DISPLAY WITH AN EXTERNAL DISPLAY REQUEST.
#   2. INTERNAL DISPLAYS CAN NOT BE SENT OUT WHEN THE ASTRONAUT IS USING THE DSKY.
#   3. PRIORITY DISPLAYS INTERRUPT ALL OTHER TYPES OF INTERNAL DISPLAYS. A PRIORI
#   PRIORITY DISPLAY WILL CAUSE AN ABORT UNLESS BIT14 IS SET FOR THE LINUS ROUT
#   4. A MARK DISPLAY INTERRUPTS ANY NORMAL DISPLAY.
#   5. A MARK THAT INTERRUPTS A MARK COMPLETELY REPLACES IT.
#
# ORDER OF WAITING DISPLAYS --
#
#   1. ASTRONAUT
#   2. PRIORITY
#   3. INTERRUPTED MARK
#   4. INTERRUPTED NORMAL
#   5. MARK TO BE REQUESTED (SEE DESCRIPTION OF ENDMARK)
#   6. MARK WAITING
#   7. NORMAL WAITING
#
# Page 1453
# THE DISPLAY ROUTINES ARE INTENDED TO SERVE AS AN INTERFACE BETWEEN THE USER AND PINBALL. THE
# FOLLOWING STATEMENTS CAN BE MADE ABOUT NORMAL DISPLAYS AND PRIORITY DISPLAYS (A DESCRIPTION C
# WILL FOLLOW LATER):
#
#   1. ALL ROUTINES THAT END IN R HAVE AN IMMEDIATE RETURN TO THE USER. FOR ALL FLASHING
#   IS TO THE USER'S CALL CADR +4. FOR THE ONLY NON-FLASHING IMMEDIATE RETURN DISPLAY
#   IS TO THE USER'S CALLING LOC +1.
#
#   2. ALL ROUTINES NOT ENDING IN R DO NOT DO AN IMMEDIATE RETURN TO THE USER.
#
#   3. ALL ROUTINES THAT END IN R START A SEPARATE JOB (MAKEPLAY) WITH USER'S JOB PRIORITY
#
#   4. ALL ROUTIENS NOT ENDING IN R BRANCH DIRECTLY TO MAKEPLAY WHICH MAKES THESE DISPLAYS
#   USER'S JOB.
#
#   5. ALL DISPLAY ROUTIENS ARE CALLED VIA BANKCALL.
#
#   6. TO RESTART A DISPLAY THE USER WILL GENERALLY USE A PHASE OF ONE WITH DESIRED RESTAR
#   DESCRIPTION OF RESTARTS).
#
#   7. ALL FLASHING DISPLAYS HAVE 3 RETURNS TO THE USER FROM ASTRONAUT RESPOSES. A TERMIN
#   TO THE USER'S CALL CADR +1. A PROCEED (V33) BRANCHES TO THE USER'S CALL CADR +2.
#   (V32) BRANCHES TO THE USER'S CALL CADR +3.

```

```

#      8.  ALL ROUTINES MUST BE USED UNDER EXECUTIVE CONTROL
#
# A DESCRIPTION OF EACH ROUTINE WITH AN EXAMPLE FOLLOWS:

#      GODSP IS USED TO DISPLAY A VERB NOUN ARRIVING IN A.  NO RETURN IS MADE TO THE
#      1.  GODSP IS NOT RESTARTABLE
#      2.  A VERB PASTE WITH GODSP ALWAYS TURNS ON THE FLASH.
#
#              CAF      VXXNYY
#              TC       BANKCALL
#              CADR     GODSP
#      VXXNYY OCT      OXXYY

#      GODSPR IS THE SAME AS GODSP ONLY RETURN IS TO THE USER.
#
#              CAF      VXXNYY
#              TC       BANKCALL
#              CADR     GODSPR
#              ...      ...      # IMMEDIATE RETURN OF GODSPR

#      GOFLASH DISPLAYS A FLASHING VERB NOUN WITH NO IMMEDIATE RETURN TO THE USER.
#      THE ASTRONAUT (SEE NO. 7 ABOVE).
#
#              CAF      VXXNYY      # VXX NYY WILL BE A FLASHING
#              TC       BANKCALL
#              CADR     GOFLASH
#              ...      ...      # TERMINATE RETURN
#              ...      ...      # PROCEED RETURN
#              ...      ...      # ENTER OR RECYCLE RETURN

#      GOPERF1 IS ENTERED WITH DESIRED CHECKLIST VALUE IN A.  GOPERF1 WILL DISPLAY T
# Page 1454
#      V01 N25.  A FLASHING PLEASE PERFORM ON CHECKLIST (V50 N25) IS THEN DISPLAYED
#      USER (SEE NO. 7 ABOVE).
#      GOPERF1 BLANKS REGISTERS R2 AND R3
#
#              CAF      OCTXX      # CODE FOR CHECKLIST VALUE XX
#              TC       BANKCALL
#              CADR     GOPERF1
#              ...      ...      # TERMINATE RETURN
#              ...      ...      # PROCEED RETURN
#              ...      ...      # ENTER RETURN

#      GOPERF2 IS ENTERED WITH A VARIABLE NOUN AND V01 (V00 FOR N10 OR N11) IN A.  C
#      REQUESTED NOUN BY MEANS OF A V01NYY OR A VOONYY.  PLEASE PERFORM ON NOUN (V50
#      DISPLAY.  NO IMMEDIATE RETURN IS MADE TO THE USER (SEE NO. 7 ABOVE).
#      GOPERF2 DOES NOT BLANK ANY REGISTERS
#
#              CAF      VXXNYY      # VARIABLE NOUN YY.  XX=0 OR C
#              TC       BANKCALL
#              CADR     GOPERF2
#              ...      ...      # TERMINATE RETURN

```



```

#           ...      ...      # PROCEED RETURN
#           ...      ...      # ENTER RETURN
# GOPERF3 IS USED FOR A PLEASE PERFORM ON A PROGRAM NUMBER.  THE DESIRED PROGRAM NO. IS E
# DISPLAYS THE NO. BY MEANS OF A V06 N07 FOLLOWED BY A FLASHING V50 N07 FOR A PLEASE PER
# IS MADE TO THE USER (SEE NO. 7 ABOVE).
# GOPERF3 BLANKS REGISTERS R2 AND R3
#           CAF      DECXX      # REQUEST PERFORM ON PXX
#           TC       BANKCALL
#           CADR     GOPERF3
#           ...      ...      # TERMINATE RETURN
#           ...      ...      # PROCEED RETURN
#           ...      ...      # ENTER RETURN

# GOPERF4 IS USED FOR A PLEASE PERFORM ON AN OPTION.  THE DESIRED OPTION IS ENTERED IN A
# GOPERF4 DISPLAYS R1 AND R2 BY MEANS OF A V04N06 FOLLOWED BY A FLASHING V50N06 FOR A PL
# IMMEDIATE RETURN IS MADE TO THE USER (SEE NO. 7 ABOVE).
#           CAF      OCTXX      # REQUEST PERFORM ON OPTION XX
#           TC       BANKCALL
#           CADR     GOPERF4
#           ...      ...      # TERMINATE RETURN
#           ...      ...      # PROCEED RETURN
#           ...      ...      # ENTER RETURN
# GOPERF4 BLANKS REGISTER R3.
#
# Page 1455
# GODSPRET IS USED TO DISPLAY A VERB NOUN ARRIVING IN A WITH A RETURN TO THE USER AFTER T
# OUT.
#           CAF      VXXNYY
#           TC       BANKCALL
#           CADR     GODSPRET
#           ...      ...      # RETURN TO USER.

# REGODSP IS USED TO DISPLAY A VERB NOUN ARRIVING IN A.  REGODSP IS THE SAME AS GODSP ONL
# ACTIVE NORMAL DISPLAY IF ONE WAS ACTIVE.
#           CAF      VXXNYY
#           TC       BANKCALL
#           CADR     REGODSP

# REFLASH IS THE SAME AS GOFLASH ONLY REFLASH REPLACES ANY ACTIVE NORMAL DISPLAY IF ONE W
#           CAF      VXXNYY      # VXX NYY WILL BE A FLASHING VERB NOUN
#           TC       BANKCALL
#           CADR     REFLASH
#           ...      ...      # TERMINATE RETURN
#           ...      ...      # PROCEED RETURN
#           ...      ...      # ENTER RETURN

```

```

#      GOFLASHR IF SAME AS GOFLASH ONLY AN IMMEDIATE RETURN IS MADE TO THE USER'S CA
#      CAF      VXXNYY
#      TC      BANKCALL
#      CADR     GOFLASHR
#      ...      ...      # TERMINATE RETURN
#      ...      ...      # PROCEED RETURN
#      ...      ...      # ENTER OR RECYCLE RETURN
#      ...      ...      # IMMEDIATE RETURN FROM GOFLA

#      GOPERF1R IS THE SAME AS GOPERF1 ONLY GOPERF1R HAS AN IMMEDIATE RETURN TO USER
#      GOPERF1R BLANKS REGISTERS R2 AND R3
#      CAF      OCTXX      # CODE FOR CHECKLIST VALUE XX
#      TC      BANKCALL
#      CADR     GOPERF1R
#      ...      ...      # TERMINATE RETURN
#      ...      ...      # PROCEED RETURN
#      ...      ...      # ENTER RETURN
#      ...      ...      # IMMEDIATE RETURN FROM GOPER

#      GOPERF2R IS THE SAME AS GOPERF2 ONLY AN IMMEDIATE RETURN IS MADE TO USER'S CA
# Page 1456
#      GOPERF2R DOES NOT BLANK ANY REGISTERS
#      CAF      VXXXNYY      # VARIABLE NOUN YY REQUESTED
#      TC      BANKCALL
#      CADR     GOPERF2R
#      ...      ...      # TERMINATE RETURN
#      ...      ...      # PROCEED RETURN
#      ...      ...      # ENTER RETURN
#      ...      ...      # IMMEDIATE RETURN HERE FROM

#      GOPERF3R IS THE SAME AS GOPERF3 ONLY AN IMMEDIATE RETURN IS MADE TO USER'S CA
#      GOPERF3R BLANKS REGISTERS R2 AND R3
#      CAF      PROGXX      # PERFORM PROGRAM XX
#      TC      BANKCALL
#      CADR     GOPERF3R
#      ...      ...      # TERMINATE RETURN
#      ...      ...      # PROCEED RETURN
#      ...      ...      # ENTER RETURN
#      ...      ...      # GOPERF3R IMMEDIATELY RETURN

#      GOPERF4R IS THE SAME AS GOPERF4 ONLY AN IMMEDIATE RETURN IS MADE TO USER'S CA
#      CAF      OCTXX      # REQUEST PERFORM ON OPTIONXX
#      TC      BANKCALL
#      CADR     GOPERF4R
#      ...      ...      # TERMINATE RETURN
#      ...      ...      # PROCEED RETURN

```

```

#           ...      ...      # ENTER RETURN
#           ...      ...      # IMMEDIATE RETURN TO USER
# GOPERF4R BLANKS REGISTER R3.
#
# REFLASHR IS THE SAME AS REFLASH ONLY AN IMMEDIATE RETURN IS MADE TO THE USER'S CALL CAD
#           CAF      VXXNYY      # VXX NYY WILL BE A FLASHING VERB NOUN
#           TC      BANKCALL
#           CADR     REFLASHR
#           ...      ...      # TERMINATE RETURN
#           ...      ...      # PROCEED RETURN
#           ...      ...      # ENTER RETURN
#           ...      ...      # IMMEDIATE RETURN TO USER

# REGODSPR IS THE SAME AS REGODSP ONLY A RETURN (IMMEDIATE) IS MADE TO THE USER.
# Page 1457
#           CAF      VXXNYY
#           TC      BANKCALL
#           CADR     REGODSPR
#           ...      ...      # IMMEDIATE RETURN TO USER

# Page 1458
# GOMARK IS USED TO DISPLAY A MARK VERB NOUN ARRIVING IN A. NO RETURN IS MADE TO THE USER
# GOXDSP = GOMARK
#           CAF      VXXNYY      # VXXNYY CONTAINS VERB AND NOUN
#           TC      BANKCALL
#           CADR     GOMARK      # OTHER EXTENDED VERBS USE CADR GOXDSP

# GOMARKR IS THE SAME AS GOMARK ONLY RETURN IS TO THE USER.
# GOXDSPR = GOMARKR
#           CAF      VXXNYY
#           TC      BANKCALL
#           CADR     GOMARKR      # OTHER EXTENDED VERBS USE CADR GOXDSPR
#           ...      ...      # IMMEDIATE RETURN OF GOMARKR

# GOMARKF DISPLAYS A FLASHING MARK VERB NOUN WITH NO IMMEDIATE RETURN TO THE USER. 3 RET
# THE ASTRONAUT (SEE NO. 7 ABOVE).
# GOXDSPF = GOMARKF
#           CAF      VXXNYY      # VXXNYY WILL BE A FLASHING MARK VERB N
#           TC      BANKCALL
#           CADR     GOMARKFR      # OTHER EXTENDED VERBS USE CADR GOXDSPFR
#           ...      ...      # TERMINATE RETURN
#           ...      ...      # PROCEED RETURN
#           ...      ...      # ENTER OR RECYCLE RETURN
#           ...      ...      # IMMEDIATE RETURN TO THE USER

# GOMARKFR IS THE SAME AS GOMARKF ONLY AN IMMEDIATE RETURN IS MADE TO THE USER CALL CADR

```

```

#      GOXDSPFR = GOMARKFR
#
#      CAF      VXXNYY      # FLASHING MARK VERB NOUN
#      TCF      BANKCALL
#      CADR     GOMARKFR    # OTHER EXTENDED VERBS USE CA
#      ...      ...        # TERMINATE RETURN
#      ...      ...        # PROCEED RETURN
#      ...      ...        # ENTER OR RECYCLE RETURN
#
#      ...      ...        # IMMEDIATE RETURN TO THE USI

#      GOMARK1 IS USED FOR A PLEASE PERFORM ON A MARK REQUEST WITH ONLY 1 ASTRONAUT
#      RETURN IS MADE.  THE DESIRED MARK PLEASE PERFORM VERB AND DESIRED NOUN IS EN
#      MEANS OF A V05NYY FOLLOWED BY A FLASHING V5XNYY FOR A PLEASE PERFORM.  THE AS
#      OR MARK REJECT OR AN ENTER.  THE ENTER IS THE ONLY ASTRONAUT RESPONSE THAT W
#      CAF      V5XNYY      # X=1,2,3,4      YY=NOUN
#      TC       BANKCALL
#
# Page 1459
#      CADR     GOMARK1
#      ...      ...        # ENTER RETURN

#      *** IF BLANKING DESIRED ON NON-R ROUTINES, NOTIFY DISPLAYER.
#
#      GOMARK1R IS THE SAME AS A GOMARK1 ONLY AN IMMEDIATE RETURN IS MADE TO THE USI
#      CAF      V5XNYY      # X=1,2,3,4      YY=NOUN
#      TC       BANKCALL
#      CADR     GOMARK1R
#      ...      ...        # ASTRONAUT ENTER RETURN
#      ...      ...        # IMMEDIATE RETURN TO USER

#      GOMARK2 IS THE SAME AS GOMARK1 ONLY 3 RETURNS ARE MADE TO THE USER FROM THE A
#      CAF      V5XNYY      # X=1,2,3,4      YY=NOUN
#      TC       BANKCALL
#      CADR     GOMARK2
#      ...      ...        # TERMINATE RETURN
#      ...      ...        # PROCEED RETURN
#      ...      ...        # ENTER RETURN

#      GOMARK2R IS THE SAME AS GOMARK1R ONLY 3 ASTRONAUT RETURNS ARE MADE TO THE USI
#      CAF      V5XNYY      # X=0,1,2,3,4      YY=NOUN
#      TCF      BANKCALL
#      CADR     GOMARK24
#      ...      ...        # TERMINATE RETURN
#      ...      ...        # PROCEED RETURN
#      ...      ...        # ENTER RETURN
#      ...      ...        # IMMEDIATE RETURN TO THE USI

```

```

# GOMARK3 IS USED FOR A PLEASE PERFORM ON A MARK REQUEST WITH A 3 COMP. DEC DISPLAY. THE
# PERFORM VERB AND NOUN ARE ENTERED IN A. GOMARK3 DISPLAYS R1, R2, R3 BY MEANS OF A VO6N
# V5XNYY FOR A PLEASE PERFORM. GOMARK3 HAS 3 ASTRONAUT RETURNS TO THE USER WITH NO IMMEDIATE
# CAF V5XNYY # X=1,2,3,4 YY=NOUN
# TC BANKCALL
# CADR GOMARK3
# ... .. # TERMINATE RETURN
# ... .. # PROCEED RETURN
# ... .. # ENTER RETURN

# GOMARK4 IS THE SAME AS GOMARK3 ONLY R2 AND R3 ARE BLANKED AND R1 IS DISPLAYED IN OCTAL.
# CAF V5XNYY # X=1,2,3,4 YY=NOUN
# TC BANKCALL
# CADR GOMARK4
# ... .. # TERMINATE RETURN
# ... .. # PROCEED RETURN
# Page 1460
# ... .. # ENTER RETURN
# EXDSPRET IS USED TO DISPLAY A VERB NOUN ARRIVING IN A WITH A RETURN MADE TO THE USER AFTER
# SENT OUT.
# CAF VXNYY
# TC BANKCALL
# CADR EXDSPRET
# ... .. # RETURN TO USER

# KLEENEX CLEANS OUT ALL MARK DISPLAYS (ACTIVE AND INACTIVE). A RETURN IS MADE TO THE USER
# HAVE BEEN CLEANED OUT.
# TC BANKCALL
# CADR KLEENEX
# ... .. # RETURN TO USER

# MARKBRAN IS A SPECIAL PURPOSE ROUTINE USED FOR SAVING JOB VAC AREAS (SEE DESCRIPTION OF
# TC BANKCALL
# CADR MARKBRAN
# ... .. # BAD RETURN IF MARK DISPLAY NOT ACTIVE
# ... .. # (GOOD RETURN TO IMMEDIATE RETURN LOCATION)
# ... .. # LAST FLASHING MARK R ROUTINE)

# PINBRNCH REESTABLISHES THE LAST ACTIVE FLASHING DISPLAY. IF THERE IS NO ACTIVE FLASHING
# BLANKED AND CONTROL IS SENT TO ENDOFJOB.
# TC POSTJUMP
# CADR PINBRNCH

# PRIODSP IS USED AS A PRIORITY DISPLAY. IT WILL DISPLAY A GOFLASH TYPE DISPLAY WITH THREE
# THE ASTRONAUT (SEE NO. 7 ABOVE).
#

```

```

# THE MAIN PURPOSE OF PRIODSP IS TO REPLACE THE PRESENT DISPLAY WITH A DISPLAY
# PROVIDE A MEANS FOR RESTORING THE OLD DISPLAY WHEN THE PRIORITY DISPLAY
# IS RESPONDED TO BY THE ASTRONAUT.
#
# THE FORMER DISPLAY IS RESTORED BY AN AUTOMATIC BRANCH TO WAKE UP THE DISPLAY
# PRIO DISPLAY
#
# CAF VXXNYY # VXXNYY WILL BE A FLASHING V
# TC BANKCALL
# CADR PRIODSP
# ... ... # TERMINATE RETURN
# ... ... # PROCEED RETURN
# Page 1461
# ... ... # ENTER OR RECYCLE RETURN

# PRIODSPR IS THE SAME AS PRIODSP ONLY AN IMMEDIATE RETURN IS MADE TO THE USER
# CAF VXXNYY # VXXNYY WILL BE A FLASHING V
# TC BANKCALL
# CADR PRIODSPR
# ... ... # TERMINATE ACTION
# ... ... # PROCEED RETURN
# ... ... # ENTER OR RECYCLE RETURN
# ... ... # IMMEDIATE RETURN

# PRIOLARM DOES A V05N09 PRIODSPR.
#
# CLEANDSP CLEANS OUT ALL NORMAL DISPLAYS (ACTIVE AND INACTIVE). A RETURN IS M
# DISPLAYS ARE CLEANED OUT.
# TC BANKCALL
# CADR CLEANDSP
# ... ... # RETURN TO USER
# Page 1462
#
# GENERAL INFORMATION
# -----
#
# ALARM OR ABORT EXIT MODE --
# PRIOBORT TC ABORT
# OCT 1502
#
# PRIOBORT IS BRANCHED TO WHEN (1) A NORMAL DISPLAY IS REQUESTED AND ANOTHER NO
# (REFLASH AND REGODSP ARE EXCEPTIONS) OR (2) A PRIORITY DISPLAY IS REQUESTED W
# ALREADY ACTIVE (A PRIORITY WITH LINUS BIT14 IS AN EXCEPTION).
#
# ERASABLE INITIALIZATION REQUIRED --
# ACCOMPLISHED BY FRESH START -- 1. FLAGWRD4 (USED EXCLUSIVELY BY DISPLAY IN
# 2. NVSAVE = NORMAL VERB AND NOUN REGISTER.

```

```

#                                     3. EBANKTEM = NORMAL INACTIVE FLAGWORD (ALSO CONTAINS
#                                     5. R1SAVE = MARKBRAN CONTROL WORD
#                                     4. RESTREG = PRIORITY 30 AND SUPERBANK 3.
#
# OUTPUT --
#     NVWORD = PRIO VERB AND NOUN
#     NVWORD +1 (MARKNV) = MARK VERB AND NOUN
#     NVWORD +2 (NVSAVE) = NORMAL VERB AND NOUN
#     DSPFLG (EBANKSAV) = PRIO FLAGWORD (INCLUDING EBANK)
#     DSPFLG +1 (MARKEBAN) = MARK FLAGWORD (INCLUDING EBANK)
#     DSPFLG +2 (EBANKTEM) = NORMAL FLAGWORD (INCLUDING EBANK)
#     CADRFLSH = PRIO USER'S CALL CADR +1 LOCATION
#     CADRFLSH +1 (MARKFLSH) = MARK USER'S CALL CADR +1 LOCATION
#     CADRFLSH +2 (TEMPFLSH) = NORMAL USER'S CALL CADR +1 LOCATION
#     PRIOTIME = TIME EACH PRIO REQUEST FIRST SENT OUT
#     OPTION1 = DESIRED OPTION FROM GOPERF4
#     FLAGWRD4 = BIT INFO FOR CONTROL OF ALL DISPLAY ROUTINES
#     DSPTEM1 = R1 INFO FOR ASTRONAUT FROM PERFORM DISPLAYS (NORMAL)
#
# SUBROUTINES USED -- NVSUB, FLAGUP, FLAGDOWN, ENDOFJOB, BLANKSUB, ABORT, JOBWAKE, JOBSLEEP, FI
#     JAMTERM, NVSUBUSY, FLASHON, ENDIDLE, CHANG1, BANKJUMP, MAKECADR, NOVAC
#
# DEBRIS -- (STORED INTO)
#     TEMPORARY TEMPORARIES -- A, Q, L, MPAC +2, MPAC +3, MPAC +4, MPAC +5, MPAC +6, RUPREG2,
#     EBANK, RUPTREG4, LOC, BANKSET, MODE, MPAC, MPAC +1, FACEREG
#     ERASABLES (SHARED AND USED WITH OTHER PROGRAMS) -- CADRSTOR, DSPLIST, LOC, DSPTEM1, OPT
#     ERASABLES (USED ONLY BY DISPLAY ROUTINES) -- NVWORD,+1,+2, DSPFLAG,+1,+2, CADRFLSH,+1,+
# Page 1463
#     R1SAVE, MARK2PAC
#
# DEBRIS -- (USED BUT NOT STORED INTO) -- NOUNREG, VERBREG, LOCCTR, MONSAVE1
#
# FLAGWORD DESCRIPTIONS --
#     FLAGWRD4 -- SEE DESCRIPTION UNDER LOG SECTION ERASABLE ASSIGNMENTS
#
#     DSPFLG, DSPFLG+1, DSPFLG+2
#     -----
#     BITS 1  BLANK R1
#           2  BLANK R2
#           3  BLANK R3
#           4  FLASHING DISPLAY REQUESTED
#           5  PERFORM DISPLAY REQUESTED
#           6  -----          EXDSPRET          GODSPRET
#           7  PRIO DISPLAY          -----          -----
#           8  -----          DEC MARK PERFORM          -----
#           9  EBANK

```

```

#          10 EBANK
#          11 EBANK
#          12 -----
#          13 2ND PART OF PERFORM
#          14 REFLASH OR REDO
#          15 -----
#                                     MARK REQUEST
#                                     -----
#
# RESTARTING DISPLAYS --
#
# RULES FOR THE DSKY OPERATOR --
#   1. PROCEED AND TERMINATE SERVE AS RESPONSES TO REQUESTS FOR OPERATOR RESPONSES
#       AS THERE IS ANY REQUEST AWAITING OPERATOR RESPONSE, ANY USE OF PROCEED OR
#       RESPONSES TO THAT REQUEST. CARE SHOULD BE EXERCISED IN ATTEMPTING TO KILL
#       WITH PROCEED AND TERMINATE FOR THIS REASON.
#   2. THE ASTRONAUT MUST RESPOND TO A PRIORITY DISPLAY NO SOONER THAN 2 SECONDS
#       PROGRAM SENT OUT THE REQUEST FOR OPERATOR RESPONSE (THE ASTRONAUT WOULD S
#       DUE TO TIME IT TAKES TO GET DISPLAY SENT OUT.) IF THE ASTRONAUT RESPONDS
#       IS SENT OUT AGAIN -- AND AGAIN UNTIL AN ACCUMULATED 2 SECS FROM THE TIME
#       OUT. THE SAME 2 SEC. DELAY WILL OCCUR AT 163.84 SECS OR IN ANY MULTIPLE
#       CONSIDERATION.
#   3. KEY RELEASE BUTTON --
#       A) IF THE KEY RELEASE LIGHT IS ON, IT SIMPLY RELEASES THE KEYBOARD AND I
#       B) IF THE KEY RELEASE LIGHT IS OFF, AND IF SOME REQUEST FOR OPERATOR RES
#           AWAITING RESPONSE THEN IT RE-ESTABLISHES THE DISPLAYS THAT ORIGINALLY
#       IF AN OPERATOR WANTS THEREFORE TO RE-ESTABLISH BUT CONDITION (A) IS ENCOU
#       KEY RELEASE BUTTON MAY BE NECESSARY.
#   4. IT IS IMPORTANT TO ANSWER ALL REQUESTS FOR OPERATOR RESPONSE.
#   5. IT IS ALWAYS GOOD PRACTICE TO TERMINATE AN EXTENDED VERB BEFORE ASKING FO
#       OVER AGAIN.
#
# SPECIAL CONSIDERATONS --
# Page 1464
#   1. MPAC +2 SAVED ONLY IN MARK DISPLAYS
#   2. GODSP(R), REGODSP(R), GOMARK(R) ALWAYS TURN ON THE FLASH IF ENTERED WITH
#   3. ALL NORMAL DISPLAYS ARE RESTARTABLE EXCEPT GODSP(R), REGODSP(R)
#   4. ALL EXTENDED VERBS WITH DISPLAYS SHOULD START WITH A TC TESTXACT AND FINI
#   5. GODSP(R) AND REGODSP(R) MUST BE IN THE SAME EBANK AND SUPERBANK AS THE LA
#       BY A .1 RESTART PHASE CHANGE.
#   6. IN ORDER TO SET UP A NON DISPLAY .1 RESTART POINT, THE USER MUST MAKE CER
#       CORRECT PRIORITY AND SUPERBANK AND THAT EBANKTEM CONTAINS THE CO
#   7. IF CLEANDSP IS RESTARTED VIA A .1 PHASE CHANGE, CAF ZERO SHOULD BE EXECUT
#
# Page 1465
# CALLING SEQUENCE FOR BLANKING
#           CAF      BITX          # X=1,2,3 BLANK R1,R2,R3 RESPECTIVELY
#           TC       BLANKET

```



```

#           ...           # RETURN TO USER HERE
# IN ORDER TO USE BLANKET CORRECTLY, THE USER MUST USE A DISPLAY ROUTINE THAT ENDS IN R FIRST P
# TO BLANKET AT THE IMMEDIATE RETURN LOC.
      BLOCK   02
      SETLOC  FFTAG4
      BANK

      COUNT   02/DSPLA

BLANKET      TS      MPAC +6
             CS      PLAYTEM4
             MASK     MPAC +6
             INDEX    MPAC +5
             ADS      PLAYTEM4

             TC      Q

ENDMARK      TC      POSTJUMP
             CADR     MARKEND

CLEARMRK     CAF      ZERO
             TS      EXTVBACT

             INHINT
             CS      BIT1
             MASK     FLAGWRD4
             TS      FLAGWRD4

             RELINT
             TC      Q

# *** ALL EXTENDED VERB ROUTINES THAT HAVE AT LEAST ONE FLASHING DISPLAY MUST TCF ENDMARK OR TO
# FINISHED.

      BANK     10
      SETLOC   DISPLAYS
      BANK

      COUNT    10/DSPLA

# NTERONLY IS USED TO DIFFERENTIATE THE MARK ROUTINE WITH ONLY ONE RETURN TO THE USER FROM THE
# 3 RETURNS TO THE USER.  THIS ROUTINE IS ONLY USED BY GOMARK1 AND GOMARK1R.

MARKEND      TC      CLEARMRK
             TCF     MARKOVER

```

Page 1466

GOMARK	TS	PLAYTEM1	# ENTRANCE FOR MARK GODSP
GOMARS	CAF TCF	BIT15 GOFLASH2	# BIT15 SET FOR ALL MARK REQUESTS
KLEENEX	CAF	ZERO	# CLEAN OUT EXTENDED VERBS
GOMARKF	TS	PLAYTEM1	# ENTRANCE FOR MARK GOFLASH
	CAF TCF	MARKFMSK GOFLASH2	# MARK, FLASH
GOMARK2	TS	PLAYTEM1	# MARK GOPERFS-3 AST. RETURNS
MARKFORM	CAF TCF	MPERFMSK GOFLASH2	# MARK, PERFORM, FLASH
GOMARK3	TS CAF TCF	PLAYTEM1 MARK3MSK GOFLASH2	# USED FOR 3COMP DECIMAL PERFORM
GOMARK4	TS CAF TCF	PLAYTEM1 MARK4MSK GOFLASH2	# MARK, PERFORM, FLASH, BLANK
GOMARKR	TS	PLAYTEM1	# ENTRANCE FOR MARK GODSPR
	CAF TCF	BIT15 GODSPR2	
GOMARKFR	TS	PLAYTEM1	# ENTRANCE FOR MARK GOFLASHR
	CAF TCF	MARKFMSK GODSPRS	
GOMARK2R	TS CAF TCF	PLAYTEM1 MPERFMSK GODSPRS	# MARK GOPERFS-3 AST. RETS+ IMMEDIATE RET. # MARK, PERFORM, FLASH
GOMARK3R	TS CAF TCF	PLAYTEM1 MARK3MSK GODSPRS	
MAKEMARK	CAF TC	ONE COPIES	
	CA	FLAGWRD4	# IS NORM OR PRIO BUSY OR WAITING

July 12, 2016

363

```

                                MASK    OCT34300
                                CCS      A
                                TCF      CHKPRI0

                                CA        FLAGWRD4      # IS MARK SLEEPING DUE TO ASTRO BUSY?
# Page 1467
                                MASK     BIT9

                                EXTEND
                                BZF       MARKPLAY      # NO
                                TCF       ENDOFJOB

MARKPLAY                       INHINT
                                CS        FIVE          # RESET MARK OVER NORM, SET MARK
                                MASK      FLAGWRD4
                                AD        ONE
                                TS        FLAGWRD4
                                RELINT

GOGOMARK                       CS        MARKFLAG      # PERFORM
                                MASK      BIT5
                                CCS      A
                                TCF      MARKCOP
                                CS        MARKNV
                                TS        MARKNV

MARKCOP                        CAF        ONE          # MARK INDEX
                                TCF      PRIOPLAY

COPYTOGO                       CA        MPAC2SAV
                                TS        MPAC +2

COPYPACS                       INDEX     COPINDEX
                                CAF      PRIOOCT
                                TS        GENMASK

                                INDEX     COPINDEX
                                CA        EBANKSAV      # Was CAF --- RSB 2004.
                                TS        TEMPOR2        # ACTIVE EBANK AND FLAG

                                TS        EBANK

                                TC        Q
```

PINCHEK CHECKS TO SEE IF THE CURRENT MARK REQUEST IS MADE BY THE ASTRONAUT WHILE INTERRUPTING

```

# (A NORMAL OR A PRIO). IF THE ASTRONAUT TRIES TO MARK DURING A PRIO, THE CHECK FAIL
# REQUEST IS ENDED. IF HE TRIES TO MARK DURING A NORM, THE MARK IS ALLOWED. IN THIS
# UNTIL ALL MARKING IS FINISHED.
#
# IF THE MARK REQUEST COMES FROM THE PROGRAM DURING A TIME THE ASTRONAUT IS NOT INTER
# PRIO, THE MARK REQUEST IS PUT TO SLEEP UNTIL THE PRESENT ACTIVE DISPLAY IS RESPONDI

```

```

CHKPRIO      CA      FLAGWRD4      # MARK ATTEMPT DURING PRIO
              MASK    OCT24100
              CCS     A
              TCF     MARSLEEP

```

```

# Page 1468

```

```

              CS      FLAGWRD4
              MASK    BIT3          # SET MARK OVER NORM
              INHINT
              ADS     FLAGWRD4
              TCF     SETNORM

```

```

MARKPERF     CA      MARKNV
              MASK    VERBMASK
              TCF     NV50DSP

```

```

GODSP        TS      PLAYTEM1

```

```

GODSP2       CAF     ZERO
              TCF     GOFLASH2

```

```

GODSPRET     TS      PLAYTEM1      # ENTRANCE FOR A GODSP WITH A PASTE
              CAF     BIT6          # SET BIT6 TO GO BACK TO USER AFTER NVSUB
              TCF     GOFLASH2

```

```

GODSPR       TS      PLAYTEM1

```

```

GODSPR1      CAF     ZERO
GODSPR2      TS      PLAYTEM4

```

```

              CAF     ZERO          # * DON'T MOVE
              TCF     GODSPRS1

```

```

# CLEANDSP IS USED FOR CLEARING OUT A NORMAL DISPLAY THAT IS PRESENTLY ACTIVE OR A NO
# SET UP TO BE STARTED OR RESTARTED.
#
# NORMALLY THE USER WILL NOT NEED TO USE THIS ROUTINE SINCE A NEW NORMAL DISPLAY AUTO

```

```

# OLD DISPLAY.
#
# CALLING SEQUENCE FOR CLEANDSP --
#
#           TC      BANKCALL
#           CADR     CLEANDSP

CLEANDSP    CAF      ZERO
REFFLASH    TS       PLAYTEM1

           CAF      REDOMASK      # FLASH AND PERMIT
           TCF      GOFLASH2

REGODSP     TS       PLAYTEM1

           CAF      BIT14
           TCF      GOFLASH2

# Page 1469

REGODSPR    TS       PLAYTEM1
           CAF      BIT14
           TCF      GODSPR2

CLOCPLAY    TS       PLAYTEM1
           CAF      CLOCKCON
           TCF      GOFLASH2
GOFLASH     TS       PLAYTEM1

           CAF      BIT4          # LEAVE ONLY FLASH BIT SET
GOFLASH2    TS       PLAYTEM4

           TC       SAVELOCS

           RELINT

           TCF      MAKEPLAY      # BRANCH DIRECT WITH NO SEPARATE JOB CALL

PRIODSPR    TS       PLAYTEM1

           CAF      BITS7+4
           TCF      GODSPRS

PRIODSP     TS       PLAYTEM1

SETPRIO     CAF      BITS7+4

```

	TCF	GOFLASH2	
MAKEPRIO	CAF	ZERO	
	TS	COPINDEX	
	TC	LINUSCHR	
	TCF	HIPRIO	# LINUS RETURN
	CA	FLAGWRD4	
	MASK	OCT20100	# IS PRIO IN ENDIDLE OR BUSY
	CCS	A	
	TCF	PRIOBORT	# YES, ABORT
HIPRIO	CA	FLAGWRD4	# MARK ACTIVE
	MASK	OCT40400	
	EXTEND		
	BZF	ASKIFNRM	# NO
SETMARK	CAF	ZERO	
	TCF	JOBXCHS	
ASKIFNRM	CA	FLAGWRD4	# NORMAL ACTIVE
	MASK	OCT10200	# BITS 13+8
	EXTEND		
# Page 1470			
	BZF	OKTOCOPY	# NO
SETNORM	CAF	ONE	
	TCF	JOBXCHS	
OKTOCOPY	TC	COPYNORM	
	TC	WITCHONE	
	TC	JOBWAKE	
	TC	XCHTOEND	
REDOPRIO	CA	TIME1	# SAVE TIME PRIODSP SENT OUT
	TS	PRIOTIME	
KEEPPRIO	CAF	ZERO	# START UP PRIO DISPLAY
	TCF	PRIOPLAY	
MAKEPLAY	CA	PRIORITY	# SAVE USER'S PRIORITY
	MASK	PRI037	
	TS	USERPRIO	

July 12, 2016

367

	CAF	PRI033	# RAISE PRIORITY FOR FAST JOBS AFTER WAKE
	TC	PRI0CHNG	
	CA	PLAYTEM4	# IS IT MARK OR PRIO OR NORM
	MASK	BITS15+7	
	CCS	A	
	TCF	MAKEPRIO	# ITS PRIO
	TCF	IFLEGAL	
	TCF	MAKEMARK	# ITS MARK
IFLEGAL	CAF	TWO	
	TS	COPINDEX	
	TC	LINUSCHR	
	TCF	OKTOPLAY	# LINUS RETURN
	CS	EBANKTEM	
	MASK	BIT4	
	CCS	A	
	TCF	OKTOPLAY	# NO
	CA	FLAGWRD4	# WAS NORM ASLEEP
	MASK	NBUSMASK	# ARE ANY NORMS ASLEEP
	EXTEND		
	BZF	OKTOPLAY	# NO
PRIOBORT	TC	P00D00	
	OCT	1502	
# Page 1471			
OKTOPLAY	TC	COPIES2	
	CA	USERPRIO	
	EXTEND		
	ROR	SUPERBNK	
	TS	RESTREG	
	CA	FLAGWRD4	# PRIO OR MARK GOING
	MASK	PMMASK	
	CCS	A	
	TCF	GOSLEEPS	# MARK GOING
	TCF	+2	
	TCF	GOSLEEPS	
#	COULD PUT NORM BUSY CHECK HERE TO SAVE TIME		

	TC	WITCHONE	# IS IT NVSUB BUSY, ENDIDLE OR NOONE
	TC	JOBWAKE	
	TC	XCHTOEND	
PLAYJUM1	CAF	TWO	
PRIOPLAY	TS	COPINDEX	
	TCF	GOPLAY	
EXDSPRET	TS	PLAYTEM1	
	CAF	BIT15+6	
	TCF	GOFLASH2	
GOPERF1	TS	NORMTEM1	# STORE DESIRED CHECKLIST VALUE
	CAF	V01N25	# USED TO DISPLAY CHECKLIST VALUE IN R1
GOPERFS	TS	PLAYTEM1	
	CAF	PERFMASK	# LEAVE ONLY FLASH, PERFORM, BLANKING
	TCF	GOFLASH2	
GOPERF2	TS	PLAYTEM1	# DESIRED VERB-NOUN TO DISPLAY R1,R2,R3
	CAF	PERF2MSK	
	TCF	GOFLASH2	
GOPERF4	TC	PURRS4	
	TCF	GOFLASH2	
GOFLASHR	TS	PLAYTEM1	
# Page 1472	CAF	BIT4	# LEAVE ONLY FLASH BIT SET
GODSPRS	TS	PLAYTEM4	
	CAF	THREE	
GODSPRS1	INHINT		# IMMEDIATE RETURN IS CALL CADR +4
	TS	RUPTREG3	
	CA	PRIORITY	# MAKE DISPLAY ONE HIGHER THAN USER
	MASK	PRI037	
	TS	NEWPRIO	

	CA	PLAYTEM4	# IS THIS A FLASHING R DISPLAY
	MASK	BIT4	
	CCS	A	
	TCF	VACDSP	# YES, MAKE DSPLAY JOB A VAC
	CA	NEWPRIO	# NO, MAKE DSPLAY JOB A NOVAC
	TC	NOVAC	
	EBANK=	WHOCARES	
	2CADR	MAKEPLAY	
	TCF	BOTHJOBS	
VACDSP	CA	BBANK	
	EXTEND		
	ROR	SUPERBNK	
	TS	L	
	CAF	MAKEGEN	
	TC	SPVAC	
BOTHJOBS	TC	SAVELOCS	# COPY TEMPS INTO PERMANENT REGISTERS
	EXTEND		# SAVE NVWORD AND USER'S MPAC +2
	DCA	MPAC +1	
	INDEX	LOCCTR	
	DXCH	MPAC +1	
	EXTEND		# SAVE USER'S CADR, FLAGS AND EBANK
	DCA	MPAC +3	
	INDEX	LOCCTR	
	DXCH	MPAC +3	
	CA	LOCCTR	
	TS	MPAC +5	
	TC	SAVELOCR	
	RELINT		
	TCF	BANKJUMP	# CALL CADR +4
# Page 1473			
GOPERF1R	TS	NORMTEM1	# DESIRED CHECKLIST VALUE
	CAF	V01N25	# DISPLAYS CHECKLIST VALUE IN R1
GOPERFRS	TS	PLAYTEM1	
	CAF	PERFMASK	# LEAVE ONLY FLASH, PERFORM, BLANKING
	TCF	GODSPRS	

GOPERF2R	TS	PLAYTEM1	# DESIRED VERB-NOUN TO DISPLAY R1,R2,R3
	CAF	PERF2MSK	
	TCF	GODSPRS	
GOPERF4R	TC	PURRS4	
	TCF	GODSPRS	
PURRS4	TS	OPTION1	# DESIRED OPTION CODE
	CAF	V04N06	
	TS	PLAYTEM1	
	CAF	PERF4MSK	# FLASH, PERFORM AND EBANK R3
	TC	Q	
SAVELOCS	INHINT		
	CS	OCT3400	# EBANK BITS
	MASK	PLAYTEM4	
	AD	EBANK	
	TS	PLAYTEM4	
SAVELOCR	LXCH	Q	
	TC	MAKECADR	
	TS	PLAYTEM3	
	AD	RUPTREG3	# NOT USED FOR NON R ROUTINES
	TC	L	
COPYNORM	CAF	ZERO	
COPIES	TS	COPINDEX	
COPIES2	INHINT		
	CA	PLAYTEM4	# FLAGWORD
	INDEX	COPINDEX	
	TS	EBANKSAV	# EQUIV TO DSPFLG
	MASK	CADRMASK	# FLASH AND GODSPRET
	EXTEND		
# Page 1474	BZF	SKIPADD	
	CA	PLAYTEM3	

July 12, 2016

371

	INDEX	COPINDEX	
	TS	CADRFLSH	
SKIPADD	CA	PLAYTEM1	# VERB NOUN
	INDEX	COPINDEX	
	TS	NVWORD	
	TCF	RELINTQ	
GOSLEEPS	INDEX	COPINDEX	
	CA	PRIOOCT	
	MASK	WAITMASK	
	TC	UPENT2	
WAITMASK	OCT	3004	
	CS	ONE	
	AD	COPINDEX	
	TS	FACEREG	
XCHSLEEP	INDEX	FACEREG	
	CAF	WAKECADR	
	INHINT		
	TC	JOBWAKE	# FIND CADR IN JOB AREA
	TC	XCHTOEND	# CAUSES AWAKENED JOB TO GO TO ENDOFJOB
	INDEX	FACEREG	# REPLACE SAME CADR BUT NEW JOB AREA
	CAF	WAKECADR	
	TCF	JOBSLEEP	
JOBXCHS	TS	FACEREG	# CONTROLS TYPE OF DISPLAY PUT TO SLEEP
	TC	WITCHONE	
	TC	JOBWAKE	
	CA	FACEREG	
	INDEX	LOCCTR	
	TS	FACEREG	
	CAF	XCHQADD	
	TC	XCHNYLOC	
	INDEX	FACEREG	
	CA	MARKOCT	
	MASK	IDLESLEP	
	TC	DOWNENT2	
IDLEMASK	OCT	74004	# * DON'T MOVE
	INDEX	FACEREG	# BIT SHOWS PRIO INTERRUPTED NORM OR MARK

```

# Page 1475
XCHQADD    CA    BIT5          # BIT5 FOR MARK, BIT4 FOR NORMAL
           AD    FOUR
           TC    UPENT2        # FLAG ROUTINE DOES RELINT
           GENADR XCHSLEEP     # * DON'T MOVE
           CA    FLAGWRD4
           MASK  BIT3          # MARK OVER NORM?
           CCS   A
GENMARK     TC    MARKPLAY     # USED AS GENADR FOR JOBWAKE
           TCF   OKTOCOPY

MARKWAKE    CAF   ZERO
WAKEPLAY    TS    TEMPOR2

           INDEX  TEMPOR2
           CA    BITS5+11
           AD    FOUR
           TC    DOWNENT2
MARKFMSK    OCT   40010        # *** DON'T MOVE

           INDEX  TEMPOR2
           CAF   WAKECADR
           INHINT
           TC    JOBWAKE

           TCF   ENDRET

```

```

# ALL .1 RESTARTS BRANCH DIRECTLY TO INITDSP.  NORMAL DISPLAYS ARE THE ONLY DISPLAYS
# INITDSP FIRST RESTORES THE EBANK AND THE SUPERBANK TO THE MOST RECENT NORMAL EBANK
#
# IF THE MOST RECENT NORMAL DISPLAY REQUEST WAS NOT FINISHED, CONTROL IS SENT BACK TO
# OTHERWISE THE NORMAL DISPLAY SET UP IN THE NORMAL DISPLAY REGS IS STARTED UP IMMEDIATELY

```

```

INITDSP     CA    EBANKTEM     # RESTORE MOST RECENT NORMAL EBANK
           TS    EBANK

           CA    RESTREG       # SUPERBANK AND JOB PRIORITY
           TC    SUPERSW       # RESTORE SUPERBANK

           MASK  PRI037
           TC    PRIOCHNG

           CS    THREE
           AD    TEMPFLSH
           TCF   BANKJUMP

```

July 12, 2016

373

```
PINBRNCH      RELINT      # FOR GOPIN USERS
               CA      MARK2PAC  # NEEDED TO SAVE MPAC +2 FOR MARK USERS
               TS      MPAC +2   # ONLY

               CA      FLAGWRD4  # PINBRANCH CONDITION
               MASK     PINMASK
               CCS      A

# Page 1476    TCF      +3
               TCF      ERASER   # ** NOTHING IN ENDIDLE
               TCF      MARKPLAY

NORMBNCH      TC      UPFLAG     # SET PINBRANCH BIT
               ADRES   PINBRFLG

               CAF      BIT14     # PRIO INTERRUPTED
               MASK     FLAGWRD4
               CCS      A
               TCF      KEEPPRIO

               TCF      PLAYJUM1

NVDSP         TC      COPYPACS

               CA      TEMPOR2    # SET UP BLANK BITS FOR NVMONOPT IN CASE
               MASK     SEVEN      # USER REQUESTS BLANKING MONITOR
               TS      L

               CS      BIT13
               INDEX   COPINDEX
               MASK     DSPFLG
               INDEX   COPINDEX
               TS      DSPFLG

               MASK     BIT8       # BIT8 SET IF DEC MARK PERFORM DISPLAY
               TS      TEM1

               CA      MPAC +2
               TS      MPAC2SAV

               TS      MARK2PAC    # * FOR DISK ONLY *
               INDEX   COPINDEX
               CCS      NVWORD
               TCF      NVDSP1
               TCF      CLEANEND
               CS      MARKNV
```

	TS	MARKNV	# IN CASE MARKPLAY AWAKENED AFTER SLEEPING
	MASK	LOW7	
	AD	VO5NOOM1	
	AD	TEM1	
NVDSP1	AD	ONE	
NV50DSP	TC	NVMONOPT	
	TCF	REST	# IF BUSY
	TC	FLASHOFF	# IN CASE OF EXTENDED VERB NON-FLASH
	TC	COPYTOGO	# MPACS DESTROYED BY NVSUB
	TC	DOWNFLAG	# UNSET SLEEPING BITS
	ADRES	MRKNVFLG	
# Page 1477	TC	DOWNFLAG	
	ADRES	NRMNVFLG	
	TC	DOWNFLAG	
	ADRES	PRONVFLG	
BLANKCHK	CA	TEMPOR2	# BLANK BITS 1,2,3 IF SET
	TC	BLANKSUB	
	TCF	NVDSP	
PERFCHEK	CAF	BIT5	# BIT 5 FOR PERFORM
	MASK	TEMPOR2	
	CCS	A	# IS THIS A GOPERF DISPLAY
	TCF	1STOR2ND	# YES
GOANIDLE	CAF	BIT4	
	MASK	TEMPOR2	
	CCS	A	
	TCF	FLASHSUB	# IT IS
	CS	TEMPOR2	# IS THIS A GODSPRET
	MASK	BIT6	
	CCS	A	
	TCF	ISITN00	
	INDEX	COPINDEX	
	CA	CADRFLSH	
	TS	MPAC +3	
	TCF	ENDIT	
ISITN00	INDEX	COPINDEX	# IS THIS A PASTE
	CA	NVWORD	
	MASK	LOW7	# CHECK MADE FOR PINBRNCH AND PRIO ON MARK
	EXTEND		
	BZF	FLASHSUB	# YES, ASSUME PASTE ALWAYS ON FLASH

July 12, 2016

375

	TCF	ENDOFJOB	# NOT FLASH, NOT GOPERF, THEREFORE EXIT
1STOR2ND	CA	TEMPOR2	
	MASK	BIT13	
	CCS	A	
	TCF	GOANIDLE	# SECOND
	CA	BIT13	
	INDEX	COPINDEX	
	ADS	DSPFLG	
	ZL		
	EXTEND		# IS IT MARK
	BZMF	MARKPERF	# YES
	MASK	BIT12	
	EXTEND		
# Page 1478	BZF	V50PASTE	
	CS	NVWORD1	# NVWORD1= -0 IS V97. NVWORD1= -400 IS V99
	AD	V97N00	
	TCF	NV50DSP	
V50PASTE	CAF	V50N00	
	TCF	NV50DSP	# DISPLAY SECOND PART OF GOPERF
WITCHONE	CS	BIT5	# TURN OFF KEY RELEASE LIGHT
	EXTEND		
	WAND	DSALMOUT	
	CA	FLAGWRD4	
	MASK	NVBUSMSK	# IS IT NVSUB ALEEP
	CCS	A	
	CAF	ONE	
	TS	L	
	CAF	ZERO	
	INDEX	L	
	XCH	CADRSTOR	
	INHINT		
	TC	Q	
XCHTOEND	CAF	ENDINST	# TC ENDOFJOB REPLACES GENADR IN LOC FOR
XCHNYLOC	XCH	LOCCTR	# WAS THIS ADDRESS SLEEPING
	EXTEND		
	BZMF	RELINTQ	# NO
	XCH	LOCCTR	# YES

	INDEX	LOCCTR	
	TS	LOC	
RELINTQ	RELINT		
	TC	Q	# BACK TO USER
CLEANEND	CAF	PRI032	# ONE LOWER THAN DISPLAYS SLEEPING
	TC	FINDVAC	
	EBANK=	NVSAVE	
	2CADR	JAMTERM	
	TCF	FLASHSUB +1	
ISITPRIO	CA	FLAGWRD4	
	MASK	ITISMASK	# IS PINBRFLG, MARKIDFLG SET
	EXTEND		
	BZF	PRIOBORT	
	TCF	ENDOFJOB	
REST	CCS	CADRSTOR	# IS SOMEONE IN ENDIDLE
	TCF	ENDOFJOB	# YES
# Page 1479			
	TCF	RESTSLEP	
	TCF	ENDOFJOB	
RESTSLEP	CA	GENMASK	# SET NVSLEEP BITS
	MASK	ASTROMSK	
	TC	UPENT2	
OCT24100	OCT	24100	# *** DON'T MOVE
	INDEX	COPINDEX	
	CAF	NVCADR	
	TC	NVSUBUSY	# BUSY OR ABORT IF ILLEGAL
FLASHSUB	TC	FLASHON	
	CA	COPINDEX	# COPINDEX DESTROYED BY ENDIDLE
	TS	COPMPAC	
	CA	GENMASK	
	MASK	IDLEMASK	
	TC	UPENT2	
ITISMASK	OCT	40040	# *** ENDIDLE ALLOW *** DON'T MOVE
	CA	R1SAVE	# IS THIS A REPEAT AND RETURN DISPLAY
	INDEX	COPINDEX	

July 12, 2016

377

	MASK	BIT3	
	CCS	A	
	TCF	UNSETR1	# YES
	CCS	CADRSTOR	# SEE IF SOMEONE ALREADY IN ENDIDLE
	TCF	ISITPRIO	
	TCF	+2	
	TCF	ISITPRIO	
IDLERET1	TC	ENDIDLE	
	TCF	TERMATE	
	TCF	PROCEED	# ENDIDLE RETURNS HERE ON PROCEED
	CS	LOWLOAD	
	AD	MPAC	# VERBREG
	EXTEND		
	DIM	A	
	EXTEND		
	BZF	LOADITIS	# V21 OR V22 OR V23 ON DSKY
OKTOENT	CAF	TWO	
ENDOUT	TS	OUTHERE	
	CA	FLAGWRD4	# CHECK NATURE OF ENDIDLE RETURN
	MASK	OCT60000	
# Page 1480	CCS	A	
	TCF	TIMECHEK	# PRIO ENDIDLE RETURN
	TCF	NORMRET	# NORMAL ENDIDLE RETURN
	TCF	MARKRET	# MARK ENDIDLE RETURN
TIMECHEK	CS	TIME1	
	AD	PRIOTIME	
	CCS	A	
	COM		
	AD	OCT37776	
	AD	ONE	
	AD	-2SEC	
	EXTEND		
	BZMF	KEEPPRIO	
	TCF	NORMRET	
NORMWAKE	CAF	ONE	
	TCF	WAKEPLAY	

ENDRET	CCS	OUTHERE	
	AD	ONE	
	TCF	+2	# NORMAL ENDIDLE EXIT
	TCF	ENDOFJOB	
	INDEX	COPMPAC	
	AD	CADRFLSH	
	TS	MPAC +3	
	CA	GENMASK	# REMOVE ENDIDLE AND PINBRANCH BITS
	MASK	PINIDMSK	
	TC	DOWNT2	
PINIDMSK	OCT	74044	# *** DON'T MOVE
	CS	THREE	# BLANK EVERYTHING EXCEPT MM
	TC	NVSUB	
	TCF	+1	
ENDIT	CA	USERPRIO	# RETURN TO USER'S PRIORITY
	MASK	PRI037	
	TC	PRI0CHNG	
	CA	MPAC +3	
	TCF	BANKJUMP	
UNSETR1	INDEX	COPINDEX	# RESET REPEAT AND RETURN REQUEST
	CS	BIT3	
	MASK	R1SAVE	
	TS	R1SAVE	
	CAF	ZERO	# *** 205 ONLY MARKBRAN USERS IN
	TC	SUPERSW	# SUPERBANK 0
# Page 1481			
-1	CAF	THREE	# RETURN TO USER'S IMMEDIATE RETURN LOC
IMMEDRET	INDEX	COPINDEX	
	AD	CADRFLSH	
	TCF	BANKJUMP	
TERMATE	CAF	ZERO	# ASTRONAUT TERMINATE (V34) RETURNS TO
	TCF	ENDOUT	
LINUSCHR	CS	PLAYTEM4	# IS THIS A LINUS
	MASK	BIT14	
	CCS	A	
	TCF	Q+1	# NO
	CS	PLAYTEM3	# YES, IS IT ALREADY IN ENDIDLE
	INDEX	COPINDEX	
	AD	CADRFLSH	

```

EXTEND
BZF      +2          # YES

TC       Q           # NO
CCS      DSPLOCK     # IS THE ASTRONAUT BUSY
TC       ENDOFJOB     # END THE NEW DISPLAY, IT'S ALREADY ACTIVE
TC       Q

# MORE LOGIC COULD BE INCORPORATED HERE TO MAKE SURE A RECYCLE IS A RECYCLE AND CONVERSELY THAT

PROCEED  CAF      ONE          # ASTRONAUT PROCEED (V33) RETURNS
          TCF      ENDOUT

# LASTPLAY CHECKS TO SEE IF (1) THE LAST NORMAL DISPLAY WAS EITHER INTERRUPTED BY A PRIO OR A M
# COULD ONLY HAPPEN DURING PINBRANCH) OR IF (2) THE LAST NORMAL DISPLAY WAS REQUESTED WHILE A H
# DISPLAY WAS GOING, RESULTING IN THE NORMAL BEING PUT TO SLEEP.
#
# IF EITHER OF THE ABOVE 2 CONDITIONS EXISTS, THE NORMAL DISPLAY IS AWAKENED TO GO TO PLAYJUM1
# UP THE MOST RECENT VALID NORMAL DISPLAY. IF THESE 2 CONDITIONS DO NOT EXIST, CONTROL GOES TO
# STARTED IMMEDIATELY WITH THE ASSUMPTION THAT THE MOST RECENT NORMAL DISPLAY IS ALREADY IN END
# PINBRNCH) OR THAT A RESTART HAS OCCURRED AND THE DISPLAY CAN BE STARTED AS A .1 RESTART.

MARKRET  CS        SIX
          MASK      FLAGWRD4
          INHINT    # *** MAY MOVE DISPLAY FLAGWORD OUT OF
          TS        FLAGWRD4

          RELINT    # INHINT REALM
          TCF      ENDRET

MARKOVER CAF      MINUS1      # RUPTREG2 IS - MEANS ENDOFJOB TO ENDRET
          TS        OUTHERE
          CA        FLAGWRD4   # IS ENDIDFLG SET
          MASK      PRIO30     # IS NORMAL OR PRIO IN ENDIDLE
          CCS      A

# Page 1482
          TCF      NORMBNCH

NORMRET  CA        FLAGWRD4    # IS MARK SLEEPING
          MASK      BITS5+11    # OR WAITING
          CCS      A
          TCF      MARKWAKE

          CA        FLAGWRD4    # NO
          MASK      BITS4+10    # IS NORMAL INTERRUPTED OR WAITING
          CCS      A

```

	TCF	NORMWAKE	# YES
	CA	EBANKTEM	# NO, WAS IT A FLASH REQUEST
	MASK	OCT50	# OR A GODSPRET
	CCS	A	
	TCF	ENDRET	# YES
	CA	NVSAVE	
	EXTEND		
	BZF	ENDRET	
	CAF	PRI015	
	INHINT		
	TC	NOVAC	
	EBANK=	NVWORD	
	2CADR	PLAYJUM1	
	TCF	ENDRET	
MARSLEEP	CA	FLAGWRD4	# IS MARK ALREADY ON
	MASK	BITS5+11	
	CCS	A	
	TCF	ENDOFJOB	# YES
	TCF	GOSLEEPS	
LOADITIS	INDEX	COPMPAC	
	CA	NVWORD	
	MASK	LOW7	
	COM		
	AD	MPAC +1	# NOUNREG
	EXTEND		
	BZF	OKTOENT	# NO, THEN LOAD IS VALID
	TCF	PINBRNCH	# YES, ACCEPT LOAD BUT ASK FOR LAST AGAIN
ERASER	CS	THREE	# BLANK EVERYTHING EXCEPT MM
	TC	NVSUB	
	TCF	ENDOFJOB	
	TCF	ENDOFJOB	
PERFMASK	OCT	0036	# FLASH, PERFORM, BLANK R2 AND R3
# Page 1483			
V01N25	VN	00125	
V06N07	VN	00607	# GOPERF3 VN DISPLAY BEFORE V50
V50N00	VN	5000	
PERF2MSK	OCT	00030	# FLASH, PERFORM
V04N06	VN	00406	
PERF4MSK	OCT	14	# FLASH, BLANK R3

July 12, 2016

381

GOAGIN	EQUALS	PINBRNCH	
REDOMASK	OCT	20010	# BITS 4 AND 14
MARK3MSK	OCT	40230	# MARK, DECIMAL NOUN, PERFORM, FLASH
MARK4MSK	OCT	40036	# MARK, PERFORM, FLASH, BLANK 2 AND 3
NVCADR	CADR	REDOPRIO	
WAKECADR	CADR	MARKPLAY	
	CADR	PLAYJUM1	
OCT3400	OCT	3400	# EBANK MASK
NBUSMASK	OCT	11210	
PMMASK	OCT	66521	
VERBMASK	=	MID7	# (OCT 37600)
V05N00M1	OCT	1177	# V05 MINUS ONE
GOXDSP	EQUALS	GOMARK	
GOXDSPR	EQUALS	GOMARKR	
GOXDSPF	EQUALS	GOMARKF	
GOXDSPFR	EQUALS	GOMARKFR	
ENDEXT	EQUALS	ENDMARK	
MPAC2SAV	EQUALS	BANKSET	
NVBUSMSK	OCT	700	
ASTROMSK	OCT	704	
MPERFMSK	OCT	40030	# BIT 15,5,4 FOR MARK,PERFORM,FLASH
OCT34300	OCT	34300	
BITS15+7	OCT	40100	
BITS7+4	OCT	110	
DSPFLG	EQUALS	EBANKSAV	
MARKFLAG	EQUALS	MARKEBAN	
SAVEFLAG	EQUALS	EBANKTEM	
BITS5+11	OCT	2020	# * DON'T MOVE
BITS4+10	OCT	1010	# * DON'T MOVE
LOWLOAD	DEC	22	
BUSYMASK	OCT	77730	
CADRMASK	OCT	50	
PINMASK	EQUALS	13,14,15	
GOPLAY	EQUALS	NVDSP	
PRIOSAVE	EQUALS	R1SAVE	
COPMPAC	EQUALS	MPAC +3	
TEMPOR2	EQUALS	MPAC +4	
OUTHERE	EQUALS	MPAC +5	
COPINDEX	EQUALS	LOC	
USERPRIO	EQUALS	MODE	
GENMASK	EQUALS	MPAC +6	
PRIOOCT	OCT	20144	# PRIO
MARKOCT	OCT	42424	# MARK
# Page 1484			
	OCT	11254	# NORM

IDLESLEP	OCT	74704	
OCT67777	OCT	67777	
LINUS	EQUALS	BLANKET	
FACEREG	EQUALS	MPAC	
PLAYTEM1	EQUALS	MPAC +1	
PLAYTEM3	EQUALS	MPAC +3	
PLAYTEM4	EQUALS	MPAC +4	
OCT40420	OCT	40420	
MAKEGEN	GENADR	MAKEPLAY	
OCT10200	OCT	10200	
V97N00	VN	09700	# PASTE FOR V97 OR V99
OCT20100	OCT	20100	
CLOCKCON	OCT	24030	

This code is written to file `src/DISPLAY-INTERFACE-ROUTINES.s`.

B.24 DOWNLINK LISTS

383

<src/DOWNLINK-LISTS.s 383>≡

```
# Copyright:   Public domain.
# Filename:    DOWNLINK_LISTS.agc
# Purpose:    Part of the source code for Comanche, build 055. It
#             is part of the source code for the Command Module's
#             (CM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:  yaYUL
# Reference:   pp. 170-180
# Contact:    Ron Burkey <info@sandroid.org>,
#             Fabrizio Bernardini <fabrizio@spacecraft.it>
# Website:    http://www.ibiblio.org/apollo.
# Mod history: 10/05/09 FB      Transcription of Batch FB-1 Assignment.
#
# The contents of the "Comanche055" files, in general, are transcribed
# from scanned documents.
#
# Assemble revision 055 of AGC program Comanche by NASA
# 2021113-051. April 1, 1969.
#
# This AGC program shall also be referred to as Colossus 2A
#
# Prepared by
#             Massachussets Institute of Technology
#             75 Cambridge Parkway
#             Cambridge, Massachusetts
#
# under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information. Please report any errors to info@sandroid.org.
```

Page 170

```
BANK      22
SETLOC    DOWNTELM
BANK
```

```
EBANK=    DNTMBUFF
```

SPECIAL DOWNLINK OP CODES

#	OP CODE	ADDRESS (EXAMPLE)	SENDS...	BIT 15	BITS 14
#	-----	-----	-----	-----	-----
#	1DNADR	TIME2	(2 AGC WDS)	0	0
#	2DNADR	TEPHEM	(4 AGC WDS)	0	1
#	3DNADR	VGBODY	(6 AGC WDS)	0	2

```

#      4DNADR      STATE      (8 AGC WDS)      0
#      5DNADR      UPBUFF     (10 AGC WDS)     0
#      6DNADR      DSPTAB     (12 AGC WDS)     0
#      DNCHAN      30         CHANNELS        0
#
#      DNPTR       NEXTLIST   POINTS TO NEXT   0
#                               LIST
#
# DOWNLIST FORMAT DEFINITIONS AND RULES --
# 1. END OF A LIST = -XDNADR (X = 1 TO 6), -DNPTR, OR -DNCHAN.
# 2. SNAPSHOT SUBLIST = LIST WHICH STARTS WITH A -1DNADR.
# 3. SNAPSHOT SUBLIST CAN ONLY CONTAIN 1DNADRS.
# 4. TIME2 1DNADR MUST BE LOCATED IN THE CONTROL LIST OF A DOWNLIST.
# 5. ERASABLE DOWN TELEMETRY WORDS SHOULD BE GROUPED IN SEQUENTIAL
#     LOCATIONS AS MUCH AS POSSIBLE TO SAVE STORAGE USED BY DOWNLINK LISTS.
# 6. THE DOWNLINK LISTS (INCLUDING SUBLISTS) ARE ORGANIZED SUCH THAT THE ITEMS LISTED
#     SENT FIRST. EXCEPTION--- SNAPSHOT SUBLISTS. IN THE SNAPSHOT SUBLISTS THE DATA
#     11 1DNADRS IS PRESERVED (IN ORDER) IN DNTMBUFF AND SENT BY THE NEXT 11 DOWNRUPTS
#     LIST IS SENT IMMEDIATELY.

```

```

COUNT 05/DLIST
ERASZERO EQUALS 7
SPARE EQUALS ERASZERO # USE SPARE TO INDICATE AVAIL
LOWIDCOD OCT 77340 # LOW ID CODE

NOMDNLST EQUALS CMCSTADL # FRESH START AND POST P27 DO
UPDNLST EQUALS CMENRDL # UPDATE PROGRAM (P27) DOWNL

```

Page 171

CSM POWERED FLIGHT DOWNLIST

#

----- CONTROL LIST -----

```

CMPOWEDL EQUALS
DNPTR CMPOWE01 # COLLECT SNAPSHOT
6DNADR DNTMBUFF # SEND SNAPSHOT
DNPTR CMPOWE02 # COLLECT SECOND SNAPSHOT
4DNADR DNTMBUFF # SEND SNAPSHOT
DNPTR CMPOWE03 # COMMON DATA
1DNADR TIG # TIG,+1
1DNADR DELLT4 # DELLT4,+1
3DNADR RTARG # RTARG,+1,+2,...+5
1DNADR TGO # TGO,+1
1DNADR PIPTIME1 # PIPTIME1,+1
3DNADR DELV # DELV,+1,...,+4,+5
1DNADR PACTOFF # PACTOFF,YACTOFF

```


July 12, 2016

385

1DNADR	PCMD	#	PCMD,YCMD	
1DNADR	CSTEER	#	CSTEER,+1	
3DNADR	DELVEET1	#	CSI DELTA VELOCITY COMPONENTS	(31-3
6DNADR	REFSMMAT	#	REFSMMAT,+1,...+10,+11	
DNPTR	CMPOWE04	#	COMMON DATA	
1DNADR	TIME2	#	TIME2,TIME1	
DNPTR	CMPOWE05	#	COMMON DATA	
6DNADR	DNTMBUFF	#	SEND SNAPSHOT	
DNPTR	CMPOWE02	#	COLLECT SNAPSHOT	
4DNADR	DNTMBUFF	#	SEND SNAPSHOT	
DNPTR	CMPOWE03	#		
DNPTR	CMPOWE06	#	COMMON DATA	
1DNADR	ELEV	#	ELEV,+1	
1DNADR	CENTANG	#	CENTANG,+1	
1DNADR	DELTAR	#	DELTAR,+1	
1DNADR	STATE +10D	#	FLAGWORDS 10 AND 11	
1DNADR	TEVENT	#	TEVENT,+1	
1DNADR	PCMD	#	PCMD,YCMD	
1DNADR	OPTMODES	#	OPTMODES,HOLDFLAG	
DNPTR	CMPOWE07	#	COMMON DATA	
3DNADR	VGITIG	#	VGITIG,+1,...,+4,+5	
-3DNADR	DELVEET2	#	CDH DELTA VELOCITY COMPONENTS	(98-1

----- SUB LISTS -----

CMPOWE01	-1DNADR RN +2	# RN +2,+3	SNAPSHO
	1DNADR RN +4	# RN +4,+5	
	1DNADR VN	# VN, +1	
	1DNADR VN +2	# VN +2,+3	
	1DNADR VN +4	# VN +4,+5	
	1DNADR PIPTIME	# PIPTIME, +1	
	-1DNADR RN	# RN, +1	
CMPOWE02	-1DNADR CDUZ	# CDUZ,CDUT	SNAPSHO
# Page 172			
	1DNADR ADOT	# ADOT,+1/OGARATE,+1	
	1DNADR ADOT +2	# ADOT+2,+3/OMEGAB+2,+3	
	1DNADR ADOT +4	# ADOT+4,+5/OMEGAB+4,+5	
	-1DNADR CDUX	# CDUX,CDUY	
CMPOWE03	2DNADR AK	# AK,AK1,AK2,RCSFLAGS	COMMON
	-2DNADR THETADX	# THETADX,THETADY,THETADZ,GARBAGE	
CMPOWE04	5DNADR STATE	# FLAGWRD0 THRU FLAGWRD9	COMMON
	-6DNADR DSPTAB	# DISPLAY TABLES	

CMPOWE05	-1DNADR R-OTHER +2	# R-OTHER+2,+3
	1DNADR R-OTHER +4	# R-OTHER+4,+5
	1DNADR V-OTHER	# V-OTHER,+1
	1DNADR V-OTHER +2	# V-OTHER+2,+3
	1DNADR V-OTHER +4	# V-OTHER+4,+5
	1DNADR T-OTHER	# T-OTHER,+1
	-1DNADR R-OTHER	# R-OTHER,+1
CMPOWE06	1DNADR RSBBQ	# RSBBQ,+1
	3DNADR CADRFLSH	# CADRFLSH,+1,+2,FAILREG,+1,-
	-2DNADR CDUS	# CDUS,PIPAX,PIPAY,PIPAZ
CMPOWE07	1DNADR LEMMASS	# LEMMASS,CSMMASS
	1DNADR DAPDATR1	# DAPDATR1,DAPDATR2
	2DNADR ERRORX	# ERRORX,ERRORY,ERRORZ,GARBA
	3DNADR WBODY	# WBODY,...+5/OMEGAC,...+5
	2DNADR REDOCTR	# REDOCTR,THETAD,+1,+2
	1DNADR IMODES30	# IMODES30,IMODES33
	DNCHAN 11	# CHANNELS 11,12
	DNCHAN 13	# CHANNELS 13,14
	DNCHAN 30	# CHANNELS 30,31
	-DNCHAN 32	# CHANNELS 32,33

Page 173

CSM COAST AND ALIGNMENT DOWNLIST

----- CONTROL LIST -----

CMCSTADL	EQUALS	# SEND ID BY SPECIAL CODING
	DNPTR CMCSTAO1	# COLLECT SNAPSHOT
	6DNADR DNTMBUFF	# SEND SNAPSHOT
	DNPTR CMCSTAO2	# COLLECT SECOND SNAPSHOT
	4DNADR DNTMBUFF	# SEND SNAPSHOT
	DNPTR CMCSTAO3	# COMMON DATA
	1DNADR TIG	# TIG,+1
	1DNADR BESTI	# BESTI,BESTJ
	4DNADR MARKDOWN	# MARKDOWN,+1,...+5,+6,GARBA
	4DNADR MARK2DWN	# MARK2DWN,+1,...+5,+6
	2DNADR HAPOX	# APOGEE AND PERIGEE FROM R30
	1DNADR PACTOFF	# PACTOFF, YACTOFF
	3DNADR VGTIG	# VGTIG,...+5
	6DNADR REFSMMAT	# REFSMMAT,+1,...+10,+11
	DNPTR CMCSTAO4	# COMMON DATA
	1DNADR TIME2	# TIME2,TIME1
	DNPTR CMCSTAO5	# COLLECT SNAPSHOT

July 12, 2016

387

6DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMCSTA02	# COLLECT SNAPSHOT
4DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMCSTA03	# COMMON DATA
DNPTR	CMCSTA06	# COMMON DATA
3DNADR	OGC	# OGC,+1,IGC,+1,MGC,+1
1DNADR	STATE +10D	# FLAGWRDS 10 AND 11
1DNADR	TEVENT	# TEVENT,+1
1DNADR	LAUNCHAZ	# LAUNCHAZ,+1
1DNADR	OPTMODES	# OPTMODES,HOLDFLAG
DNPTR	CMCSTA07	# COMMON DATA
-6DNADR	DSPTAB	# DISPLAY TABLES

----- SUB LISTS -----

CMCSTA01	EQUALS	CMPOWE01	# COMMON DOWNLIST DATA
CMCSTA02	EQUALS	CMPOWE02	# COMMON DOWNLIST DATA
CMCSTA03	EQUALS	CMPOWE03	# COMMON DOWNLIST DATA
CMCSTA04	EQUALS	CMPOWE04	# COMMON DOWNLIST DATA
CMCSTA05	EQUALS	CMPOWE05	# COMMON DOWNLIST DATA
CMCSTA06	EQUALS	CMPOWE06	# COMMON DOWNLIST DATA
CMCSTA07	EQUALS	CMPOWE07	# COMMON DOWNLIST DATA

Page 174

Page 175

CSM RENDEZVOUS AND PRETHRUST LIST

----- CONTROL LIST -----

CMRENDDL	EQUALS	# SEND ID BY SPECIAL CODING
DNPTR	CMREND01	# COLLECT SNAPSHOT
6DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMREND02	# COLLECT SECOND SNAPSHOT
4DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMREND03	# COMMON DATA
1DNADR	TIG	# TIG,+1
1DNADR	DELLT4	# DELLT4,+1
3DNADR	RTARG	# RTARG,+1,...+4,+5
1DNADR	VHFTIME	# VHFTIME,+1

4DNADR	MARKDOWN	# MARKTIME(DP) , YCDU , SCDU , ZCDU
1DNADR	VHFCNT	# VHFCNT , +1
1DNADR	TTPI	# TTPI , +1
1DNADR	ECSTEER	# ECSTEER , +1
1DNADR	DELVTPI	# DELVTPI , +1
2DNADR	TCDH	# CDH AND CSI TIME
1DNADR	TPASS4	# TPASS4 , +1
3DNADR	DELVSLV	# DELVSLV , +1 . . . +4 , +5
2DNADR	RANGE	# RANGE , +1 , RRATE , +1
DNPTR	CMREND04	# COMMON DATA
1DNADR	TIME2	# TIME2 , TIME1
DNPTR	CMREND05	# COLLECT SNAPSHOT
6DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMREND02	# COLLECT SNAPSHOT
4DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMREND03	# COMMON DATA
DNPTR	CMREND06	# COMMON DATA
1DNADR	DIFFALT	# CDH DELTA ALTITUDE
1DNADR	CENTANG	# CENTANG , +1
1DNADR	DELTAR	# DELTAR , +1
3DNADR	DELVEET3	# DELVEET3 , +1 , . . . , +4 , +5
1DNADR	OPTMODES	# OPTMODES , HOLDFLAG
DNPTR	CMREND07	# COMMON DATA
1DNADR	RTHETA	# RTHETA , +1
2DNADR	LAT(SPL)	# LAT(SPL) , LNG(SPL) , +1
2DNADR	VPRED	# VPRED , +1 , GAMMAEI , +1
-1DNADR	STATE +10D	# FLAGWRDS 10 AND 11

----- SUB LISTS -----

CMREND01	EQUALS	CMPOWE01	# COMMON DOWNLIST DATA
CMREND02	EQUALS	CMPOWE02	# COMMON DOWNLIST DATA
CMREND03	EQUALS	CMPOWE03	# COMMON DOWNLIST DATA
CMREND04	EQUALS	CMPOWE04	# COMMON DOWNLIST DATA
# Page 176			
CMREND05	EQUALS	CMPOWE05	# COMMON DOWNLIST DATA
CMREND06	EQUALS	CMPOWE06	# COMMON DOWNLIST DATA
CMREND07	EQUALS	CMPOWE07	# COMMON DOWNLIST DATA

Page 177

CSM ENTRY AND UPDATE DOWNLIST

----- CONTROL LIST -----

CMENTRDL	EQUALS	# SEND ID BY SPECIAL CODING
	DNPTR CMENTRO1	# COLLECT SNAPSHOT
	6DNADR DNTMBUFF	# SEND SNAPSHOT
	DNPTR CMENTRO2	# COLLECT SNAPSHOT
	4DNADR DNTMBUFF	# SEND SNAPSHOT
	DNPTR CMENTRO3	# COMMON DATA
	2DNADR CMDAPMOD	# CMDAPMOD,PREL,QREL,RREL
	1DNADR L/D1	# L/D1,+1
	6DNADR UPBUFF	# UPBUFF,+1,...+10,+11
	4DNADR UPBUFF +12D	# UPBUFF+12,13,...+18,+19D
	2DNADR COMPNUMB	# COMPNUMB,UPOLDMOD,UPVERB,UPCOUNT
	1DNADR PAXERR1	# PAXERR1,ROLLTM
	3DNADR LATANG	# LATANG,+1,RDOT,+1,THETAH,+1
	2DNADR LAT(SPL)	# LAT(SPL),+1,LNG(SPL),+1
	1DNADR ALFA/180	# ALFA/180,BETA/180
	DNPTR CMENTRO4	# COMMON DATA
	1DNADR TIME2	# TIME2,TIME1
	DNPTR CMENTRO5	# COLLECT SNAPSHOT
	6DNADR DNTMBUFF	# SEND SNAPSHOT
	DNPTR CMENTRO2	# COLLECT SNAPSHOT
	4DNADR DNTMBUFF	# SENT SNAPSHOT
	2DNADR AK	# AK,AK1,AK2,RCSFLAGS
	3DNADR ERRORX	# ERRORX/Y/Z,THETADX/Y/Z
	2DNADR CMDAPMOD	# CMDAPMOD,PREL,QREL,RREL
	6DNADR UPBUFF	# UPBUFF+0,+1,...+10,+11D
	4DNADR UPBUFF +12D	# UPBUFF+12,+13,...+18,+19D
	1DNADR LEMMASS	# LEMMASS,CSMASS
	1DNADR DAPDATR1	# DAPDATR1,DAPDATR2
	1DNADR ROLLTM	# ROLLTM,ROLLC
	1DNADR OPTMODES	# OPTMODES,HOLDFLAG
	3DNADR WBODY	# WBODY,...+5/OMEGAC,...+5
	2DNADR REDOCTR	# REDOCTR,THETAD+0,+1,+2
	1DNADR IMODES30	# IMODES30,IMODES33
	DNCHAN 11	# CHANNELS 11,12
	DNCHAN 13	# CHANNELS 13,14
	DNCHAN 30	# CHANNELS 30,31
	DNCHAN 32	# CHANNELS 32,33
	1DNADR RSBBQ	# RSBBQ,+1
	3DNADR CADRFLSH	# CADRFLSH,+1,+2,FAILREG,+1,+2
	1DNADR STATE +10D	# FLAGWRDS 10 AND 11
	-1DNADR GAMMAEI	# GAMMAEI,+1

----- SUB LISTS -----

CMENTR01	EQUALS	CMPOWE01	# COMMON DOWNLIST DATA
# Page 178			
CMENTR02	EQUALS	CMPOWE02	# COMMON DOWNLIST DATA
CMENTR03	EQUALS	CMPOWE03	# COMMON DOWNLIST DATA
CMENTR04	EQUALS	CMPOWE04	# COMMON DOWNLIST DATA
CMENTR05	-1DNADR	DELV	# DELV,+1
	1DNADR	DELV +2	# DELV+2,+3
	1DNADR	DELV +4	# DELV+4,+5
	1DNADR	TTE	# TTE,+1
	1DNADR	VIO	# VIO,+1
	1DNADR	VPRED	# VPRED,+1
	-1DNADR	PIPTIME1	# PIPTIME1,+1
CMENTR07	EQUALS	CMPOWE07	# COMMON DOWNLIST DATA

Page 179

----- CONTROL LIST -----

CMPG22DL	EQUALS		# SEND ID BY SPECIAL CODING
	DNPTR	CMPG2201	# COLLECT SNAPSHOT
	6DNADR	DNTMBUFF	# SEND SNAPSHOT
	DNPTR	CMPG2202	# COLLECT SNAPSHOT
	4DNADR	DNTMBUFF	# SEND SNAPSHOT
	DNPTR	CMPG2203	# COMMON DATA
	6DNADR	SVMRKDAT	# LANDING SITE MARK DATA
	6DNADR	SVMRKDAT +12D	# SVMRKDAT+0,...+34
	6DNADR	SVMRKDAT +24D	# LANDING SITE MARK DATA
	1DNADR	LANDMARK	# LANDMARK,GARBAGE
	1DNADR	SPARE	
	1DNADR	SPARE	
	1DNADR	SPARE	
	DNPTR	CMPG2204	# COMMON DATA
	1DNADR	TIME2	# TIME2,TIME1
	DNPTR	CMPG2205	# COLLECT SNAPSHOT
	2DNADR	DNTMBUFF	# SEND SNAPSHOT
	1DNADR	SPARE	
	1DNADR	SPARE	
	1DNADR	SPARE	
	1DNADR	SPARE	

July 12, 2016

391

DNPTR	CMPG2202	# COLLECT SNAPSHOT
4DNADR	DNTMBUFF	# SEND SNAPSHOT
DNPTR	CMPG2203	# COMMON DATA
DNPTR	CMPG2206	# COMMON DATA
1DNADR	8NN	# 8NN,GARBAGE
1DNADR	STATE +10D	# FLAGWRDS 10 AND 11
3DNADR	RLS	# RLS,+1,...+4,+5
1DNADR	SPARE	
1DNADR	OPTMODES	# OPTMODES,HOLDFLAG
DNPTR	CMPG2207	# COMMON DATA
1DNADR	SPARE	
1DNADR	SPARE	
1DNADR	SPARE	
1DNADR	SPARE	
1DNADR	SPARE	
-1DNADR	SPARE	

----- SUB LISTS -----

CMPG2201	EQUALS	CMPOWE01	# COMMON DOWNLIST DATA
CMPG2202	EQUALS	CMPOWE02	# COMMON DOWNLIST DATA
CMPG2203	EQUALS	CMPOWE03	# COMMON DOWNLIST DATA
# Page 180			
CMPG2204	EQUALS	CMPOWE04	# COMMON DOWNLIST DATA
CMPG2205	-1DNADR	LONG	# LONG,+1
	1DNADR	ALT	# ALT,+1
	-1DNADR	LAT	# LAT,+1
CMPG2206	EQUALS	CMPOWE06	# COMMON DOWNLIST DATA
CMPG2207	EQUALS	CMPOWE07	# COMMON DOWNLIST DATA

DNTABLE	GENADR	CMCSTADL
	GENADR	CMENTRDL
	GENADR	CMRENDDL
	GENADR	CMPOWEDL
	GENADR	CMPG22DL

This code is written to file src/DOWNLINK-LISTS.s.

B.25 DOWN TELEMETRY PROGRAM

```

392  <src/DOWN-TELEMETRY-PROGRAM.s 392>≡
# Copyright:      Public domain.
# Filename:       DOWN-TELEMETRY-PROGRAM.agc
# Purpose:        Part of the source code for Luminary 1A build 099.
#                It is part of the source code for the Lunar Module's (LM)
#                Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:     yaYUL
# Contact:        Ron Burkey <info@sandroid.org>.
# Website:        www.ibiblio.org/apollo.
# Pages:          988-997
# Mod history:    2009-05-24 RSB   Adapted from the corresponding
#                                Luminary131 file, using page
#                                images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum. The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum. Many thanks to both. The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo. If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
#
# Page 988
# PROGRAM NAME -- DOWN TELEMETRY PROGRAM
# MOD NO. -- 0          TO COMPLETELY REWRITE THE DOWN TELEMETRY PROGRAM AND DOWNLINK
#                      PURPOSE OF SAVING APPROXIMATELY 150 WORDS OF CORE STORAGE.
#                      THIS CHANGE REQUIRES AN ENTIRELY NEW METHOD OF SPECIFYING DOWN
#                      LINKS LOG SECTION FOR MORE DETAILS. HOWEVER THIS CHANGE WILL
#                      OF DOWN TELEMETRY DATA.
# MOD BY -- KILROY, SMITH, DEWITT
# DATE -- 02 OCT 67
# AUTHORS -- KILROY, SMITH, DEWITT, DEWOLF, FAGIN
# LOG SECTION -- DOWN-TELEMETRY PROGRAM
#
# FUNCTIONAL DESCRIPTION -- THIS ROUTINE IS INITIATED BY TELEMETRY END
# PULSE FROM THE DOWNLINK TELEMETRY CONVERTER. THIS PULSE OCCURS
# AT 50 TIMES PER SEC (EVERY 20 MS) THEREFORE DODOWNTM IS

```



```

#       EXECUTED AT THESE RATES.  THIS ROUTINE SELECTS THE APPROPRIATE
#       AGC DATA TO BE TRANSMITTED DOWNLINK AND LOADS IT INTO OUTPUT
#       CHANNELS 34 AND 35.  THE INFORMATION IS THEN GATED OUT FROM THE
#       LGC IN SERIAL FASHION.
#
#       THIS PROGRAM IS CODED FOR A 2 SECOND DOWNLIST.  SINCE DOWNRUPTS
#       OCCUR EVERY 20 MS AND 2 AGC COMPUTER WORDS CAN BE PLACED IN
#       CHANNELS 34 AND 35 DURING EACH DOWNRUPT THE PROGRAM IS CAPABLE
#       OF SENDING 200 AGC WORDS EVERY 2 SECONDS.
#
# CALLING SEQUENCE -- NONE
#       PROGRAM IS ENTERED VIA TCF DODOWNTM WHICH IS EXECUTED AS A
#       RESULT OF A DOWNRUPT.  CONTROL IS RETURNED VIA TCF RESUME WHICH
#       IN EFFECT IS A RESUME.
#
# SUBROUTINES CALLED -- NONE
#
# NORMAL EXIT MODE -- TCF RESUME
#
# ALARM OR ABORT EXIT MODE -- NONE
#
# RESTART PROTECTION:
#       ON A FRESH START AND RESTART THE 'STARTSUB' SUBROUTINE WILL INITIALIZE THE DOWNLIST POINT
#       DNTMGOTO) TO THE BEGINNING OF THE CURRENT DOWNLIST (I.E., CURRENT CONTENTS OF DNLSTADR)
#       EFFECT OF IGNORING THE REMAINDER OF THE DOWNLIST WHICH THE DOWN-TELEMETRY PROGRAM WAS W
#       THE RESTART (OR FRESH START) OCCURRED AND RESUME DOWN TELEMETRY FROM THE BEGINNING OF T
#       DOWNLIST.
#
#       ALSO OF INTEREST IS THE FACT THAT ON A RESTART THE AGC WILL ZERO DOWNLINK CHANNELS 13,
#
# DOWNLINK LIST SELECTION:
#       THE APPROPRIATE DOWNLINK LISTS ARE SELECTED BY THE FOLLOWING:
#       1.      FRESH START
#       2.      V37EXXE WHERE XX = THE MAJOR MODE BEING SELECTED.
#       3.      UPDATE PROGRAM (P27)
#       4.      NON-V37 SELECTABLE TYPE PROGRAMS (E.G., AGS INITIALIZATION (SUNDANCE, LUMINARY)
#               TRANSITION (COLOSSUS) ETC.).
#
# DOWNLINK LIST RULES AND LIMITATIONS:
#       READ SECTION(S) WHICH FOLLOW 'DEBRIS' WRITEUP.
#
# OUTPUT -- EVERY 2 SECONDS 100 DOUBLE PRECISION WORDS (I.E., 200 LGC
#       COMPUTER WORDS) ARE TRANSMITTED VIA DOWNLINK.
#
# ERASABLE INITIALIZATION REQUIRED -- NONE
#       'DNTMGOTO' AND 'DNLSTADR' ARE INITIALIZED BY THE FRESH START PROGRAM.

```

```

#
# DEBRIS (ERASABLE LOCATIONS DESTROYED BY THIS PROGRAM) --
#       LDATA1ST, DNTMBUFF TO DNTMBUFF +21D, TMINDEX, DNQ.
# Page 989 (empty page)
# Page 990
# DODOWNTM IS ENTERED EVERY 20 MS BY AN INTERRUPT TRIGGERED BY THE
# RECEIPT OF AN ENDPULSE FROM THE SPACECRAFT TELEMETRY PROGRAMMER.
#
# NOTES REGARDING DOWNLINK LISTS ASSOCIATED WITH THIS PROGRAM:
# 1.   DOWNLISTS. DOWNLISTS MUST BE COMPILED IN THE SAME BANK AS THE
#       DOWN-TELEMETRY PROGRAM. THIS IS DONE FOR EASE OF CODING, FASTER
#       EXECUTION.
# 2.   EACH DOWNLINK LIST CONSISTS OF A CONTROL LIST AND A NUMBER OF
#       SUBLISTS.
# 3.   A SUBLIST REFERS TO A SNAPSHOT OR DATA COMMON TO THE SAME OR OTHER
#       DOWNLINK LISTS. ANY SUBLIST CONTAINING COMMON DATA NEEDS TO BE
#       CODED ONLY ONCE FOR THE APPLICABLE DOWNLINK LISTS.
# 4.   SNAPSHOT SUBLISTS REFER SPECIFICALLY TO HOMOGENEOUS DATA WHICH MUST BE
#       SAVED IN A BUFFER DURING ONE DOWNRUPT.
# 5.   THE 1DNADR FOR THE 1ST WORD OF SNAPSHOT DATA IS FOUND AT THE END
#       OF EACH SNAPSHOT SUBLIST, SINCE THE PROGRAM CODING SENDS THIS DP WORD
#       IMMEDIATELY AFTER STORING THE OTHERS IN THE SNAPSHOT BUFFER.
# 6.   ALL LISTS ARE COMBINATIONS OF CODED ERASABLE ADDRESS CONSTANTS
#       CREATED FOR THE DOWNLIST PROGRAM.
#       A.      1DNADR              1-WORD DOWNLIST ADDRESS.
#               SAME AS ECADR, BUT USED WHEN THE WORD ADDRESSED IS THE LEFT
#               HALF OF A DOUBLE-PRECISION WORD FOR DOWN TELEMETRY.
#       B.      2DNADR - 6DNADR      N-WORD DOWNLIST ADDRESS, N = 2 - 6.
#               SAME AS 1DNADR, BUT WITH THE 4 UNUSED BITS OF THE ECADR FORMAT
#               FILLED IN WITH 0001-0101. USED TO POINT TO A LIST OF N DOUBLE-
#               PRECISION WORDS, STORED CONSECUTIVELY, FOR DOWN TELEMETRY.
#       C.      DNCHAN              DOWNLIST CHANNEL ADDRESS.
#               SAME AS 1DNADR, BUT WITH PREFIX BITS 0111. USED TO POINT TO
#               A PAIR OF CHANNELS FOR DOWN TELEMETRY.
#       D.      DNPTR              DOWN-TELEMETRY SUBLIST POINTER.
#               SAME AS CAF BUT TAGGED AS A CONSTANT. USED IN CONTROL LIST TO POINT
#               CAUTION --- A DNPTR CANNOT BE USED IN A SUBLIST.
# 7.   THE WORD ORDER CODE IS SET TO ZERO AT THE BEGINNING OF EACH DOWNLIST (I.E.,
#       A '1DNADR TIME2' IS DETECTED IN THE CONTROL LIST (ONLY).
# 8.   IN THE SNAPSHOT SUBLIST ONLY, THE DNADR'S CANNOT POINT TO THE FIRST WORD OF A
#
# DOWNLIST LIST RESTRICTIONS:
# (THE FOLLOWING POINTS MAY BE LISTED ELSEWHERE BUT ARE LISTED HERE SO IT IS CLEAR TH
# DONE)
# 1.   SNAPSHOT DOWNLIST:
#       (A) CANNOT CONTAIN THE FOLLOWING ECADRS (I.E., 1DNADR'S): Q, 400, 1000, 1400.

```

```

#      (B) CAN CONTAIN ONLY 1DNADR'S
# 2.    ALL DOWNLINKED DATA (EXCEPT CHANNELS) IS PICKED UP BY A DCA SO DOWNLINK LISTS CANNOT CO
#      EQUIVALENT OF THE FOLLOWING ECADRS (I.E., 1DNADRS): 377, 777, 1377, 1777, 2377, 2777, 3
#      (NOTE: THE TERM 'EQUIVALENT' MEANT THAT THE 1DNADR TO 6DNADR WILL BE PROCESSED LIKE 1 T
# 3.    CONTROL LISTS AND SUBLISTS CANNOT HAVE ENTRIES = OCTAL 00000 OR OCTAL 77777
# Page 991
# 4.    THE '1DNADR TIME2' WHICH WILL CAUSE THE DOWNLINK PROGRAM TO SET THE WORDER CODE TO 3 MU
#      CONTROL SECTION OF THE DOWNLIST.
# 5.    'DNCHAN 0' CANNOT BE USED.
# 6.    'DNPTR 0' CANNOT BE USED.
# 7.    DNPTR CANNOT APPEAR IN A SUBLIST.
#
# EBANK SETTINGS
#      IN THE PROCESS OF SETTING THE EBANK (WHEN PICKING UP DOWNLINK DATA) THE DOWN TELEMETRY
#      'GARBAGE' INTO BITS15-12 OF EBANK.  HUGH BLAIR-SMITH WARNS US THAT BITS15-12 OF EBANK M
#      SIGNIFICANT SOMEDAY IN THE FUTURE.  IF/WHEN THAT HAPPENS, THE PROGRAM SHOULD INSURE (BY
#      THAT BITS 15-12 OF EBANK ARE ZERO.
#
#      INITIALIZATION REQUIRED -- TO INTERRUPT CURRENT LIST AND START A NEW ONE.
#          1. ADRES OF DOWNLINK LIST INTO DNLSTADR
#          2. NEGONE INTO SUBLIST
#          3. NEGONE INTO DNECADR

```

	BANK	22	
	SETLOC	DOWNTELM	
	BANK		
	EBANK=	DNTMBUFF	
	COUNT*	\$\$/DPROG	
DODOWNTM	TS	BANKRUPT	
	EXTEND		
	QXCH	QRUPT	# SAVE Q
	TCF	WOTEST	
W01	EXTEND		# SET WORD ORDER BIT TO 1 ONLY IF IT
	WOR	CHAN13	# ALREADY ISN'T
	TC	DNTMGOTO	# GOTO APPROPRIATE PHASE OF PROGRAM
DNPHASE1	CA	NEGONE	# INITIALIZE ALL CONTROL WORDS
	TS	SUBLIST	# WORDS TO MINUS ONE
	TS	DNECADR	
	CA	LDNPHAS2	# SET DNTMGOTO = 0 ALL SUSEQUENT DOWRUPTS
	TS	DNTMGOTO	# GO TO DNPHASE2
	TCF	NEWLIST	
DNPHASE2	CCS	DNECADR	# SENDING OF DATA IN PROGRESS
DODNADR	TC	FETCH2WD	# YES -- THEN FETCH THE NEXT 2 SP WORDS

```

MINTIME2      -1DNADR TIME2      # NEGATIVE OF TIME2 1DNADR
TCF           +1                  # (ECADR OF 3776 + 74001 = 77777)

                                CCS   SUBLIST      # IS THE SUBLIST IN CONTROL
                                TCF   NEXTINSL     # YES

# Page 992
DNADRDCR      OCT   74001        # DNADR COUNT AND ECADR DECREMENTER

CHKLIST       CA      CTLIST
EXTEND
BZMF          NEWLIST            # IT WILL BE NEGATIVE AT END OF LIST
TCF           NEXTINCL
NEWLIST       INDEX  DNLSTCOD
CA            DNTABLE            # INITIALIZE CTLIST WITH
TS            CTLIST             # STARTING ADDRESS OF NEW LIST
CS            DNLSTCOD
TCF           SENDID  +3
NEXTINCL      INDEX  CTLIST
CA            0
CCS           A
INCR          CTLIST            # SET POINTER TO PICK UP NEXT CTLIST WORD
TCF           +4                # ON NEXT ENTRY TO PROG. (A SHOULD NOT =0)
XCH           CTLIST            # SET CTLIST TO NEGATIVE AND PLACE(CODING)
COM           UNCOMPLEMENTED DNADR INTO A. (FOR LA)
XCH           CTLIST            # (ST IN )
+4            INCR   A           # (CTLIST)
TS            DNECADR           # SAVE DNADR
AD            MINTIME2          # TEST FOR TIME2 (NEG. OF ECADR)
CCS           A
TCF           SETWO   +1        # DON'T SET WORD ORDER CODE
MINB1314      OCT   47777       # MINUS BIT 13 AND 14 (CAN'T GET HERE)
TCF           SETWO   +1        # DON'T SET WORD ORDER CODE
SETWO         TC      WOZERO     # GO SET WORD ORDER CODE TO ZERO.
+1            CA      DNECADR     # RELOAD A WITH THE DNADR.
+2            AD      MINB1314    # IS THIS A REGULAR DNADR?
EXTEND
BZMF          FETCH2WD          # YES. (A MUST NEVER BE ZERO)
AD            MINB12            # NO. IS IT A POINTER (DNPTR) OR A
EXTEND        CHANNEL(DNCHAN)  # CHANNEL(DNCHAN)
BZMF          DODNPTR           # IT'S A POINTER. (A MUST NEVER BE ZERO)

DODNCHAN      TC      6          # (EXECUTED AS EXTEND) IT'S A CHANNEL
INDEX         DNECADR
INDEX         0      -4000      # (EXECUTED AS READ)
TS            L
TC            6                # (EXECUTED AS EXTEND)

```

```

INDEX  DNECADR
INDEX  0      -4001  # (EXECUTED AS READ)
TS     DNECADR      # SET DNECADR
CA     NEGONE       #          TO MINUS
XCH    DNECADR      #          WHILE PRESERVING A.
TCF    DNTMEXIT     # GO SEND CHANNELS

WOZERO  CS     BIT7
        EXTEND
WAND    CHAN13      # SET WORD ORDER CODE TO ZERO

# Page 993
TC      Q           # RETURN TO CALLER

DODNPTR INDEX  DNECADR      # DNECADR CONTAINS ADRES OF SUBLIST
        0      0          # CLEAR AND ADD LIST ENTRY INTO A.
CCS     A          # IS THIS A SNAPSHOT SUBLIST
CA      DNECADR    # NO, IT IS A REGULAR SUBLIST.
TCF     DOSUBLST   # A MUST NOT BE ZERO.

XCH     DNECADR    # YES. IT IS A SNAPSHOT SUBLIST.
TS      SUBLIST    # C(DNECADR) INTO SUBLIST
CAF     ZERO       #          A      INTO      A
XCH     TMINDEX    # (NOTE: TMINDEX = DNECADR)

# THE FOLLOWING CODING (FROM SNAPLOOP TO SNAPEND) IS FOR THE PURPOSE OF TAKING A SNAPSHOT OF 12
# THIS IS DONE BY SAVING 11 DP REGISTERS IN DNTMBUFF AND SENDING THE FIRST DP WORD IMMEDIATELY.
# THE SNAPSHOT PROCESSING IS THE MOST TIME CONSUMING AND THEREFORE THE CODING AND LIST STRUCTURE
# TO MINIMIZE TIME. THE TIME OPTIMIZATION RESULTS IN RULES UNIQUE TO THE SNAPSHOT PORTION OF THE
# THESE RULES ARE .....
#      1. ONLY 1DNADR'S CAN APPEAR IN THE SNAPSHOT SUBLIST
#      2. THE 1DNADR'S CANNOT REFER TO THE FIRST LOCATION IN ANY BANK.

SNAPLOOP TS     EBANK      # SET EBANK
        MASK    LOW8      # ISOLATE RELATIVE ADDRESS
        EXTEND
        INDEX   A
        EBANK=  1401
        DCA     1401      # PICK UP 2 SNAPSHOT WORDS.
        EBANK=  DNTMBUFF
        INDEX   TMINDEX
        DXCH    DNTMBUFF  # STORE 2 SNAPSHOT WORDS IN BUFFER
        INCR    TMINDEX   # SET BUFFER INDEX FOR NEXT 2 WORDS.
        INCR    TMINDEX
SNAPAGN  INCR    SUBLIST   # SET POINTER TO NEXT 2 WORDS OF SNAPSHOT
        INDEX   SUBLIST
        0      0          # = CA SSSS (SSSS = NEXT ENTRY IN SUBLIST)

```

	CCS	A	# TEST FOR LAST TWO WORDS OF SNAPSHOT.
	TCF	SNAPLOOP	# NOT LAST TWO.
LDNPHAS2	GENADR	DNPHASE2	
	TS	SUBLIST	# YES, LAST. SAVE A.
	CA	NEGONE	# SET DNECADR AND
	TS	DNECADR	# SUBLIST POINTERS
	XCH	SUBLIST	# TO NEGATIVE VALUES
	TS	EBANK	
	MASK	LOW8	
	EXTEND		
	INDEX	A	
	EBANK=	1401	
	DCA	1401	# PICK UP FIRST 2 WORDS OF SNAPSHOT.
# Page 994	EBANK=	DNTMBUFF	
SNAPEND	TCF	DNTMEXIT	# NOW TO SEND THEM.
FETCH2WD	CA	DNECADR	
	TS	EBANK	# SET EBANK
	MASK	LOW8	# ISOLATE RELATIVE ADDRESS
	TS	L	
	CA	DNADRDRCR	# DECREMENT COUNT AND ECADR
	ADS	DNECADR	
	EXTEND		
	INDEX	L	
	EBANK=	1400	
	DCA	1400	# PICK UP 2 DATA WORDS
	EBANK=	DNTMBUFF	
	TCF	DNTMEXIT	# NOW GO SEND THEM.
DOSUBLST	TS	SUBLIST	# SET SUBLIST POINTER
NEXTINSL	INDEX	SUBLIST	
	0	0	# = CA SSSS (SSSS = NEXT ENTRY IN SUBLIST)
	CCS	A	# IS IT THE END OF THE SUBLIST
	INCR	SUBLIST	# NO --
	TCF	+4	
	TS	SUBLIST	# SAVE A.
	CA	NEGONE	# SET SUBLIST TO MINUS
	XCH	SUBLIST	# RETRIEVE A.
+4	INCR	A	
	TS	DNECADR	# SAVE DNADR
	TCF	SETWO +2	# GO USE COMMON CODING (PROLEMS WOULD # OCCUR IF THE PROGRAM ENCOUNTERED A # DNPTR NOW)
DNTMEXIT	EXTEND		# DOWN-TELEMETRY EXIT

July 12, 2016

399

```

                                WRITE  DNTM1          # TO SEND A + L TO CHANNELS 34 + 35
                                CA      L              # RESPECTIVELY
TMEXITL                        EXTEND
                                WRITE  DNTM2
TMRESUME                        TCF      RESUME        # EXIT TELEMTRY PROGRAM VIA RESUME.

MINB12                         EQUALS  -1/8
DNECADR                        EQUALS  TMINDEX
CTLIST                         EQUALS  LDATALST
SUBLIST                        EQUALS  DNQ
```

Page 995

SUBROUTINE NAME -- DNDUMP

#

FUNCTIONAL DESCRIPTION -- TO SEND (DUMP) ALL ERASABLE STORAGE 'N' TIMES. (N=1 TO 4). BANKS A
EACH BANK IS PRECEDED BY AN ID WORD, SYNCH BITS, ECADR AND TIME1 FOLLOWED BY THE 256D W
EBANK. EBANKS ARE DUMPED IN ORDER (I.E., EBANK 0 FIRST, THEN EBANK1 ETC.)

#

CALLING SEQUENCE -- THE GROUND OR ASTRONAUT BY KEYING V74E CAN INITIALIZE THE DUMP.

AFTER KEYING IN V74E THE CURRENT DOWNLIST WILL BE IMMEDIATELY TERMINATED AND THE DOWNLI
WILL BEGIN.

#

ONCE INITITIATED THE DOWNLINK ERASABLE DUMP CAN BE TERMINATED (AND INTERRUPTED DOWNLIST
BY THE FOLLOWING:

#

1. A FRESH START

2. COMPLETION OF ALL DOWNLINK DUMPS REQUESTED (ACCORDING TO BITS SET IN DUMPCNT).
CAN BE ALTERED BY A V21N01.

3. AND INVOLUNTARILY BY A RESTART.

#

NORMAL EXIT MODE -- TCF DNPHEASE1

#

ALARM OR ABORT MODE -- NONE

#

*SUBROUTINES CALLED -- NONE

#

ERASABLE INITIALIZATION REQUIRED --

DUMPCNT OCT 20000 IF 4 COMPLETE ERASABLE DUMPS ARE DESIRED

DUMPCNT OCT 10000 IF 2 COMPLETE ERASABLE DUMPS ARE DESIRED

DUMPCNT OCT 04000 IF 1 COMPLETE ERASABLE DUMP IS DESIRED

#

DEBRIS -- DUMPLOC, DUMPSW, DNTMGOTO, EBANK, AND CENTRAL REGISTERS

#

TIMING -- TIME (IN SECS) = ((NO.DUMPS)*(NO.EBANKS)*(WDSPEREBANK + NO.IDWDS)) / NO.WDSPERS

TIME (IN SECS) = (4)*(8)*(256 + 4) / 100

THUS TIME (IN SECS TO SEND DUMP OF ERASABLE 4 TIMES VIA DOWNLINK) = 83.2 SECONDS

```

#
# STRUCTURE OF ONE EBANK AS IT IS SENT BY DOWNLINK PROGRAM --
# (REMINDER -- THIS ONLY DESCRIBES ONE OF THE 8 EBANKS X 4 (DUMPS) = 32 EBANKS
#
# DOWNLIST
# WORD TAKEN FROM CONTENTS OF EXAMPLE 0 COMMENTS
# 1 ERASID 0177X 0 DOWNLIST I.D. FOR DOWNLINK EBANK
# 2 LOWIDCOD 77340 1 DOWNLINK SYNCH BITS. (SAME COUNTER AS DUMPLOC)
# 3 DUMPLOC 13400 1 (SEE NOTES ON DUMPLOC) 1 = 3RD WORD OF EBANK
# 4 TIME1 14120 1 TIME IN CENTISECONDS
# 5 FIRST WORD OF EBANK X 03400 1 IN THIS EXAMPLE THIS WORD = 0
# 6 2ND WORD OF EBANK X 00142 1 IN THIS EXAMPLE THIS WORD = 0
# 7 3RD WORD OF EBANK X 00142 1 IN THIS EXAMPLE THIS WORD = 0
# .
# .
# .
# 260D 256TH WORD OF EBANK X 03777 1 IN THIS EXAMPLE THIS WORD = 0
#
# NOTE -- DUMPLOC CONTAINS THE COUNTER AND ECADR FOR EACH WORD BEING SENT.
# THE BIT STRUCTURE OF DUMPLOC IS FOLLOW --
# X = NOT USED
# X ABC EEE RRRRRRRR ABC = ERASABLE DUMP COUNTER (I.E. ABC = 0,1,2,3,4,5,6,7)
# COMPLETE ERASABLE DUMP NUMBER 1,2,3,4,5,6,7
# EEE = EBANK BITS
# RRRRRRRR = RELATIVE ADDRESS WITHIN AN EBANK
#
# Page 996
DNDUMPI CA ZERO # INITIALIZE DOWNLINK
TS DUMPLOC # ERASABLE DUMP
+2 TC SENDID # GO SEND ID AND SYNCH BITS
CA LDNDUMP1 # SET DNTMGOTO
TS DNTMGOTO # TO LOCATION FOR NEXT PASS
CA TIME1 # PLACE TIME1
XCH L # INTO L
CA DUMPLOC # AND ECADR OF THIS EBANK INTO A
TCF DNTMEXIT # SEND DUMPLOC AND TIME1

LDNDUMP ADRES DNDUMP
LDNDUMP1 ADRES DNDUMP1

DNDUMP CA TWO # INCREMENT ECADR IN DUMPLOC
ADS DUMPLOC # TO NEXT DP WORD TO BE
MASK LOW8 # DUMPED AND SAVE IT.
CCS A # IS THIS THE BEGINNING OF A NEW EBANK
TCF DNDUMP2 # NO -- THEN CONTINUE DUMPING
CA DUMPLOC # YES -- IS THIS THE END OF THE

```


	MASK	DUMPCNT	# N TH (N = 1 TO 4) COMPLETE ERASABLE
	MASK	PRI034	# DUMP (BIT14 FOR 4, BIT13 FOR 2 OR BIT12
	CCS	A	# FOR 1 COMPLETE ERASABLE DUMP(S)).
	TCF	DNPASE1	# YES -- START SENDING INTERRUPTED DOWNLIST
			# AGAIN
	TCF	DNDUMPI +2	# NO -- GO BACK AND INITIALIZE NEXT BANK
DNDUMP1	CA	LDNDUMP	# SET DNTMGOTO
	TS	DNTMGOTO	# FOR WORDS 3 TO 256D OF CURRENT EBANK
DNDUMP2	CA	DUMPLOC	
	TS	EBANK	# SET EBANK
	MASK	LOW8	# ISOLATE RELATIVE ADDRESS.
	TS	Q	# (NOTE: MASK INSTRUCTION IS USED TO PICK
	CA	NEGO	# UP ERASABLE REGISTERS SO THAT EDITING
	TS	L	# REGISTERS 20-23 WILL NOT BE ALTERED.)
	INDEX	Q	
	EBANK=	1400	# PICK UP LOW ORDER REGISTER OF PAIR
	MASK	1401	# OF ERASABLE REGISTERS.
	XCH	L	
	INDEX	Q	# PICK UP HIGH ORDER REGISTER OF PAIR
	MASK	1400	# OF ERASABLE REGISTERS.
	EBANK=	DNTMBUFF	
	TCF	DNTMEXIT	# GO SEND THEM
SENDID	EXTEND		# ** ENTRANCE USED BY ERASABLE DUMP PROG. **
	QXCH	DNTMGOTO	# SET DNTMGOTO SO NEXT TIME PROG WILL GO
	CAF	ERASID	# TO LOCATION FOLLOWING 'TC SENDID'
	TS	L	# ** ENTRANCE USED BY REGULAR DOWNLINK PG **
# Page 997	TC	WOZERO	# GO SET WORD ORDER CODE TO ZERO
	CAF	LOWIDCOD	# PLACE SPECIAL ID CODE INTO L
	XCH	L	# AND ID BACK INTO A
	TCF	DNTMEXIT	# SEND DOWNLIST ID CODE(S).
WOTEST	CA	BIT7	# AT THE BEGINNING OF THE LIST THE WORD
	EXTEND		# ORDER BIT WILL BE SET BACK TO ZERO
	RAND	CHAN13	
	CCS	A	
	TC	DNTMGOTO	
	CA	BIT7	
	TCF	WO1	

B.26 DOWN-TELEMETRY PROGRAM

```

402  <src/DOWN-TELEMETRY-PROGRAM.s 402>≡
      # Copyright:    Public domain.
      # Filename:     DOWN-TELEMETRY_PROGRAM.agc
      # Purpose:      Part of the source code for Comanche, build 055. It
      #               is part of the source code for the Command Module's
      #               (CM) Apollo Guidance Computer (AGC), Apollo 11.
      # Assembler:    yaYUL
      # Reference:     pp. 1093-1102
      # Contact:       Ron Burkey <info@sandroid.org>
      # Website:       http://www.ibiblio.org/apollo.
      # Mod history:   2009-05-08 RSB   Adapted from Colossus249/ file of same name
      #               and page images. Corrected various typos
      #               in the transcription of program comments,
      #               and these should be back-ported to
      #               Colossus249.
      #
      # The contents of the "Comanche055" files, in general, are transcribed
      # from scanned documents.
      #
      # Assemble revision 055 of AGC program Comanche by NASA
      # 2021113-051. April 1, 1969.
      #
      # This AGC program shall also be referred to as Colossus 2A
      #
      # Prepared by
      #
      #               Massachussets Institute of Technology
      #               75 Cambridge Parkway
      #               Cambridge, Massachusetts
      #
      # under NASA contract NAS 9-4065.
      #
      # Refer directly to the online document mentioned above for further
      # information. Please report any errors to info@sandroid.org.
      #
      # Page 1093
      # PROGRAM NAME -- DOWN TELEMETRY PROGRAM
      # MOD NO. -- 0          TO COMPLETELY REWRITE THE DOWN TELEMETRY PROGRAM AND DOWNLINE
      #                       PURPOSE OF SAVING APPROXIMATELY 150 WORDS OF CORE STORAGE.
      #                       THIS CHANGE REQUIRES AN ENTIRELY NEW METHOD OF SPECIFYING DOWN
      #                       LISTS LOG SECTION FOR MORE DETAILS. HOWEVER THIS CHANGES WITH
      #                       OF DOWN TELEMETRY DATA.
      # MOD BY -- KILROY, SMITH, DEWITT
      # DATE -- 02 OCT 67
      # AUTHORS -- KILROY, SMITH, DWITT, DEWOLF, FAGIN

```

```

# LOG SECTION -- DOWN-TELEMETRY PROGRAM
#
# FUNCTIONAL DESCRIPTION -- THIS ROUTINE IS INITIATED BY TELEMETRY END
#     PULSE FROM THE DOWNLINK TELEMETRY CONVERTER.  THIS PULSE OCCURS
#     AT 50 TIMES PER SEC (EVERY 20 MS) THEREFORE DODOWNTM IS
#     EXECUTED AT THESE RATES.  THIS ROUTINE SELECTS THE APPROPRIATE
#     AGC DATA TO BE TRANSMITTED DOWNLINK AND LOADS IT INTO OUTPUT
#     CHANNELS 34 AND 35.  THE INFORMATION IS THEN GATED OUT FROM THE
#     LGC IN SERIAL FASHION.
#
#     THIS PROGRAM IS CODED FOR A 2 SECOND DOWNLIST.  SINCE DOWNRUPTS
#     OCCUR EVERY 20 MS AND 2 AGC COMPUTER WORDS CAN BE PLACED IN
#     CHANNELS 34 AND 35 DURING EACH DOWNRUPT THE PROGRAM IS CAPABLE
#     OF SENDING 200 AGC WORDS EVERY 2 SECONDS.
#
# CALLING SEQUENCE -- NONE
#     PROGRAM IS ENTERED VIA TCF DODOWNTM WHICH IS EXECUTED AS A
#     RESULT OF A DOWNRUPT.  CONTROL IS RETURNED VIA TCF RESUME WHICH
#     IN EFFECT IS A RESUME.
#
# SUBROUTINES CALLED -- NONE
#
# NORMAL EXIT MODE -- TCF RESUME
#
# ALARM OR ABORT EXIT MODE -- NONE
#
# RESTART PROTECTION:
#     ON A FRESH START AND RESTART THE 'STARTSUB' SUBROUTINE WILL INITIALIZE THE DOWNLIST POINT
#     DNTMGOTO) TO THE BEGINNING OF THE CURRENT DOWNLIST (I.E., CURRENT CONTENTS OF DNLSTADR)
#     EFFECT OF IGNORING THE REMAINDER OF THE DOWNLIST WHICH THE DOWN-TELEMETRY PROGRAM WAS W
#     THE RESTART (OR FRESH START) OCCURRED AND RESUME DOWN TELEMETRY FROM THE BEGINNING OF T
#     DOWNLIST.
#
#     ALSO OF INTEREST IS THE FACT THAT ON A RESTART THE AGC WILL ZERO DOWNLINK CHANNELS 13,
#
# DOWNLINK LIST SELECTION:
#     THE APPROPRIATE DOWNLINK LISTS ARE SELECTED BY THE FOLLOWING:
#     1.      FRESH START
#     2.      V37EXXE WHERE XX = THE MAJOR MODE BEING SELECTED.
#     3.      UPDATE PROGRAM (P27)
#     4.      NON-V37 SELECTABLE TYPE PROGRAMS (E.G., AGS INITIALIZATION (SUNDANCE, LUMINARY)
#             TRANSITIONS (COLOSSUS) ETC.).
#
# DOWNLINK LIST RULES AND LIMITATIONS:
#     READ SECTION(S) WHICH FOLLOW 'DEBRIS' WRITEUP.
#

```

```

# OUTPUT -- EVERY 2 SECONDS 100 DOUBLE PRECISION WORDS (I.E., 200 LGC
#       COMPUTER WORDS) ARE TRANSMITTED VIA DOWNLINK.
#
# ERASABLE INITIALIZATION REQUIRED -- NONE
#       'DNTMGOTO' AND 'DNLSTADR' ARE INITIALIZED BY THE FRESH START PROGRAM.
#
# DEBRIS (ERASABLE LOCATIONS DESTROYED BY THIS PROGRAM) --
#       LDATA1ST, DNTMBUFF TO DNTMBUFF +21D, TMINDEX, DNQ.
# Page 1094
# (No source on this page of the original assembly listing.)

# Page 1095
# DODOWNTM IS ENTERED EVERY 20 MS BY AN INTERRUPT TRIGGERED BY THE
# RECEIPT OF AN ENDPULSE FROM THE SPACECRAFT TELEMETRY PROGRAMMER.
#
# NOTES REGARDING DOWNLINK LISTS ASSOCIATED WITH THIS PROGRAM:
# 1.  DOWNLISTS.  DOWNLISTS MUST BE COMPILED IN THE SAME BANK AS THE
#       DOWN-TELEMETRY PROGRAM.  THIS IS DONE FOR EASE OF CODING, FASTER
#       EXECUTION.
# 2.  EACH DOWNLINK LIST CONSISTES OF A CONTROL LIST AND A NUMBER OF
#       SUBLISTS.
# 3.  A SUBLIST REFERS TO A SNAPSHOT OR DATA COMMON TO THE SAME OR OTHER
#       DOWNLINK LISTS.  ANY SUBLIST CONTAINING COMMON DATA NEEDS TO BE
#       CODED ONLY ONCE FOR THE APPLICABLE DOWNLINK LISTS.
# 4.  SNAPSHOT SUBLISTS REFER SPECIFICALLY TO HOMOGENEOUS DATA WHICH MUST BE
#       SAVED IN A BUFFER DURING ONE DOWNRUPT.
# 5.  THE 1DNADR FOR THE 1ST WORD OF SNAPSHOT DATA IS FOUND AT THE END
#       OF EACH SNAPSHOT SUBLIST, SINCE THE PROGRAM CODING SENDS THIS DP WORD
#       IMMEDIATELY AFTER STORING THE OTHERS IN THE SNAPSHOT BUFFER.
# 6.  ALL LISTS ARE COMBINATIONS OF CODED ERASABLE ADDRESS CONSTANTS
#       CREATED FOR THE DOWNLIST PROGRAM.
#   A.  1DNADR 1-WORD DOWNLIST ADDRESS.
#         SAME AS ECADR, BUT USED WHEN THE WORD ADDRESSED IS THE LEFT
#         HALF OF A DOUBLE-PRECISION WORD FOR DOWN TELEMETRY.
#   B.  2DNADR - 6DNADR N-WORD DOWNLIST ADDRESS, N = 2 - 6.
#         SAME AS 1DNADR, BUT WITH THE 4 UNUSED BITS OF THE ECADR FORMAT
#         FILLED IN WITH 0001-0101.  USED TO POINT TO A LIST OF N DOUBLE-
#         PRECISION WORDS, STORED CONSECUTIVELY, FOR DOWN TELEMETRY.
#   C.  DNCHAN DOWNLIST CHANNEL ADDRESS.
#         SAME AS 1DNADR, BUT WITH PREFIX BITS 0111.  USED TO POINT TO
#         A PAIR OF CHANNELS FOR DOWN TELEMETRY.
#   D.  DNPTR DOWN-TELEMETRY SUBLIST POINTER.
#         SAME AS CAF BUT TAGGES AS A CONSTANT.  USED IN CONTROL LIST TO POINT
#         CAUTION --- A DNPTR CANNOT BE USED IN A SUBLIST.
# 7.  THE WORD ORDER CODE IS SET TO ZERO AT THE BEGINNING OF EACH DOWNLIST (I.E., C
#       A '1DNADR TIME2' IS DETECTED IN THE CONTROL LIST (ONLY).

```

```

# 8.    IN THE SNAPSHOT SUBLIST ONLY, THE DNADR'S CANNOT POINT TO THE FIRST WORD OF ANY EBANK.
#
# DOWNLIST LIST RESTRICTIONS:
# (THE FOLLOWING POINTS MAY BE LISTED ELSEWHERE BUT ARE LISTED HERE SO IT IS CLEAR THAT THESE T
# DONE)
# 1.    SNAPSHOT DOWNLIST:
#        (A) CANNOT CONTAIN THE FOLLOWING ECADRS (I.E., 1DNADR'S): Q, 400, 1000, 1400, 2000, 240
#        (B) CAN CONTAIN ONLY 1DNADR'S
# 2.    ALL DOWNLINKED DATA (EXCEPT CHANNELS) IS PICKED UP BY A DCA SO DOWNLINK LISTS CANNOT CO
#        EQUIVALENT OF THE FOLLOWING ECADRS (I.E., IDNADRS): 377, 777, 1377, 1777, 2377, 2777, 3
#        (NOTE: TE TERM 'EQUIVALENT' MEANT THAT THE IDNADR TO 6DNADR WILL BE PROCESSED LIKE 1 TO
# 3.    CONTROL LISTS AND SUBLISTS CANNOT HAVE ENTRIES = OCTAL 00000 OR OCTAL 77777
# Page 1096
# 4.    THE '1DNADR TIME2' WHICH WILL CAUSE THE DOWNLINT PROGRAM TO SET THE WORDER CODE TO 3 MU
#        CONTROL SECTION OF THE DOWNLIST.
# 5.    'DNCHAN 0' CANNOT BE USED.
# 6.    'DNPTR 0' CANNOT BE USED.
# 7.    DNPTR CANNOT APPEAR IN A SUBLIST.
#
# EBANK SETTINGS
# IN THE PROCESS OF SETTING THE EBANK (WHEN PICKING UP DOWNLINK DATA) THE DOWN TELEMETRY
# 'GARBAGE' INTO BITS15-12 OF EBANK.  HUGH BLAIR-SMITH WARNS US THAT BITS15-12 OF EBANK M
# SIGNIFICANT SOMEDAY IN THE FUTURE.  IF/WHEN THAT HAPPENS, THE PROGRAM SHOULD INSURE (BY
# THAT BITS 15-12 OF EBANK ARE ZERO.
#
# INITIALIZATION REQUIRED -- TO INTERRUPT CURRENT LIST AND START A NEW ONE.
#     1. ADRES OF DOWNLINK LIST INTO DNLSTADR
#     2. NEGONE INTO SUBLIST
#     3. NEGONE INTO DNECADR
#
# BANK      22
# SETLOC    DOWNTELM
# BANK
#
# EBANK=    DNTMBUFF
#
# COUNT     05/DPROG
#
# DODOWNTM  TS      BANKRUPT
#           EXTEND
#           QXCH    QRUPT      # SAVE Q
#           CA      BIT7      # SET WORD ORDER CODE TO 1.  EXCEPTION: AT
#           EXTEND      # THE BEGINNING OF EACH LIST THE WORD
#           WOR      CHAN13    # CODE WILL BE SET BACK TO 0.
#           TC       DNTMGOTO  # GOTO APPROPRIATE PHASE OF PROGRAM

```

DNPHASE1	CA	NEGONE	# INITIALIZE ALL CONTROL WORDS
	TS	SUBLIST	# WORDS TO MINUS ONE
	TS	DNECADR	
	CA	LDNPHAS2	# SET DNTMGOTO = 0 ALL SUSEQUENT DOWRUPTS
	TS	DNTMGOTO	# GO TO DNPHEASE2
	TCF	NEWLIST	
DNPHASE2	CCS	DNECADR	# SENDING OF DATA IN PROGRESS
DODNADR	TC	FETCH2WD	# YES -- THEN FETCH THE NEXT 2 SP WORDS
MINTIME2	-1DNADR	TIME2	# NEGATIVE OF TIME2 1DNADR
	TCF	+1	# (ECADR OF 3776 + 74001 = 77777)
	CCS	SUBLIST	# IS THE SUBLIST IN CONTROL
# Page 1097	TCF	NEXTINSL	# YES
DNADRDCR	OCT	74001	# DNADR COUNT AND ECADR DECREMENTER
CHKLIST	CA	CTLIST	
	EXTEND		
	BZMF	NEWLIST	# IT WILL BE NEGATIVE AT END OF LIST
	TCF	NEXTINCL	
NEWLIST	INDEX	DNLSTCOD	
	CA	DNTABLE	# INITIALIZE CTLIST WITH
	TS	CTLIST	# STARTING ADDRESS OF NEW LIST
	CS	DNLSTCOD	
	TCF	SENDID +3	
NEXTINCL	INDEX	CTLIST	
	CA	0	
	CCS	A	
	INCR	CTLIST	# SET POINTER TO PICK UP NEXT CTLIST WORD
	TCF	+4	# ON NEXT ENTRY TO PROG. (A SHOULD NOT =0)
	XCH	CTLIST	# SET CTLIST TO NEGATIVE AND PLACE(CODING)
	COM		# UNCOMPLEMENTED DNADR INTO A. (FOR LA)
	XCH	CTLIST	# (ST IN)
+4	INCR	A	# (CTLIST)
	TS	DNECADR	# SAVE DNADR
	AD	MINTIME2	# TEST FOR TIME2 (NEG. OF ECADR)
	CCS	A	
	TCF	SETWO +1	# DON'T SET WORD ORDER CODE
MINB1314	OCT	47777	# MINUS BIT 13 AND 14 (CAN'T GET HERE)
	TCF	SETWO +1	# DON'T SET WORD ORDER CODE
SETWO	TC	WOZERO	# GO SET WORD ORDER CODE TO ZERO.
+1	CA	DNECADR	# RELOAD A WITH THE DNADR.
+2	AD	MINB1314	# IS THIS A REGULAR DNADR?
	EXTEND		
	BZMF	FETCH2WD	# YES. (A MUST NEVER BE ZERO)
	AD	MINB12	# NO. IS IT A POINTER (DNPTR) OR A

407

```
WOZERO      CS      BIT7
            EXTEND
```

```
WAND    CHAN13      # SET WORD ORDER CODE TO ZERO
TC      Q           # RETURN TO CALLER
```

```
# THE FOLLOWING CODING (FROM SNAPLOOP TO SNAPEND) IS FOR THE PURPOSE OF TAKING A SNAPSHOT OF 12
# THIS IS DONE BY SAVING 11 DP REGISTERS IN DNTMBUFF AND SENDING THE FIRST DP WORD IMMEDIATELY.
# THE SNAPSHOT PROCESSING IS THE MOST TIME CONSUMING AND THEREFORE THE CODING AND LIST STRUCTURE
# TO MINIMIZE TIME.  THE TIME OPTIMIZATION RESULTS IN RULES UNIQUE TO THE SNAPSHOT PORTION OF T
# THESE RULES ARE .....
#      1.      ONLY 1DNADR'S CAN APPEAR IN THE SNAPSHOT SUBLIST
#      2.      THE 1DNADR'S CANNOT REFER TO THE FIRST LOCATION IN ANY BANK.
```

SNAPLOOP	TS	EBANK	# SET EBANK
	MASK	LOW8	# ISOLATE RELATIVE ADDRESS
	EXTEND		
	INDEX	A	
	EBANK=	1401	
	DCA	1401	# PICK UP 2 SNAPSHOT WORDS.

	EBANK=	DNTMBUFF	
	INDEX	TMINDEX	
	DXCH	DNTMBUFF	# STORE 2 SNAPSHOT WORDS IN BUFFER
	INCR	TMINDEX	# SET BUFFER INDEX FOR NEXT 2 WORDS.
	INCR	TMINDEX	
SNAPAGN	INCR	SUBLIST	# SET POINTER TO NEXT 2 WORDS OF SNAPSHOT
	INDEX	SUBLIST	
	0	0	# = CA SSSS (SSSS = NEXT ENTRY IN SUBLIST)
	CCS	A	# TEST FOR LAST TWO WORDS OF SNAPSHOT.
	TCF	SNAPLOOP	# NOT LAST TWO.
LDNPHAS2	GENADR	DNPHASE2	
	TS	SUBLIST	# YES, LAST. SAVE A.
	CA	NEGONE	# SET DNECADR AND
	TS	DNECADR	# SUBLIST POINTERS
	XCH	SUBLIST	# TO NEGATIVE VALUES
	TS	EBANK	
	MASK	LOW8	
	EXTEND		
	INDEX	A	
	EBANK=	1401	
# Page 1099			
	DCA	1401	# PICK UP FIRST 2 WORDS OF SNAPSHOT.
	EBANK=	DNTMBUFF	
SNAPEND	TCF	DNTMEXIT	# NOW GO SEND THEM.
FETCH2WD	CA	DNECADR	
	TS	EBANK	# SET EBANK
	MASK	LOW8	# ISOLATE RELATIVE ADDRESS
	TS	L	
	CA	DNADRD CR	# DECREMENT COUNT AND ECADR
	ADS	DNECADR	
	EXTEND		
	INDEX	L	
	EBANK=	1400	
	DCA	1400	# PICK UP 2 DATA WORDS
	EBANK=	DNTMBUFF	
	TCF	DNTMEXIT	# NOW GO SEND THEM.
DOSUBLST	TS	SUBLIST	# SET SUBLIST POINTER
NEXTINSL	INDEX	SUBLIST	
	0	0	# = CA SSSS (SSSS = NEXT ENTRY IN SUBLIST)
	CCS	A	# IS IT THE END OF THE SUBLIST
	INCR	SUBLIST	# NO --
	TCF	+4	
	TS	SUBLIST	# SAVE A.
	CA	NEGONE	# SET SUBLIST TO MINUS

July 12, 2016

409

```

+4      XCH      SUBLIST      # RETRIEVE A.
        INCR     A
        TS       DNECADR      # SAVE DNADR
        TCF      SETWO +2     # GO USE COMMON CODING (PROBLEMS WOULD
                                # OCCUR IF THE PROGRAM ENCOUNTERED A
                                # DNPTR NOW)

DNTMEXIT      EXTEND          # DOWN-TELEMETRY EXIT
              WRITE  DNTM1     # TO SEND A + L TO CHANNELS 34 + 35
              CA      L        # RESPECTIVELY

TMEXITL       EXTEND
              WRITE  DNTM2
TMRESUME      TCF      RESUME  # EXIT TELEMTRY PROGRAM VIA RESUME.

MINB12        EQUALS  -1/8
DNECADR       EQUALS  TMINDEX
CTLIST        EQUALS  LDATA1ST
SUBLIST       EQUALS  DNQ
```

Page 1100

SUBROUTINE NAME -- DNDUMP

#

FUNCTIONAL DESCRIPTION -- TO SEND (DUMP) ALL 8 BANKS OF ERASABLE STORAGE TWICE. BANKS ARE SE

EACH BANK IS PRECEDED BY AN ID WORD, SYNCH BITS, ECADR AND TIME1 FOLLOWED BY THE 256D W

EBANK. EBANKS ARE DUMPED IN ORDER (I.E., EBANK 0 FIRST, THEN EBANK1 ETC.)

#

CALLING SEQUENCE -- THE GROUND OR ASTRONAUT BY KEYING V74E CAN INITIALIZE THE DUMP.

AFTER KEYING IN V74E THE CURRENT DOWNLIST WILL BE IMMEDIATELY TERMINATED AND THE DOWNLI

WILL BEGIN.

#

ONCE INITITIATED THE DOWNLINK ERASABLE DUMP CAN BE TERMINATED (AND INTERRUPTED DOWNLIST

BY THE FOLLOWING:

#

1. A FRESH START

2. COMPLETION OF ALL DOWNLINK DUMPS REQUESTED (ACCORDING TO BITS SET IN DUMPCNT).

CAN BE ALTERED BY A V21N01.

3. AND INVOLUNTARILY BY A RESTART.

#

NORMAL EXIT MODE -- TCF DNPHASE1

#

ALARM OR ABORT MODE -- NONE

#

*SUBROUTINES CALLED -- NONE

#

ERASABLE INITIALIZATION REQUIRED --

DUMPCNT OCT 20000 IF 4 COMPLETE ERASABLE DUMPS ARE DESIRED

```

#          DUMPCNT OCT 10000          IF 2 COMPLETE ERASABLE DUMPS ARE DESIRED
#          DUMPCNT OCT 04000          IF 1 COMPLETE ERASABLE DUMP  IS  DESIRED
#
# DEBRIS -- DUMPLOC, DUMPSW, DNTMGOTO, EBANK, AND CENTRAL REGISTERS
#
# TIMING --      TIME (IN SECS) = ((NO.DUMPS)*(NO.EBANKS)*(WDSPEREBANK + NO.IDWDS)) /
#                  TIME (IN SECS) = (  4  )*(  8  )*( 256  +  4  ) /
#                  THUS TIME (IN SECS TO SEND DUMP OF ERASABLE 4 TIMES VIA DOWNLINK) = 83.2 S
#
# STRUCTURE OF ONE EBANK AS IT IS SENT BY DOWNLINK PROGRAM --
# (REMINDER -- THIS ONLY DESCRIBES ONE OF THE 8 EBANKS X 4 (DUMPS) = 32 EBANKS
#
# DOWNLIST
#          WORD  TAKEN FROM CONTENTS OF  EXAMPLE 0      COMMENTS
#          1    ERASID                    0177X  0      DOWNLIST I.D. FOR DOWNLINK ER
#          2    LOWIDCOD                  77340  1      DOWNLINK SYNCH BITS. (SAME C
#          3    DUMPLOC                   13400  1      (SEE NOTES ON DUMPLOC) 1 = 3B
#          4    TIME1                     14120  1      TIME IN CENTISECONDS
#          5    FIRST WORD OF EBANK X     03400  1      IN THIS EXAMPLE THIS WORD = C
#          6    2ND  WORD OF EBANK X     00142  1      IN THIS EXAMPLE THIS WORD = C
#          7    3RD  WORD OF EBANK X     00142  1      IN THIS EXAMPLE THIS WORD = C
#          .
#          .
#          .
#          260D  256TH WORD OF EBANK X    03777  1      IN THIS EXAMPLE THIS WORD = C
#
# NOTE --      DUMPLOC CONTAINS THE COUNTER AND ECADR FOR EACH WORD BEING SENT.
#              THE BIT STRUCTURE OF DUMPLOC IS FOLLOW --
#
#              X = NOT USED
#              X ABC EEE RRRRRRRR      ABC = ERASABLE DUMP COUNTER (I.E. ABC =
#              COMPLETE ERASABLE DUMP NUMBER 1,2
#              EEE = EBANK BITS
#              RRRRRRRR = RELATIVE ADDRESS WITHIN AN EBANK
#
# Page 1101
DNDUMPI      CA      ZERO                # INITIALIZE DOWNLINK
              TS      DUMPLOC            # ERASABLE DUMP
              +2     TC      SENDID       # GO SEND ID AND SYNCH BITS
              CA      LDNDUMP1           # SET DNTMGOTO
              TS      DNTMGOTO           # TO LOCATION FOR NEXT PASS
              CA      TIME1              # PLACE TIME1
              XCH     L                  # INTO L
              CA      DUMPLOC            # AND ECADR OF THIS EBANK INTO A
              TCF     DNTMEXIT           # SEND DUMPLOC AND TIME1

LDNDUMP      ADRES  DNDUMP

```

LDNDUMP1	ADRES	DNDUMP1	
DNDUMP	CA	TWO	# INCREMENT ECADR IN DUMPLOC
	ADS	DUMPLOC	# TO NEXT DP WORD TO BE
	MASK	LOW8	# DUMPED AND SAVE IT.
	CCS	A	# IS THIS THE BEGINNING OF A NEW EBANK
	TCF	DNDUMP2	# NO -- THEN CONTINUE DUMPING
	CA	DUMPLOC	# YES -- IS THIS THE END OF THE
	MASK	DUMPCNT	# N-TH(N = 1 TO 4) COMPLETE ERASABLE
	MASK	PRI034	# DUMP(BIT14 FOR 4, BIT13 FOR 2 OR BIT12
	CCS	A	# FOR 1 COMPLETE ERASABLE DUMP(S)).
	TCF	DNPBASE1	# YES -- SEND DOWNLIST AGAIN
			# AGAIN
	TCF	DNDUMPI +2	# NO -- GO BACK AND INITIALZE NEXT BANK
DNDUMP1	CA	LDNDUMP	# SET DNTMGOTO
	TS	DNTMGOTO	# FOR WORDS 3 TO 256D OF CURRENT EBANK
DNDUMP2	CA	DUMPLOC	
	TS	EBANK	# SET EBANK
	MASK	LOW8	# ISOLATE RELATIVE ADDRESS.
	TS	Q	# (NOTE: MASK INSTRUCTION IS USED TO PICK
	CA	NEGO	# UP ERASABLE REGISTERS TO THAT EDITING
	TS	L	# REGISTERS 20-23 WILL NOT BE ALTERED.)
	INDEX	Q	
	EBANK=	1400	# PICK UP LOW ORDER REGISTER OF PAIR
	MASK	1401	# OF ERASABLE REGISTERS.
	XCH	L	
	INDEX	Q	# PICK UP HIGH ORDER REGISTER OF PAIR
	MASK	1400	# OF ERASABLE REGISTERS.
	EBANK=	DNTMBUFF	
	TCF	DNTMEXIT	# GO SEND THEM
SENDID	EXTEND		# ** ENTRANCE USED BY ERASABLE DUMP PROG. **
	QXCH	DNTMGOTO	# SET DNTMGOTO SO NEXT TIME PROG WILL GO
	CAF	ERASID	# TO LOCATION FOLLOWING 'TC SENDID'
	TS	L	# ** ENTRANCE USED BY REGULAR DOWNLINK PG **
# Page 1102	TC	WOZERO	# GO SET WORD ORDER CODE TO ZERO
	CAF	LOWIDCOD	# PLACE SPECIAL ID CODE INTO L
	XCH	L	# AND ID BACK INTO A
	TCF	DNTMEXIT	# SEND DOWNLIST ID CODE(S).

This code is written to file `src/DOWN-TELEMETRY-PROGRAM.s`.

B.27 ENTRY LEXICON

```

413  <src/ENTRY-LEXICON.s 413>≡
# Copyright:   Public domain.
# Filename:    ENTRY_LEXICON.agc
# Purpose:     Part of the source code for Comanche, build 055. It
#              is part of the source code for the Command Module's
#              (CM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:  yaYUL
# Reference:   pp. 837-843
# Contact:     Ron Burkey <info@sandroid.org>
# Website:     http://www.ibiblio.org/apollo.
# Mod history: 2009-05-12 RSB   Adapted from Colossus249 file of the same
#                               name and Comanche 055 page images.
#
# The contents of the "Comanche055" files, in general, are transcribed
# from scanned documents.
#
# Assemble revision 055 of AGC program Comanche by NASA
# 2021113-051. April 1, 1969.
#
# This AGC program shall also be referred to as Colossus 2A
#
# Prepared by
#
#           Massachussets Institute of Technology
#           75 Cambridge Parkway
#           Cambridge, Massachusetts
#
# under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information. Please report any errors to info@sandroid.org.

# Page 837
# VARIABLE      DESCRIPTION                      MAXIMUM VALUE *      COMPUTER NAME
# -----      -
# -
# URTO          INITIAL TARGET VECTOR            2 (UNIT VECTOR)      = RTINIT
# -
# UZ            UNIT VECTOR NORTH                 1                     = UNITW
# -
# V            VELOCITY VECTOR                    2 VSAT                = VEL
# -
# R            POSITION VECTOR                      2 EXP 29 METERS       = RN
# -
# VI           INERTIAL VELOCITY                  128 M/CENTISEC        = VN

```

# _				
# RTE	VECTOR EAST AT INITIAL TARGET	2		= RTE
# _				
# UTR	NORMAL TO RTE AND UZ	2		= RTE
# _				
# URT	TARGET VECTOR	2		= RTE
# _				
# UNI	UNIT NORMAL TO TRAJECTORY PLANE	2		
# _				
# DELV	INTEGRATED ACCEL. FROM PIPAS	5.85 16384 CM/S		
# _				
# G	GRAVITY VECTOR	128 M/CENTISEC		= GDT
#				
# AO	INITIAL DRAG FOR UPCTRL	805 FPSS		FPSS
#				
# AHOOKDV	TERM IN GAMMAL CALC. = AHOOK DVL	16		
#				
# A1	DRAG VALUE IN FACTOR CALCULATION	805 FPSS		
#				
# ALP	CONST FOR UPCTRL	1		
#				
# ASKEP	KEPLER RANGE	21600 NM		NM =
#				
# ASP1	FINAL PHASE RANGE	21600 NM		
#				
# ASPUP	UP-RANGE	21600 NM		
#				
# ASP3	GAMMA CORRECTION	21600 NM		
#				
# ASPDOWN	RANGE DOWN TO PULL-UP	21600 NM		
#				
# ASP	PREDICTED RANGE	21600 NM		NOT S
#				
# COSG	COSINE(GAMMAL)	2		= COS
#				
# C/DO	RECIPROCAL DRAG, -4/DO B-8	64/FPSS		
#				
# D	TOTAL ACCELERATION	805 FPSS		
#				
# DO	CONTROLLED CONSTANT D	805 FPSS		
#				
# DHOOK	TERM IN GAMMAL COMPUTATION	805 FPSS		
#				
# DIFF	THETNM-ASP (RANGE DIFFERENCE)	21600 NM		
#				
# DIFFOLD	PREVIOUS VALUE OF DIFF	21600 NM		

July 12, 2016

415

#				
#	DLEWD	CHANGE IN LEWD	1	
#				
#	DR	REFERENCE DRAG FOR DOWNCONTROL	805 FPSS	NOT STORED
#				
#	DREFR	REFERENCE DRAG	805 FPSS	NOT STORED
#				
#	DVL	VS1-VL	2 VSAT	
#				
#	E	ECCENTRICITY	4	NOT STORED
#				
#	F1	DRANGE/D DRAG (FINAL PHASE)	2700/805	= FX +5
#				
#	F2	DRANGE/D RDOT (FINAL PHASE)	2700/2VS NM/FPS	= FX +4
#	Page 838			
#				
#	F3	DRANGE/D (L/D)	2700 NM	= FX
#				
#	FACT1	CONST FOR UPCONTRL	805 FPSS	
#				
#	FACT2	CONST FOR UPCONTRL	1/805 FPSS	
#				
#	FACTOR	USED IN UPCONTRL	1	* MAXIMUM VALUE
#				VARIABLE VALU
#	GAMMAL	FLIGHT PATH ANGLE AT VL	1 RADIAN	VARIABLE HAS
#				
#	GAMMAL1	SIMPLE FORM OF GAMMAL	1 RADIAN	
#	Page 839			
#				
#	HEADSUP	INDICATOR FOR INITIAL ROLL	1	
#				
#	KA	DRAG TO LIFT UP IF DOWN	805 FPSS	= KAT
#				
#	KLAT	LATERAL SWITCH GAIN	1	(NOM = .0125)
#				
#	K2ROLL	INDICATOR FOR ROLL SWITCH		
#				
#	LAD	MAX L/D (MIN ACTUAL VEHICLE L/D)	1	
#				
#	LADPAD	NOMINAL VEHICLE L/D, SP PAD LOAD	1	(NOM = 0.3)
#				
#	LATANG	LATERAL RANGE	4 RADIANS	
#				
#	LEQ	EXCESS C.F. OVER GRAV=(VSQ-1)GS	128.8 FPSS	
#				
#	LEWD	UPCONTROL REFERENCE L/D	1	

#				
#	LOD	FINAL PHASE L/D	1	(NOM
#				
#	LODPAD	FINAL PHASE L/D, SP PAD LOAD	1	
#				
#	L/D	DESIRED LIFT TO DRAG RATIO	1	
#		(VERTICAL PLANE)		
#				
#	L/D1	TEMP STORAGE FOR L/D IN LATERAL	1	
#				
#	L/DCMINR	LAD COS(15DEG)	1	(NOM
#				
#	PREDANGLE	PREDICTED RANGE (FINAL PHASE)	2700 NM	= PR
#				
#	Q2	FINAL PHASE RANGE -23500 Q3	21600 NM	
#		Q2 = FCN(LAD)		
#				
#	Q7	MINIMUM DRAG FOR UPCONTROL	805FPSS	
#				
#	RDOT	ALTITUDE RATE	2 VSAT	
#				
#	RDOTREF	REFERENCE RDOT FOR UPCONTROL	2 VSAT	
#				
#	RDTR	REFERENCE RDOT FOR DOWNCONT	2 VSAT	NOT S
#				
#	ROLLC	ROLL COMMAND	1 REVOLUTION	
#				
#	RTOGO	RANGE TO GO (FINAL PHASE)	2700 NM	= FX
#				
#	SL	SINE OF LATITUDE	1	NOT S
#				
#	T	TIME	B 28 CENTISEC	= TI
#				
#	THETA	DESIRED RANGE (RADIAN)	2 PI RADIAN	= TH
#				
#	THETNM	DESIRED RANGE (NM)	21600 NM	NON P
#				
#	V	VELOCITY MAGNITUDE	2 VSAT	
#				
#	V1	INITIAL VELOCITY FOR UPCONTROL	2 VSAT	
#				
#	VL	EXIT VELOCITY FOR UPCONTROL	2 VSAT	
#				
#	VREF	REFERENCE VELOCITY FOR UPCONTROL	2 VSAT	
#				
#	VS1	VSAT OR V1, WHICHEVER IS SMALLER	2 VSAT	

July 12, 2016

417

#	2 2		
# VBARS	VL /VSAT	4	
#		2 2	
# VSQ	NORMALIZED VEL. SQUARED = V /VSAT	4	= VSQUARE
#			
# WT	EARTH RATE TIMES TIME	1 REVOLUTION	NOT SAVED
#			
# X	INTERMEDIATE VARIABLE IN G-LIMITER	2 VSAT	NOT SAVED
#			
# Y	LATERAL MISS LIMIT	4 RADIANS	NOT SAVED

Page 840

EXTRA COMPUTER ERASABLE LOCATIONS NOT SHOWN ON FLOW CHARTS

# VARIABLE	DESCRIPTION	MAXIMUM VALUE
# -----	-----	-----
#		
# GOTOADDR	ADDRESS SELECTED BY SEQUENCER	
#		
# XPIPBUF	BUFFER TO STORE X PIPA COUNTS	
#		
# YPIPBUF	BUFFER TO STORE Y PIPA COUNTS	
#		
# ZPIPBUF	BUFFER TO STORE Z PIPA COUNTS	
#		
# PIPCTR	COUNTS PASSES THRU PIPA READ ROUTINE	
#		
# JJ	INDEX IN FINAL PHASE TABLE LOOK-UP	
#		
# MM	INDEX IN FINAL PHASE TABLE LOOK-U	
#		
# GRAD	INTERPOLATION FACTOR IN FINAL PHASE	
#		
# FX	DRANGE/D L/D = F3	2700 NM
#		
# FX +1	AREF	805 FPSS
#		
# FX +2	RTOGO	2700 NM
#		
# FX +3	RDOTREF	VSAT/4
#		
# FX +4	DRANGE/D RDOT = F2	21600/2VS NM/FPS
#		
# FX +5	DRANGE/D DRAG = F1	2700/805 NM/FPSS
#		

```

# TEM1B          TEMPORARY LOCATION
#
# TIME/RTO       TIME OF INITIAL TARGET RTINIT          B 28 CENTISEC
#
# DTEAROT        EST TIME BETWEEN RTINIT AND RT          B 28 CENTISEC
# -
# UNITV          UNIT V VECTOR                          2
# -
# UNITR          UNIT R VECTOR                          2
# -
# -VREL          NEGATIVE VELOCITY REL TO ATMOSP         2 VSAT

# COMPUTER SWITCHES          INITIAL STATE          CM/FT
# -----
#
# ENTRYDSP        DO ENTRY DISPLAY, IF SET              NON-BRANCH (1)          92D,
# GONEPAST        INDICATES OVERSHOOT OF TARGET          NON-BRANCH (0)          95D,
# RELVELSW        RELATIVE VELOCITY SWITCH              NON-BRANCH (0)          96D,
# EGSW            FINAL PHASE SWITCH                    NON-BRANCH (0)          97D,
# FIRSTPAS        INITIAL PASS THRU HUNTEST             NON-BRANCH (0)          98D,
# HIND            INDICATES ITERATION IN HUNTEST         NON-BRANCH (0)          99D,
# INRLSW          INDICATES INIT ROLL ATTITUDE SET       NON-BRANCH (0)          100D,
# LATSW           INHIBIT DOWNLIFT SWITCH IF NOT SET     BRANCH (1)             101D,
# .05GSW          INDICATES DRAG EXCEEDS .05 GS         BRANCH (0)             102D,
#
# GONEBY          INDICATES GONE PAST TARGET (SET)       SELF-INITIALIZING      112D,

# Page 841
# CONSTANTS AND GAINS          VALUE
# -----
#
# C1              FACTOR IN ALP COMPUTATION              1.25
# C16             CONSTD GAIN ON DRAG                    .01
# C17             CONSTD GAIN ON RDOT                    .001
# C18             BIAS VEL. FOR FINAL PHASE START        500          FPS
# C20             MAX DRAG FOR DOWN-LIFT                  175          FPSS
# CHOOK           FACTOR IN AHOOK COMPUTATION            .25
# CH1             FACTOR IN GAMMAL COMPUTATION           1.0
# COS15           COS( 15 DEG )                         .965
# DLEWDO          INITIAL VARIATION IN LEWD              -.05
# D2              DRAG TO CHANGE LEWD                    175          FPSS
# DT              COMPUTATION CYCLE TIME INTERVAL        2            SEC.
# GMAX            MAXIMUM ACCELERATION                    257.6        FPSS
# KA1             FACTOR IN KA CALC                      1.3          GS
# KA2             FACTOR IN KA CALC                      .2           GS
# KA3             FACTOR IN DO CALC                      90           FPSS

```

# KA4	FACTOR IN D0 CALC	40	FPSS
# KB1	OPTIMIZED UPCONTROL GAIN	3.4	
# KB2	OPTIMIZED UPCONTROL GAIN	.0034	
# KDMIN	INCREMENT ON Q7 TO DETECT END OF KEPLER PHASE	.5	FPSS
# KTETA	TIME OF FLIGHT CONSTANT	1000	
# KLAT1	FACTOR IN KLAT CALC	1/24	
# K44	GAIN USED IN INITIAL ROLL SECTION	19749550	FPS
# LATBIAS	LATERAL SWITCH BIAS TERM	.41252961	NM
# LEWD1	NOMINAL UPCONTROL L/D	.15	
# POINT1	FACTOR TO REDUCE UPCONTROL GAIN	.1	
# Q2	FINAL PHASE RANGE - 23500 Q3	-1002	NM
# Q3	FINAL PHASE DRANGE/D V	.07	NM/FPS
# Q5	FINAL PHASE DRANGE/D GAMMA	7050	NM/RAD
# Q6	FINAL PHASE INITIAL FLIGHT PATH ANGLE	.0349	RAD
# Q7F	MIN DRAG FOR UPCONTROL	6	FPSS
# Q7MIN	IN VALUE FOR Q7 IN FACTOR CALCULATION	40	FPSS
# Q19	FACTOR IN GAMMAL1 CALCULATION	.5	
# Q21	FACTOR IN Q2 CALCULATION	1000	NM
# Q22	FACTOR IN Q2 CALCULATION	-1302	NM
# VFINAL1	VELOCITY TO START FINAL PHASE ON INITIAL ENTRY	27000	FPS
# VFINAL	FACTOR IN INITIAL UP-DOWN CALC	26600	FPS
# VLMIN	MINIMUM VL	18000	FPS
# VMIN	VELOCITY TO SWITCH TO RELATIVE VEL	VSAT/2	
# VRCONTRL	RDOT TO START INTO HUNTEST	700	FPS
#	VRCONT=COMPUTER NAME		
# 25NM	TOLERANCE TO STOP RANGE ITERATION	25	NM
# VQUIT	VELOCITY TO STOP STEERING	1000	FPS

Page 842

CONVERSION FACTORS AND SCALING CONSTANTS

#

# ATK	ANGLE IN RAD TO NM	3437.7468	NM/RAD
# G5	NOMINAL G VALUE FOR SCALING	32.2	FPSS
# H5	ATMOSPHERE SCALE HEIGHT	28500	FT
# J	GRAVITY HARMONIC COEFFICIENT	.00162346	
# KWE	EQUATORIAL EARTH RATE	1546.10168	FPS
# MUE	EARTH GRAVITATIONAL CONSTANT	3.986032233 E14	CUBIC M
# RE	EARTH RADIUS	21202900	FT
# REQ	EARTH EQUATORIAL RADIUS	20925738.2	FT
# VSAT	SATELLITE VELOCITY AT RE	25766.1973	FPS
# WIE	EARTH RATE	.0000729211505	RAD/SEC

(END GSOP AS-278, VOL 1, FIG. 5.6-3 CONSTANTS, GAINS, ETC.)

DISPLAY QUANTITIES

```

# -----
#
# (SEE SECTION 4 OF THE GSOP FOR SIGN CONVENTIONS.)
#
# VARIABLE      DESCRIPTION      MAXIMUM VALUE
# -----      -----
#
# QMAX          PREDICTED MAXIMUM ENTRY ACCEL      163.84 GS      N 60
# VPRED         PREDICTED VELOCITY AT ALTITUDE     128 M/CENTISEC N 60
#              400K FT ABOVE FISCHER RADIUS.
# GAMMAEI      PREDICTED GAMMA AT ALTITUDE         1 REVOLUTION  N 60
#              400K FT ABOVE FISCHER RADIUS
# D            DRAG ACCELERATION                   805 FPSS      N 64
# VMAGI        INERTIAL VELOCITY MAGNITUDE         128 M/CENTISEC N 64, N 68
# THETAH       DESIRED RANGE ANGLE NM              1 REVOLUTION  N 64, N 67
# LAT          PRESENT LATITUDE                    1 REVOLUTION  N 67
# LONG         PRESENT LONGITUDE                    1 REVOLUTION  N 67
# RTOGO        RANGE ANGLE TO SPLASH FROM           1 REVOLUTION  N 67
#              EMSALT FT ABOVE FISCHER RADIUS (IN NM)
# VIO          PREDICTED VELOCITY AT ALTITUDE     128 M/CENTISEC N 63
#              EMSALT FT ABOVE FISCHER RADIUS.
# TTE          TIME OF FREE FALL TO ALT             B 28 CENTISEC N 63
#              EMSALT FT ABOVE FISCHER RADIUS
# ROLLC        ROLL COMMAND                        1 REVOLUTION  N 68, N 68, 1
# LATANG       CROSS-RANGE ERROR (XRNERR)          4 RADIANS     N 66
# DNRNGERR     DOWN RANGE ERROR                    1 REVOLUTION  N 66
#              (PREDDANG - THETAH IN NM)
# HDOT         ALTITUDE RATE                       128 M/CENTISEC N 68
# QT           MINIMUM DRAG FOR UP-CONTROL          805 FPSS      N 69
# VL           EXIT VELOCITY FOR UP-CONTROL         2 VSAT        N 69

```

Page 843

BODY ATTITUDE QUANTITIES (CM/POSE)

```

# -----
#
# VARIABLE      DECEIPTION      MAXIMUM VALUE
# -----      -----
#
# -
# -VREL         NEGATIVE VELOCITY REL TO ATMOS.     2 VSAT
# -
# OLDUYA       USED FOR UYA BELOW 1000 FPS          2
# -
# UXA/2        UNIT VECTOR TRIAD                    2
# -
# UYA/2        BASED ON                              2
# -

```

July 12, 2016

421

# UZA/2	THE TRAJECTORY	2
# _		
# UBX/2	UNIT VECTOR	2
# _		
# UBY/2	BODY TRIAD	2
# _		
# UBZ/2	FOR CM.	2

This code is written to file `src/ENTRY-LEXICON.s`.

B.28 ERASABLE ASSIGNMENTS

```

422  <src/ERASABLE-ASSIGNMENTS.s 422>≡
      # Copyright:    Public domain.
      # Filename:     ERASABLE_ASSIGNMENTS.agc
      # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
      #               It is part of the source code for the Command Module's (CM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Mod history:   2009-05-06 RSB Started adapting from the Colossus249/ file
      #               of the same name, using Comanche055 page
      #               images. Only through page 51 so far.
      #               2009-05-07 RSB Through page 92 so far.
      #               2009-05-07 RSB (Again!) First draft completed.
      #               2009-05-20 RSB Fixed some bugs uncovered in trial assemblies:
      #               EMDOT, STATEXIT, VGDISP, DVPREV, POSTCDH,
      #               RETROFLG not defined correctly, changed the
      #               typing of labels 9X9LOC1 and 9X9LOC2,
      #               R32FLBIT -> R31FLBIT.
      #               2009-05-21 RSB Corrected definition of DELBRTMP, which
      #               chained to quite a lot of off-by-one errors.
      #               Changed a +8 to a +8D.
      #
      # This source code has been transcribed or otherwise adapted from digitized
      # images of a hardcopy from the MIT Museum. The digitization was performed
      # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
      # thanks to both. The images (with suitable reduction in storage size and
      # consequent reduction in image quality as well) are available online at
      # www.ibiblio.org/apollo. If for some reason you find that the images are
      # illegible, contact me at info@sandroid.org about getting access to the
      # (much) higher-quality images which Paul actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      # Assemble revision 055 of AGC program Comanche by NASA
      # 2021113-051. 10:28 APR. 1, 1969
      #
      # This AGC program shall also be referred to as
      # Colossus 2A

      # Page 37
      # CONVENTIONS AND NOTATIONS UTILIZED FOR ERASABLE ASSIGNMENTS.

```

```

#      EQUALS IS USED IN TWO WAYS. IT IS OFTEN USED TO CHAIN A GROUP
#      OF ASSIGNMENTS SO THAT THE GROUP MAY BE MOVED WITH THE
#      CHANGING OF ONLY ONE CARD. EXAMPLE:
#
#          X      EQUALS  START
#          Y      EQUALS  X      +SIZE.X
#          Z      EQUALS  Y      +SIZE.Y
#
#      (X, Y, AND Z ARE CONSECUTIVE AND BEGIN AT START.
#      SIZE.X AND SIZE.Y ARE THE RESPECTIVE SIZES OF X AND Y.
#      USUALLY NUMERIC, IE. 1, 2, 6, 18D, ETC.)
#
#      EQUALS OFTEN IMPLIES THE SHARING OF REGISTERS (DIFFERENT NAMES
#      AND DIFFERENT DATA). EXAMPLE:
#
#          X      EQUALS  Y
#
#      =      MEANS THAT MULTIPLE NAMES HAVE BEEN GIVEN TO THE SAME DATA.
#      (THIS IS LOGICAL EQUIVALENCE, NOT SHARING.) EXAMPLE:
#
#          X      =      Y
#
#      THE SIE AND UTILIZATION OF AN ERASABLE ARE OFTEN INCLUDED IN
#      THE COMMENTS IN THE FOLLOWING FORM: M(SIZE)N.
#
#      M      REFERS TO THE MOBILITY OF THE ASSIGNMENT.
#      B      MEANS THAT THE SYMBOL IS REFERENCED BY BASIC
#      INSTRUCTIONS AND THUS IS E-BANK SENSITIVE.
#      I      MEANS THAT THE SYMBOL IS REFERENCED ONLY BY
#      INTERPRETIVE INSTRUCTIONS, AND IS THUS E-BANK
#      INSENSITIVE AND MAY APPEAR IN ANY E-BANK.
#
#      SIZE   IS THE NUMBER OF REGISTERS INCLUDED BY THE SYMBOL.
#
#      N      INDICATES THE NATURE OF PERMANENCE OF THE CONTENTS.
#      PL     MEANS THAT THE CONTENTS ARE PAD LOADED.
#      DSP    MEANS THAT THE REGISTER IS USED FOR A DISPLAY.
#      PRM    MEANS THAT THE REGISTER IS PERMANENT. IE., IT
#      IS USED DURING THE ENTIRE MISSION FOR ONE
#      PURPOSE AND CANNOT BE SHARED.
#      TMP    MEANS THAT THE REGISTER IS USED TEMPORARILY OR
#      IS A SCRATCH REGISTER FOR THE ROUTINE TO WHICH
#      IT IS ASSIGNED. THAT IS, IT NEED NOT BE SET
#      PRIOR TO INVOCATION OF THE ROUTINE NOR DOES IT
#      CONTAIN USEFUL OUTPUT TO ANOTHER ROUTINE. THUS
#
# Page 38

```

```

#           IT MAY BE SHARED WITHANY OTHER ROUTINE WHICH
#           IS NOT ACTIVE IN PARALLEL
#           IN      MEANS INPUT TO THE ROUTINE AND IT IS PROBABLY
#           TEMPORARY FOR A HIGHER-LEVEL ROUTINE/PROGRAM.
#           OUT    MEANS OUTPUT FROM THE ROUTINE, PROBABLY
#           TEMPORARY FOR A HIGHER-LEVEL ROUTINE/PROGRAM.

```

```

# Page 39

```

```

# SPECIAL REGISTERS.

```

```

A           EQUALS 0
L           EQUALS 1           # L AND Q ARE BOTH CHANNELS AND REGISTERS
Q           EQUALS 2
EBANK       EQUALS 3
FBANK       EQUALS 4
Z           EQUALS 5           # ADJACENT TO FBANK AND BBANK FOR DXCH Z
BBANK       EQUALS 6           # (DTCB) AND DXCH FBANK (DTCF).
                                   # REGISTER 7 IS A ZERO-SOURCE, USED BY ZL.

ARUPT       EQUALS 10          # INTERRUPT STORAGE
LRUPT       EQUALS 11
QRUPT       EQUALS 12
SAMPTIME    EQUALS 13          # SAMPLED TIME 1 & 2.
ZRUPT       EQUALS 15          # (13 AND 14 ARE SPARES.)
BANKRUPT    EQUALS 16          # USUALLY HOLDS FBANK OR BBANK.
BRUPT       EQUALS 17          # RESUME ADDRESS AS WELL.

CYR         EQUALS 20
SR          EQUALS 21
CYL         EQUALS 22
EDOP        EQUALS 23          # EDITS INTERPRETIVE OPERATION CODE PAIRS.

TIME2       EQUALS 24
TIME1       EQUALS 25
TIME3       EQUALS 26
TIME4       EQUALS 27
TIME5       EQUALS 30
TIME6       EQUALS 31
CDUX        EQUALS 32
CDUY        EQUALS 33
CDUZ        EQUALS 34
CDUT        EQUALS 35          # OPTICS TRUNNION CDU (WAS OPTY).
OPTY        =          CDUT
CDUS        EQUALS 36          # OPTICS SHAFT CDU (WAS OPTX).
OPTX        =          CDUS

```


PIPAX	EQUALS	37
PIPAY	EQUALS	40
PIPAZ	EQUALS	41
BMAGX	EQUALS	42
BMAGY	EQUALS	43
BMAGZ	EQUALS	44
INLINK	EQUALS	45
RNRAD	EQUALS	46
GYROCTR	EQUALS	47
GYROCMD	EQUALS	47
CDUXCMD	EQUALS	50
CDUYCMD	EQUALS	51

Page 40

CDUZCMD	EQUALS	52	
CDUTCMD	EQUALS	53	# OPTICS TRUNNION COMMAND (WAS OPTYCMD)
OPTYCMD	=	CDUTCMD	
TVCYAW	EQUALS	CDUTCMD	# SPS YAW COMMAND IN TVC MODE
CDUSCMD	EQUALS	54	# OPTICS SHAFT COMMAND (WAS OPTXCMD).
TVCPITCH	EQUALS	CDUSCMD	# SPS PITCH COMMAND IN TVC MODE
OPTXCMD	=	CDUSCMD	
EMSD	EQUALS	55	
THRUST	EQUALS	55	
LEMONM	EQUALS	56	
LOCALARM	EQUALS	57	
BANKALRM	EQUALS	60	

INTERPRETIVE REGISTERS ADDRESSED RELATIVE TO VAC AREA.

LVSQUARE	EQUALS	34D	# SQUARE OF VECTOR INPUT TO ABVAL AND UNIT
LV	EQUALS	36D	# LENGTH OF VECTOR INPUT TO UNIT.
X1	EQUALS	38D	# INTERPRETIVE SPECIAL REGISTER RELATIVE
X2	EQUALS	39D	# TO THE WORK AREA.
S1	EQUALS	40D	
S2	EQUALS	41D	
QPRET	EQUALS	42D	

Page 41

INPUT/OUTPUT CHANNELS

*** CHANNEL ZERO IS TO BE USED IN AN INDEXED OPERATION ONLY. ***

LCHAN	EQUALS	L
QCHAN	EQUALS	Q
HISCALAR	EQUALS	3

LOSCALAR	EQUALS	4
PYJETS	EQUALS	5
ROLLJETS	EQUALS	6
SUPERBNK	EQUALS	7
OUTO	EQUALS	10
DSALMOUT	EQUALS	11
CHAN12	EQUALS	12
CHAN13	EQUALS	13
CHAN14	EQUALS	14
MNKEYIN	EQUALS	15
NAVKEYIN	EQUALS	16
CHAN30	EQUALS	30
CHAN31	EQUALS	31
CHAN32	EQUALS	32
CHAN33	EQUALS	33
DNTM1	EQUALS	34
DNTM2	EQUALS	35

END OF CHANNEL ASSIGNMENTS

Page 42

FLAGWORDS

#

# FLAGWRD0	STATE +0	(000-014)
# FLAGWRD1	STATE +1	(015-029)
# FLAGWRD2	STATE +2	(030-044)
# FLAGWRD3	STATE +3	(045-059)
# FLAGWRD4	STATE +4	(060-074)
# FLAGWRD5	STATE +5	(075-089)
# FLAGWRD6	STATE +6	(090-104)
# FLAGWRD7	STATE +7	(105-119)
# FLAGWRD8	STATE +8D	(120-134)
# FLAGWRD9	STATE +9D	(135-149)

SORTED LIST OF

INTERPRETIVE SWITCH BIT ASSIGNMENTS

INTERPRETIVE SWITCH BIT ASSIGNMENTS

#

# FLAGWORD	DEC NUM	BIT & FLAG	EQUIVALENT FLAGWORDS
#			
# 22DSPFLG	032D	BIT 13 FLAG 2	
# 360SW	134D	BIT 1 FLAG 8	
# 3AXISFLG	084D	BIT 6 FLAG 5	
# ADVTRK	125D	BIT 10 FLAG 8	

# AMOONFLG	13D	BIT 2 FLAG 0	
# APSESW	130D	BIT 5 FLAG 8	
# ASTNFLAG	108D	BIT 12 FLAG 7	
# ATTCHFLG	118D	BIT 2 FLAG 7	
# AVEGFLAG	029D	BIT 1 FLAG 1	
# AVEMIDSW	149D	BIT 1 FLAG 9	
# AVFLAG	040D	BIT 5 FLAG 2	
# CALCMAN2	043D	BIT 2 FLAG 2	
# CMDAPARM	093D	BIT 12 FLAG 6	
# CMOONFLG	123D	BIT 12 FLAG 8	
# CM/DSTBY	103D	BIT 2 FLAG 6	
# COGAFLAG	131D	BIT 4 FLAG 8	
# COMPUTER	082D	BIT 8 FLAG 5	
# CPHIFLAG	000D	BIT 15 FLAG 0	
# CULTFLAG	053D	BIT 7 FLAG 3	
# D6OR9FLG	058D	BIT 2 FLAG 3	
# DAPBIT1	090D	BIT 15 FLAG 6	
# DAPBIT2	091D	BIT 14 FLAG 6	
# DIMOFLAG	059D	BIT 1 FLAG 3	
# DMENFLAG	081D	BIT 9 FLAG 5	
# DRIFTFLG	030D	BIT 15 FLAG 2	
# DSKYFLAG	075D	BIT 15 FLAG 5	
# EGSW	097D	BIT 8 FLAG 6	KNOTNFLAG R57FLAG

Page 43

# ENG1FLAG	018D	BIT 12 FLAG 1	
# ENG2FLAG	019D	BIT 11 FLAG 1	
# ENGONFLG	083D	BIT 7 FLAG 5	
# ERADFLAG	017D	BIT 13 FLAG 1	
# ETPIFLAG	038D	BIT 7 FLAG 2	FIRSTFLG OPTNSW
# F2RTE	010D	BIT 5 FLAG 0	
# FINALFLG	039D	BIT 6 FLAG 2	
# FIRSTFLG	038D	BIT 7 FLAG 2	ETPIFLAG OPTNSW
# FREEFLAG	012D	BIT 3 FLAG 0	
# GAMDIFSW	094D	BIT 11 FLAG 6	
# GLOKFAIL	046D	BIT 14 FLAG 3	
# GONEBY	112D	BIT 8 FLAG 7	
# GONEPAST	095D	BIT 10 FLAG 6	
# GRRBKFLG	085D	BIT 5 FLAG 5	
# GUESSW	028D	BIT 2 FLAG 1	
# GYMDIFSW	104D	BIT 1 FLAG 6	
# .05GSW	102D	BIT 3 FLAG 6	
# HIND	099D	BIT 6 FLAG 6	
# IDLEFAIL	024D	BIT 6 FLAG 1	
# IDLEFLAG	113D	BIT 7 FLAG 7	

# IGNFLAG	107D	BIT 13 FLAG 7	
# IMPULSW	036D	BIT 9 FLAG 2	
# IMUSE	007D	BIT 8 FLAG 0	
# INCORFLG	079D	BIT 11 FLAG 5	
# INFINFLG	128D	BIT 7 FLAG 8	
# INRLSW	100D	BIT 5 FLAG 6	
# INTFLAG	151D	BIT 14 FLAG 10	
# INTYPFLG	056D	BIT 4 FLAG 3	
# ITSWICH	106D	BIT 14 FLAG 7	
# KFLAG	014D	BIT 1 FLAG 0	
# KNOWNFLG	097D	BIT 8 FLAG 6	EGSW R57FLAG
# LATSW	101D	BIT 4 FLAG 6	
# LMOONFLG	124D	BIT 11 FLAG 8	
# LUNAFLAG	048D	BIT 12 FLAG 3	
# MAXDBFLG	138D	BIT 12 FLAG 9	
# MGLVFLAG	088D	BIT 2 FLAG 5	
# MID1FLAG	147D	BIT 3 FLAG 9	
# MIDAVFLG	148D	BIT 2 FLAG 9	
# MIDFLAG	002D	BIT 13 FLAG 0	
# MKOVFLAG	072D	BIT 3 FLAG 4	
# MOONFLAG	003D	BIT 12 FLAG 0	
# MRKIDFLG	060D	BIT 15 FLAG 4	
# MRKNVFLG	066D	BIT 9 FLAG 4	
# MRUPTFLG	070D	BIT 5 FLAG 4	
# MWAITFLG	064D	BIT 11 FLAG 4	
# N22ORN17	144D	BIT 6 FLAG 9	
# NEEDLFLG	006D	BIT 9 FLAG 0	
# NEWIFLG	122D	BIT 13 FLAG 8	
# NJETSFLG	015D	BIT 15 FLAG 1	
# NODOFLAG	044D	BIT 1 FLAG 2	

Page 44

# NODOP01	018D	BIT 12 FLAG 1	
# NORFHOR	004D	BIT 11 FLAG 0	
# NORMSW	110D	BIT 10 FLAG 7	
# NOSWITCH	098D	BIT 7 FLAG 6	
# NRMIDFLG	062D	BIT 13 FLAG 4	
# NRMNVFLG	067D	BIT 8 FLAG 4	
# NRUPTFLG	071D	BIT 4 FLAG 4	
# NWAITFLG	065D	BIT 10 FLAG 4	
# OPTNSW	038D	BIT 7 FLAG 2	ETPIFLAG FIRSTFLG
# ORBWFLAG	054D	BIT 6 FLAG 3	
# ORDERSW	129D	BIT 6 FLAG 8	
# P21FLAG	033D	BIT 12 FLAG 2	
# P22MKFLG	049D	BIT 11 FLAG 3	

# P39/79SW	126D	BIT 9 FLAG 8	
# PDSPFLAG	063D	BIT 12 FLAG 4	
# PFRATFLG	041D	BIT 4 FLAG 2	
# PINBRFLG	069D	BIT 6 FLAG 4	
# PRECIFLG	052D	BIT 8 FLAG 3	
# PRFTRKAT	060D	BIT 10 FLAG 5	
# PRIODFLG	061D	BIT 14 FLAG 4	
# PRONVFLG	068D	BIT 7 FLAG 4	
# QUITFLAG	145D	BIT 5 FLAG 9	
# R21MARK	031D	BIT 14 FLAG 2	
# R22CAFLG	143D	BIT 7 FLAG 9	
# R23FLG	021D	BIT 9 FLAG 1	
# R31FLAG	146D	BIT 4 FLAG 9	
# R53FLAG	009D	BIT 6 FLAG 0	
# R57FLAG	097D	BIT 8 FLAG 6	KNOWNFLG EGSW
# R60FLAG	086D	BIT 4 FLAG 5	
# REFSMFLG	047D	BIT 13 FLAG 3	
# REINTFLG	158D	BIT 7 FLAG 10	
# RELVELSW	096D	BIT 9 FLAG 6	
# RENDWFLG	089D	BIT 1 FLAG 5	
# RNDVZFLG	008D	BIT 7 FLAG 0	
# RPQFLAG	120D	BIT 15 FLAG 6	
# RVSW	111D	BIT 9 FLAG 7	
# SAVECFLG	140D	BIT 10 FLAG 9	
# SKIPVHF	035D	BIT 10 FLAG 2	
# SLOPESW	027D	BIT 3 FLAG 1	
# SOLNSW	087D	BIT 3 FLAG 5	
# SOURCFLG	142D	BIT 8 FLAG 9	
# STATEFLG	055D	BIT 5 FLAG 3	
# STEERSW	034D	BIT 11 FLAG 2	
# STIKFLAG	016D	BIT 14 FLAG 1	
# STRULLSW	092D	BIT 13 FLAG 6	
# SURFFLAG	127D	BIT 8 FLAG 8	
# SWTOVER	135D	BIT 15 FLAG 9	
# TARG1FLG	020D	BIT 10 FLAG 1	

Page 45

# TARG2FLG	021D	BIT 9 FLAG 1
# TERMIFLG	105D	BIT 15 FLAG 7
# TFFSW	119D	BIT 1 FLAG 7
# TIMRFLAG	109D	BIT 11 FLAG 7
# TRACKFLG	025D	BIT 5 FLAG 1
# TRM03FLG	026D	BIT 4 FLAG 1
# TRUNFLAG	011D	BIT 4 FLAG 0
# UPDATFLG	023D	BIT 7 FLAG 1

# UNLOCKFL	116D	BIT 4 FLAG 7
# V37FLAG	114D	BIT 6 FLAG 7
# V59FLAG	078D	BIT 12 FLAG 5
# V67FLAG	136D	BIT 14 FLAG 9
# V82EMFLG	137D	BIT 13 FLAG 9
# V94FLAG	139D	BIT 11 FLAG 9
# V96ONFLG	132D	BIT 3 FLAG 8
# VEHUPFLG	022D	BIT 8 FLAG 1
# VERIFLAG	117D	BIT 3 FLAG 7
# VFLAG	050D	BIT 10 FLAG 3
# VHFRFLAG	141D	BIT 9 FLAG 9
# VINTFLAG	057D	BIT 3 FLAG 3
# XDELVFLG	037D	BIT 8 FLAG 2
# XDSPFLAG	074D	BIT 1 FLAG 4

Page 46

INTERPRETIVE SWITCH BIT ASSIGNMENTS

FLAGWRDO	=	STATE +0	# (000-014)	
			# (SET)	(RESET)
# BIT 15 FLAG 0				
CPHIFLAG	=	000D	# OUTPUT OF CALCGA IS	OUTPUT OF CALCGA IS
			# CPHIX	THETAD
CPHIBIT	=	BIT15		
# BIT 14 FLAG 0				
JSWITCH	=	001D	# INTEGRATION OF W	INTEGRATION OF STATE
			# MATRIX	VECTOR
JSWCHBIT	=	BIT14		
# BIT 13 FLAG 0				
MIDFLAG	=	002D	# INTEGRATION WITH	INTEGRATION WITHOUT
			# SOLAR PERTURBATIONS	SOLAR PERTURBATIONS
MIDFLBIT	=	BIT13		
# BIT 12 FLAG 0				
MOONFLAG	=	003D	# MOON IS SPHERE OF	EARTH IS SPHERE OF
			# INFLUENCE	INFLUENCE
MOONBIT	=	BIT12		
# BIT 11 FLAG 0				
NORFHOR	=	004D	# FAR HORIZON	NEAR HORIZON
NORFBIT	=	BIT11		

# BIT 10 FLAG 0 ZMEASURE	=	005D	# MEASUREMENT PLANET # AND PRIMARY PLANET # DIFFERENT	MEASUREMENT PLANET AND PRIMARY PLANET SAME
ZMEASBIT	=	BIT10		
# BIT 9 FLAG 0 NEEDLFLG	=	006D	# TOTAL ATTITUDE # ERROR DISPLAYED	A/P FOLLOWING ERROR DISPLAYED
NEEDLBIT	=	BIT9		
# BIT 8 FLAG 0 IMUSE	=	007D	# IMU IN USE	IMU NOT IN USE
# Page 47				
IMUSEBIT	=	BIT8		
# BIT 7 FLAG 0 RNDVZFLG	=	008D	# P20 RUNNING	P20 NOT RUNNING
RNDVZBIT	=	BIT7		
# BIT 6 FLAG 0 R53FLAG	=	009D	# V51 INITIATED	V51 NOT INITIATED
R53FLBIT	=	BIT8		
# BIT 5 FLAG 0 F2RTE	=	010D	# IN TIME CRITICAL # MODE	NOT IN TIME CRITICAL MODE
F2RTEBIT	=	BIT5		
# BIT 4 FLAG 0 TRUNFLAG	=	011D	# DRIVING OF TRUNNION # ALLOWED	DRIVING OF TRUNNION NOT ALLOWED
TRUNBIT	=	BIT4		
# BIT 3 FLAG 0 FREEFLAG	=	012D	# (TEMPORARY FLAG USED IN MANY ROUTINES)	

FREEFBIT	=	BIT3		
# BIT 2 FLAG 0				
AMOONFLG	=	13D	# STATE VECTOR IN	STATE VECTOR IN
AMOONBIT	=	BIT2	# LUNAR SPHERE AT	EARTH SPHERE AT
			# MIDTOAVE	MIDTOAVE
# BIT 1 FLAG 0				
KFLAG	=	014D	# SEARCH SECTOR MORE	SEARCH SECTOR LESS
			# THAN 180 DEGREES	THAN 180 DEGREES
KBIT	=	BIT1		
FLAGWRD1	=	STATE +1	# (015-029)	
			# (SET)	(RESET)
# BIT 15 FLAG 1				
NJETSFLG	=	015D	# TWO JET RCS BURN	FOUR JET RCS BURN
NJETSBIT	=	BIT15		
# Page 48				
# BIT 14 FLAG 1				
STIKFLAG	=	016D	# RHC CONTROL	CMC CONTROL
STIKBIT	=	BIT14		
# BIT 13 FLAG 1				
ERADFLAG	=	017D	# EARTH, COMPUTE	EARTH, USED FIXED
			# FISCHER ELLIPSOID	RADIUS
			# RADIUS	
			# MOON, USE FIXED	MOON, USE RLS FOR
			# RADIUS	LUNAR RADIUS
ERADFBIT	=	BIT13		
# BIT 12 FLAG 1				
NODOP01	=	018D	# P01 NOT ALLOWED	P01 ALLOWD
NOP01BIT	=	BIT12		
# BIT 11 FLAG 1				
ENG2FLAG	=	019D	# RCS BURN	SPS BURN
ENG2BIT	=	BIT11		
# BIT 10 FLAG 1				

July 12, 2016

433

TARG1FLG	=	020D	# SIGHTING LEM	NOT SIGHTING LEM
TARG1BIT	=	BIT10		
# BIT 9 FLAG 1				
TARG2FLG	=	021D	# SIGHTING LANDMARK	SIGHTING STAR
TARG2BIT	=	BIT9		
# BIT 9 FLAG 1				
R23FLG	=	021D	# R23 MARKING	
R23BIT	=	BIT9		
# BIT 8 FLAG 1				
VEHUPFLG	=	022D	# CSM STATE VECTOR # BEING UPDATED	LEM STATE VECTOR BEING UPDATED
VEHUPBIT	=	BIT8		
# BIT 7 FLAG 1				
UPDATFLG	=	023D	# UPDATING BY MARKS # ALLOWED	UPDATING BY MARKS NOT ALLOWED
# Page 49				
UPDATBIT	=	BIT7		
# BIT 6 FLAG 1				
IDLEFAIL	=	024D	# INHIBIT R41	ENABLE R41 (ENGFAIL)
IDLEBIT	=	BIT6		
# BIT 5 FLAG 1				
TRACKFLG	=	025D	# TRACKING ALLOWED	TRACKING NOT ALLOWED
TRACKBIT	=	BIT5		
# BIT 4 FLAG 1				
TRM03FLG	=	026D	# REQUEST TO # TERMINATE P03 HAS # BEEN ENTERED	NO REQUEST TO TERMINATE P03 HAS BEEN ENTERED
TRM03BIT	=	BIT4		
# BIT 3 FLAG 1				
SLOPESW	=	027D	# ITERATE WITH BIAS	ITERATE WITH REGULA

			# METHOD IN ITERATOR	FALSI METHOD IN
			#	ITERATOR
SLOPEBIT	=	BIT3		
# BIT 2 FLAG 1				
GUESSW	=	028D	# NO STARTING VALUE	STARTING VALUE FOR
			# FOR ITERATION	ITERATION EXISTS
GUESSBIT	=	BIT2		
# BIT 1 FLAG 1				
AVEGFLAG	=	029D	# AVERAGEG (SERVICER)	AVERAGEG (SERVICER)
			# TO CONTINUE	TO CEASE
AVEGBIT	=	BIT1		
FLAGWRD2	=	STATE +2	# (030-044)	
			# (SET)	(RESET)
# BIT 15 FLAG 2				
DRIFTFLG	=	030D	# T3RUPT CALLS GYRO	T3RUPT DOES NO GYRO
			# COMPENSATION	COMPENSATION
DRFTBIT	=	BIT15		
# Page 50				
# BIT 14 FLAG 2				
R21MARK	=	031D	# OPTION ONE FOR	OPTION TWO FOR
			# MARKRUPT	MARKRUPT
R21BIT	=	BIT14		
# BIT 13 FLAG 2				
22DSPFLG	=	032D	# DISPLAY DR,DV	DO NOT DISPLAY DR,DV
22DSPBIT	=	BIT13		
# BIT 12 FLAG 2				
P21FLAG	=	033D	# SUCCEEDING PASS	1ST PASS THRU P21,
			# THRU P21, USE BASE	CALCULATE BASE
P21BIT	=	BIT12	# VECTOR FOR CALC.	VECTOR
STEERSW	=	034D	# STEERING TO BE DONE	STEERING OMITTED

STEERBIT	=	BIT11		
# BIT 10 FLAG 2				
SKIPVHF	=	035D	# DISREGARD RADAR	RADAR READ TO
			# READ BECAUSE OF	PROCEED NORMALLY
SKIPVBIT	=	BIT10	# SFTWRE OR HDWRE	
			# RESTART	
# BIT 9 FLAG 2				
IMPULSW	=	036D	# MINIMUM IMPULSE	STEERING BURN (NO
			# BURN (CUTOFF TIME	CUTOFF TIME YET
			# SPECIFIED)	AVAILABLE)
IMPULBIT	=	BIT9		
# BIT 8 FLAG 2				
XDELVFLG	=	037D	# EXTERNAL DELTAV VG	LAMBERT (AIMPOINT)
			# COMPUTATION	VG COMPUTATION
XDELVBIT	=	BIT8		
# BIT 7 FLAG 2				
ETPIFLAG	=	038D	# ELEVATION ANGLE	TPI TIME SUPPLIED
			# SUPPLIED FOR P34,74	FOR P34,74
# BIT 7 FLAG 2				
FIRSTFLG	=	ETPIFLAG	# FIRST PASS	SUCCEEDING PASS THRU
			# THRU S40.9	S40.9
FIRSTBIT	=	BIT7		
# BIT 7 FLAG 2				
# Page 51				
OPTNSW	=	ETPIFLAG	# SOI PHASE P38/P78	SOR PHASE OF P38/P78
FINALBIT	=	BIT6		
# BIT 6 FLAG 2				
FINALFLG	=	039D	# LAST PASS THROUGH	INTERIM PASS THROUGH
			# RENDEZVOUS PROGRAM	RENDEZVOUS PROGRAM
			# COMPUTATIONS	COMPUTATIONS
AVFLBIT	=	BIT5		
# BIT 5 FLAG 2				

AVFLAG	=	040D	# LEM IS ACTIVE # VEHICLE	CSM IS ACTIVE VEHICLE
# BIT 4 FLAG 2 PFRATFLG	=	041D	# PREFERRED ATTITUDE # COMPUTED	PREFERRED ATTITUDE NOT COMPUTED
PFRATBIT	=	BIT4		
# BIT 3 FLAG 2	=	042D		
# BIT 2 FLAG 2 CALCMAN2	=	043D	# PERFORM MANEUVER # STARTING PROCEDURE	BYPASS STARTING PROCEDURE
CALC2BIT	=	BIT2		
# BIT 1 FLAG 2 NODOFLAG	=	044D	# V37 NOT PERMITTED	V37 PERMITTED
NODOBIT	=	BIT1		
FLAGWRD3	=	STATE +3	# (045-059) # (SET)	(RESET)
# BIT 15 FLAG 3	=	045D		
# BIT 14 FLAG 3 GLOKFAIL	=	046D	# GIMBAL LOCK HAS # OCCURRED	NOT IN GIMBAL LOCK
GLOKFBIT	=	BIT14		
# Page 52				
# BIT 13 FLAG 3 REFSMFLG	=	047D	# REFSMMAT GOOD	REFSMMAT NO GOOD
REFSMBIT	=	BIT13		
# BIT 12 FLAG 3 LUNAFLAG	=	048D	# LUNAR LAT-LONG	EARTH LAT-LONG
LUNABIT	=	BIT12		

July 12, 2016

437

# BIT 11 FLAG 3 P22MKFLG	=	049D	# P22 DOWNLINKED MARK # DATA WAS JUST TAKEN	P22 DOWNLINK MARK DATA NOT JUST TAKEN
P22MKBIT	=	BIT11		
# BIT 10 FLAG 3 VFLAG	=	050D	# LESS THAN TWO STARS # IN FIELD OF VIEW	TWO STARS IN FIELD OF VIEW
VFLAGBIT	=	BIT10		
# BIT 9 FLAG 3	=	051D		
# BIT 8 FLAG 3 PRECIFLG	=	052D	# CSMPREC OR LEMPREC # OR INTEGRVS CALLED	INTEGRV CALLED
PRECIBIT	=	BIT8		
# BIT 7 FLAG 3 CULTFLAG	=	053D	# STAR OCCULTED	STAR NOT OCCULTED
CULTBIT	=	BIT7		
# BIT 6 FLAG 3 ORBWFLAG	=	054D	# W MATRIX VALID FOR # ORBITAL NAVIGATION	W MATRIX INVALID FOR ORBITAL NAVIGATION
ORBWFBIT	=	BIT6		
# BIT 5 FLAG 3 STATEFLG	=	055D	# PERMANENT STATE # VECTOR UPDATED	PERMANENT STATE VECTOR NOT UPDATED
STATEBIT	=	BIT5		
# BIT 4 FLAG 3 INTYPFLG # Page 53 INTYBIT	=	056D BIT4	# CONIC INTEGRATION	ENCKE INTEGRATION
# BIT 3 FLAG 3 VINTFLAG	=	057D	# CSM STATE VECTOR	LEM STATE VECTOR

			# BEING INTEGRATED	BEING INTEGRATED
VINTFBIT	=	BIT3		
# BIT 2 FLAG 3 D6OR9FLG	=	058D	# DIMENSION OF W IS 9 # FOR INTEGRATION	DIMENSION OF W IS 6 FOR INTEGRATION
D6OR9BIT	=	BIT2		
# BIT 1 FLAG 3 DIMOFLAG	=	059D	# W MATRIX IS TO BE # USED	W MATRIX IS NOT TO BE USED
FLAGWRD4	=	STATE +4	# (060-074) # (SET)	(RESET)
DIMOBIT	=	BIT1		
# BIT 15 FLAG 4 MRKIDFLG	=	060D	# MARK DISPLAY IN # ENDIDLE	NO MARK DISPLAY IN ENDIDLE
MRKIDBIT	=	BIT15		
# BIT 14 FLAG 4 PRIODFLG	=	061D	# PRIORITY DISPLAY IN # ENDIDLE	NO PRIORITY DISPLAY IN ENDIDLE
PRIODBIT	=	BIT14		
# BIT 13 FLAG 4 NRMIDFLG	=	062D	# NORMAL DISPLAY IN # ENDIDLE	NO NORMAL DISPLAY IN ENDIDLE
NRMIDBIT	=	BIT13		
# BIT 12 FLAG 4 PDSPFLAG	=	063D	# CAN'T INTERRUPT # PRIORITY DISPLAY	SEE M. HAMILTON
PDSPFBIT	=	BIT12		
# BIT 11 FLAG 4 MWAITFLG	=	064D	# HIGHER PRIORITY	NO HIGHER PRIORITY
# Page 54				

			# DISPLAY OPERATING # WHEN MARK DISPLAY # INITIATED	DISPLAY OPERATING WHEN MARK DISPLAY INITIATED
MWAITBIT	=	BIT11		
# BIT 10 FLAG 4 NWAITFLG	=	065D	# HIGHER PRIORITY # DISPLAY OPERATING # WHEN NORMAL # DISPLAY INITIATED	NO HIGHER PRIORITY DISPLAY OPERATING WHEN NORMAL DISPLAY INITIATED
NWAITBIT	=	BIT10		
# BIT 9 FLAG 4 MRKNVFLG	=	066D	# ASTRONAUT USING # KEYBOARD WHEN MARK # DISPLAY INITIATED	ASTRONAUT NOT USING KEYBOARD WHEN MARK DISPLAY INITIATED
MRKNVBIT	=	BIT9		
# BIT 8 FLAG 4 NRMNVFLG	=	067D	# ASTRONAUT USING # KEYBOARD WHEN # NORMAL DISPLAY # INITIATED	ASTRONAUT NOT USING KEYBOARD WHEN NORMAL DISPLAY INITIATED
NRMNVBIT	=	BIT8		
# BIT 7 FLAG 4 PRONVFLG	=	068D	# ASTRONAUT USING # KEYBOARD WHEN # PRIORITY DISPLAY # INITIATED	ASTRONAUT NOT USING KEYBOARD WHEN PRIORITY DISPLAY INITIATED
PRONVBIT	=	BIT7		
# BIT 6 FLAG 4 PINBRFLG	=	069D	# ASTRONAUT HAS # INTERFERED WITH # EXISTING DISPLAY	ASTRONAUT HAS NOT INTERFERED WITH EXISTING DISPLAY
PINBRBIT	=	BIT6		
# BIT 5 FLAG 4 MRUPTFLG	=	070D	# MARK DISPLAY # INTERRUPTED BY	MARK DISPLAY NOT INTERRUPTED BY

			# PRIORITY DISPLAY	PRIORITY DISPLAY
MRUPTBIT	=	BIT5		
# Page 55				
# BIT 4 FLAG 4				
NRUPTFLG	=	071D	# NORMAL DISPLAY # INTERRUPTED BY # PRIORITY OR MARK # DISPLAY	NORMAL DISPLAY NOT INTERRUPTED BY PRIORITY OR MARK DISPLAY
NRUPTBIT	=	BIT4		
# BIT 3 FLAG 4				
MKOVFLAG	=	072D	# MARK DISPLAY OVER # NORMAL	NO MARK DISPLAY OVER NORMAL
MKOVBIT	=	BIT3		
# BIT 2 FLAG 4				
	=	073D	# DISPLAY BIT # CLEARED AT INTERVALS	
# BIT 1 FLAG 4				
XDSPFLAG	=	074D	# MARK DISPLAY NOT TO # BE INTERRUPTED	NO SPECIAL MARK INFORMATION
XDSPBIT	=	BIT1		
FLAGWRD5	=	STATE +5	# (075-099) # (SET)	(RESET)
# BIT 15 FLAG 5				
DSKYFLAG	=	075D	# DISPLAYS SENT TO # DSKY	NO DISPLAYS TO DSKY
DSKYBIT	=	BIT15		
# BIT 14 FLAG 5				
RETROFLG	=	076D	# P37 PREMANEUVER	ORBIT NOT RETROGRADE
RETROBIT	=	BIT14	# ORBIT IS RETROGRADE	
# BIT 13 FLAG 5				
SLOWFLG	=	077D	# P37 TRANSEARTH	SLOW DOWN IS NOT
SLOWBIT	=	BIT13	# COAST SLOW DOWN # IS DESIRED	DESIRED
# BIT 12 FLAG 5				

July 12, 2016

441

V59FLAG	=	078D	# CALIBRATING FOR # P23	NORMAL MARKING FOR P23
V59FLBIT	=	BIT12		
# BIT 11 FLAG 5 # Page 56				
INCORFLG	=	079D	# FIRST INCORPORATION	SECOND INCORPORATION
INCORBIT	=	BIT11		
# BIT 10 FLAG 5				
RNGSCFLG	=	080D	# ANOTHER TAG FOR PRFTRKAT	
# BIT 10 FLAG 5				
PRFTRKAT	=	RNGSCFLG	# PREF TRACK ATT	+K AXIS TRACK ATT
PRFTRBIT	=	BIT10		
# BIT 9 FLAG 5				
DMENFLG	=	081D	# DIMENSION OF W IS 9 # FOR INCORPORATION	DIMENSION OF W IS 6 FOR INCORPORATION
DMENFBIT	=	BIT9		
# BIT 8 FLAG 5				
COMPUTER	=	082D	# COMPUTER IS CMC	COMPUTER IS LGC
COMPTBIT	=	BIT8		
# BIT 7 FLAG 5				
ENGONFLG	=	083D	# ENGINE TURNED ON	ENGINE TURNED OFF
ENGONBIT	=	BIT7		
# BIT 6 FLAG 5				
3AXISFLG	=	084D	# MANEUVER SPECIFIED # BY THREE AXES	MANEUVER SPECIFIED BY ONE AXIS
3AXISBIT	=	BIT6		
# BIT 5 FLAG 5				
GRRBKFLG	=	085D	# BACKUP GRR RECEIVED #	BACKUP GRR NOT RECEIVED
GRRBKBIT	=	BIT5		

# BIT 4 FLAG 5 R60FLAG	=	086D	# R61 MUST USE R60	NORMAL R61
R60FLBIT	=	BIT4		
# BIT 3 FLAG 5 SOLNSW # Page 57	=	087D	# LAMBERT DOES NOT # CONVERGE, OR TIME- # RADIUS NEARLY CIRC.	LAMBERT CONVERGES OR TIME-RADIUS NON CIRCULAR.
SOLNSBIT	=	BIT3		
# BIT 2 FLAG 5 MGLVFLAG	=	088D	# LOCAL VERTICAL # COORDINATES # COMPUTED	MIDDLE GIMBAL ANGLE COMPUTED
MGLVFBIT	=	BIT2		
# BIT 1 FLAG 5 RENDWFLG	=	089D	# W MATRIX VALID # FOR RENDEZVOUS # NAVIGATION	W MATRIX INVALID FOR RENDEZVOUS NAVIGATION
RENDWBIT	=	BIT1		
FLAGWRD6	=	STATE +6	# (090-104) # (SET)	(RESET)
# BIT 15 FLAG 6 DAPBIT1	=	090D	# 1 SATURN 1 TVC	0 RCS 0 NO
DAP1BIT	=	BIT15		
# BIT 14 FLAG 6 DAPBIT2	=	091D	# 1 A/P 0 A/P	1 A/P 0 A/P
DAP2BIT	=	BIT14		
# BIT 13 FLAG 6 STRULLSW	=	092D	# DO STEERULL	DO ULAGEOFF ONLY
STRULBIT	=	BIT13		
# BIT 13 FLAG 6				

July 12, 2016

443

ENTRYDSP	=	STRULLSW	# DO ENTRY DISPLAY # VIA ENTRYVN.	OMIT ENTRY DISPLAY
ENDSPBIT	=	BIT13		
# BIT 12 FLAG 6 CMDAPARM	=	093D	# ALLOW ENTRY FIRINGS # AND CALCULATIONS	INHIBIT ENTRY FIRING AND CONTROL FUNCTION
# Page 58 CMDARMBIT	=	BIT12		
# BIT 11 FLAG 6 GAMDIFSW	=	094D	# CALCULATE GAMDOT #	GAMDOT NOT TO BE CALCULATED
GMDIFBIT	=	BIT11		
# BIT 10 FLAG 6 GONEPAST	=	095D	# LATERAL CONTROL # CALCULATIONS TO BE # OMITTED	LATERAL CONTROL CALCULATIONS TO BE DONE
GONEBIT	=	BIT10		
# BIT 9 FLAG 6 RELVELSW	=	096D	# TARGETING USES # EARTH-RELATIVE # VELOCITY.	TARGETING USES INERTIAL VELOCITY
RELVBIT	=	BIT9		
# BIT 8 FLAG 6 EGSW	=	097D	# IN FINAL PHASE	NOT IN FINAL PHASE
EGFLGBIT	=	BIT8		
# BIT 8 FLAG 6 KNOWNFLG	=	EGSW	# LANDMARK KNOWN	LANDMARK UNKNOWN
KNOWNBIT	=	BIT8		
# BIT 8 FLAG 6 R57FLAG	=	KNOWNFLG	# DO NOT DO R57 # TRUNION BIAS HAS # BEEN OBTAINED.	DO R57, TRUNION BIAS NEEDED

R57BIT = BIT8

BIT 7 FLAG 6
NOSWITCH = 098D

LATERAL ROLL LATERAL ROLL MANEUVER
MANEUVER INHIBITED PERMITTED IN ENTRY
IN ENTRY

NOSWBIT = BIT7

BIT 6 FLAG 6
HIND = 099D

ITERATING HUNTEST ITERATING OF HUNTEST
CALCULATIONS TO BE CALCULATIONS TO BE
DONE AFTER RANGE OMITTED AFTER RANGE
PREDICTION PREDICTION

Page 59

HINDBIT = BIT6

BIT 5 FLAG 6
INRLSW = 100D

INITIAL ROLL INITIAL ROLL
V(LV) V(LV)

INRLBIT = BIT5

ATTITUDE NOT HELD ATTITUDE HELD

BIT 4 FLAG 6
LATSW = 101D

DOWNLIFT NOT DOWNLIFT INHIBITED
INHIBITED

LATSWBIT = BIT4

BIT 3 FLAG 6
.05GSW = 102D

DRAG OVER .05G DRAG LESS THAN .05G

.05GBIT = BIT3

BIT 3 FLAG 6
= 102D

BIT 2 FLAG 6
CM/DSTBY = 103D

ENTRY DAP ACTIVATED ENTRY DAP NOT
ACTIVATED

CM/DSBIT = BIT2

July 12, 2016

445

# BIT 1 FLAG 6 GYMDIFSW	=	104D	# CDU DIFFERENCES AND # BODY RATES COMPUTED #	CDU DIFFERENCES AND BODY RATES NOT COMPUTED
GYMDIBIT	=	BIT1		
FLAGWRD7	=	STATE +7	# (105-119) # (SET)	(RESET)
# BIT 15 FLAG 7 TERMIFLG	=	105D	# TERMINATE R52	DO NOT TERMINATE R52
TERMIBIT	=	BIT15		
# BIT 14 FLAG 7 ITSWICH	=	106D	# ACCEPT NEXT LAMBERT # TPI SEARCH SOLUTION	TEST LAMBERT ANSWER AGAINST LIMITS
# Page 60				
ITSWBIT	=	BIT14		
# BIT 13 FLAG 7 IGNFLAG	=	107D	# TIG HAS ARRIVED	TIG HAS NOT ARRIVED
IGNFLBIT	=	BIT13		
# BIT 12 FLAG 7 ASTNFLAG	=	108D	# ASTRONAUT HAS # OKAYED IGNITION	ASTRONAUT HAS NOT OKAYED IGNITION
ASTNBIT	=	BIT12		
# BIT 11 FLAG 7 TIMRFLAG	=	109D	# CLOKTASK OPERATING	CLOKTASK INOPERATIVE
TIMRBIT	=	BIT11		
# BIT 10 FLAG 7 NORMSW	=	110D	# UNIT NORMAL INPUT # TO LAMBERT.	LAMBERT COMPUTE ITS OWN UNIT NORMAL.
NORMSBIT	=	BIT10		
# BIT 9 FLAG 7 RVSW	=	111D	# DO NOT COMPUTE FINAL	COMPUTE FINAL STATE

			# STATE VECTOR IN	VECTOR IN TIME-THETA
			# TIME-THETA	
RVSWBIT	=	BIT9		
# BIT 8 FLAG 7				
GONEBY	=	112D	# PASSED TARGET	APPROACHING TARGET
GONBYBIT	=	BIT8		
# BIT 7 FLAG 7				
	=	113D		
# BIT 6 FLAG 7				
V37FLAG	=	114D	# AVERAGEG (SERVICER)	AVERAGEG (SERVICER)
			# RUNNING	OFF
V37FLBIT	=	BIT6		
# BIT 5 FLAG 7				
	=	115D		
# Page 61				
	=	BIT5		
# BIT 4 FLAG 7				
UPLOCKFL	=	116D	# K-KBAR-K FAIL	NO K-KBAR-K FAIL
UPLOCBIT	=	BIT4		
# BIT 3 FLAG 7				
VERIFLAG	=	117D	# CHANGED WHEN V33E OCCURS AT END OF P27	
VERIFBIT	=	BIT3		
# BIT 2 FLAG 7				
ATTCHFLG	=	118D	# LM,CM ATTACHED	LM,CM NOT ATTACHED
ATTCHBIT	=	BIT2		
# BIT 1 FLAG 7				
TFFSW	=	119D	# CALCULATE TPERIGEE	CALCULATE TFF
TFFSWBIT	=	BIT1		
FLAGWRD8	=	STATE +8D	# (120-134)	
			# (SET)	(RESET)

July 12, 2016

447

# BIT 15 FLAG 8				
RPQFLAG	=	120D	# RPQ NOT COMPUTED	RPQ COMPUTED
RPQFLBIT	=	BIT15		
# BIT 14 FLAG 8				
	=	121D		
# BIT 13 FLAG 8				
NEWIFLG	=	122D	# FIRST PASS THROUGH	SUCCEEDING ITERATION
			# INTEGRATION	OF INTEGRATION
NEWIBIT	=	BIT13		
# BIT 12 FLAG 8				
CMOONFLG	=	123D	# PERMANENT CSM STATE	PERMANENT CSM STATE
			# IN LUNAR SPHERE	IN EARTH SPHERE
CMOONBIT	=	BIT12		
# BIT 11 FLAG 8				
LMOONFLG	=	124D	# PERMANENT LM STATE	PERMANENT LM STATE
			# IN LUNAR SPHERE	IN EARTH SPHERE
LMOONBIT	=	BIT11		
# Page 62				
# BIT 10 FLAG 8				
ADVTRK	=	125D	# ADVANCE GROUND TRACK	NOT ADVANCED
			# SIGHTING WANTED	GROUND TRACK
ADVTKBIT	=	BIT10		
# BIT 9 FLAG 8				
P39/79SW	=	126D	# P39/79 OPERATING	P38/78 OPERATING
P39SWBIT	=	BIT9		
# BIT 8 FLAG 8				
SURFFLAG	=	127D	# LM ON LUNAR SURFACE	LM NOT ON LUNAR
			#	SURFACE
SURFFBIT	=	BIT8		

# BIT 7 FLAG 8 INFINFLG	=	128D	# NO CONIC SOLUTION # (CLOSURE THROUGH # INFINITY REQUIRED).	CONIC SOLUTION EXISTS.
INFINBIT	=	BIT7		
# BIT 6 FLAG 8 ORDERSW	=	129D	# ITERATOR USES 2ND # ORDER MINIMUM MODE	ITERATOR USES 1ST ORDER STANDARD MODE
ORDERBIT	=	BIT6		
# BIT 5 FLAG 8 APSESW	=	130D	# RDESIRED OUTSIDE # PERICENTER-APOCENTER # RANGE IN TIME-RAD	RDESIRED INSIDE PERICENTER-APOCENTER RANGE IN TIME-RADIUS
APSESBIT	=	BIT5		
# BIT 4 FLAG 8 COGAFLAG	=	131D	# NO CONIC SOLUTION # TOO CLOSE TO # RECTILINEAR (COGA # OVERFLOWS).	CONIC SOLUTION EXISTS (COGA DOES NOT OVERFLOW).
COGAFBIT	=	BIT4		
# Page 63				
# BIT 3 FLAG 8 V96ONFLG	=	132D	# P00 INTEGRATION HAS # BEEN INHIBITED BY # V96	P00 INTEGRATION IS PROCEEDING REGULARLY
# BIT 2 FLAG 8	=	133D		
# BIT 1 FLAG 8 360SW	=	134D	# TRANSFER ANGLE NEAR # 360 DEGREES	TRANSFER ANGLE NOT NEAR 360 DEGREES
360SWBIT	=	BIT1		
FLAGWRD9	=	STATE +9D	# (135-149) # (SET)	(RESET)

July 12, 2016

449

# BIT 15 FLAG 9 SWTOVER	=	135D	# SWITCHOVER HAS # OCCURRED.	NO SWITCHOVER YET
SWTOVBIT	=	BIT15		
# BIT 14 FLAG 9	=	136D		
V67FLBIT	=	BIT14		
# BIT 13 FLAG 9 V82EMFLG	=	137D	# MOON VICINITY	EARTH VICINITY
V82EMBIT	=	BIT13		
# BIT 12 FLAG 9 MAXDBFLG	=	138D	# MAX DB SELECTED	MIN DB SELECTED
MAXDBBIT	=	BIT12		
# BIT 11 FLAG 9 V94FLAG	=	139D	# V94 ALLOWED DURING # P23	V94 NOT ALLOWED
V94FLBIT	=	BIT11		
# BIT 10 FLAG 9 SAVECFLG	=	140D	# P23 DISPLAY AND # DATA STORAGE AFTER	P23 DISPLAY AND DATA STORAGE BEFORE
# Page 64			# MARK IS DONE	MARK IS DONE
SAVECBIT	=	BIT10		
# BIT 9 FLAG 9 VHFRFLAG	=	141D	# ALLOW R22 TO # ACCEPT RANGE # DATA	STOP ACCEPTANCE OF RANGE DATA
VHFRBIT	=	BIT9		
# BIT 8 FLAG 9 SOURCFLG	=	142D	# SOURCE OF INPUT # DATA IS FROM # VHF RADAR	SOURCE OF INPUT DATA IS FROM OPTICS MARK

SOURCBIT	=	BIT8		
# BIT 7 FLAG 9 R22CAFLG	=	143D	# R-22 CALCULATIONS # ARE GOING ON	R-22 CALCULATIONS ARE NOT GOING ON
R22CABIT	=	BIT7		
# BIT 6 FLAG 9 N22ORN17	=	144D	# COMPUTE TOTAL # ATTITUDE ERRORS # W.R.T. N22 (V62)	COMPUTE TOTAL ATTITUDE ERRORS W.R.T. N17 (V63)
N2217BIT	=	BIT6		
# BIT 5 FLAG 9 QUITFLAG	=	145D	# TERMINATE AND EXIT	CONTINUE INTEGRATION
QUITBIT	=	BIT5	# FROM INTEGRATION	
# BIT 4 FLAG 9 R31FLAG	=	146D	# R31 SELECTED (V63)	R34 SELECTED (V65)
R31FLBIT	=	BIT4		
# BIT 3 FLAG 9 MID1FLAG	=	147D	# INTEGRATE TO TDEC #	INTEGRATE TO THE THEN-PRESENT TIME
MID1FBIT	=	BIT3		
# BIT 2 FLAG 9 MIDAVFLG	=	148D	# INTEGRATION ENTERED # FROM ONE OF MIDTOAV # PORTALS	INTEGRATION WAS NOT ENTERED VIA MIDTOAV
# Page 65				
MIDAVBIT	=	BIT2		
# BIT 1 FLAG 9 AVEMIDSW	=	149D	# AVETOMID CALLING # FOR W MATRIX INTEGR # DON'T WRITE OVER RN, # VN,PIPTIME	NO AVETOMID W INTEGE ALLOW SET UP RN,VN, PIPTIME
AVEMDBIT	=	BIT1		

FLGWRD10	=	STATE +10D	# (150-164) # (SET)	(RESET)
RASFLAG	=	STATE +10D		
# BIT 15 FLAG 10	=	150D		
# BIT 14 FLAG 10 INTFLAG	=	151D	# INTEGRATION IN # PROGRESS	INTEGRATION NOT IN PROGRESS
INTFLBIT	=	BIT14		
# BIT 13 FLAG 10	=	152D		
# BIT 12 FLAG 10	=	153D		
# BIT 11 FLAG 10	=	154D		
# BIT 10 FLAG 10	=	155D		
# BIT 9 FLAG 10	=	156D		
# BIT 8 FLAG 10	=	157D		
# Page 66				
# BIT 7 FLAG 10 REINTFLG	=	158D	# INTEGRATION ROUTINE # TO BE RESTARTED	INTEGRATION ROUTINE NOT TO BE RESTARTED
REINTBIT	=	BIT7		
# BIT 6 FLAG 10	=	159D		
# BIT 5 FLAG 10	=	160D		
# BIT 4 FLAG 10				

	=	161D		
# BIT 3 FLAG 10				
	=	162D		
# BIT 2 FLAG 10				
	=	163D		
# BIT 1 FLAG 10				
	=	164D		
FLGWRD11	=	STATE	+11D	# (165-179) # (SET) (RESET)
# BIT 15 FLAG 11				
S32.1F1	=	165D		# DELTAN AT CSI TIME DVT1 LESS THAN MAX
S32BIT1	=	BIT15		# ONE EXCEEDS MAX
# BIT 14 FLAG 11				
S32.1F2	=	166D		# FIRST PASS OF REITERATION OF
S32BIT2	=	BIT14		# NEWTON INTEGRATION NEWTON
# BIT 13 FLAG 11				
S32.1F3A	=	167D		# BIT 13 AND BIT 12 FUNCTION AS AN ORDERED
S32BIT3A	=	BIT13		# PAIR (13,12) INDICATING THE POSSIBLE OC-
# BIT 12 FLAG 11				# CURRENCE OF 2NEWTON ITERATIONS FOR S32.1
S32.1F3B	=	168D		# IN THE PROGRAM IN THE FOLLOWING ORDER:
# Page 67				# (0,1) (I.E. BIT 13 RESET, BIT 12 SET)
S3229T3B	=	BIT12		# = FIRST NEWTON ITERATION BEING DONE
				# (0,0)= FIRST PASS OF 2ND NEWTON ITER.
				# (1,1)= 50 FPS STAGE OF 2ND NEWT ITER.
				# (1,0)= REMAINDER OF 2ND NEWT ITER.
# BIT 11 FLAG 11				
	=	169D		
# BIT 10 FLAG 11				
	=	170D		
# BIT 9 FLAG 11				
	=	171D		
# BIT 8 FLAG 11				
	=	172D		

BIT 7 FLAG 11
= 173D

BIT 6 FLAG 11
= 174D

BIT 5 FLAG 11
= 175D

BIT 4 FLAG 11
= 176D

BIT 3 FLAG 11
= 177D

BIT 2 FLAG 11
= 178D

BIT 1 FLAG 11
= 179D

Page 68

GENERAL ERASABLE ASSIGNMENTS

SETLOC 61

INTERRUPT TEMPORARY STORAGE POOL. (11D)

(ITEMP1 THROUGH RUPTREG4)

ANY OF THESE MAY BE USED AS TEMPORARIES DURING INTERRUPT OR WITH INTERRUPT INHIBITED. THE ITE
IS USED DURING CALLS TO THE EXECUTIVE AND WAITLIST -- THE RUPTREGS ARE NOT.

ITEMP1 ERASE
WAITEXIT EQUALS ITEMP1
EXECTEM1 EQUALS ITEMP1

ITEMP2 ERASE
WAITBANK EQUALS ITEMP2
EXECTEM2 EQUALS ITEMP2

ITEMP3 ERASE
RUPTSTOR EQUALS ITEMP3
WAITADR EQUALS ITEMP3
NEWPRIO EQUALS ITEMP3

ITEMP4 ERASE

LOCCTR	EQUALS	ITEMP4	
WAITTEMP	EQUALS	ITEMP4	
ITEMP5	ERASE		
NEWLOC	EQUALS	ITEMP5	
ITEMP6	ERASE		
NEWLOC+1	EQUALS	ITEMP6	# DP ADDRESS.
	SETLOC	67	
NEWJOB	ERASE		# MUST BE AT LOC 67 DUE TO WIRING.
RUPTREG1	ERASE		
RUPTREG2	ERASE		
RUPTREG3	ERASE		
RUPTREG4	ERASE		
KEYTEMP1	EQUALS	RUPTREG4	
DSRUPTM	EQUALS	RUPTREG4	
# FLAGWORD RESERVATIONS.			(12D)
STATE	ERASE	+11D	
# PAD LOAD FOR DAPS			
EMDOT	ERASE		# I(1)PL (SPS FLOW RATE, SC AT B+3 KG/CS)
# Page 69			
# EXIT FOR V83			
STATEXIT	ERASE		# I(1) STQ ADDRESS FOR STATEXIT
# UNUSED ERASABLES ***** (2)			
ERASFILL	ERASE	+1	
# EXEC TEMPORARIES WHICH MAY BE USED BETWEEN CCS NEWJOBS			
# (INTB15+ THROUGH RUPTMXTM)			(32D)
INTB15+	ERASE		# REFLECTS 15TH BIT OF INDEXABLE ADDRESSES
DSEXIT	EQUALS	INTB15+	# RETURN FOR DSPIN
EXITEM	EQUALS	INTB15+	# RETURN FOR SCALE FACTOR ROUTINE SELECT
BLANKRET	EQUALS	INTB15+	# RETURN FOR 2BLANK
INTBIT15	ERASE		# SIMILAR TO ABOVE.
WRDRET	EQUALS	INTBIT15	# RETURN FOR 5BLANK.
WDRET	EQUALS	INTBIT15	# RETURN FOR DSPWD
DECRET	EQUALS	INTBIT15	# RETURN FOR PUTCOM(DEC LOAD)
21/22REG	EQUALS	INTBIT15	# TEMP FOR CHARIN

THE REGISTERS BETWEEN ADDRWD AND PRIORITY MUST STAY IN THE FOLLOWING ORDER FOR INTERPRETIVE T

ADDRWD	ERASE		# 12 BIT INTERPRETIVE OPERAND SUB-ADDRESS.
POLISH	ERASE		# HOLDS CADR MADE FROM POLISH ADDRESSE.
UPDATRET	EQUALS	POLISH	# RETURN FOR UPDATNN, UPDATVB
CHAR	EQUALS	POLISH	# TEMP FOR CHARIN
ERCNT	EQUALS	POLISH	# COUNTER FOR ERROR LIGHT RESET
DECOUNT	EQUALS	POLISH	# COUNTER FOR SCALING AND DISPLAY (DEC)
FIXLOC	ERASE		# WORK AREA ADDRESS
OVFIND	ERASE		# SET NON-ZERO ON OVERFLOW.
VBUF	ERASE	+5	# TEMPORARY STORAGE USED FOR VECTORS.
SGNON	EQUALS	VBUF	# TEMP FOR +,- ON
NOUNTEM	EQUALS	VBUF	# COUNTER FOR MIXNOUN FETCH
DISTEM	EQUALS	VBUF	# COUNTER FOR OCTAL DISPLAY VERB
DECTEM	EQUALS	VBUF	# COUNTER FOR FETCH (DEC DISPLAY VERBS)
SGNOFF	EQUALS	VBUF +1	# TEMP FOR +,- ON
NVTEMP	EQUALS	VBUF +1	# TEMP FOR NVSUB
SFTEMP1	EQUALS	VBUF +1	# STORAGE FOR SF CONST HI PART (=SFTEMP2-1)
HITEMIN	EQUALS	VBUF +1	# TEMP FOR LOAD OF HRS,MIN,SEC
			# MUST = LOTEMIN-1.
CODE	EQUALS	VBUF +2	# FOR DSPIN
SFTEMP2	EQUALS	VBUF +2	# STORAGE FOR SF CONST LO PART (=SFTEMP1+1)
LOTEMIN	EQUALS	VBUF +2	# TEMP FOR LOAD OF HRS,MIN,SEC
# Page 70			
			# MUST = HITEMIN+1
MIXTEMP	EQUALS	VBUF +3	# FOR MIXNOUN DATA
SIGNRET	EQUALS	VBUF +3	# RETURN FOR +,- ON

ALSO MIXTEMP+1 = VBUF+4, MIXTEMP+2 = VBUF+5

BUF	ERASE	+2	# TEMPORARY SCALAR STORAGE
BUF2	ERASE	+1	
INDEXLOC	EQUALS	BUF	# CONTAINS ADDRESS OF SPECIFIED INDEX.
SWWORD	EQUALS	BUF	# ADDRESS OF SWITCH WORD
SWBIT	EQUALS	BUF +1	# SWITCH BIT WITHIN THE SWITCH WORD
MPTEMP	ERASE		# TEMPORARY USED IN MULTIPLY AND SHIFT
DMPNTEMP	EQUALS	MPTEMP	# DMPSUB TEMPORARY
DOTINC	ERASE		# COMPONENT INCREMENT FOR DOT SUBROUTINE
DVSIGN	EQUALS	DOTINC	# DETERMINES SIGN OF DDV RESULT
ESCAPE	EQUALS	DOTINC	# USED IN ARCSIN/ARCCOS.
ENTRET	EQUALS	DOTINC	# EXIT FROM ENTER

DOTRET	ERASE		# RETURN FROM DOT SUBROUTINE
DVNORMCT	EQUALS	DOTRET	# DIVIDENT NORMALIZATION COUNT IN DDV.
ESCAPE2	EQUALS	DOTRET	# ALTERNATE ARCSIN/ARCCOS SWITCH
WDCNT	EQUALS	DOTRET	# CHAR COUNTER FOR DSPWD
INREL	EQUALS	DOTRET	# INPUT BUFFER SELECTION (X,Y,Z, REG)
MATINC	ERASE		# VECTOR INCREMENT IN MXV AND VXM
MAXDVSW	EQUALS	MATINC	# +0 IF DP QUOTIENT IS NEAR ONE -- ELSE -1.
POLYCNT	EQUALS	MATINC	# POLYNOMIAL LOOP COUNTER
DSPMMTEM	EQUALS	MATINC	# DSPCOUNT SAVE FOR DSPMM
MIXBR	EQUALS	MATINC	# INDICATOR FOR MIXED OR NORMAL NOUN
TEM1	ERASE		# EXEC TEMP
POLYRET	EQUALS	TEM1	
DSREL	EQUALS	TEM1	# REL ADDRESS FOR DSPIN
TEM2	ERASE		# EXEC TEMP
DSMAG	EQUALS	TEM2	# MAGNITUDE STORE FOR DSPIN
IDADITEM	EQUALS	TEM2	# MIXNOUN INDIRECT ADDRESS STORAGE
TEM3	ERASE		# EXEC TEMP
COUNT	EQUALS	TEM3	# FOR DSPIN
TEM4	ERASE		# EXEC TEMP
LSTPTR	EQUALS	TEM4	# LIST POINTER FOR GRABUSY
RELRET	EQUALS	TEM4	# RETURN FOR RELDSP
FREERET	EQUALS	TEM4	# RETURN FOR FREEDSP
DSPWDRET	EQUALS	TEM4	# RETURN FOR DSPSIGN
SEPSCRET	EQUALS	TEM4	# RETURN FOR SEPSEC
SEPMNRET	EQUALS	TEM4	# RETURN FOR SEPMIN
TEM5	ERASE		# EXEC TEMP
# Page 71			
NOUNADD	EQUALS	TEM5	# TEMP STORAGE FOR NOUN ADDRESS
NNADTEM	ERASE		# TEMP FOR NOUN ADDRESS TABLE ENTRY
NNTYPTTEM	ERASE		# TEMP FOR NOUN TYPE TABLE ENTRY
IDAD1TEM	ERASE		# TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
			# MUST = IDAD2TEM-1, = IDAD3TEM-2
IDAD2TEM	ERASE		# TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
			# MUST = IDAD1TEM+1, IDAD3TEM-1.
IDAD3TEM	ERASE		# TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
			# MUST = IDAD1TEM+2, IDAD2TEM+1.
RUTMXTEM	ERASE		# TEMP FOR SF ROUT TABLE ENTRY (MIXNN ONLY)


```

# AX*SR*T STORAGE.                                (6D)
DEXDEX      EQUALS  TEM2      # B(1)TMP
DEX1        EQUALS  TEM3      # B(1)TMP
DEX2        EQUALS  TEM4      # B(1)TMP
RTNSAVER    EQUALS  TEM5      # B(1)TMP
TERM1TMP    EQUALS  BUF2      # B(2)TMP

```

```

DEXI        =          DEX1

```

```

# Page 72

```

```

# DYNAMICALLY ALLOCATED CORE SETS FOR JOBS          (84D)

```

```

MPAC        ERASE    +6          # MULTI-PURPOSE ACCUMULATOR.
MODE        ERASE          # +1 FOR TP, +0 FOR DP, OR -1 FOR VECTOR.
LOC         ERASE          # LOCATION ASSOCIATED WITH JOB.
BANKSET     ERASE          # USUALLY CONTAINS BBANK SETTING.
PUSHLOC     ERASE          # WORD OF PACKED INTERPRETIVE PARAMETERS.
PRIORITY    ERASE          # PRIORITY OF PRESENT JOB AND WORK AREA.

                ERASE    +71D      # SEVEN SETS OF 12 REGISTERS EACH

```

```

# SPECIAL DOWNLINK BUFFER. -- OVERLAYED BY P27 STORAGE --

```

```

# P27 (UPDATE PROGRAM) STORAGE. -- OVERLAYS SPEC DNLNK BUFF -- (24D)

```

```

COMPNUMB    ERASE    +23D          # B(1)TMP NUMBER OF ITEMS TO BE UPLINKED.
UPOLDMOD    EQUALS  COMPNUMB +1    # B(1)TMP HOLDS INTERRUPTED PROGRAM NUMBER
UPVERB      EQUALS  UPOLDMOD +1    # B(1)TMP VERB NUMBER
UPCOUNT    EQUALS  UPVERB +1     # B(1)TMP UPBUFF INDEX
UPBUFF      EQUALS  UPCOUNT +1     # B(20D)

```

```

# MORE P27 STORAGE.                                (2D)

```

```

UPTMP       ERASE          # B(1)TMP SCRATCH
UPVERBSV    ERASE          # B(1)TMP
INTWAK1Q    EQUALS  UPTMP      # (06D)
# (20 REGISTERS OF ENTRY DOWNLINK WILL GO HERE.)

```

```

# THE FOLLOWING ARE INDEXED FOR TM. IN ENTRY DAP.

```

```

CMTMTIME    =          UPBUFF      # B(1) (VEHICLE BODY RATE INFO IS
SW/NDX      =          CMTMTIME +1 # B(1)  TELEMETERED EACH 0.2 SEC. DURING
ENDBUF      =          CMTMTIME +16D # B(1)  ENTRY.)

```

```

V1          =          ENDBUF +1    # I(2) REENTRY, P64-P65
A0          =          V1 +2        # I(2) REENTRY, P64-P65

```

HI-ORDER WORD ONLY ON DNLNK.

ALIGNMENT STORAGE. (5D)
(CANNOT SHARE WITH PRECISION INTEGRATION OR KEPLER STORAGE.)

QMAJ	EQUALS	COMPNUMB	# B(1)TMP
MARKINDX	EQUALS	QMAJ +1	# B(1)TMP
BESTI	EQUALS	MARKINDX +1	# I(1)TMP
BESTJ	EQUALS	BESTI +1	# I(1)TMP
STARIND	EQUALS	BESTJ +1	# I(1)TMP

Page 73

ALIGNMENT/S40.2,3 COMMON STORAGE. (18D)

XSMD	EQUALS	UPBUFF +2	# I(6)TMP
YSMD	EQUALS	XSMD +6	# I(6)TMP
ZSMD	EQUALS	YSMD +6	# I(6)TMP

XSCREF	=	XSMD	# SPACE CRAFT AXES IN REF COORDS.
YSCREF	=	YSMD	
ZSCREF	=	ZSMD	
ZPRIME	=	22D	
PDA	=	22D	
COSTH	=	16D	
SINTH	=	18D	
THETA	=	20D	
STARM	=	32D	

DOWNLINK STORAGE (18D)

DNLSTADR	EQUALS	DNLSTCOD	# CONTENTS NO LONGER AN ADDR BUT A CODE
DNLSTCOD	ERASE		# B(1)PRM ID CODE OF DOWNLIST
DUMPCNT	ERASE		# B(1)PRM
LDATA1ST	ERASE		# B(1)
DNTMGOTO	ERASE		# B(1)
TMINDEX	ERASE		# B(1)
DUMPLOC	EQUALS	TMINDEX	# CONTAINS ECADR OF AGC DP WORD BEING DUMPED
			# AND COUNT OF COMPLETE DUMPS ALREADY
			# SENT.
DNQ	ERASE		# B(1)
DNTMBUFF	ERASE	+11D	# B(12)PRM DOWNLINK SNAPSHOT BUFFER

OPTICS MARKING, UNSHARED. (8D)

MKNDX	ERASE
-------	-------

```

MKT2T1      ERASE   +1
MKCDUY      ERASE
MKCDUS      ERASE
MKCDUZ      ERASE
MKCDUT      ERASE
MKCDUX      ERASE

```

```

# FOR EXCLUSIVE USE OF SYS TEST STANDARD LEAD INS      (2)
EBUF2       ERASE   +1      # B(2) UNSHARED

```

```

# Page 74

```

```

# UNSWITCHED FOR DISPLAY INTERFACE ROUTINES.      (10D)

```

```

RESTREG      ERASE      # B(1)PRM FOR DISPLAY RESTARTS.
NVWORD       ERASE
MARKNV       ERASE
NVSAVE       ERASE
# (RETAIN THE ORDER OF CADRFLSH TO FAILREG +2 FOR DOWNLINK PURPOSES)
CADRFLSH     ERASE      # B(1)TMP
CADRMARK     ERASE      # B(1)TMP
TEMPFLSH     ERASE      # B(1)TMP
FAILREG      ERASE   +2  # B(3)PRM 3 ALARM CODE REGISTERS

```

```

      SETLOC 400

```

```

# VAC AREAS. -- BE CAREFUL OF PLACEMENT --      (220D)

```

```

VAC1USE      ERASE      # B(1)PRM
VAC1         ERASE   +42D  # B(43)PRM
VAC2USE      ERASE      # B(1)PRM
VAC2         ERASE   +42D  # B(43)PRM
VAC3USE      ERASE      # B(1)PRM
VAC3         ERASE   +42D  # B(43)PRM
VAC4USE      ERASE      # B(1)PRM
VAC4         ERASE   +42D  # B(43)PRM
VAC5USE      ERASE      # B(1)PRM
VAC5         ERASE   +42D  # B(43)PRM

```

```

# WAITLIST REPEAT FLAG.      (1D)
RUPTAGN      ERASE      # B(1)PRM
KEYTEMP2     =          RUPTAGN

```

```

# STARALIGN ERASABLES.      (13D)

```

```

STARCODE     ERASE      # B(1)DSP NOUN 70 FOR P22,51 AND R52,53
STARALGN     ERASE   +11D

```

SINCDU	=	STARALGN
COSCDU	=	STARALGN +6

SINCDUX	=	SINCDU +4
SINCDUY	=	SINCDU
SINCDUZ	=	SINCDU +2
COSCDUX	=	COSCDU +4
COSCDUY	=	COSCDU
COSCDUZ	=	COSCDU +2

PHASE TABLE AND RESTART COUNTERS (12D)

Page 75

-PHASE1	ERASE	# B(1)PRM
PHASE1	ERASE	# B(1)PRM
-PHASE2	ERASE	# B(1)PRM
PHASE2	ERASE	# B(1)PRM
-PHASE3	ERASE	# B(1)PRM
PHASE3	ERASE	# B(1)PRM
-PHASE4	ERASE	# B(1)PRM
PHASE4	ERASE	# B(1)PRM
-PHASE5	ERASE	# B(1)PRM
PHASE5	ERASE	# B(1)PRM
-PHASE6	ERASE	# B(1)PRM
PHASE6	ERASE	# B(1)PRM

A**SR*T STORAGE (6D)

CDUSPOT	ERASE	+5	# B(6)
---------	-------	----	--------

CDUSPOTY	=	CDUSPOT
CDUSPOTZ	=	CDUSPOT +2
CDUSPOTX	=	CDUSPOT +4

VERB 37 STORAGE (2D)

MINDEX	ERASE	# B(1)TMP INDEX FOR MAJOR MODE
MMNUMBER	ERASE	# B(1)TMP MAJOR MODE REQUESTED BY V37

PINBALL INTERRUPT ACTION (1D)

DSPCNT	ERASE	# B(1)PRM COUNTER FOR DSPOUT
--------	-------	------------------------------

PINBALL EXECUTIVE ACTION (44D)

DSPCOUNT	ERASE	# DISPLAY POSITION INDICATOR
----------	-------	------------------------------

DECBRNCH	ERASE		# +DEC, -DEC, OCT INDICATOR
VERBREG	ERASE		# VERB CODE
NOUNREG	ERASE		# NOUN CODE
XREG	ERASE		# R1 INPUT BUFFER
YREG	ERASE		# R2 INPUT BUFFER
ZREG	ERASE		# R3 INPUT BUFFER
XREGLP	ERASE		# LO PART OF XREG (FOR DEC CONV ONLY)
YREGLP	ERASE		# LO PART OF YREG (FOR DEC CONV ONLY)
HITEMOUT	=	YREGLP	# TEMP FOR DISPLAY OF HRS,MIN,SEC # MUST = LITEMOUT-1.
ZREGLP	ERASE		# LO PART OF ZREG (FOR DEC CONV ONLY)
LOTEMOUT	=	ZREGLP	# TEMP FOR DISPLAY OF HRS,MIN,SEC # MUST = HITEMOUT+1
# Page 76			
MODREG	ERASE		# MODE CODE
DSPLOCK	ERASE		# KEYBOARD/SUBROUTINE CALL INTERLOCK
REQRET	ERASE		# RETURN REGISTER FOR LOAD
LOADSTAT	ERASE		# STATUS INDICATOR FOR LOADTST
CLPASS	ERASE		# PASS INDICATOR FOR CLEAR
NOUT	ERASE		# ACTIVITY COUNTER FOR DSPTAB
NOUNCADR	ERASE		# MACHINE CADR FOR NOUN
MONSAVE	ERASE		# N/V CODE FOR MONITOR. (= MONSAVE1-1)
MONSAVE1	ERASE		# NOUNCADR FOR MONITOR (MATBS) = MONSAVE+1
MONSAVE2	ERASE		# B(1)PRM NVMONOPT OPTIONS
DSPTAB	ERASE	+11D	# 0-100, DISPLAY PANEL BUFF. 11D, C/S LTS.
NVQTEM	ERASE		# NVSUB STORAGE FOR CALLING ADDRESS # MUST = NVBNKTEM-1.
NVBNKTEM	ERASE		# NVSUB STORAGE FOR CALLING BANK # MUST = NVQTEM+1
VERBSAVE	ERASE		# NEEDED FOR RECYCLE
CADRSTOR	ERASE		# ENDIDLE STORAGE
DSPLIST	ERASE		# WAITING REG FOR DSP SYST INTERNAL USE
EXTVBACT	ERASE		# EXTENDED VERB ACTIVITY INTERLOCK
DSPTM1	ERASE	+2	# BUFFER STORAGE AREA 1 (MOSTLY FOR TIME)
DSPTM2	ERASE	+2	# BUFFER STORAGE AREA 2 (MOSTLY FOR DEG)
DSPTMX	EQUALS	DSPTM2 +1	# B(2) S-S DISPLAY BUFFER FOR EXT. VERBS
NORMTEM1	EQUALS	DSPTM1	# B(3)DSP NORMAL DISPLAY REGISTERS.
# DISPLAY FOR EXTENDED VERBS			(2D)
OPTIONX	EQUALS	DSPTMX	# B(2) EXTENDED VERB OPTION CODE N12(V82)
# TBASE'S AND PHSPRDT'S.			(12D)
TBASE1	ERASE		# B(1)PRM

PHSPRDT1	ERASE	# B(1)PRM
TBASE2	ERASE	# B(1)PRM
PHSPRDT2	ERASE	# B(1)PRM
TBASE3	ERASE	# B(1)PRM
PHSPRDT3	ERASE	# B(1)PRM
TBASE4	ERASE	# B(1)PRM
PHSPRDT4	ERASE	# B(1)PRM
TBASE5	ERASE	# B(1)PRM
PHSPRDT5	ERASE	# B(1)PRM
TBASE6	ERASE	# B(1)PRM
PHSPRDT6	ERASE	# B(1)PRM

UNSWITCHED FOR DISPLAY INTERFACE ROUTINES. (5D)

Page 77

EBANKSAV	ERASE
MARKEBAN	ERASE
EBANKTEM	ERASE
MARK2PAC	ERASE
R1SAVE	ERASE

IMU COMPENSATION UNSWITCHED ERASABLE. (1D)

1/PIPADT	ERASE	# B(1)PRM
OLDBT1	=	1/PIPADT

SINGLE PRECISION SUBROUTINE TEMPORARIES (3D)

SPSIN, SPCOS, SPROOT VARIABLES.

DO NOT SHARE. THESE ARE USED BY DAPS IN I

AND CURRENTLY ARE NOT PROTECTED. IF OTHER

MATERIALIZE, THEN THIS CAN BE CHANGED.

HALFY	ERASE
ROOTRET	ERASE
SQRARG	ERASE
TEMK	EQUALS HALFY
SQ	EQUALS ROOTRET

Page 78

UNSWITCHED FOR ORBIT INTEGRATION (21D)

TDEC	ERASE	+20D	# I(2)
COLREG	EQUALS	TDEC +2	# I(1)
LAT	EQUALS	COLREG +1	# I(2)DSP NOUN 43,67 FOR P20,22,51 R52,53.
LANDLAT	=	LAT	# NOUN 89 FOR P22.
LONG	EQUALS	LAT +2	# I(2)DSP NOUN 43,67 FOR P20,22,51 R52,53
ALT	EQUALS	LONG +2	# I(2)DSP NOUN 43 FOR P20,22,51 R52,53.

July 12, 2016

463

YV	EQUALS	ALT +2	# I(6)
ZV	EQUALS	YV +6	# I(6)
# MARK STORAGE.			(2)
VHFCNT	ERASE		# B(1)PRM NO. OF VHF MARKS (P20 (R22)).
TRKMKCNT	ERASE		# B(1)PRM NO. OF VHF MARKS (P20 (R22)).
MARKCTR	=	TRKMKCNT	# B(1) MARK COUNTER USED BY R32
# MISCELLANEOUS UNSWITCHED.			(16D)
IRETURN1	ERASE		# B(1) RET ADDR USED BY MIDTOAV1 AND 2
			# CALLED BY P40,P41,P42, P61,P62
RATEINDX	ERASE		# (1) USED BY KALCMANU
OPTION1	ERASE		# B(1) NOUN 06 USES THIS.
OPTION2	ERASE		# B(1) NOUN 06 USES THIS.
LONGCADR	ERASE	+1	# B(2) LONGCALL REGISTER.
LONGBASE	ERASE	+1	# B(2) LONGCALL REGISTER.
LONGTIME	ERASE	+1	# B(2) LONGCALL REGISTER.
DELAYLOC	ERASE	+3	
NVWORD1	ERASE		# B(1)
TEMPR60	ERASE		# B(1)
PRIOTIME	ERASE		# B(1)
P30/RET	EQUALS	IRETURN1	
# MISC. INCLUDING RESTART COUNTER, GIMBAL ANGLE SAVE AND			
# STANDBY VERB ERASABLES. REDOCTR BEFORE THETAD (DWNLNK)			(16D)
TIME2SAV	ERASE	+1	# B(2)TMP
SCALSAVE	ERASE	+1	# B(2)TMP
REDOCTR	ERASE		# B(1)PRM CONTAINS NUMBER OF RESTARTS
THETAD	ERASE	+2	# B(3)PRM DESIRED GIM ANGLES FOR MANEUVER
CPHI	=	THETAD	# (OUTER)
CTHETA	=	THETAD +1	# (INNER)
CPSI	=	THETAD +2	# (MIDDLE)
# Page 79			
# ENTRY VARIABLES SHARED FOR TM.			
RDOTREF	=	THETAD	# I(2) P65
VREF	=	RDOTREF +2	# I(2) P65 HI-ORDER WORD ONLY DNLNK'D
DESOPTT	ERASE		# B(1)DSP NOUN 92 FOR P20,22,52, R52.
DESOPTS	ERASE		# B(1)DSP NOUN 92 FOR P20,22,52, R52.
DELV	ERASE	+5	# I(6)
DELVX	=	DELV	
DELVY	=	DELV +2	

```

DELVZ          =          DELV      +4

# P20, CONICS   (SHARING WITH TIME 2 SAV AND SCAL SAV ONLY)      (3D)
POINTEX        EQUALS    TIME2SAV      # I(1) POINT AXS EXIT
VHFTIME        EQUALS    POINTEX +1     # I(2) DOWNLINK OF VHF RANGE TIME +1M

# PERM STATE VECTORS FOR BOOST AND DOWNLINK -- WHOLE MISSION -- (14D)

RN             ERASE     +5             # B(6)PRM
VN             ERASE     +5             # B(6)PRM
PIPTIME        ERASE     +1             # B(2)PRM (MUST BE FOLLOWED BY GDT/2)

# SERVICER STORAGE.                                           (45D)

# (SERVICER STORAGE AND P11 STORAGE IN UNSWITCHED SHOULD NOT
# OVERLAY EACH OTHER AND THE TOTAL ERASABLE REQUIRED SHOULD NOT
# EXCEED THE ERASABLE STORAGE REQUIRED BY RENDEZVOUS GUIDANCE.)

GDT/2          EQUALS    PIPTIME +2     # B(6)TMP      ** MUST FOLLOW PIPTIME **
GOBL/2         EQUALS    GDT/2  +6     # B(6)TMP
AVEGEXIT       EQUALS    GOBL/2  +6     # B(2)TMP
AVEGEXIT       =        AVEGEXIT
TEMX           EQUALS    AVEGEXIT +2    # B(1)TMP
TEMY           EQUALS    TEMX  +1       # B(1)TMP
TEMZ           EQUALS    TEMY  +1       # B(1)TMP
PIPCTR         EQUALS    TEMZ  +1       # B(1)TMP
PIPAGE         EQUALS    PIPCTR +1      # B(1)TMP
RN1            EQUALS    PIPAGE +1      # B(6)TMP
VN1            EQUALS    RN1  +6        # B(6)TMP
PIPTIME1       EQUALS    VN1  +6        # B(2)TMP
GDT1/2         EQUALS    PIPTIME1 +2    # B(6)TMP
GOBL1/2        EQUALS    GDT1/2 +6     # B(6)TMP

# Page 80
# ENTRY STORAGE                                           (1D)
ENTRYVN        EQUALS    GOBL1/2 +6     # B(1)TMP VN CODE FOR ENTRY DISPLAYS P60'S.

# P11 STORAGE.                                           (9D)
PADLONG        EQUALS    ENTRYVN        # (2)PL LONGITUDE OF LAUNCH PAD.
LIFTTEMP       EQUALS    PADLONG +2     # (2)TMP
TEPHEM1        EQUALS    LIFTTEMP +2    # (3)TMP
PGNCSALT       EQUALS    TEPHEM1 +3     # (2)PL ALTITUDE

# RENDEZVOUS NAVIGATION STORAGE. (SEE COMMENT IN SERVICER STORAGE) (58D)
CSMPOS         ERASE     +57D          # I(6)TMP
LEMPOS         EQUALS    CSMPOS  +6     # I(6)TMP

```


RCL	EQUALS	LEMPOS	+6	# I(2)TMP
MARKTIME	EQUALS	RCL	+2	# B(2)TMP
VTEMP	EQUALS	MARKTIME	+2	# B(6)TMP
UM	EQUALS	VTEMP	+6	# I(6)TMP
MARKDATA	EQUALS	UM	+6	# B(2)TMP
USTAR	EQUALS	MARKDATA	+2	# I(6)TMP
WIXA	EQUALS	USTAR	+6	# B(1)TMP
WIXB	EQUALS	WIXA	+1	# B(1)TMP
ZIXA	EQUALS	WIXB	+1	# B(1)TMP
ZIXB	EQUALS	ZIXA	+1	# B(1)TMP
DELTAX	EQUALS	ZIXB	+1	# I(18)TMP
VHFRANGE	EQUALS	DELTAX		# (2)
UCL	EQUALS	DELTAX	+12D	# (6) LM-CSM LINE OF SIGHT 1/2 UNIT V
# ***** CONICSEX (MEAS INC) *****				
TRIPA	EQUALS	DELTAX		
TEMPVAR	EQUALS	DELTAX	+3	
TEMPOR1	ERASE	+1		# B(2)TMP
# T4RUPT ERASABLE				(6D)
DSRUPTSW	ERASE			
OPTIND	ERASE			
LGYRO	ERASE			
COMMANDO	ERASE	+1		
# Page 81				
ZONE	ERASE			# B(1)PRM USED IN SHAFT STOP MONITOR
LASTYCND	=	OPTY		# DUMMY TO MAKE RR BENCH TEST ASSEMBLE
LASTXCND	=	OPTY		# DUMMY TO MAKE RR BENCH TEST ASSEMBLE
# UNSWITCHED DAP ERASABLE.				(4D)
T6LOC	ERASE			
T6ADR	ERASE			
T5LOC	ERASE	+1		
# MODE SWITCHING ERASABLE				(14D)
SWSAMPLE	ERASE			# B(1)PRM
DESOPMOD	ERASE			# B(1)PRM
WTOPTION	ERASE			# B(1)PRM
ZOPTCNT	ERASE			# B(1)PRM
IMODES30	ERASE			# B(1)PRM
IMODES33	ERASE			# B(1)PRM
MODECADR	ERASE	+2		# B(3)TMP

```

IMUCADR      =      MODECADR
OPTCADR      =      MODECADR +1
RADCADR      =      MODECADR +2
ATTCADR      ERASE   +2      # B(3)PRM
ATTPRIO      =      ATTCADR +2
MARKSTAT     ERASE
OPTMODES     ERASE      # B(1)PRM

# RCSDAP ERASABLE      (1D)
HOLDFLAG     ERASE      # B(1)PRM

# CRS61.1 STORAGE.  -- USED IN R63 (VERB 89) -- (5D)
CPHIX        ERASE   +2      # B(3)DSP NOUN 95 CALCULATED BY CRS61.1

TEVENT       ERASE   +1      # B(2) TIME OF EVENT FOR DOWNLIST
TLIFTOFF     =      TEVENT

# Page 82
# P34-P35 STORAGE      (1D)
NORMEX       ERASE

# SELF-CHECK ASSIGNMENTS      (17D)

SELFERAS     ERASE   1357 - 1377  # *** MUST NOT BE MOVED *** #
SFMAIL       EQUALS  SELFERAS      # B(1)
ERESTORE     EQUALS  SFMAIL +1      # B(1)
SELFRET      EQUALS  ERESTORE +1    # B(1) RETURN
SMODE        EQUALS  SELFRET +1     # B(1)
ALMCADR      EQUALS  SMODE +1       # B(2) ALARM ABORD USER'S 2CADR
ERCOUNT     EQUALS  ALMCADR +2     # B(1)
SCOUNT       EQUALS  ERCOUNT +1     # B(3)
SKEEP1       EQUALS  SCOUNT +3      # B(1)
SKEEP2       EQUALS  SKEEP1 +1      # B(1)
SKEEP3       EQUALS  SKEEP2 +1      # B(1)
SKEEP4       EQUALS  SKEEP3 +1      # B(1)
SKEEP5       EQUALS  SKEEP4 +1      # B(1)
SKEEP6       EQUALS  SKEEP5 +1      # B(1)
SKEEP7       EQUALS  SKEEP6 +1      # B(1)

# USED BY P30 ROUTINES TO WRITE ONLY NEVER READ IN COLOSSUS

DISPDEX      EQUALS  A

# ERASABLE FOR SXTMARK CDU CHECK DELAY.  -- PAD LOADED --      (1D)

CDUCHKWD     ERASE      # B(1)PL

```

R57 STORAGE. -- MUST BE UNSHARED EXCEPT IN BOOST OR ENTRY -- (1D)

TRUNBIAS ERASE # B(1)PRM RESULT OF R57 CALIBR OF TRUNION

KEPLER STORAGE (6D)

XMODULO ERASE +1 # I(2) GREATER 2PI KEPLER

TMODULO ERASE +1 # I(2) GREATER 2 KEPLER

EPSILONT ERASE +1 # I(2)TMP

Page 83

P37 ** RETURN TO EARTH (PAD LOAD **** (2D)

RTED1 ERASE +1 # I(2)PL VGAMMA POLY COEF B-3

P40 *** STEERING ROUTINE *** PAD LOAD (1D)

DVTHRESH ERASE # I(1)PL DELTA VTHRESHOLD FOR LOW THRUST
ROUTINE B-2

P23 *** PAD LOAD **** (2D)

HORIZALT ERASE +1 # I(2)PL HORIZON ALTITUDE M B-29

P20 ALTERNATE LOS VARIANCE PAD LOAD **** (1D)

-16

ALTVAR ERASE # I(2)PL MILLARD, SQUARED SCALED 2

END-UE EQUALS SELFERS +16D # LAST USED UNSWITCHED ERASABLE

Page 84

EBANK-3 ASSIGNMENTS

SETLOC 1400

WAITLIST TASK LISTS. (26D)

LST1 ERASE +7 # B(8D)PRM DELTA T'S.

LST2 ERASE +17D # B(18D)PRM TASK 2CADR ADDRESSES.

RESTART STORAGE. (2D)

RSBBQ ERASE +1 # B(2)PRM SAVE BB AND Q FOR RESTARTS

MORE LONGCALL STORAGE. (MUST BE IN LST1'S BANK. (2D)

LONGEXIT	ERASE	+1	# B(2)TMP MAY BE SELDOM OVERLAYED
# PHASE-CHANGE LISTS PART II.			(12D)
PHSNAME1	ERASE		# B(1)PRM
PHSBB1	ERASE		# B(1)PRM
PHSNAME2	ERASE		# B(1)PRM
PHSBB2	ERASE		# B(1)PRM
PHSNAME3	ERASE		# B(1)PRM
PHSBB3	ERASE		# B(1)PRM
PHSNAME4	ERASE		# B(1)PRM
PHSBB4	ERASE		# B(1)PRM
PHSNAME5	ERASE		# B(1)PRM
PHSBB5	ERASE		# B(1)PRM
PHSNAME6	ERASE		# B(1)PRM
PHSBB6	ERASE		# B(1)PRM
# IMU COMPENSATION PARAMETERS			(22D)
PBIASX	ERASE		# B(1) PIPA BIAS, PIPA SCALE FACTOR TERMS
PIPABIAS	=	PBIASX	# INTERMIXED.
PIPASCFX	ERASE		
PIPASCF	=	PIPASCFX	
PBIASY	ERASE		
PIPASCFY	ERASE		
PBIASZ	ERASE		
PIPASCFZ	ERASE		
NBDX	ERASE		# GYRO BIAS DRIFT
GBIASX	=	NBDX	
NBDY	ERASE		
# Page 84			
NBDZ	ERASE		
ADIAX	ERASE		# ACCELERATION SENSITIVE DRIFT ALONG THE
ADIAY	ERASE		# INPUT AXIS
ADIAZ	ERASE		
ADSRAX	ERASE		# ACCELERATION SENSITIVE DRIFT ALONG THE
ADSRAY	ERASE		# SPIN REFERENCE AXIS
ADSRAZ	ERASE		
GCOMP	ERASE	+5	# CONTAINS COMPENSATING TORQUES
GCOMP SW	ERASE		

```

COMMAND      EQUALS  GCOMP
CDUIND        EQUALS  GCOMP  +3

```

```

# STATE VECTORS FOR ORBIT INTEGRATION.  (44D)

```

```

#           (DIFEQCNT THUR XKEP MUST BE IN THE SAME
#           EBANK AS RRECTCSM, RRECTLEM ETC
#           BECAUSE THE COPY CYCLES (ATOPCSM,
#           PTOACSM ETC) ARE EXECUTED IN BASIC.
#           ALL OTHER REFERENCES TO THIS GROUP
#           ARE BY INTERPRETIVE INSTRUCTIONS.)
#

```

```

DIFEQCNT      ERASE  +43D          # B(1)TMP
# (UPSVFLAG...XKEP MUST BE KEPT IN ORDER).

```

```

UPSVFLAG      EQUALS  DIFEQCNT +1    # B(1)PRM UPDATE FLAG
RRECT          EQUALS  UPSVFLAG +1    # B(6)TMP POS AT RECT      KM*2(-14)
VRECT          EQUALS  RRECT  +6      # B(6)TMP VEL AT RECT      KM(-1/2)*2(6)
TET            EQUALS  VRECT  +6      # B(2)TMP TIME OF STATE VECT  CSPCS*2(-28)
TDELTA V      EQUALS  TET    +2      # B(6)TMP POSITION DEVIATION  KM*2(14)
TNUV           EQUALS  TDELTA V +6    # B(6)TMP VEL DEVIATION     KM(-1/2)*2(14)
RCV            EQUALS  TNUV   +6      # B(6)TMP CONIC POSITION      KM*2(-14)
VCV            EQUALS  RCV    +6      # B(6)TMP CONIC VELOCITY     KM(-1/2)*2(6)
TC             EQUALS  VCV    +6      # B(2)TMP TIME SINCE RECITIFICATION
XKEP           EQUALS  TC     +2      # B(2)TMP ROOT OF KEPLER EQ  KM(1/2)*2(-10)

```

```

# **** TEMP -- IN VAC AREA ****

```

```

RRECT1        EQUALS  18D
VRECT1        EQUALS  24D
TET1          EQUALS  30D

```

```

# PERMANENT STATE VECTORS AND TIMES.  (101D)

```

```

# (DO NOT OVERLAY WITH ANYTHING AFTER BOOST)

```

```

# Page 86

```

```

# (RRECTCSM...XKEPCSM MUST BE KEPT IN THIS ORDER)

```

```

RRECTCSM      ERASE  +5          # B(6)PRM CSM VARIABLES
RRECTHIS      =      RRECTCSM
VRECTCSM      ERASE  +5          # B(6)PRM
TETCSM        ERASE  +1          # B(2)PRM
TETHIS        =      TETCSM
DELTACSM      ERASE  +5          # B(6)PRM
NUVCSM        ERASE  +5          # B(6)PRM

```

RCVCSM	ERASE	+5	# B(6)PRM
VCVCSM	ERASE	+5	# B(6)PRM
TCCSM	ERASE	+1	# B(2)PRM
XKEPCSM	ERASE	+1	# B(2)PRM

(RRECTLEM...XKEPLEM MUST BE KEPT IN THIS ORDER)

RRECTLEM	ERASE	+5	# B(6)PRM LEM VARIABLES
RRECTOTH	=	RRECTLEM	
VRECTLEM	ERASE	+5	# B(6)PRM
TETLEM	ERASE	+1	# B(2)PRM
TETOTHER	=	TETLEM	
DELTALEM	ERASE	+5	# B(6)PRM
NUVLEM	ERASE	+5	# B(6)PRM
RCVLEM	ERASE	+5	# B(6)PRM
VCVLEM	ERASE	+5	# B(6)PRM
TCLEM	ERASE	+1	# B(2)PRM
XKEPLEM	ERASE	+1	# B(2)PRM

X789	ERASE	+5	
TEPHEM	ERASE	+2	
AZO	ERASE	+1	
UNITW	ERASE	+5	
-AYO	EQUALS	UNITW	# (2)
AXO	EQUALS	UNITW +2	# (2)

STATE VECTORS FOR DOWNLINK (12D)

R-OTHER	ERASE	+5	# B(6)PRM POS VECT (OTHER VECH) FOR DNLINK
V-OTHER	ERASE	+5	# B(6)PRM VEL VECT (OTHER VECH) FOR DNLINK
T-OTHER	=	TETLEM	# TIME (OTHER VECH) FOR DNLINK

REFSMMAT. (18D)

REFSMMAT	ERASE	+17D	# I(18D)PRM
----------	-------	------	-------------

Page 87

AVERAGEG INTEGRATOR STORAGE. (8D)

UNITR	ERASE	+5
RMAG	ERASE	+1

P40 PAD LOADS (6D)

EK1VAL	ERASE	+1	# I(2)PL 1-SEC SPS IMPULSE NEWTSEC/100/B23
EK2VAL	ERASE	+1	# I(2)PL B+23 NEWTON-SEC/E+2
EK3VAL	ERASE		# I(1)PL B+09 NEWTONS/E+4

July 12, 2016

471

FANG ERASE # I(1)PL SPS THRUST USED BY IMPULSIVE BURN

*****LUNAR MODULE CHANGE *****

E3J22R2M EQUALS FANG +2

E32C31RM EQUALS E3J22R2M +1

**** CONICSEX (PLANETARY INERT. ORIEN.) ****

TIMSUBO EQUALS TEPHEM # CSEC B-14 (TRIPLE PREC)

END-E3 EQUALS E32C31RM # NEXT UNUSED E3 ADDRESS

Page 88

EBANK-4 ASSIGNMENTS

SETLOC 2000

P20 STORAGE. -- PAD LOADED -- (4D)

WRENDPOS	ERASE	# B(1)PL	M B-14
WRENDVEL	ERASE	# B(1)PL	M/CSECBO
RMAX	ERASE	# B(1)PL	METERS*2(-19)
VMAX	ERASE	# B(1)PL	M/CSEC*2(-7)

P22 STORAGE. -- PAD LOADED -- (5D)

WORBPOS	ERASE	# B(1)PL	M B-14
WORBVEL	ERASE	# B(1)PL	M/CSECBO
S22WSUBL	ERASE	# B(1)PL	M B-14
RPVAR	ERASE +1	# B(2)PL	

CONISEX STORAGE. -- PAD LOADED -- (6D)

504LM ERASE +5 # I(6) MOON LIBRATION VECTOR

ENTRY STORAGE -- PAD LOADED -- (2D)

EMSALT ERASE +1 # I(2)PL

P35 CONSTANTS. -- PAD LOADED -- (4D)

ATIGINC ERASE +1 # B(2)PL

PTIGINC ERASE +1 # B(2)PL

LUNAR LANDING SIGHT DATA. -- PAD LOADED -- (6D)

(USED BY INTEGRATION INITIALIZATION, LAT-LONG SUBROUTINES, P30'S)

RLS ERASE +5 # I(6) LANDING SIGHT VECTOR

CONISEX (LUNAR AND SOLAR EPHEM) STORAGE. -- PAD LOADED -- (77D)

TIMEMO ERASE +76D

```

VECOEM          EQUALS  TIMEMO  +3
RESO            EQUALS  VECOEM  +60D
# Page 89
VESO            EQUALS  RESO    +6
OMEGAES         EQUALS  VESO    +6

```

```
# INTEGRATION STORAGE. (95D)
```

```

PBODY          ERASE          # I(1)
ALPHAV         EQUALS  PBODY  +1  # I(6)TMP
BETAV          EQUALS  ALPHAV +6  # I(6)TMP
PHIV           EQUALS  BETAV +6  # I(6)TMP
PSIV           EQUALS  PHIV  +6  # I(6)TMP
FV             EQUALS  PSIV  +6  # I(6)TMP
BETAM          EQUALS  FV    +6  # I(6)TMP
H              EQUALS  BETAM +2  # I(2)TMP
GMODE          EQUALS  H      +2  # I(1)TMP
IRETURN        EQUALS  GMODE +1  # I(1)TMP
NORMGAM        EQUALS  IRETURN +1 # I(1)TMP
VECTAB         EQUALS  NORMGAM +1 # I(36)TMP
RPQV           EQUALS  VECTAB +36D # (6)TMP VECTOR PRIMARY TO SECONDARY BODY
ORIGEX         EQUALS  RPQV  +6  # B(1)TMP QSAVE FOR COORD. SWITCH ROUTINE
KEPRTN         EQUALS  ORIGEX    # QSAVE FOR KEPLER
RQVV           EQUALS  ORIGEX +1  # (6) SEC. BODY TO VEH.VETOR (USED P23)
RPSV           EQUALS  RQVV  +6  # (6)TMP SUN TO PRIMARY BODY VECTOR
XKEPNEW        EQUALS  RPSV  +6  # (2)TMP ROOT OF KEPLER'S EQU FOR T

```

```
# THESE PROBABLY CAN SHARE INTEGRATION VARIABLES (9D)
```

```

VACX           EQUALS  VECTAB +6  # I(2)TMP
VACY           EQUALS  VACX  +2  # I(2)TMP
VACZ           EQUALS  VACY  +2  # I(2)TMP

ERADM          EQUALS  VECTAB +18D # I(2)TMP
INCORPEX       EQUALS  ERADM  +2  # I(1)TMP

```

```
# R31 (V83) STORAGE. -- SHARES WITH INTEGRATION STORAGE -- (24D)
```

```

BASEOTP        EQUALS  VECTAB +6  # I(6) BASE POS VECTOR OTHER VEH
BASEOTV        EQUALS  VECTAB +18D # I(6) BASE VEL VECTOR OTHER VEH
BASETHP        EQUALS  VECTAB +30D # I(6) BASE POS VECTOR THIS VEH
BASETHV        EQUALS  RPQV      # I(6) BASE VEL VECTOR THIS VEH

```

```
# CONIC INTEGRATION STORAGE. -- MAY NOT SHARE WITH SERVICER -- (6D)
```

```

ALPHAM         EQUALS  XKEPNEW +2  # I(2)TMP
TAU.           EQUALS  ALPHAM  +2  # I(2)TMP

```


473

```
URONE          EQUALS  V82FLAGS          # I(6) SAVE ACTUAL FOR CALCULATIONS
```

```

# V82 DISPLAY                                (4D)
HAPOX                EQUALS  THETA(1) +2      # I(2)DSP NOUN 44
HPERX                EQUALS  HAPOX  +2        # I(2)DSP NOUN 44

# P22 DISPLAY REGISTERS                      (06D)
AOPTIME              EQUALS  HPERX  +2        # I(2)TMP FOR SR52.1.ADVTRACK
LANDLONG             EQUALS  AOPTIME +2       # I(2)DSP NOUN 89 FOR P22
LANDALT              EQUALS  LANDLONG +2     # I(2)DSP NOUN 89 FOR P22

# S34/35.5,P34-P35 STORAGE.                 (6D)
KT                   EQUALS  LANDALT +2       # B(2)
VERBNOUN             EQUALS  KT      +2       # B(1)TMP
QSAVED               EQUALS  VERBNOUN +1     # B(1)TMP HOLDS RETURN
RTRN                 EQUALS  QSAVED  +1      # B(1) RETURN
SUBEXIT              EQUALS  RTRN   +1       # B(1)TMP
RGEXIT               EQUALS  SUBEXIT          # RGEXIT CAN'T SHARE WITH HPER,HAPO
                                           # I(1)TMP Q SAVE MODE 1 AND 2 TO RTRN MAIN

# P30 DISPLAY                                (4D)
HAPO                 EQUALS  KT              # I(2)DSP NOUN 42, FOR P30.
HPER                 EQUALS  HAPO   +2       # I(2)DSP NOUN 42, FOR P30.

# SOME P34 STORAGE.      (OVERLAYS P35.1 STORAGE)      (2D)
NOMTPI               EQUALS  KT              # I(2)TMP NOMINAL TPI TIME FOR RECYCLE.

# THE FOLLOWING ARE ERASABLES USED BY THE SYSTEM TESTS.  205 USES TRANSM1.  G'S ARE 1
# WHILE 504 USES TRANSM1 AND ALFDK.
# Page 92
# RSB 2009.  The definition of TRANSM1 was previously just "TRANSM1 EQUALS 2000",
# this messes up the label typing system in yaYUL.
                SETLOC  2000
TRANSM1              EQUALS                      # (18) INITIALIZATION FOR IMU TESTS
ALFDK                =      TRANSM1 +18D      # (144) ERASABLE LOAD IN 504

# END OF PERF. TEST ERASABLE IN BANK 4

# *--* V82 *--*                                (6D)
VONE'                EQUALS  RGEXIT  +1      # I(6)TMP NORMAL VELOCITY VONE/ SQ RT MU

# PAD LOAD INTEGRATION ERROR INCLUDED IN VARIANCE BY P20      (1D)
INTVAR               EQUALS  VONE'   +6      # I(1)PL          SQUARE OF EXPECTED INTEGRATION
                                           #              POSITION EXTRAPOLATION ERROR
                                           #              SCALED METERS(2) 2(15)
END-E4               EQUALS  INTVAR          # LAST USED ERASABLE IN E4.

```

Page 93
EBANK-5 ASSIGNMENTS

SETLOC 2400

*- - *- - *- OVERLAY 1 IN EBANK 5 *- - *- - *

W-MATRIX STORAGE. (162D)

RSB 2009. The following 3 lines have been replaced to be consistent with yaYUL's
label-typing system. They *were* "W EQUALS 2400", "9X9LOC1 EQUALS 2444",
"9X9LOC2 EQUALS 2532".

W EQUALS # B(162)

9X9LOC1 EQUALS W +44

9X9LOC2 EQUALS 9X9LOC1 +66

EMATRIX = W +120D # B(42) USED TO CONVERT W TO 6X6

END-W EQUALS W +162D # **NEXT AVAILABLE LOC AFTER W MATRIX**

AUTO-OPTICS STORAGE -R52-

DO NOT MOVE FROM E5,1554. A DELICATE BALANCE EXISTS BETWEEN THIS AND P03

XNB1 EQUALS W +108D # B(6D)TMP

YNB1 EQUALS XNB1 +6 # B(6)TMP

ZNB1 EQUALS YNB1 +6 # B(6)TMP

SAVQR52 EQUALS ZNB1 +6 # I(2)TMP

PLANVEC EQUALS SAVQR52 +2 # B(6) S-S SIGHTING VECTOR IN REF. COOR.

TSIGHT EQUALS PLANVEC +6 # B(2) S-S TIME OF SIGHTING

RENDEZVOUS -P34-35 (26D)

DVLOS EQUALS TSIGHT +2 # I(6) S-S DELTA VELOCITY, LOS COORD-DISPLAY

DELTAR EQUALS DVLOS # I(2)

TINTSOI EQUALS DELTAR # I(2) INTERCEPT TIME FOR SOI MANEUVER

DELTTIME EQUALS DVLOS +2 # I(2)

TARGTIME EQUALS DVLOS +4 # I(2)

UNRM EQUALS DVLOS +6 # I(6) S-S

ULOS EQUALS UNRM +6 # I(6) S-S UNIT LINE OF SIGHT VECTOR

ACTCENT EQUALS ULOS +6 # I(2) S-S CENTRAL ANGLE BETWEEN ACTIVE

VEH AT TPI IGNITION TIME AND

TARGET VECTOR.

DELVTPI EQUALS ACTCENT +2 # I(2) NOUN 58 FOR P34

DELVTPF EQUALS DELVTPI +2 # I(2) NOUN 58,59 FOR P34,35

POSTTPI EQUALS DELVTPF +2 # I(2) NOUN 58 FOR P34.

TDEC2 EQUALS DELVTPI # (2)

```

# ALIGNMENT                                     (12D)
# Page 94
STARSAV1      EQUALS  DVLOS                      # I(6)TMP RESTART STAR SAVE.
STARSAV2      EQUALS  STARSAV1 +6                # I(6)TMP RESTART STAR SAVE.
US            =       STARSAV2                    # (CISLUNAR TAG FOR STARSAV2).

# TPI SEARCH                                     (26D)
IT            EQUALS  DVLOS                      # (6)
THETZERO      EQUALS  IT      +6                 # (2)
TFI           EQUALS  THETZERO +2               # (2)
DELVEE        EQUALS  TFI      +2               # (2)
HP            EQUALS  DELVEE  +2               # (2)
TFO           EQUALS  HP      +2               # (2)
HPO           EQUALS  TFO     +2               # (2)
DELVEO        EQUALS  HPO     +2               # (2)
MAGVTPI       EQUALS  DELVEO  +2               # I(2)TMP MAG OF DELTAVTPI OR VMID
RELDELV       EQUALS  MAGVTPI +2               # I(2)TMP MAG OF DELTAVTPF
T3TOT4        EQUALS  RELDELV +2               # I(2)DSP NOUN 39 FOR P34,35. TPI TO TINT
                                           # (CANNOT SHARE WITH RETURN TO EARTH)

# Page 95
# ALIGNMENT/SYSTEST/CALCSMSC/CRS61.1 COMMON STORAGE      (36D)
# (CALCSMSC IS A SUBSET OF S41.1 AT LEAST)
# (CRS61.1 IS A SUBSET OF P20)

XSM           EQUALS  END-W   +23D             # B(6)
YSM           EQUALS  XSM     +6               # B(6)TMP
ZSM           EQUALS  YSM     +6               # B(6)TMP

XDC           EQUALS  ZSM     +6               # B(6)TMP
YDC           EQUALS  XDC     +6               # B(6)TMP
ZDC           EQUALS  YDC     +6               # B(6)TMP

XNB           =       XDC
YNB           =       YDC
ZNB           =       ZDC

# OVERLAYS WITHIN ALIGNMENT/SYSTEST/CALCSMSC COMMON STORAGE

-COSB         EQUALS  XSM     +2               # (2)TMP
SINB          EQUALS  -COSB   +2               # (2)TMP

# ALIGNMENT/SYSTEST COMMON STORAGE      (18D)

STARAD        EQUALS  ZDC     +6               # I(18D)TMP

# ALIGNMENT/SYSTEST/AUTO OPTICS COMMON STORAGE. (17D)

```

OGC	EQUALS	STARAD	+18D	# I(2)TMP
IGC	EQUALS	OGC	+2	# I(2)TMP
MGC	EQUALS	IGC	+2	# I(2)TMP
STAR	EQUALS	MGC	+2	# I(6)TMP
SAC	EQUALS	STAR	+6	# I(2)TMP
PAC	EQUALS	SAC	+2	# I(2)TMP
QMIN	EQUALS	PAC	+2	# B(1)TMP

**** COLP50'S **** (1D)
 CULTRIX EQUALS VEARTH # VEARTH, VSUN, VMOON

OVERLAYS WITHIN ALIGNMENT/SYSTEST COMMON STORAGE (24D)

VEARTH	EQUALS	STARAD		# (6)TMP
VSUN	EQUALS	VEARTH	+6	# (6)TMP
VMOON	EQUALS	VSUN	+6	# (6)TMP
SAX	EQUALS	VMOON	+6	# (6)TMP

Page 96

*-***- OVERLAY NUMBER 2 IN EBANK 5 -***-*

CONICS ROUTINE STORAGE. (87D)

DELX	EQUALS	END-W		# I(2)TMP
DELT	EQUALS	DELX	+2	# I(2)TMP
URRECT	EQUALS	DELT	+2	# I(6)TMP
RCNORM	EQUALS	URRECT	+6	# I(2)TMP
XPREV	EQUALS	XKEP		# I(2)TMP
R1VEC	EQUALS	RCNORM	+2	# I(6)TMP
R2VEC	EQUALS	R1VEC	+6	# I(6)TMP
TDESIRE	EQUALS	R2VEC	+6	# I(2)TMP
GEOMSGN	EQUALS	TDESIRE	+2	# I(1)TMP
UN	EQUALS	GEOMSGN	+1	# I(6)TMP
VTARGET	EQUALS	UN	+6	# I(1)TMP
VTARGET	EQUALS	VTARGET	+1	# I(6)TMP
RTNLAMB	EQUALS	VTARGET	+6	# I(1)TMP
U2	EQUALS	RTNLAMB	+1	# I(6)TMP
MAGVEC2	EQUALS	U2	+6	# I(2)TMP
UR1	EQUALS	MAGVEC2	+2	# I(6)TMP
SNTH	EQUALS	UR1	+6	# I(2)TMP
CSTH	EQUALS	SNTH	+2	# I(2)TMP
1-CSTH	EQUALS	CSTH	+2	# I(2)TMP
CSTH-RHO	EQUALS	1-CSTH	+2	# I(2)TMP
P	EQUALS	CSTH-RHO	+2	# I(2)TMP
R1A	EQUALS	P	+2	# I(2)TMP

RVEC	EQUALS	R1VEC		# I(6)TMP
VVEC	EQUALS	R1A	+2	# I(6)TMP
RTNTT	EQUALS	RTNLAMB		# I(1)TMP
ECC	EQUALS	VVEC	+6	# I(2)TMP
RTNTR	EQUALS	RTNLAMB		# I(1)TMP
RTNAPSE	EQUALS	RTNLAMB		# I(1)TMP
R2	EQUALS	MAGVEC2		# I(2)TMP
RTNPRM	EQUALS	ECC	+2	# I(1)TMP
SGNRDOT	EQUALS	RTNPRM	+1	# I(1)TMP
RDESIRED	EQUALS	SGNRDOT	+1	# I(2)TMP
DELDEP	EQUALS	RDESIRED	+2	# I(2)TMP
DEPREV	EQUALS	DELDEP	+2	# I(2)TMP
TERRLAMB	EQUALS	DELDEP		# I(2)TMP
TPREV	EQUALS	DEPREV		# I(2)TMP

Page 97

*-***- OVERLAY NUMBER 3 IN EBANK 5 -***-*

MEASUREMENT INCORPORATION STORAGE. (66D)

(CALLED BY P20, P22, P23)

OMEGAM1	EQUALS	END-W		# I(6)TMP
OMEGAM2	EQUALS	OMEGAM1	+6	# I(6)TMP
OMEGAM3	EQUALS	OMEGAM2	+6	# I(6)TMP
HOLDW	EQUALS	OMEGAM3	+6	# I(18)TMP
TDPOS	EQUALS	HOLDW	+18D	# I(6)TMP
TDVEL	EQUALS	TDPOS	+6	# I(6)TMP

ZI	EQUALS	TDVEL	+6	# I(18)
----	--------	-------	----	---------

P22-P23 STORAGE. (8D)

22SUBSCL	EQUALS	ZI	+18D	# DE OF ABCDE LANDMARK ID NO.
CXOFF	EQUALS	22SUBSCL	+1	# B OF ABCDE OFFSET INDICATOR
8KK	EQUALS	CXOFF	+1	# B(1)TMP INDEX OF PRESENT MARK
8NN	EQUALS	8KK	+1	# B(1)TMP
S22LOC	EQUALS	8NN	+1	# I(1)TMP MARK DATA LOC
LANDMARK	EQUALS	S22LOC	+1	# B(1)DSP NOUN 70 FOR P22,51, R52,53
HORIZON	EQUALS	LANDMARK	+1	# B(1)DSP NOUN 70 FOR P22,51, R52,53
IDOFLMK	EQUALS	HORIZON	+1	# B(1)

*****P23*** (1D)

TRUNION	EQUALS	IDOFLMK	+1	# B(1)
---------	--------	---------	----	--------

Page 98

*- - *- - *- OVERLAY NUMBER 0 IN EBANK 5 *- - *- - *

SYSTEM TEST STORAGE. (174)

AZIMUTH	ERASE	+1
LATITUDE	ERASE	+1
TRUNA	EQUALS	DESOPTT
SHAFTA	EQUALS	DESOPTS
ERVECTOR	ERASE	+5
LENGTHOT	ERASE	
LOSVEC	ERASE	+5
SXTOPTN	=	LOSVEC
NDXCTR	ERASE	
PIPINDEX	ERASE	
POSITON	ERASE	
QPLAC	ERASE	
QPLACE	ERASE	
QPLACES	ERASE	
RUN	ERASE	
STOREPL	ERASE	
SOUTHDR	ERASE	
TARG1/2	=	SOUTHDR
TAZEL1	ERASE	+5
TEMPTIME	ERASE	+1
TMARK	ERASE	+1
GENPL	ERASE	+134D
CDUTIMEI	=	GENPL
CDUTIMEF	=	GENPL +2
IMU/OPT	=	GENPL +4
CDUREADF	=	GENPL +5
CDUREADI	=	GENPL +6
CDULIMIT	=	GENPL +7
TEMPADD	=	GENPL +4
TEMP	=	GENPL +5
NOBITS	=	GENPL +6
CHAN	=	GENPL +7
LOS1	=	GENPL +8D
LOS2	=	GENPL +14D
CALCDIR	EQUALS	GENPL +20D
CDFLAG	EQUALS	GENPL +21D

GYTOBETQ	EQUALS	GENPL	+22D	
OPTNREG	EQUALS	GENPL	+23D	
SAVE	EQUALS	GENPL	+24D	# THREE CONSEC LOC
SFCONST1	EQUALS	GENPL	+27D	
# Page 99				
TIMER	EQUALS	GENPL	+28D	
DATAPL	EQUALS	GENPL	+30D	
RDSP	EQUALS	GENPL		# FIX LATER POSSIBLY KEEP1
MASKREG	EQUALS	GENPL	+64D	
CDUNDX	EQUALS	GENPL	+66D	
RESULTCT	EQUALS	GENPL	+67D	
COUNTPL	EQUALS	GENPL	+70D	
CDUANG	EQUALS	GENPL	+71D	
AINLA	=	GENPL		# OPTIMUM CALIB. AND ALIGNMENT
WANGO	EQUALS	AINLA		
WANGI	EQUALS	AINLA	+2D	
WANGT	EQUALS	AINLA	+4D	
TORQNDX	=	WANGT		
DRIFTT	EQUALS	AINLA	+6D	
ALX1S	EQUALS	AINLA	+8D	
CMPX1	EQUALS	AINLA	+9D	
ALK	EQUALS	AINLA	+10D	
VLAUNS	EQUALS	AINLA	+22D	
THETAX	=	ALK	+2	
WPLATO	EQUALS	AINLA	+24D	
INTY	EQUALS	AINLA	+28D	
THETAN	=	THETAX	+6	
ANGZ	EQUALS	AINLA	+30D	
INTZ	EQUALS	AINLA	+32D	
ANGY	EQUALS	AINLA	+34D	
ANGX	EQUALS	AINLA	+36D	
DRIFTO	EQUALS	AINLA	+38D	
DRIFTI	EQUALS	AINLA	+40D	
VLAUN	EQUALS	AINLA	+44D	
FILDELV	=	THETAN	+6	
ACCWD	EQUALS	AINLA	+46D	
INTVEC	=	FILDELV	+2	
POSNV	EQUALS	AINLA	+52D	
DPIPAY	EQUALS	AINLA	+54D	
DPIPAZ	EQUALS	AINLA	+58D	
ALTIM	EQUALS	AINLA	+60D	
ALTIMS	EQUALS	AINLA	+61D	
ALDK	EQUALS	AINLA	+62D	
DELM	EQUALS	AINLA	+76D	


```

WPLATI      EQUALS  AINLA  +84D
RESTARPT    =      AINLA  +91D
GEOSAVED    =      AINLA  +117D
PREMTRXC    =      AINLA  +118D
LAUNCHAZ    =      AINLA  +119D
NEWAZMTH    =      AINLA  +121D
OLDAZMTH    =      AINLA  +123D
# Page 100
TOLDAZMT    =      AINLA  +125D
GEOCOMPS    =      AINLA  +127D
1SECXT      =      AINLA  +128D
GTSXTLST    =      AINLA  +129D
ERECTIME    =      AINLA  +130D
ERCOMP      =      AINLA  +131D
ZERONDX     =      AINLA  +137D
GTSOPNDZ    =      ZERONDX

```

THE FOLLOWING TAGS ARE USED BY THE 504 IMU CALIBRATION AND ALIGNMENT PROGRAM ONLY.

```

THETAX1     EQUALS  ALK      +2
THETAN1     EQUALS  THETAX1 +6
FILDELV1    EQUALS  THETAN1 +6
INTVEC1     EQUALS  FILDELV1 +2
GEOSAVE1    EQUALS  AINLA   +117D
PREMTRX1    EQUALS  AINLA   +118D
LUNHAZ1     EQUALS  AINLA   +119D
NEWAZ1      EQUALS  LUNHAZ1 +2
OLDAZ1      EQUALS  LUNHAZ1 +4
TOLDAZ1     EQUALS  LUNHAZ1 +6
GEOCOMP1    EQUALS  AINLA   +127D
1SECXT1     EQUALS  AINLA   +128D
GTSWTLT1    EQUALS  AINLA   +129D
ERECTIM1    EQUALS  AINLA   +130D
ERCOMP1     EQUALS  AINLA   +131D  # I(6)
ZERONDX1    EQUALS  AINLA   +137D
PERFDLAY    EQUALS  AINLA   +138D  # B(2).....

```

END OF 504 + ALIGN ERASE.

Page 101

```

# *-*-*-* OVERLAY 4 IN EBANK 5 *-*-*-*
#
# P32 --- P33 (26D)
UP1          EQUALS  DVLOS      # I(6)
VPASS2       EQUALS  UP1      +6  # I(6)

```

RPASS2	EQUALS	VPASS2	+6	# I(6)
DIFFALT	EQUALS	RPASS2	+6	# I(2)
TCDH	EQUALS	DIFFALT	+2	# I(2)
TCSI	EQUALS	TCDH	+2	# I(2)
TTPIO	EQUALS	TCSI	+2	# I(2)

```
# P32,P33 STORAGE OVERLAYING 9X9 W-MATRIX LOCATIONS      (26D)
DELVEET1      EQUALS  9X9LOC1      # I(6) DELV FOR CSI
RACT2         EQUALS  DELVEET1 +6   # I(6) POS. ACTIVE VEH. AT CDH TIME
VACT2         EQUALS  9X9LOC2      # I(6) VEL. ACTIVE VEH. AT CDH TIME
RACT1         EQUALS  VACT2  +6     # I(6) POS. ACTIVE VEH. AT CSI TIME
T1TOT2        EQUALS  RACT1  +6     # I(2) TCDH - TCSI
END-E5        EQUALS  QMIN          # LAST USED E5 ADDRESS
```

Page 102

EBANK-6 ASSIGNMENTS.

```
                SETLOC  3000
# P23  PAD LOADS ***                (2D).
WMIDPOS        ERASE                # I(1)PL INITIAL VALUES FOR W-MATRIX IN
WMIDVEL        ERASE                # I(1)PL Cislunar (P23) NAVIGATION
```

```
# R22  PAD LOADS                (5D).
RVAR          ERASE  +1          # I(2)PL VHF RADAR
RVARMIN       ERASE  +2          # I(3)PL VHF RADAR
```

```
# ***** PAD LOADED ENTRY DAP STEERING VARIABLES ***** (3D)
LADPAD        ERASE                # I(1)PL FOR ENTRY. HOLDS CM NOMINAL L/D
LODPAD        ERASE                # I(1)PL FOR ENTRY. HOLDS CM NOMINAL LOD
ALFAPAD       ERASE                # B(1)PL ALFA TRIM / 180
```

```
# ***** PAD LOADED TVC DAP VARIABLES ***** (26D)
ETDECAY       ERASE                # I(1)PL
ESTROKER      ERASE                # B(1)PL
EKPRIME       ERASE  +1            # B(2)PL
EKTLX/I       ERASE  +2            # B(3)PL
EREPRAC       ERASE  +1            # B(2)PL
PACTOFF       ERASE                # B(1)PL, DSP N48 R01 = PTRIM, R02 = YTRIM
YACTOFF       ERASE                # B(1)PL, CONSECUTIVE WITH PACTOFF
HBN10         ERASE                # B(1)
HBN11/2       ERASE                # B(1)
HBN12         ERASE                # B(1)
HBD11/2       ERASE                # B(1)
HBD12         ERASE                # B(1)
HBN20         ERASE                # B(1)
```

HBN21/2	ERASE	# B(1)
HBN22	ERASE	# B(1)
HBD21/2	ERASE	# B(1)
HBD22	ERASE	# B(1)
HBN30	ERASE	# B(1)
HBN31/2	ERASE	# B(1)
HBN32	ERASE	# B(1)
#Page 103		
HBD31/2	ERASE	# B(1)
HBD32	ERASE	# B(1)

**** EXCLUSIVE TVC DAP VARIABLES. ***** (5D)

V97VCNTR	ERASE	# B(1)
TEMPDAP	ERASE +1	# B(2)
MRKRTMP	= TEMPDAP	# ((B(1)))
CNTR	ERASE	# B(1)
OGAD	ERASE	# B(1)

**** EXCLUSIVE RCS DAP VARIABLES ***** (13D)

RWORD1	ERASE +12D	# B(1)
RWORD2	EQUALS RWORD1 +1	# B(1)
PWORD1	EQUALS RWORD2 +1	# B(1)
PWORD2	EQUALS PWORD1 +1	# B(1)
YWORD1	EQUALS PWORD2 +1	# B(1)
YWORD2	EQUALS YWORD1 +1	# B(1)
BLAST	EQUALS YWORD2 +1	# B(2)
BLAST1	EQUALS BLAST +2	# B(2)
BLAST2	EQUALS BLAST1 +2	# B(2)
T5PHASE	EQUALS BLAST2 +2	# B(1)

**** RCS/TVC DAP COMMON STORAGE. ***** (16D)

DAPDATR1	ERASE	# B(1)DSP NOUN 46(R1)
DAPDATR2	ERASE	# B(1)DSP NOUN 46(R2)

IXX	ERASE	# B(1) CONSECUTIVE WITH IAVG, IAVG/TLX FOR
IAVG	ERASE	# B(1) MASSPROP
IAVG/TLX	ERASE	# B(1)

LEMMASS	ERASE	# B(1)DSP NOUN 47 (R2)
CSMMASS	ERASE	# B(1)DSP NOUN 47 (R1)
WEIGHT/G	ERASE	# B(1)
MASS	= WEIGHT/G	

AK	ERASE
AK1	ERASE
AK2	ERASE

```

RCSFLAGS      ERASE                # B(1) CONSECUTIVE WITH AK2 DOWNLINK
T5TEMP        ERASE                # B(1)
EDRIVEX       ERASE
EDRIVEY       ERASE
# Page 104
EDRIVEZ       ERASE

```

```

# INTMP THRU INTMP+14D ARE RESERVED FOR OVERLAYED TVC/RCS INTERUP TRUE TEMPORARIES
INTTEMP      ERASE    +14D        # (15)

```

```

# TVC/RCS THRU TVCRCS +11D RESERVED FOR DOWNLINKED VARIABLES
TVCRCS      ERASE    +11D        # (12)
                                     # RCS (WBODYS,ADOTS)
                                     # TVC(OMEGACS,OMEGABS)

```

```

# TVC DAP TEMPORARY VARIABLES*****

```

```

# TVC DAP INTERRUPT TRUE TEMPORARIES*****

```

```

PHI333      EQUALS  INTTEMP      # B(1) TEMPORARY REGISTER
PSI333      EQUALS  PHI333  +1    # B(1) COUNTING REGISTER
TEMP333     EQUALS  PSI333  +1    # B(1) COUNTING REGISTER
VARST0      EQUALS  TEMP333  +1    # B(8) BREAKPOINTS AND SLOPES
VARST5      =      VARST0  +5
LASTMASP    EQUALS  VARST0  +9D    # LAST VARSTO WORD
TVCTMP1     EQUALS  LASTMASP +1    # B(1)

```

```

# *****REGULAR TVC TEMPORARIES*****

```

```

# TVC ZEROING STARTS HERE

```

```

OMEGAC      EQUALS  TVCRCS      # I(6)
OMEGAXC     =      OMEGAC
OMEGAYC     =      OMEGAC  +2
OMEGAZC     =      OMEGAC  +4

```

```

OMEGAB      EQUALS  TVCRCS  +6    # B(6)
OMEGAXB     =      OMEGAB
OMEGAYB     =      OMEGAB  +2
OMEGAZB     =      OMEGAB  +4

```

```

PTMP1      EQUALS  OMEGAC  +12D   # B(2)
PTMP2      EQUALS  PTMP1   +2     # B(2)
PTMP3      EQUALS  PTMP2   +2     # B(2)
PTMP4      EQUALS  PTMP3   +2     # B(2)

```

PTMP5	EQUALS	PTMP4	+2	# B(2)
# Page 105				
PTMP6	EQUALS	PTMP5	+2	# B(2)
YTMP1	EQUALS	PTMP6	+2	# B(2)
YTMP2	EQUALS	YTMP1	+2	# B(2)
YTMP3	EQUALS	YTMP2	+2	# B(2)
YTMP4	EQUALS	YTMP3	+2	# B(2)
YTMP5	EQUALS	YTMP4	+2	# B(2)
YTMP6	EQUALS	YTMP5	+2	# B(2)
ROLLFIRE	EQUALS	YTMP6	+2	# B(1)
ROLLWORD	EQUALS	ROLLFIRE	+1	# B(1)
TEMREG	EQUALS	ROLLWORD	+1	# B(1)
STROKER	EQUALS	TEMREG	+1	# B(1)
PERRB	EQUALS	STROKER	+1	# B(2)
YERRB	EQUALS	PERRB	+2	# B(2)
DELPBAR	EQUALS	YERRB	+2	# B(2)
DELYBAR	EQUALS	DELPBAR	+2	# B(2)
PDELOFF	EQUALS	DELYBAR	+2	# B(2)
YDELOFF	EQUALS	PDELOFF	+2	# B(2)
# TVC ZEROING LOOP ENDS HERE				
TTMP1	EQUALS	YDELOFF	+2	# B(2)
TTMP2	EQUALS	TTMP1	+2	# B(2)
DAP1	EQUALS	TTMP2	+2	# B(2)
DAP2	EQUALS	DAP1	+2	# B(2)
DAP3	EQUALS	DAP2	+2	# B(2)
PCMD	EQUALS	DAP3	+2	# B(1)
YCMD	EQUALS	PCMD	+1	# B(1), CONSECUTIVE WITH PCMD
T5TVCDT	EQUALS	YCMD	+1	# B(1)
MDT	EQUALS	T5TVCDT	+1	# I(6)
KPRIMEDT	EQUALS	MDT	+6	# I(2)
KTLX/I	EQUALS	KPRIMEDT	+2	# B(1)
TENMDOT	EQUALS	KTLX/I	+1	# B(1)
1/CONACC	EQUALS	TENMDOT	+1	# B(1)
VARK	EQUALS	1/CONACC	+1	# B(1)
REPFRAC	EQUALS	VARK	+1	# B(1)
VCNTR	EQUALS	REPFRAC	+1	# B(1)
TVCPHASE	EQUALS	VCNTR	+1	# B(1)
PCDUYPST	EQUALS	TVCPHASE	+1	# B(1)

PCDUZPST	EQUALS	PCDUYPST +1	# B(1)
MCDUYDOT	EQUALS	PCDUZPST +1	# B(1)
MCDUZDOT	EQUALS	MCDUYDOT +1	# B(1)

Page 106

TVCEXPHS	EQUALS	MCDUZDOT +1	# B(1)	
MASSTMP	EQUALS	TVCEXPHS +1	# B(1)	PROTECT
VCNTRTMP	EQUALS	MASSTMP +1	# B(1)	*PROTECT***

STROKE TEST VARIABLES

STRKTIME	EQUALS	VCNTRTMP +1	# B(1)
CADDY	EQUALS	STRKTIME +1	# B(1)
N	EQUALS	CADDY +1	# B(1)
BUNKER	EQUALS	N +1	# B(1)
REVS	EQUALS	BUNKER +1	# B(1)
CARD	EQUALS	REVS +1	# B(1)

TVC ROLL DAP VARIABLES

OGANOW	EQUALS	CARD +1	# B(1)
OGAPAST	EQUALS	OGANOW +1	# B(1)
OGA	EQUALS	OGAPAST +1	# B(1)TMP
OGAERR	=	OGA	# (ROLL DAP USES OGA, MEANS OGAERROR)
DELOGART	EQUALS	OGA +1	# B(1)TMP
SGNRT	EQUALS	DELOGART +1	# SIGN OF CGA RATE
DELOGA	EQUALS	SGNRT +1	# USED IN ROLL LOGIC
I	EQUALS	DELOGA +1	# USED IN ROLL LOGIC
IOGARATE	EQUALS	I +1	# USED IN ROLL LOGIC

TVC DAP RESTART TEMPORARIES.

PACTTMP	EQUALS	IOGARATE +1	# B(2)
YACTTMP	EQUALS	PACTTMP +2	# B(2)
CNTRTMP	EQUALS	YACTTMP +2	# B(1)
STRKTTMP	EQUALS	CNTRTMP +1	# B(1)
DELBRTMP	EQUALS	STRKTTMP +1	# B(2)
ERRBTMP	EQUALS	DELBRTMP +2	# B(2)
CMDTMP	EQUALS	ERRBTMP +2	# B(2)

TMP1	EQUALS	CMDTMP +2	# B(2)
TMP2	EQUALS	TMP1 +2	# B(2)
TMP3	EQUALS	TMP2 +2	# B(2)
TMP4	EQUALS	TMP3 +2	# B(2)
TMP5	EQUALS	TMP4 +2	# B(2)
TMP6	EQUALS	TMP5 +2	# B(2)

TVC DAP FILTER COEFFICIENTS TEMPORARIES

COEFFADR	EQUALS	TMP6 +2	# B(1)
N10	EQUALS	COEFFADR +1	# I(15)

Page 107

OVERLAYS WITHIN TVC DAP

OGARATE	=	OMEGAB	# B(2)
PHASETMP	=	TTMP1	# B(1) RESTART FOR CSM/LM V46 SWITCH-OVER
RTRNLOC	=	TTMP2	# B(1) RESTART FOR CSM/LM V46 SWITCH-OVER
BZERO	=	ERRBTMP	
CZERO	=	ERRBTMP	
JZERO	=	CMDTMP	
YZERO	=	CMDTMP	

540.9 STORAGE

NBRCYCLS	EQUALS	N10	+15D	# B(1) COUNTER FOR P40,41 STEERING
NBRCYCLP	EQUALS	NBRCYCLS	+1	# B(1) MAINTAIN ORDER
DELVSUM	EQUALS	NBRCYCLP	+1	# I(6) P40,P41
DELVSUMP	EQUALS	DELVSUM	+6	# I(6) P40,P41

Page 108

**** RCS DAP TEMPORARY VARIABLES. ***** (95D)

** RCS INTERRUPT TRUE TEMPS ***** (15D)

SPNDX	EQUALS	INTTEMP	# B(1)
DPNDX	EQUALS	SPNDX	+1 # B(1)TMP
KMPAC	EQUALS	DPNDX	+1 # B(2)TMP
KMPTEMP	EQUALS	KMPAC	+2 # B(1)TMP
XNDX1	EQUALS	KMPTEMP	+1 # B(1)TMP XNDX1 THRU NYJETS ARE OVERLAYED
XNDX2	EQUALS	XNDX1	+1 # B(1)TMP BY OTHER DAP ERASABLES SO
YNDX	EQUALS	XNDX2	+1 # B(1)TMP SHOULD ALWAYS BE DEFINED IN
ZNDX	EQUALS	YNDX	+1 # B(1)TMP A BLOCK
RINDEX	EQUALS	ZNDX	+1 # B(1)TMP
PINDEX	EQUALS	RINDEX	+1 # B(1)TMP
YINDEX	EQUALS	PINDEX	+1 # B(1)TMP
NRJETS	EQUALS	YINDEX	+1 # B(1)TMP
NPJETS	EQUALS	NRJETS	+1 # B(1)TMP
NYJETS	EQUALS	NPJETS	+1 # B(1)TMP
WTEMP	EQUALS	XNDX1	# B(2)TMP WTEMP THRU DELTEMPZ OVERLAY
DELTEMPX	EQUALS	WTEMP	+2 # B(2)TMP XNDX1 THRU NRJETS AND EDOT THRU
DELTEMPY	EQUALS	DELTEMPX	+2 # B(2)TMP ADBVEL
DELTEMPZ	EQUALS	DELTEMPY	+2 # B(2)TMP
EDOT	EQUALS	YNDX	# B(2)TMP EDOT THRU ADBVEL OVERLAY

```

AERR          EQUALS  EDOT      +2      # B(1)TMP YNDX THRU NPJETS AND DELTEMPX
EDOTVEL       EQUALS  AERR      +1      # B(2)TMP THRU DELTEMPZ
AERRVEL       EQUALS  EDOTVEL   +2      # B(1)TMP
ADBVEL        EQUALS  AERRVEL   +1      # B(1)TMP

```

```

# *** REGULAR RCS TEMPS ***** ( ).

```

```

# *** RCS ZEROING LOOP STARTS HERE ***** (37)

```

```

WBODY          EQUALS  TVCRCS      # B(2)TMP
WBODY1         EQUALS  WBODY      +2  # B(2)TMP
WBODY2         EQUALS  WBODY      +4  # B(2)TMP
ADOT           EQUALS  WBODY2     +2  # B(2)TMP
ADOT1          EQUALS  ADOT       +2  # B(2)TMP
ADOT2          EQUALS  ADOT1      +2  # B(2)TMP

```

```

MERRORX        EQUALS  ADOT2      +2  # (2)
MERRORY        EQUALS  MERRORX    +2  # (2)
MERRORZ        EQUALS  MERRORY    +2  # (2)
DFT            EQUALS  MERRORZ    +2  # B(1)TMP
DFT1           EQUALS  DFT        +1  # B(1)TMP
DFT2           EQUALS  DFT1       +1  # B(1)TMP
DRHO           EQUALS  DFT2       +1  # B(2)TMP
DRHO1          EQUALS  DRHO       +2  # B(2)TMP

```

```

# Page 109

```

```

DRHO2          EQUALS  DRHO1      +2  # B(2)TMP
ATTSEC         EQUALS  DRHO2      +2  # B(1)TMP
TAU            EQUALS  ATTSEC     +1  # B(1)TMP
TAU1           EQUALS  TAU        +1  # B(1)TMP
TAU2           EQUALS  TAU1       +1  # B(1)TMP
BIAS           EQUALS  TAU2       +1  # B(1)TMP
BIAS1          EQUALS  BIAS       +1  # B(1)TMP
BIAS2          EQUALS  BIAS1      +1  # B(1)TMP
ERRORX         EQUALS  BIAS2      +1  # B(1)TMP
ERRORY         EQUALS  ERRORX     +1  # B(1)TMP
ERRORZ         EQUALS  ERRORY     +1  # B(1)TMP

```

```

# RCS ZERO LOOP ENDS HERE

```

```

# MORE RCS (69D)

```

```

THETADX        EQUALS  ERRORZ    +1  # B(1)TMP MUST BE CONSECUTIVE WITH ERRORZ
THETADY        EQUALS  THETADX   +1  # B(1)TMP
THETADZ        EQUALS  THETADY   +1  # B(1)TMP

```

```

DELCDUX        EQUALS  THETADZ   +1  # B(2)TMP
DELCDUY        EQUALS  DELCDUX   +2  # B(2)TMP
DELCDUZ        EQUALS  DELCDUY   +2  # B(2)TMP

```


DCDU	EQUALS	DELCDUZ	+2	# B(6)TMP USED DURING P20
DTHETASM	EQUALS	DCDU	+6	# B(6)TMP STEER LOW OUTPUT.
ATTKALMN	EQUALS	DTHETASM	+6	# B(1)TMP
KMJ	EQUALS	ATTKALMN	+1	# B(1)TMP
KMJ1	EQUALS	KMJ	+1	# B(1)TMP
KMJ2	EQUALS	KMJ1	+1	# B(1)TMP
J/M	EQUALS	KMJ2	+1	# B(1)TMP
J/M1	EQUALS	J/M	+1	# B(1)TMP
J/M2	EQUALS	J/M1	+1	# B(1)TMP
RACFAIL	EQUALS	J/M2	+1	# B(1)TMP
RBDFAIL	EQUALS	RACFAIL	+1	# B(1)TMP
ACORBD	EQUALS	RBDFAIL	+1	# B(1)TMP
XTRANS	EQUALS	ACORBD	+1	# B(1)TMP
CH31TEMP	EQUALS	XTRANS	+1	# B(1)TMP
CHANTEMP	EQUALS	CH31TEMP	+1	# B(1)TMP
T5TIME	EQUALS	CHANTEMP	+1	# B(1)TMP
RHO	EQUALS	T5TIME	+1	# B(1)TMP
RHO1	EQUALS	RHO	+1	# B(1)TMP
RHO2	EQUALS	RHO1	+1	# B(1)TMP
AMGB1	EQUALS	RHO2	+1	# B(1)TMP
AMGB4	EQUALS	AMGB1	+1	# B(1)TMP
# Page 110				
AMGB5	EQUALS	AMGB4	+1	# B(1)TMP
AMGB7	EQUALS	AMGB5	+1	# B(1)TMP
AMGB8	EQUALS	AMGB7	+1	# B(1)TMP
CAPSI	EQUALS	AMGB8	+1	# B(1)TMP
CDUXD	EQUALS	CAPSI	+1	# B(2)TMP
CDUYD	EQUALS	CDUXD	+2	# B(2)TMP
CDUZD	EQUALS	CDUYD	+2	# B(2)TMP
SLOPE	EQUALS	CDUZD	+2	# B(1)TMP
ADB	EQUALS	SLOPE	+1	# B(1)TMP
RMANNDX	EQUALS	ADB	+1	# B(1)TMP
PMANNDX	EQUALS	RMANNDX	+1	# B(1)TMP
YMANNDX	EQUALS	PMANNDX	+1	# B(1)TMP MUST BE LAST VARIABLE IN RCS

Page 111

***** ENTRY DAP TEMPORARY VARIABLES. *****

(69D)

ANGLE REGISTERS FOR ENTRY DAPS

AOG	EQUALS	BCDU		# 1P
AIG	EQUALS	AOG	+1	# 1P
AMG	EQUALS	AIG	+1	# 1P
ROLL/180	EQUALS	AMG	+1	# 1P
ALFA/180	EQUALS	ROLL/180	+1	# 1P
BETA/180	EQUALS	ALFA/180	+1	# 1P

AOG/PIP	EQUALS	BETA/180 +1	# 1P
AIG/PIP	EQUALS	AOG/PIP +1	# 1P
AMG/PIP	EQUALS	AIG/PIP +1	# 1P
ROLL/PIP	EQUALS	AMG/PIP +1	# 1P
ALFA/PIP	EQUALS	ROLL/PIP +1	# 1P
BETA/PIP	EQUALS	ALFA/PIP +1	# 1P

GYMBAL DIFFERENCES OVER INTERNAL TCDU = .1 SEC.

-DELAG	EQUALS	BETA/PIP +1	# 1P
-DELAIG	EQUALS	-DELAG +1	# 1P
-DELAMG	EQUALS	-DELAIG +1	# 1P

ESTIMATED BODY RATES

CMDAPMOD	EQUALS	-DELAMG +1	# 1P GOES BEFORE PREL FOR TM.
----------	--------	------------	-------------------------------

PREL	EQUALS	CMDAPMOD +1	# 1P P TCDU/180	(ROLDDOT)
QREL	EQUALS	PREL +1	# 1P Q TCDU/180	(PITCHDOT)
RREL	EQUALS	QREL +1	# 1P R TCDU/180	(YAWDOT)

BETADOT	EQUALS	RREL +1	# 1P MUST FOLLOW RREL. BETADOT TCDU/180
PHIDOT	EQUALS	BETADOT +1	# 1P

OLD (UNAVERAGED) BODY RATE MEASURE

OLDELP	EQUALS	PHIDOT +1	# 1P
OLDELQ	EQUALS	OLDELP +1	# 1P
OLDELR	EQUALS	OLDELQ +1	# 1P

JETAG	EQUALS	OLDELR +1	# 1P
-------	--------	-----------	------

TUSED	EQUALS	JETAG +1	# 1P ELAPSED TIME SINCE NOMINAL UPDATE.
-------	--------	----------	---

FOLLOWING 3 SP WORDS IN DOWNLINK. ROLLTM SENT EACH 1 SEC.

PAXERR1	EQUALS	TUSED +1	# 1P INTEGRATED ROLL ERROR/360.
ROLLTM	EQUALS	PAXERR1 +1	# 1P ROLL/180 FOR TM.
ROLLC	EQUALS	ROLLTM +1	# 2P ROLLCOM/360 FROM ENTRY (FOR TM)
			# KEEP ROLLC & ROLLHOLD ADJACENT FOR TP

Page 112

ROLLHOLD	EQUALS	ROLLC +2	# 1P FOR ATTITUDE HOLD IN CMDAPMOD = +1
----------	--------	----------	---

ENTRY DAP QUANTITIES THAT SHARE WITH RCS DAP.

ALFACOM	EQUALS	DCDU	# 1P KEEP ADJACENT TO BETACOM. <<
BETACOM	EQUALS	ALFACOM +1	# 1P

JET LIST. DT, JETBITS IN THIS ORDER.

TOFF	EQUALS	BETACOM +1	# 1P DP PAIR
TBITS	EQUALS	TOFF +1	# 1P
TON2	EQUALS	TBITS +1	# 1P DP PAIR

T2BITS EQUALS TON2 +1 # 1P

MISCELLANEOUS PERMANENT ERASABLE.

OUTTAG EQUALS T2BITS +1 # 1P

NUJET EQUALS OUTTAG +1 # 1P

MORE ENTRY DAP QUANTITIES THAT DO NOT SHARE WITH RCS DAP.

JETEM EQUALS ROLLHOLD +1 # 2P THIS DP USED IN RATEAVG

GAMA EQUALS JETEM +2 # 1P

GAMDOT EQUALS GAMA +1 # 1P

POSEXIT EQUALS GAMDOT +1 # 1P

CM/GYMDT EQUALS POSEXIT +1 # 1P

HEADSUP EQUALS CM/GYMDT +1 # 1P DSP NOUN 61 FOR P62,63,64,67.

P63FLAG EQUALS HEADSUP +1 # 1P INTERLOCK FOR WAKEP62

#>> SHARE BELOW WITH RCS RUPT TEMPS (< 15D) <<<

CALFA EQUALS SPNDX # 1P

SALFA EQUALS CALFA +1 # 1P

SINM EQUALS SALFA +1 # 1P

COSM EQUALS SINM +1 # 1P

SINO EQUALS COSM +1 # 1P

COSO EQUALS SINO +1 # 1P

SINOCOSM EQUALS COSO +1 # 1P

COSOCOSM EQUALS SINOCOSM +1 # 1P

#>> SHARE ABOVE WITH RCS RUPT TEMPS <<<

THE FOLLOWING FEW REGISTERS USED ONCE EACH 2 SEC

-VT/180 EQUALS NUJET +1 # 1P

LCX/360 EQUALS -VT/180 +1 # 1P

XD/360 EQUALS LCX/360 +1 # 1P

VSQ/4API EQUALS XD/360 +1 # 1P

JNDX EQUALS VSQ/4API +1 # 1P

JNDX1 EQUALS JNDX +1 # 1P

Page 113

TON1 EQUALS JNDX1 +1 # 1P DP PAIR

T1BITS EQUALS TON1 +1 # 1P

MISCELLANEOUS REGISTERS USED EACH UPDATE.

CM/SAVE EQUALS T1BITS +1 # 1P

JETEM2 EQUALS CM/SAVE +1 # 1P TEMPORARY STORAGE

DAP QUANTITIES SHARED WITH RCS DAP FOR TM & FLIGHT RECORDER.

VDT/180 = ERRORX # 1P (EDIT)

-VT/180E = ERRORY # 1P (EDIT)

PAXERR	EQUALS	AK	# 1P ROLL ERROR FOR NEEDLES
QAXERR	=	THETADX	# 1P SINCE AK1 IS ZEROED IN ATM DAP.
RAXERR	=	QAXERR +1	# 1P SINCE AK2 IS ZEROED IN TM DAP.

*** COLMANU (R60,R62) ****
 VECQTEMP EQUALS COFSKEW

Page 114

***** KALCMANU VARIABLES. (71D) *****

BCDU	EQUALS	YMANNDX +1	# B(3)TMP
KSPNDX	EQUALS	BCDU +3	# B(1)TMP
KDPNDX	EQUALS	KSPNDX +1	# B(1)TMP
TMIS	EQUALS	KDPNDX +1	# I(18) MUST BE IN THE SAME BANK AS RCS DAP
COFSKEW	EQUALS	TMIS +18D	# I(6) MUST BE IN THE SAME BANK AS RCS DAP
CAM	EQUALS	COFSKEW +6	# I(2) MUST BE IN THE SAME BANK AS RCS DAP
MIS	EQUALS	CAM +2	# I(18) (THE REST MAY GO ANYWHERE)
COF	EQUALS	MIS +18D	# I(6)TMP
SCAXIS	EQUALS	COF +6	# I(6)TMP
POINTVSM	EQUALS	SCAXIS +6	# I(6)TMP
AM	EQUALS	POINTVSM +6	# I(2)TMP
RAD	EQUALS	AM +2	# I(2)TMP

FIRST-ORDER OVERLAYS IN KALCMANU

KV1	EQUALS	TMIS	# I(6)TMP
MFISYM	EQUALS	TMIS	# I TMP
TMFI	EQUALS	TMIS	# I TMP
NCDU	EQUALS	TMIS	# B TMP
NEXTIME	EQUALS	TMIS +3	# B TMP
TTEMP	EQUALS	TMIS +4	# B TMP
KV2	EQUALS	TMIS +6	# I(6)TMP
BIASTEMP	EQUALS	TMIS +6	# B TMP
KV3	EQUALS	TMIS +12D	# I(6)TMP
CGF	EQUALS	TMIS +12D	# I TMP

BRATE	EQUALS	COFSKEW	# B TMP
TM	EQUALS	CAM	# B TMP

SECOND-ORDER OVERLAYS IN KALCMANU

P21	EQUALS	KV1	# I(2)TMP
D21	EQUALS	KV1 +2	# I(2)TMP
G21	EQUALS	KV1 +4	# I(2)TMP

SATURN BOOST STORAGE. SAVE TILL RCS DAP OPERATION. (17D)
 POLYNUM EQUALS BCDU # B(15) PAD LOADED

July 12, 2016

493

```
POLYLOC          =          POLYNUM +10D
SATRLRT          EQUALS    POLYNUM +15D    # B(2)  PAD LOADED

# MORE P11 STORAGE --PAD LOADED--          (2D)
# Page 115
# (NOTE:  THIS PAD LOAD WILL NOT BE PRESERVED THROUGHOUT THE MISSION AS IT SHARES STORAGE WITH
# ENTRY DAP AND TVC DAP)
RPSTART          EQUALS    SATRLRT +2      # B(1) PITCH ROLL START TIME
POLYSTOP          EQUALS    RPSTART +1      # B(1) POLY CUT OFF MINUS RPSTART SEC

# STORAGE FOR VHHDOT AND ATTDSP
BODY3            EQUALS    POLYSTOP +1      # B(1)OUT
BODY2            EQUALS    BODY3  +1        # B(1)OUT
BODY1            EQUALS    BODY2  +1        # B(1)OUT
SPOLYARG          EQUALS    BODY1  +1        # B(1)TMP ARGUMENT FOR POLLY

OLDBODY1         =          EDRIVEX          # 1 PULSE = 0.0432 DEGREES
OLDBODY2         =          EDRIVEY
OLDBODY3         =          EDRIVEZ

# STORAGE FOR S11.1
VDISP            EQUALS    SPOLYARG +1      # I(2)OUT          2(7) M/CS
HDISP            EQUALS    VDISP  +2        # I(2)OUT          2(29) M
HDOTDISP         EQUALS    HDISP  +2        # I(2)OUT          2(7) M/CS
BOOSTEMP         EQUALS    HDOTDISP +2      # B(3)TEMP

# P11 SATURN I/F          (9D)
SATRATE          EQUALS    BOOSTEMP +3      # B(4)PL MANEUVER RATES FOR SATURN STICK
SATSW            EQUALS    SATRATE +4      # B(1)TEM STATUS SW FOR BOOST TAKEOVER
BIASAK           EQUALS    SATSW  +1        # B(3)TEM STOR AKBIAS FOR BOOST TAKEOVER
SATSCALE         EQUALS    BIASAK  +3      # B(1) SCALE FACTOR FOR SATURN STEERING

# P21 STORAGE.          (1D)
GENRET           EQUALS    RAD    +2        # B(1)TMP

# R61CSM STORAGE.        (1D)
SAVBNK           EQUALS    GENRET  +1      # B(1) S-S SAVE EBANK FOR R61 SUBROUTINE

# CRS61.1 STORAGE FOR AUTOPILOT BANK.  (3D)
SAVEDCDU         EQUALS    SAVBNK  +1      # B(3)TMP

# R61 STORAGE.          (1D)
# Page 116
R61CNTR          EQUALS    SAVEDCDU +3      # (1)TMP

# ENTRY RESTART PROTECTION STORAGE.  --KEEP TEMPS IN ORDER--  (12D)
```

```

TEMPROLL      EQUALS  GENRET          # B(1)TMP COPY CYCLE REGISTER
TEMPALFA      EQUALS  TEMPROLL +1     # B(1)TMP COPY CYCLE REGISTER
TEMPBETA      EQUALS  TEMPALFA +1     # B(1)TMP COPY CYCLE REGISTER
60GENRET      EQUALS  TEMPBETA +1     # B(1)TMP QSAVE FOR S61.1 AND ENTRY.
S61DT         EQUALS  60GENRET +1     # B(1)TMP VARIABLE DT FOR S61.1 RESTART.

# ENTRY TM SHARING FOR ACCELERATION PROFILE.
XPIPBUFF      EQUALS  ADOT            # B(1) PIPA BUFFER FOR TM DURING ENTRY.
YPIPBUFF      EQUALS  XPIPBUFF +1     # B(1) PIPS FILED HERE EACH .5 SEC APPEAR
ZPIPBUFF      EQUALS  YPIPBUFF +1     # B(1) ON DOWNLIST ONCE PER SECOND DURING
XOLDBUFF      EQUALS  ZPIPBUFF +1     # B(1) ENTRY AFTER RCS DAP HAS BEEN DIS-
YOLDBUFF      EQUALS  XOLDBUFF +1     # B(1) ABLED.  NEWEST PIP VALUE REPLACES
ZOLDBUFF      EQUALS  YOLDBUFF +1     # B(1) PIPBUFF, WHICH IS MOVED INTO OLDBUFF.

# REENTRY VARIABLES SHARED WITH RCS DAP FOR TM & FLIGHT RECORDER.
Q7             =      THETADZ         # I(2) HI-WORD ONLY ON DNLIST.
ASPS(TM)       =      WBODY          # I(6)DWN
                                           #      ASKEP, ASP1, ASPUP, ASPDN, ASP3, ASP3

# P37 PAD LOADS
P37RANGE      EQUALS  R61CNTR +1     (1)
                                           # I(1)PL      *****

END-E6        =      P37RANGE +1     # FIRST UNUSED ERASABLE LOCATION IN E6

# Page 117
# EBANK-7 ASSIGNMENTS

SETLOC 3400

# *--*--*-- OVERLAY NUMBER 0 IN EBANK 7 --*--*--*

# EXTERNAL DELTA-V UPDATE. (21D)
# (MUST BE IN ORDER FOR UPDATE PROGRAM. ALSO ENTRY PROGRAM PICK UP 'LAT(SPL)' WITH A

LAT(SPL)      ERASE  +20D            # I(2)DSP NOUN 61 FOR P62,63,64,67
LNG(SPL)      EQUALS  LAT(SPL) +2    # I(2)DSP NOUN 61 FOR P62,63,64,67

DELVS LV      EQUALS  LNG(SPL) +2    # I(6)TMP DELTA VEL VECT, LOC VER COORDS
TIG           EQUALS  DELVS LV +6    # B(2)DSP NOUN 33 FOR X-V84(R32),P30,40.
RTARG         EQUALS  TIG +2         # I(6)IN DESIRED VEHICLE RADIUS VECTOR
DELLT4        EQUALS  RTARG +6       # I(2)IN TIME DIFFERENCE FOR INITVEL
ECSTEER       EQUALS  DELLT4 +2     # I(1)PL FOR P40'S
DELVLVC       =      DELVS LV
END-DELV      ERASE                  # *NEXT AVAIL LOC AFTER UNSHARED E7*

# SERVICER STORAGE. (13D)

```

July 12, 2016

495

DVTOTAL	EQUALS	END-DELV	# B(2)DSP NOUN 40,99 FOR P30,34,35,40
TGO	EQUALS	DVTOTAL +2	# B(2)
DVCNTR	EQUALS	TGO +2	# B(1)TMP
DELVREF	EQUALS	DVCNTR +1	# I(6)TMP

NOMTIG	EQUALS	END-KALC	# I(2) (CANNOT SHARE WITH KALCMANU # OR DELVREF)
END-SVCR	EQUALS	NOMTIG +2	# ***NEXT AVAILABLE AFTER SERVICER

# ALIGNMENT STORAGE.			(25D)
XSCD	EQUALS	END-SVCR	# I(6)TMP
YSCD	EQUALS	XSCD +6	# I(6)TMP
ZSCD	EQUALS	YSCD +6	# I(6)TMP
VEL/C	EQUALS	ZSCD +6	# I(6)TMP
R53EXIT	EQUALS	VEL/C +6	# I(1)TMP

# ALIGNMENT MARKDATA (DOWNLINK) ***** (7D)			
MARK2DWN	EQUALS	R53EXIT +1	# (7) USED BY ALIGNMENT P50'S

Page 118

--*-* OVERLAY NUMBER 1 IN EBANK 7 -*-*-*-*

# REENTRY ERASABLES.			(206D)
RTINIT	EQUALS	END-SVCR	# 6P
RTEAST	EQUALS	RTINIT +6	# 6P
RTNORM	EQUALS	RTEAST +6	# 6P
RT	EQUALS	RTNORM +6	# 6P
UNI	EQUALS	RT +6	# 6P
UNITV	EQUALS	UNI +6	# 6P
VEL	EQUALS	UNITV +6	# 6P
TIME/RTO	EQUALS	VEL +6	# 2P TIME OF INITIAL TARGET, RTO.
-VREL	EQUALS	TIME/RTO +2	# 6P
OLDUYA	EQUALS	-VREL +6	# 6P USED BY CM/POSE (ENTRY DAP)
UXA/2	EQUALS	OLDUYA +6	# 6P USED BY CM/POSE (ENTRY DAP) -UVA
URH	=	UXA/2	# P67 DISPLAY NOUN
UYA/2	EQUALS	UXA/2 +6	# 6P USED BY CM/POSE (ENTRY DAP) UYA
UZA/2	EQUALS	UYA/2 +6	# 6P USED BY CM/POSE (ENTRY DAP) UNA
UBX/2	EQUALS	UZA/2 +6	# 6P USED BY CM/POSE (ENTRY DAP)
UBY/2	EQUALS	UBX/2 +6	# 6P USED BY CM/POSE (ENTRY DAP)
UBZ/2	EQUALS	UBY/2 +6	# 6P USED BY CM/POSE (ENTRY DAP)
DTEAROT	EQUALS	UBZ/2 +6	# 2P
DIFF	EQUALS	DTEAROT +2	# 2P
DIFFOLD	EQUALS	DIFF +2	# 2P
FACTOR	EQUALS	DIFFOLD +2	# 2P

FACT1	EQUALS	FACTOR	+2	# 2P	
FACT2	EQUALS	FACT1	+2	# 2P	
#Q7	=	THETAD2		# 2P	SHARED FOR TM. P64-P66
VSQUARE	EQUALS	FACT2	+2	# 2P	
LAD	EQUALS	VSQUARE	+2	# 2P	
LOD	EQUALS	LAD	+2	# 2P	
L/DCMINR	EQUALS	LOD	+2	# 2P	
KLAT	EQUALS	L/DCMINR	+2	# 2P	
L/D	EQUALS	KLAT	+2	# 2P	
L/D1	EQUALS	L/D	+2	# 2P	
LEWD	=	VIO		# 2P	SHARED FOR TM. P64-P65
D	EQUALS	L/D1	+2	# 2P	DSP NOUN 64,66,68 FOR P63,64,67
#V1	=	ENDBUF	+1	# 2P	SHARED FOR TM. P64-P65
DLEWD	EQUALS	D	+2	# 2P	
K2ROLL	EQUALS	DLEWD	+2	# 2P	
GOTOADDR	EQUALS	K2ROLL	+2	# 1P	
TEM1B	EQUALS	GOTOADDR	+1	# 2P	
MM	EQUALS	TEM1B	+2	# 2P	
GRAD	EQUALS	MM	+1	# 2P	
FX	EQUALS	GRAD	+1	# 1P	OVERWRITES NEXT 5 LOCS IN P67
LEQ	EQUALS	FX	+1	# 2P	
DHOOK	EQUALS	LEQ	+2	# 2P	
AHOOKDV	EQUALS	DHOOK	+2	# 2P	
# Page 119					
DVL	EQUALS	AHOOKDV	+2	# 2P	
#A0	=	ENDBUF	+3	# 2P	SHARED FOR TM. (HI-WD) P84-P85
A1	EQUALS	DVL	+2	# 2P	
VBARS	EQUALS	A1	+2	# 2P	
COSG/2	EQUALS	VBARS	+2	# 2P	
#GAMMAL	=	GAMMAEI		# 2P	SHARED FOR TM. P64
GAMMAL1	=	22D		# 2P	
VS1	EQUALS	COSG/2	+2	# 2P	
VL	=	VPRED		# 2P	SHARED FOR TM. P64-P65
V	EQUALS	VS1	+2	# 2P	
#VREF	=	THETAD	+2	# 2P	SHARED FOR TM. P65
LATANG	EQUALS	V	+2	# 2P	ADJACENT FOR TM.
RDOT	EQUALS	LATANG	+2	# 2P	ADJACENT FOR TM.
THETAH	EQUALS	RDOT	+2	# 2P	DSP NOUN 64,67 FOR P63,64,67
#RDOTREF	=	THETAD		# 2P	SHARED FOR TM. P65
ALP	EQUALS	THETAH	+2	# 2P	
ASKEP	=	ASPS		# 2P)	THESE ARE STORED IN
ASP1	=	ASPS	+1	# 2P)	SEQUENCE, OVERLAPPING
ASPUP	=	ASPS	+2	# 2P)>HI-WD OF EACH<	HI-WORD ONLY APPEARING
ASPDWN	=	ASPS	+3	# 2P)	ON DOWNLIST, EXCEPT
ASP3	=	ASPS	+4	# 2P)	ASP3 IS COMPLETE.


```

C/D0          EQUALS  ALP      +2      # 2P      -1/D0
D0            EQUALS  C/D0     +2      # I(2)    CONSTANT DRAG
Q2            EQUALS  D0       +2      # 2P

# ROLL C IS LOCATED IN EBANK= AOG TO AID ENTRY DAP.
RTGO          EQUALS  Q2       +2      # 2P DSP NOUN 66 FOR P64,P67
DNRNGERR      EQUALS  RTGO     +2      # 2P DSP NOUN 66 FOR P64,67
XRNGERR       =      LATANG      #          FOR DISKY DISPLAY
KAT           EQUALS  DNRNGERR +2      # 2P
GMAX          EQUALS  KAT      +2      # 1P DSP NOUN 60 FOR P61,62,63
                                     # GMAX IS LOADED IN DOUBLE PRECISION.
L/DCALC       =      TTE                # 2P CALCULATED L/D FOR TM: P64-P67.
GAMMAL        =      GAMMAEI            # 2P SHARED FOR TM. P64.
PREDANG       =      GAMMAEI            #          FOR TM IN P67.
JJ            =      PREDANG +1          #          FOR TM IN P67.
VMAGI         EQUALS  GMAX      +1      # 2P DSP NOUN 62,64,66 FOR P11,63,64.
VIO           EQUALS  VMAGI     +2      # 2P DSP NOUN 63 FOR P61.
TTE           EQUALS  VIO       +2      # 2P DSP NOUN 63 FOR P61.
ASPS          EQUALS  TTE       +2      # I(2) HI-WORD ONLY ON DNLIST FOR TEMP
TTE1          EQUALS  ASPS      +2      # I(2) TMP HOLDS UNDECREMENTED TTE VALUE

# **** P60'S ****
RTGON64       EQUALS  RTGO                # RANGE ERRORS NEGATIVE IF FALLS SHORT
# Page 120
RTGON67       EQUALS  RTGO                # DSP NOUN 67

# REENTRY, RETURN TO EARTH COMMON DISPLAY      (4D)
VPRED         EQUALS  BETA12  +2      # DSP NOUN 60 FOR P61,62,63
GAMMAEI       EQUALS  VPRED   +2      # DSP NOUN 60 FOR P61,62,63

# DISPLAY REGISTER FOR VG      (2D)
VGDISP        EQUALS  GAMMAEI +2      # B(2)DSP N.40,42,99 FOR P30,34,35,37,40,
                                     #          41 VG DISPLAY

# SOME P11 DISPLAY REGISTERS      (6D)
ALTI          EQUALS  TTE1     +2      # 2P DSP NOUN 62 FOR P11.
HDOT          EQUALS  ALTI     +2      # 2P DSP NOUN 62 FOR P11.

# Page 121
# ***** OVERLAY NUMBER 2 IN EBANK 7 *****

# KALCMANU STORAGE.      (18D)
MFS           EQUALS  END-DELV      # I(18)
MFI           EQUALS  MFS           # I      TMP
DEL           EQUALS  MFS           # I      TMP

```

```

END-KALC          EQUALS  MFS      +18D    # **NEXT AVAIL LOC AFTER KALCMANU**

# MEASUREMENT INCORPORATION STORAGE (R22) STORAGE.      (56D)

TX789             EQUALS  END-KALC      # I(6)TMP
GAMMA             EQUALS  TX789  +6      # I(3)TMP
OMEGA             EQUALS  GAMMA  +2      # I(18)TMP
BVECTOR           EQUALS  OMEGA  +18D    # I(18)TMP
DELTAQ            EQUALS  BVECTOR +18D   # I(2)TMP
VARIANCE          EQUALS  DELTAQ  +2     # I(3)TMP
RCLP              EQUALS  VARIANCE +3    # I(6)TMP
GRP2SVQ           EQUALS  RCLP   +6     # I(1)TMP QSAVE FOR RESTARTS

# P20, P22, P23 DSP NOUN                      (5D)
N49DISP           EQUALS  BVECTOR      # B(5)TMP

# S22.1 STORAGE.                               (36D)
SVMRKDAT          EQUALS  GRP2SVQ +1     # I(36)TMP 5 SETS OF MARK DATA +PAD OF ONE

# **** CISELUNAR NAV. ERAS. (P20'S) ****        (45D)
TRUNX             EQUALS  SVMRKDAT +36D
DATATEST          EQUALS  TRUNX      # (1)
UBAR0             EQUALS  TRUNX  +1
UBAR1             EQUALS  UBAR0  +6
UBAR2             EQUALS  UBAR1  +6
RZC               EQUALS  UBAR2  +6
VZC               EQUALS  RZC    +6
UCLSTAR           EQUALS  VZC    +6
USSTAR            EQUALS  UCLSTAR +6
SRRETURN          EQUALS  USSTAR +6

# Page 122
# ***-***- OVERLAY NUMBER 3 IN EBANK 7 -***-***

# RENDEZVOUS GUIDANCE STORAGE. -- P32 ... P35 --      (8D)
DELTEEO           EQUALS  END-KALC      # I(2) S-S BACK VALUES OF DELTA TIME
DELEL             EQUALS  DELTEEO +2    # I(2) S-S
SECMAX            EQUALS  DELEL  +2     # I(2) S-S MAX STOP SIZE FOR ROUTINE
XXXALT            EQUALS  SECMAX +2     # I(2)

# S40.9 STORAGE                                   (16D)
VG                EQUALS  XXXALT  +2    # I(6)TMP
VRPREV            EQUALS  VG      +6    # I(6)
TNIT              EQUALS  VRPREV +6     # I(2)
TNITPREV          EQUALS  TNIT   +2     # I(2)

```

```

# S40.2,3 STORAGE. (1D)
AXISCODE          EQUALS  TNITPREV +2  # I(1)IN

# P30'S-P17 COMMON STORAGE. (24D)
RACT3             EQUALS  GRP2SVQ +1   # I(6)TMP POSITION OF ACTIVE AT TPI TIME.
VACT3             EQUALS  RACT3  +6    # I(6)TMP VELOCITY OF ACTIVE AT TPI TIME.
RPASS3            EQUALS  VACT3  +6    # I(6)TMP POSITION OF PASSIVE AT TPI TIME.
VPASS3            EQUALS  RPASS3 +6    # I(6)TMP VELOCITY OF PASSIVE AT TPI TIME.

# P76, N84 DISPLAY (6D)
DELVOV           EQUALS  RACT3        # I(6)DSP NOUN 84 FOR X-V84, P34-35

# INITVEL/MIDGIM STORAGE. (34D)
# (CALLED BY S34.1,2, S35.1,2, AND S40.9)
# (CALLS LAMBERT, CONIC SUBROUTINES)
RINIT            EQUALS  VPASS3 +6    # I(6)IN ACTIVE VEHICLE RADIUS VECTOR
VINIT            EQUALS  RINIT  +6    # I(6)IN ACTIVE VEHICLE VELOCITY VECTOR
RTARG1           EQUALS  VINIT  +6    # I(6)TMP SHIFTED RTARG
VIPRIME          EQUALS  RTARG1 +6    # I(6)OUT NEW VEL REQ AT INITIAL RADIUS
VTPRIME          EQUALS  VIPRIME +6   # I(6)OUT TOTAL VELOCITY AT DESIRED RADIUS
+MGA             EQUALS  VTPRIME +6   # I(2)DSP NOUN 45 FOR P30,34,35. +MID GIM.
COZY4            EQUALS  +MGA  +2    # I(2)TMP COSINE OF ANGLE WHEN ROT STARTS

# THE FOLLOWING OVERLAYS MEASUREMENT INCORP AND CANNOT SHARE WITH TPI
# Page 123
INTIME           EQUALS  AXISCODE +3
ITCTR           EQUALS  INTIME  +2    # I(1)TMP ITERATION COUNTER
END-IN/M        EQUALS  COZY4  +2    # ** NEXT AVAIL LOC AFTER INITVEL/MIDGIM **

# P34 AND P33 STORAGE. (OVERLAYS INITVEL/MIDGIM) (24D)
VAPREC          EQUALS  RINIT        # I(6) S-S PREC VEC FOR NOM TPI TIME (ACT V)
RAPREC          EQUALS  VINIT        # I(6) S-S PREC VEC FOR NOM TPI TIME (ACT V)
VPPREC          EQUALS  VIPRIME      # I(6) S-S PREC VEC FOR NOM TPI TIME (PASS)
RPPREC          EQUALS  VTPRIME      # I(6) S-S PREC VEC FOR NOM TPI TIME (PASS)

# P30, P40 INTERFACE. (20D)
RTIG            EQUALS  END-IN/M     # I(6)TMP
VTIG            EQUALS  RTIG  +6     # I(6)TMP
DELVSIN         EQUALS  VTIG  +6     # I(6)TMP
DELVEET3        EQUALS  DELVSIN     # TMP DELTA VEL VECT INERTIAL COORDS.
VGTEMP          EQUALS  DELVEET3
DELVSAB         EQUALS  DELVSIN +6   # I(2)TMP

# P35-P40 INTERFACE STORAGE. (OVERLAYS P30-P40 I/F STORAGE) (12D)
RPASS4          EQUALS  RTIG        # I(6)TMP POSITION OF PASSIVE AT INTERCEPT
VPASS4          EQUALS  RPASS4 +6    # I(6)TMP VELOCITY OF PASSIVE AT INTERCEPT

```

```

# TPI SEARCH (P17)                                (6D)
E2                                EQUALS  VPASS4  +6    # I(6)TMP

# P30-P40 COMMON STORAGE.                          (3D)
TPASS4                            EQUALS  DELVSAB +2    # I(2)TMP
TINT                             =        TPASS4      # I(2)
QTEMP                            EQUALS  TPASS4  +2    # I(1)TMP

# P30-P40 STORAGE.                                  (4D)
TTOGO                            EQUALS  QTEMP   +1    # B(2)DSP NOUN 35,40,45,59,99
                                         #          FOR P30,34,35,40,41,47, R30.
TTPI                             EQUALS  TTOGO   +2    # B(2)DSP NOUN 37 FOR P34 TPI TIME, CSECS.
# Page 124
END-P30S                         EQUALS  TTPI    +2    # ** NEXT AVAIL LOC AFTER P30-40 STORAGE. **

# P40 STORAGE.                                      (8D)
VGBODY                            EQUALS  END-P30S      # B(6)DSP NOUN 85 FOR P40,41,42 VG-SC COOR
DELVCTL                           =        VGBODY
P40TMP                            EQUALS  VGBODY  +6    # B(2)TMP

# P47 STORAGE.
DV47TEMP                         EQUALS  VG
DELVIMU                          EQUALS  P40TMP  +2    # I(6)DSP NOUN 83 FOR P47 DELTAV(IMU).

# S40.1 STORAGE.                                    (23D)
CSTEER                          EQUALS  DELVIMU +6    # I(2)IN
BDT                             EQUALS  CSTEER  +2    # I(6)IN
UT                              EQUALS  BDT    +6    # I(6)OUT THRUST DIRECTION
VGTIG                          EQUALS  UT      +6    # I(6)OUT
VGPREV                         =        VGTIG
F                              EQUALS  VGTIG  +6    # I(2)OUT S40.3 NEEDS THIS

QTEMP1                          EQUALS  F        +2    # I(1)TMP HOLDS RETURN

# R41                                                (2D)
T-TO-ADD                        EQUALS  QTEMP1  +1    # I(1D) FOR MIDTOAVE

# Page 125
# ***-**- OVERLAY NUMBER 4 IN EBANK 7 -***-**-

# S35.1 STORAGE.                                    (2D)
TSTRT                          EQUALS  END-P30S      # I(2)IN MIDCOURSE START TIME

# S34.1 STORAGE. (OVERLAYS S35.1 STORAGE)          (1)
TITER                          EQUALS  TSTRT        # I(1)TMP ITERATION COUNTER

```

```

# (P30-31 Q-SAVES) (1)
P30/31RT      EQUALS  TITER      # B(1) RETURN POINT

# P22 STORAGE. (6D)
S22WUNL      EQUALS  TSTRT  +2    # 1      WUNL W8 UNKNOWN INIT VALUE.
S22TOFF      EQUALS  S22WUNL +1    # 2      T SUB OFF
S22TPRIM     EQUALS  S22TOFF +2    # 2      SAVE TF
S22EORM      EQUALS  S22TPRIM +2   # 0 = EARTH -- NON-ZERO = MOON

# DOWNLINK ERASABLES FOR P22, P20 MARK DATA. (8D)
MARKDOWN     EQUALS  S22EORM +1    # B(1)
RM           EQUALS  S22RTNEX      # DOWNLINK OF VHF RANGE

# S22.1 (1D)
S22RTNEX     EQUALS  MARKDOWN +7   # B(1)

# P22 STORAGE (6D)
STARSAV3     EQUALS  S22RTNEX +1   # I(6)TMP

# CRS61.1 STORAGE. --A SUBSET OF P20-- (14D)
Q611         EQUALS  RM      +1     # I(1)TMP QSAVE
Q6111        EQUALS  Q611   +1     # I(1)TMP QSAVE
SAVEPOS      EQUALS  Q6111  +1     # I(6)TMP LEM POSITION VECTOR
# Page 126
SAVEVEL      EQUALS  SAVEPOS +6     # I(6)TMP LEM VELOCITY VECTOR

# ATTITUDE MANEUVER -- CALLED BY P20,R61,R63,CRS61.1 (3D)
PRAXIS       EQUALS  SAVEVEL +6     # B(3) S-S DISP RES FOR PREF AXIS N95.

# MARK ROUTINE (R21) STORAGE. -- IS SUBSET OF R22 -- (14D)
MRKBUF1      EQUALS  PRAXIS  +3     # B(7)TMP R21 MARK BUFFER.
MRKBUF2      EQUALS  MRKBUF1 +7     # B(7)TMP R21 MARK BUFFER.

# MORE CONICS STORAGE. (4)
COGA         EQUALS  3774          # I(2) COTAN OF INITIAL FLIGHT PATH ANGLE
INDEP        EQUALS  COGA          # I(1) USED BY SUBROUTINE 'ITERATOR'
EPSILONL     EQUALS  COGA  +2      # I(2)TMP

# RENDEZVOUS GUIDANCE STORAGE. -- P32...P35 -- (10D)
ELEV         EQUALS  MRKBUF2 +7     # I(2)TMP
RTX1         EQUALS  ELEV   +2     # (1)
RTX2         EQUALS  RTX1   +1     # (1)
RTMU         EQUALS  RTX2   +1     # (2)
RTSR1/MU     EQUALS  RTMU   +2     # (2)
CENTANG      EQUALS  RTSR1/MU +2    # I(2) S-S CENTRAL ANGLE COVERED (TPI-TFF)

```

```
# TPI SEARCH (S17.1, S17.2) P17 STORAGE.(10D)
DELTEE      EQUALS  MRKBUF2 +7      # I(2)
XRS         EQUALS  DELTEE +2      # I(2)
THETL       EQUALS  XRS +2         # I(2)
TF          EQUALS  THETL +2       # I(2)
DELHITE     EQUALS  TF +2         # (2)
```

```
# Page 127
```

```
# *- - *- - *- OVERLAY NUMBER 5 IN EBANK 7 - *- - *- - *
```

```
# P17,P34                                     (2D)
NN1      =      NN      # I(2)DSP NOUN 55,R1
```

```
# ***** THE FOLLOWING ARE FOR FLIGHT 504 ONLY *****
```

```
# RETURN-TO-EARTH STORAGE.                  (93D)
RTEDVD     EQUALS  END-IN/M      # I(2)IN DELTA VELOCITY DESIRED      M/CS
RTEGAM2D    EQUALS  RTEDVD +2    # I(2)IN REENTRY ANGLE DESIRED      REVS
RCON        EQUALS  RTEGAM2D +2  # I(2)TMP CONIC R2 RADIUS          M B23
R(T1)/      EQUALS  RCON +2      # I(6)TMP POSITION VECTOR AT TIG     M B23
R(T1)       EQUALS  R(T1)/ +6    # I(2)TMP MAGNITUDE OF R(T1)/      M B23
DT21PR      EQUALS  R(T1) +2     # I(2)TMP PREVIOUS DT21            CS B3
MAMAX1      EQUALS  DT21PR +2    # I(2)TMP MAJ AXIS LOW BOUND LMT    M B30
MAMAX2      EQUALS  MAMAX1 +2    # I(2)TMP MAJ AXIS UP BOUND LMT    M B30
R(T2)/      EQUALS  MAMAX2 +2    # I(6)TMP FINAL POSITION VECTOR     M B23
RD          EQUALS  R(T2)/ +6    # I(2)TMP FINAL R DESIRED          M B23
DRCON       EQUALS  RD +2        # I(2)TMP RCON SLOPE ITERATOR      M B23
RPRE'       EQUALS  DRCON +2     # I(2)TMP PREVIOUS RPRE            M B23
V(T1)/      EQUALS  RPRE' +2     # I(6)TMP VEL VECTOR AT TIG        M/CS
V2(T1)/     EQUALS  V(T1)/ +6    # I(6)TMP POST IMP VEL AT TIG      M/CS
DV          EQUALS  V2(T1)/ +6   # I(2)TMP DELTA VELOCITY AT TIG     M/CS
V(T2)/      EQUALS  DV +2        # I(6)TMP FINAL VELOCITY VECTOR     M/CS
T1          EQUALS  V(T2)/ +6    # I(2)TMP INITIAL VECTOR TIME      CS B2
PCON        EQUALS  T1 +2        # I(2)TMP SEMI-LATUS RECTUM        M B23
X(T1)       EQUALS  PCON +2      # I(2)TMP COTANGENT GAMMA1         B5
T12         EQUALS  X(T1) +2     # I(2)TMP INIT TO FINAL POSIT TIME CS B2
DELTAT      EQUALS  T12 +2      # I(2)TMP DELTA T IN SAVE PERILUNE CS B2
NN1A        EQUALS  DELTAT +2    # I(2)TMP ITERATION COUNTER 1
NN2         EQUALS  NN1A +2     # I(2)TMP ITERATION COUNTER 2
RTENCKEX    EQUALS  NN2 +2      # I(1)TMP RTENCK RETURN ADDRESS
CONICX1     EQUALS  RTENCKEX +1  # I(1)TMP CONICS MU TABLE INDEX
T2          EQUALS  CONICX1 +1   # I(2)TMP FINAL TIME              CS B2
UR1/        EQUALS  T2 +2        # I(6)TMP UNIT R(T1)/              B1
UV1/        EQUALS  UR1/ +6      # I(6)TMP UNIT V(T1)/              B1
BETA1       EQUALS  UV1/ +6      # I(2)TMP 1+X(T2)**2              B1
```

July 12, 2016

503

P(T1)	EQUALS	BETA1	+2	# I(1)TMP PRIMARY BODY STATE TIME 1	B14
CFPA	EQUALS	P(T1)	+1	# I(2)TMP COSINE FLIGHT PATH ANGLE	B1
PHI2	EQUALS	CFPA	+2	# I(2)TMP PERI OR APO INDICATOR	B2
SPRTEX	EQUALS	PHI2	+2	# I(1)TMP ROUTINE RETURN ADDRESS	
VNSTORE	EQUALS	SPRTEX	+1	# I(1)TMP VERBNOUN STORAGE	
BETA12	EQUALS	VNSTORE	+1	# I(2)TMP SIGN FOR TIMERAD	

OVERLAYS WITHIN RETURN-TO-EARTH STORAGE.

Page 128

RPRE	EQUALS	24D	# I(2)TMP COMPUTED PREC RADIUS	M B29/B27
P/RPRE	EQUALS	26D	# I(2)TMP P/R	B4
R/APRE	EQUALS	28D	# I(2)TMP R/A	B6
X(T2)PRE	EQUALS	T12	# I(2)TMP PREC COTAN GAMMA2	B0
X(T2)	EQUALS	DELTAT	# I(2)TMP COTAN GAMMA2	B0
UH/	EQUALS	UV1/	# I(2)TMP UNIT HORIZONTAL VECTOR.	B1
SPRTETIG	EQUALS	TIG	# I(2)IN TIME OF IGNITION	CS B28

Page 129

*-**-*- OVERLAY 6 IN EBANK 7 -**-*-*

P32,P33

THE FOLLOWING OVERLAY MEAS. INCORP. ARE AND IN USE ONLY WHEN (32D)

POSTCSI	EQUALS	VG	# I(2)	
DELVCSI	EQUALS	POSTCSI +2	# I(2)	
DELDV	EQUALS	DELVCSI +2	# I(2)	
GAMPREV	EQUALS	DELDV +2	# I(2)	
DVPREV	EQUALS	GAMPREV +2	# I(2)	
POSTCDH	EQUALS	DVPREV +2	# I(2)	
HAFPA1	EQUALS	POSTCDH		
VACT4	EQUALS	POSTCDH +2	# I(6)	
RDOTV	EQUALS	VACT4 +6	# I(2)	
VACT1	EQUALS	RDOTV +2	# I(6)	
VPASS1	EQUALS	VACT1 +6	# I(6) VEL. PASSIVE VEH. AT CSI TIME	

UNVEC EQUALS VACT3

T2TOT3 EQUALS TPASS4 # I(2) TPI - TCDH

CSIALRM EQUALS TITER # I(2) ALARM INDEX

DELVEET2 EQUALS S22WUNL # I(6) VACT3 - VACT2 = DVCDH REF. COORD.

ADDITIONAL CSI - CDH STORAGE. (10D)

RPASS1	EQUALS	CENTANG +2	# I(6) POS. PASSIVE VEH. AT CSI TIME.
LOOPCT	EQUALS	RPASS1 +6	# I(2) ITERATION COUNTER
NN	EQUALS	LOOPCT +2	# I(2)

```

# P21 STORAGE                                (19D)
P21ORIG      EQUALS  TRUNX                    # I(1)
P21BASER     EQUALS  P21ORIG +1              # I(6)
P21BASEV     EQUALS  P21BASER +6             # I(6)
P21ALT       EQUALS  P21BASEV +6             # I(2) NOUN 73 R1 ALTITUDE
P21VEL       EQUALS  P21ALT +2               # I(2) NOUN 73 R2 VELOCITY
P21GAM       EQUALS  P21VEL +2              # I(2) NOUN 73 R3 FLIGHT PATH ANGLE

# The following two statements had been just "WHOCARES = 3777".---RSB 2009
      SETLOC 3777
WHOCARES     EQUALS                                # A DUMMY FOR E-BANK INSENSITIVE 2CADRS.
END-E7       EQUALS  WHOCARES                   # ***** LAST LOCATION IN E7

# Page 130 ... is empty.

```

This code is written to file `src/ERASABLE-ASSIGNMENTS.s`.

B.29 EXECUTIVE

```

505  <src/EXECUTIVE.s 505>≡
    # Copyright:    Public domain.
    # Filename:     EXECUTIVE.agc
    # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
    #               It is part of the source code for the Command Module's (CM)
    #               Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:    yaYUL
    # Contact:       Ron Burkey <info@sandroid.org>.
    # Website:       www.ibiblio.org/apollo.
    # Pages:         1208-1220
    # Mod history:   2009-05-14 RSB   Adapted from the Colossus249/ file of the
    #               same name, using Comanche055 page images.
    #
    # This source code has been transcribed or otherwise adapted from digitized
    # images of a hardcopy from the MIT Museum. The digitization was performed
    # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
    # thanks to both. The images (with suitable reduction in storage size and
    # consequent reduction in image quality as well) are available online at
    # www.ibiblio.org/apollo. If for some reason you find that the images are
    # illegible, contact me at info@sandroid.org about getting access to the
    # (much) higher-quality images which Paul actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    #       Assemble revision 055 of AGC program Comanche by NASA
    #       2021113-051.  10:28 APR. 1, 1969
    #
    #       This AGC program shall also be referred to as
    #               Colossus 2A

    # Page 1208

                BLOCK    02

    # TO ENTER A JOB REQUEST REQUIRING NO VAC AREA:

                COUNT    02/EXEC

NOVAC          INHINT
                AD       FAKEPRET      # LOC(MPAC +6) - LOC(QPRET)
                TS       NEWPRIO       # PRIORITY OF NEW JOB + NOVAC C(FIXLOC)

                EXTEND
                INDEX    Q             # Q WILL BE UNDISTURBED THROUGHOUT.
                DCA      0             # 2CADR OF JOB ENTERED.

```

DXCH	NEWLOC	
CAF	EXECBANK	
XCH	FBANK	
TS	EXEITEM1	
TCF	NOVAC2	# ENTER EXECUTIVE BANK.

TO ENTER A JOB REQUEST REQUIREING A VAC AREA -- E.G., ALL (PARTIALLY) INTERPRETIVE

FINDVAC	INHINT	
	TS	NEWPRIO
	EXTEND	
	INDEX	Q
	DCA	0
SPVACIN	DXCH	NEWLOC
	CAF	EXECBANK
	XCH	FBANK
	TCF	FINDVAC2 # OFF TO EXECUTIVE SWITCHED-BANK.

TO ENTER A FINDVAC WITH THE PRIORITY IN NEWPRIO TO THE 2CADR ARRIVING IN A AND L:
USERS OF SPVAC MUST INHINT BEFORE STORING IN NEWPRIO.

SPVAC	XCH	Q
	AD	NEG2
	XCH	Q
	TCF	SPVACIN

TO SUSPEND A BASIC JOB SO A HIGHER PRIORITY JOB MAY BE SERVICED:

CHANG1	LXCH	Q
	CAF	EXECBANK
	XCH	BBANK
	TCF	CHANJOB

TO SUSPEND AN INTERPRETIVE JOB:

CHANG2	CS	LOC	# NEGATIVE LOC SHOWS JOB = INTERPRETIVE.
# ITRACE (4) REFERS TO "CHANG2"			
# Page 1209			

	TS	L
+2	CAF	EXECBANK
	TS	BBANK
	TCF	CHANJOB -1

Page 1210

TO VOLUNTARILY SUSPEND A JOB UNTIL THE COMPLETION OF SOME ANTICIPATED EVENT (I/O EV

July 12, 2016

507

JOBSLEEP	TS	LOC
	CAF	EXECBANK
	TS	FBANK
	TCF	JOBSLP1

TO AWAKEN A JOB PUT TO SLEEP IN THE ABOVE FASHION:

JOBWAKE	INHINT		
	TS	NEWLOC	
	CS	TWO	# EXIT IS VIA FINDVAC/NOVAC PROCEDURES.
	ADS	Q	
	CAF	EXECBANK	
	XCH	FBANK	
	TCF	JOBWAKE2	

TO CHANGE THE PRIORITY OF A JOB CURRENTLY UNDER EXECUTION:

PRIOCHNG	INHINT		# NEW PRIORITY ARRIVES IN A. RETURNS TO
	TS	NEWPRIO	# CALLER AS SOON AS NEW JOB PRIORITY IS
	CAF	EXECBANK	# HIGHEST. PREPARE FOR POSSIBLE BASIC-
	XCH	BBANK	# STYLE CHANGE-JOB.
	TS	BANKSET	
	CA	Q	
	TCF	PRIOCH2	

TO REMOVE A JOB FROM EXECUTIVE CONSIDERATIONS:

ENDOFJOB	CAF	EXECBANK
	TS	FBANK
	TCF	ENDJOB1

ENDFIND	CA	EXECTEM1	# RETURN TO CALLER AFTER JOB ENTRY
	TS	FBANK	# COMPLETE.
	TCF	Q+2	

EXECBANK	CADR	FINDVAC2
----------	------	----------

FAKEPRET	ADRES	MPAC -36D	# LOC(MPAC +6) - LOC(QPRET)
----------	-------	-----------	-----------------------------

Page 1211

LOCATE AN AVAILABLE VAC AREA

BANK	01
COUNT	01/EXEC

FINDVAC2	TS	EXECTEM1	# (SAVE CALLER'S BANK FIRST.)
	CCS	VAC1USE	

	TCF	VACFOUND	
	CCS	VAC2USE	
	TCF	VACFOUND	
	CCS	VAC3USE	
	TCF	VACFOUND	
	CCS	VAC4USE	
	TCF	VACFOUND	
	CCS	VAC5USE	
	TCF	VACFOUND	
	TC	BAILOUT	
	OCT	1201	# NO VAC AREAS.
VACFOUND	AD	TWO	# RESERVE THIS VAC AREA BY STORING A ZERO
	ZL		# IN ITS VAC USE REGISTER AND STORE THE
	INDEX	A	# ADDRESS OF THE FIRST WORD OF IT IN THE
	LXCH	0 -1	# LOW NINE BITS OF THE PRIORITY WORD.
	ADS	NEWPRIO	
NOVAC2	CAF	ZERO	# NOVAC ENTERS HERE. FIND A CORE SET.
	TS	LOCCTR	
	CAF	NO.CORES	# SEVEN SETS OF ELEVEN REGISTERS EACH.
NOVAC3	TS	EXECTEM2	
	INDEX	LOCCTR	
	CCS	PRIORITY	# EACH PRIORITY REGISTER CONTAINS -0 IF
	TCF	NEXTCORE	# THE CORRESPONDING CORE SET IS AVAILABLE.
NO.CORES	DEC	6	
	TCF	NEXTCORE	# AN ACTIVE JOB HAS A POSITIVE PRIORITY
			# BUT A DORMANT JOB'S PRIORITY IS NEGATIVE
# Page 1212			
CORFOUND	CA	NEWPRIO	# SET THE PRIORITY OF THIS JOB IN THE CORE
	INDEX	LOCCTR	# SET'S PRIORITY REGISTER AND SET THE
	TS	PRIORITY	# JOB'S PUSH-DOWN POINTER AT THE BEGINNING
	MASK	LOW9	# OF THE WORK AREA AND OVERFLOW INDICATOR.
	INDEX	LOCCTR	
	TS	PUSHLOC	# OFF TO PREPARE FOR INTERPRETIVE PROGRAMS.
	CCS	LOCCTR	# IF CORE SET ZERO IS BEING LOADED, SET UP
	TCF	SETLOC	# OVFINDD AND FIXLOC IMMEDIATELY.
	TS	OVFINDD	
	CA	PUSHLOC	
	TS	FIXLOC	
SPECTEST	CCS	NEWJOB	# SEE IF ANY ACTIVE JOBS WAITING (RARE).
	TCF	SETLOC	# MUST BE AWAKENED OUT UNCHANGED JOB.
	TC	CCSHOLE	

```

TC      CCSHOLE
TS      NEWJOB      # +0 SHOWS ACTIVE JOB ALREADY SET.
DXCH    NEWLOC
DXCH    LOC
TCF     ENDFIND

SETLOC   DXCH    NEWLOC      # SET UP THE LOCATION REGISTERS FOR THIS
        INDEX   LOCCTR
        DXCH    LOC
        INDEX   NEWJOB      # THIS INDEX INSTRUCTION INSURES THAT THE
        CS      PRIORITY    # HIGHEST ACTIVE PRIORITY WILL BE COMPARED
        AD      NEWPRIO     # WITH THE NEW PRIORITY TO SEE IF NEWJOB
        EXTEND   # SHOULD BE SET TO SIGNAL A SWITCH.
        BZMF    ENDFIND
        CA      LOCCTR      # LOCCTR IS LEFT SET AT THIS CORE SET IF
        TS      NEWJOB      # THE CALLER WANTS TO LOAD ANY MPAC
        TCF     ENDFIND     # REGISTERS, ETC.

NEXTCORE CAF      COREINC
        ADS      LOCCTR
        CCS      EXECTEM2
        TCF      NOVAC3
        TC       BAILOUT    # NO CORE SETS.
        OCT      1202

# Page 1213
# THE FOLLOWING ROUTINE SWAPS CORE SET 0 WITH THAT WHOSE RELATIVE ADDRESS IS IN NEWJOB.

        -2      LXCH    LOC
        -1      CAE     BANKSET      # BANKSET, NOT BBANK, HAS RIGHT CONTENTS.
CHANJOB  INHINT
        EXTEND
        ROR      SUPERBNK      # PICK UP CURRENT SBANK FOR BBCON
        XCH      L            # LOC IN A AND BBCON IN L.
        +4      INDEX   NEWJOB    # SWAP LOC AND BANKSET.
        DXCH    LOC
        DXCH    LOC

        CAE     BANKSET
        EXTEND
        WRITE   SUPERBNK      # SET SBANK FOR NEW JOB.
        DXCH    MPAC          # SWAP MULTI-PURPOSE ACCUMULATOR AREAS.
        INDEX   NEWJOB
        DXCH    MPAC
        DXCH    MPAC
        DXCH    MPAC +2
        INDEX   NEWJOB

```

	DXCH	MPAC +2	
	DXCH	MPAC +2	
	DXCH	MPAC +4	
	INDEX	NEWJOB	
	DXCH	MPAC +4	
	DXCH	MPAC +4	
	DXCH	MPAC +6	
	INDEX	NEWJOB	
	DXCH	MPAC +6	
	DXCH	MPAC +6	
	CAF	ZERO	
	XCH	OVFIND	# MAKE PUSHLOC NEGATIVE IF OVFIND NZ.
	EXTEND		
	BZF	+3	
	CS	PUSHLOC	
	TS	PUSHLOC	
	DXCH	PUSHLOC	
	INDEX	NEWJOB	
	DXCH	PUSHLOC	
	DXCH	PUSHLOC	# SWAPS PUSHLOC AND PRIORITY.
	CAF	LOW9	# SET FIXLOC TO BASE OF VAC AREA.
	MASK	PRIORITY	
	TS	FIXLOC	
	CCS	PUSHLOC	# SET OVERFLOW INDICATOR ACCORDING TO
	CAF	ZERO	
	TCF	ENDPRCHG -1	
# Page 1214			
	CS	PUSHLOC	
	TS	PUSHLOC	
	CAF	ONE	
	XCH	OVFIND	
	TS	NEWJOB	
ENDPRCHG	RELINT		
	DXCH	LOC	# BASIC JOBS HAVE POSITIVE ADDRESSES, SO
	EXTEND		# DISPATCH WITH A DTCB.
	BZMF	+2	# IF INTERPRETIVE, SET UP EBANK, ETC.
	DTCB		
# Page 1215			
	COM		# EPILOGUE TO JOB CHANGE FOR INTERPRETIVE
	AD	ONE	
	TS	LOC	# RESUME

TCF INTRSM

COMPLETE JOBSLEEP PREPARATIONS.

JOBSLP1	INHINT		
	CS	PRIORITY	# NNZ PRIORITY SHOWS JOB ASLEEP.
	TS	PRIORITY	
	CAF	LOW7	
	MASK	BBANK	
	EXTEND		
	ROR	SUPERBNK	# SAVE OLD SUPERBANK VALUE.
	TS	BANKSET	
	CS	ZERO	
JOBSLP2	TS	BUF +1	# HOLDS -- HIGHEST PRIORITY.
	TCF	EJSCAN	# SCAN FOR HIGHEST PRIORITY ALA ENDOFJOB.
NUCHANG2	INHINT		# QUICK... DON'T LET NEWJOB CHANGE TO +0.
	CCS	NEWJOB	
	TCF	+3	# NEWJOB STILL PNZ
	RELINT		# NEW JOB HAS CHANGED TO +0. WAKE UP JOB
	TCF	ADVAN +2	# VIA NUDIRECT. (VERY RARE CASE.)
	CAF	TWO	
	EXTEND		
	WOR	DSALMOUT	# TURN ON ACTIVITY LIGHT
	DXCH	LOC	# AND SAVE ADDRESS INFO FOR BENEFIT OF
	TCF	CHANJOB +4	# POSSIBLE SLEEPING JOB.

Page 1216

TO WAKE UP A JOB, EACH CORE SET IS FOUND TO LOCATE ALL JOBS WHICH ARE ASLEEP. IF THE FCADR I
 # LOC REGISTER OF ANY SUCH JOB MATCHES THAT SUPPLIED BY THE CALLER, THAT JOB IS AWAKENED. IF N
 # LOCCTR IS SET TO -1 AND NO FURTHER ACTION TAKES PLACE.

JOBWAKE2	TS	EXECTEM1	
	CAF	ZERO	# BEGIN CORE SET SCAN
	TS	LOCCTR	
	CAF	NO.CORES	
JOBWAKE4	TS	EXECTEM2	
	INDEX	LOCCTR	
	CCS	PRIORITY	
	TCF	JOBWAKE3	# ACTIVE JOB -- CHECK NEXT CORE SET.
COREINC	DEC	12	# 12 REGISTERS PER CORE SET.
	TCF	WAKETEST	# SLEEPING JOB -- SEE IF CADR MATCHES.
JOBWAKE3	CAF	COREINC	
	ADS	LOCCTR	

	CCS	EXECTEM2	
	TCF	JOBWAKE4	
	CS	ONE	# EXIT IF SLEEPING JOB NOT FOUND.
	TS	LOCCTR	
	TCF	ENDFIND	
WAKETEST	CS	NEWLOC	
	INDEX	LOCCTR	
	AD	LOC	
	EXTEND		
	BZF	+2	# IF MATCH.
	TCF	JOBWAKE3	# EXAMINE NEXT CORE SET IF NO MATCH.
	INDEX	LOCCTR	# RE-COMPLEMENT PRIORITY TO SHOW JOB AWAKE
	CS	PRIORITY	
	TS	NEWPRIO	
	INDEX	LOCCTR	
	TS	PRIORITY	
	CS	FBANKMSK	# MASK UP THE 2CADR OF THE WAKE ADDRESS
	MASK	NEWLOC	# USING THE CADR IN NEWLOC AND THE EBANK
	AD	2K	# HALF OF BBANK SAVED IN BANKSET.
	XCH	NEWLOC	
	MASK	FBANKMSK	
	INDEX	LOCCTR	
	AD	BANKSET	
	TS	NEWLOC +1	
	CCS	LOCCTR	# SPECIAL TREATMENT IF THIS JOB WAS
	TCF	SETLOC	# ALREADY IN THE RUN (0) POSITION.
	TCF	SPECTEST	

Page 1217

PRIORITY CHANGE. CHANGE THE CONTENTS OF PRIORITY AND SCAN FOR THE JOB OF HIGHEST P

PRIOCH2	TS	LOC	
	CAF	ZERO	# SET FLAG TO TELL ENDJOB SCANNER IF THIS
	TS	BUF	# JOB IS STILL HIGHEST PRIORITY.
	CAF	LOW9	
	MASK	PRIORITY	
	AD	NEWPRIO	
	TS	PRIORITY	
	COM		
	TCF	JOBSLP2	# AND TO EJSCAN.

Page 1218

RELEASE THIS CORE SET AND VAC AREA AND SCAN FOR THE JOB OF HIGHEST ACTIVE PRIORITY.

```

ENDJOB1      INHINT
              CS      ZERO
              TS      BUF +1
              XCH      PRIORITY
              MASK     LOW9
              TS      L

              CS      FAKEPRET
              AD      L

              EXTEND
              BZMF     EJSCAN      # NOVAC ENDOFJOB

              CCS      L
              INDEX    A
              TS      0

EJSCAN       CCS      PRIORITY +12D
              TC      EJ1
              TC      CCSHOLE
              TCF      +1

              CCS      PRIORITY +24D  # EXAMINE EACH PRIORITY REGISTER TO FIND
              TC      EJ1              # THE JOB OF HIGHEST ACTIVE PRIORITY.
              TC      CCSHOLE
              TCF      +1

              CCS      PRIORITY +36D
              TC      EJ1
-CCSPR       -CCS     PRIORITY
              TCF      +1

              CCS      PRIORITY +48D
              TC      EJ1
              TC      CCSHOLE
              TCF      +1

              CCS      PRIORITY +60D
              TC      EJ1
              TC      CCSHOLE
              TCF      +1

              CCS      PRIORITY +72D
              TC      EJ1

```

TC CCSHOLE
TCF +1

Page 1219

EVALUATE THE RESULTS OF THE SCAN.

CCS BUF +1 # SEE IF THERE ARE ANY ACTIVE JOBS WAITING
TC CCSHOLE
TC CCSHOLE

TCF +2
TCF DUMMYJOB
CCS BUF # BUF IS ZERO IS THIS IS A PRIOCHNG AND
TCF +2 # CHANGED PRIORITY IS STILL HIGHEST.
TCF ENDPRCHG -1

INDEX A # OTHERWISE, SET NEWJOB TO THE RELATIVE
Was CAF --- RSB 2004
CA 0 -1 # ADDRESS OF THE NEW JOB'S CORE SET.
AD -CCSPR
TS NEWJOB
TCF CHANJOB -2

EJ1 TS BUF +2
AD BUF +1 # - OLD HIGH PRIORITY.
CCS A
CS BUF +2
TCF EJ2 # NEW HIGH PRIORITY.
NOOP
INDEX Q
TC 2 # PROCEED WITH SEARCH.

EJ2 TS BUF +1
EXTEND
QXCH BUF # FOR LOCATING CCS PRIORITY + X INSTR.
INDEX BUF
TC 2

Page 1220

IDLING AND COMPUTER ACTIVITY (GREEN) LIGHT MAINTENANCE. THE IDLING ROUTIEN IS NOT A
BUT RATHER A SUBROUTINE OF THE EXECUTIVE.

EBANK= SELFRET # SELF-CHECK STORAGE IN EBANK.

DUMMYJOB CS ZERO # SET NEWJOB TO -0 FOR IDLING.
TS NEWJOB

July 12, 2016

515

```

                                RELINT
                                CS      TWO          # TURN OFF THE ACTIVITY LIGHT.
                                EXTEND
                                WAND     DSALMOUT
                                CCS      NEWJOB        # IS THE NEWJOB ACTIVE?
                                TCF      NUCHANG2      # YES... ONE REQUIRING A CHANGE JOB.
                                CAF      TWO          # NEW JOB ALREADY IN POSITION FOR
                                TCF      NUDIRECT      # EXECUTION

                                CA       SELFRET
                                TS       L            # PUT RETURN ADDRESS IN L.
                                CAF      SELF BANK
                                TCF      SUPDXCHZ +1  # AND DISPATCH JOB.

                                EBANK=   SELFRET
                                BB CON    SELFCHK
                                SELF BANK

                                NUDIRECT      EXTEND          # TURN THE GREEN LIGHT BACK ON.
                                WOR      DSALMOUT
                                DXCH     LOC            # JOBS STARTED IN THIS FASHION MUST BE
                                TCF      SUPDXCHZ

                                BLOCK     2            # IN FIXED-FIXED SO OTHERS MAY USE.

                                COUNT     02/EXEC

                                # SUPDXCHZ -- ROUTINE TO TRANSFER TO SUPERBANK.
                                # CALLING SEQUENCE:
                                #          TCF      SUPDXCHZ          # WITH 2CADR OF DESIRED LOCATION IN A + 1.

                                SUPDXCHZ      XCH      L            # BASIC.
                                +1           EXTEND
                                WRITE     SUPERBNK
                                TS        BBANK
                                TC        L

                                NEG100      OCT      77677
```

This code is written to file `src/EXECUTIVE.s`.

B.30 EXTENDED VERBS

```

516  <src/EXTENDED-VERBS.s 516>≡
      # Copyright:    Public domain.
      # Filename:     EXTENDED_VERBS.agc
      # Purpose:      Part of the source code for Comanche, build 055. It
      #               is part of the source code for the Command Module's
      #               (CM) Apollo Guidance Computer (AGC), Apollo 11.
      # Assembler:    yaYUL
      # Reference:     pp. 236-267
      # Contact:       Ron Burkey <info@sandroid.org>,
      #               Fabrizio Bernardini <fabrizio@spacecraft.it>
      # Website:       http://www.ibiblio.org/apollo.
      # Mod history:   2009-05-18 FB   Transcription Batch 3 Assignment.
      #               2009-05-20 RSB   Corrections:  P00D00 -> P00D00,
      #               GOTOP00H -> GOTOP00H, added a couple of
      #               missing instructions in Verb 96.
      #               2009-05-23 RSB   In SYSTEST, corrected TC FLAGWRD1 to
      #               CA FLAGWRD1.  Added a variety of SBANK=
      #               statements prior to 2CADDRs.  One day I'll
      #               have to figure out what yaYUL is doing
      #               wrong with those ....
      #
      # The contents of the "Comanche055" files, in general, are transcribed
      # from scanned documents.
      #
      # Assemble revision 055 of AGC program Comanche by NASA
      # 2021113-051.  April 1, 1969.
      #
      # This AGC program shall also be referred to as Colossus 2A
      #
      # Prepared by
      #
      #               Massachussets Institute of Technology
      #               75 Cambridge Parkway
      #               Cambridge, Massachusetts
      #
      # under NASA contract NAS 9-4065.
      #
      # Refer directly to the online document mentioned above for further
      # information.  Please report any errors to info@sandroid.org.

      # Page 236

      BANK      7
      SETLOC    EXTVERBS
      BANK

```

July 12, 2016

517

EBANK= OGC

COUNT* \$\$/EXTVB

FAN-OUT

GOEXTVB	INDEX	MPAC	# VERB-40 IS IN MPAC
	TC	LST2FAN	# FAN AS BEFROE.
LST2FAN	TC	VBZERO	# VB40 ZERO (USED WITH NOUN 20 ONLY)
	TC	VBCOARK	# VB41 COARSE ALIGN (USED WITH NOUN 20 OR # 91 ONLY)
	TC	IMUFINEK	# VB42 FINE ALIGN IMU
	TC	IMUATTCK	# VB43 LOAD IMU ATTITUDE ERROR METERS.
	TC	SETSURF	# VB44 SET SURFACE FLAG
	TC	RESTSRF	# VB45 RESET SURFACE FLAG
	TC	STABLISH	# VB46 ESTABLISH G+C CONTROL
	TC	LMTOCMSV	# VB47 MOVE LM STATE VECTOR INTO CM
	TC	DAPDISP	# VB48 LOAD A/P DATA
	TCF	CREWMANU	# VB49 START AUTOMATIC ATTITUDE MANEUVER
	TC	GOLOADLV	# VB50 PLEASE PERFORM
	TC	GOLOADLV	# VB51 PLEASE MARK
	TC	V52	# VB52 SET OFFSET NO. FOR P22
	TC	GOLOADLV	# VB53 PLEASE PERFORM COAS MARK
	TC	GOTOR23	# VB54 PLEASE MARK (R-21 BACKUP)
	TC	ALINTIME	# VB55 ALIGN TIME
	TC	TRACKTRM	# VB56 TERMINATE TRACKING (P20 + P25)
	TC	GOTOR21	# VB57 START R21 REND TRACK SIGHT MARK ROUT
	TC	ENATMA	# VB58 ENABLE AUTOMATIC ATTITUDE MANEUVER
	TC	GOLOADLV	# VB59 PLEASE CALIBRATE
	TC	V60	# VB60 SET CPHIX (N17) EQUAL TO CDU
	TC	V61	# VB61 SELECT MODE I
	TC	V62	# VB62 SELECT MODE II, ERROR WRT N22
	TC	V63	# VB63 SELECT MODE III, ERROR WRT N17
	TC	VB64	# VB64 CALCULATE, DISPLAY S-BAND ANT ANGLES
	TC	CKOPTVB	# V 65 E OPTICAL VERIFICATION FOR PRELAUNC
	TC	ATTACHED	# VB66 ATTACHED. MOVE THIS TO OTHER STATE
	TC	V67	# VB67 W MATRIX MONITOR
	TC	STROKON	# VB68 CSM STROKE TEST ON.
VERB69	TC	VERB69	# VB69 CAUSE RESTART
	TC	V70UPDAT	# VB70 UPDATE LIFTOFF TIME.
	TC	V71UPDAT	# VB71 UNIVERSAL UPDATE -- BLOCK ADDRESS
	TC	V72UPDAT	# VB72 UNIVERSAL UPDATE -- SINGLE ADDRESS
	TC	V73UPDAT	# VB73 UPDATE AGC TIME (OCTAL)
	TC	DNEDUMP	# VB74 INITIALIZE DOWN-TELEMETRY PROGRAM # FOR ERASABLE DUMP.

```

# Page 237
TC      LFTFLGON      # VB75 SET LIFTOFF FLAG.
TC      SETPRFLG      # VB76 SET PREFERRED ATTITUDE FLAG
TC      RESETPRF      # VB77 RESET PREFERRED ATT. FLAG
TC      CHAZFOGC      # CHANGE GYROCOMPASS LAUNCH AZIMUTH V78
TC      ALM/END        # V79 SPARE
TC      LEMVEC         # VB80 UPDATE LEM STATE VECTOR
TC      CSMVEC         # VB81 UPDATE CSM STATE VECTOR
TC      V82PERF        # VB82 REQUEST ORBIT PARAM DISPLAY (R30)
TC      V83PERF        # VB83 RANGE, RANGE RATE, +X AXIS (R31)
TC      ALM/END        # V84 SPARE
TC      V85PERF        # VB85 RANGE, RANGE RATE, SLOS (R32)
TC      V86PERF        # VB86 BACKUP MARK REJECT
TC      SETVHFLG      # VB87 SET VHF RANGE FLAG
TC      RESETVHF      # VB88 RESET VHF RANGE FLAG
TC      V89PERF        # V89-ALIGN X OR PRF CSM AXIS TO LOS (R63)
TC      V90PERF        # VB90-OUT OF PLAN PARAMETERS (R36)
TC      GOSHOSUM      # VB91 TEMP FOR HYBRID AND STG.
TC      SYSTEST        # VB92 OPERATE IMU PERFORMANCE TEST
TC      WMATRXNG      # VB93 CLEAR RENDWFLG
TC      VERB94         # VB94 DO R64
TC      ALM/END        # VB95 SPARE
TCF     VERB96         # VB96 SET QUITFLAG TO STOP INTEGRATION
TC      GOLOADLV      # VB97 PLEASE PERFORM ENGINE-FAIL (R41)
TC      ALM/END        # VB98 SPARE
TC      GOLOADLV      # VB99 PLEASE ENABLE ENGINE

# END OF EXTENDED VERB FAN

TESTXACT  CCS      EXTVBACT
          TC      ALM/END      # YES, TURN ON OPERATOR ERROR LIGHT
          CA      FLAGWRD4     # ARE PRIOS USING DSKY
          MASK     OC24100
          CCS      A
          TC      ALM/END

          CAF      OCT24      # SET BITS 3 AND 5
SETXTACT  TS      EXTVBACT    # NO. SET FLAG TO SHOW EXT VERB DISPLAY
          # SYSTEM BUSY

          CA      Q
          TS      MPAC +1

          CS      TWO      # BLANK EVERYTHING EXCEPT MM AND VERB
          TC      NVSUB
          TC      +1

```

July 12, 2016

519

	TC	MPAC +1	
XACTALM	TC	FALTON	# TURN ON OPERATOR ERROR LIGHT.
	TC	ENDEXT	# RELEASE MARK AND EXT. VERB DISPLAY SYS.
# Page 238			
TERMEXTV	EQUALS	ENDEXT	
ENDEXTVB	EQUALS	ENDEXT	
XACTO	CAF	ZERO	# RELEASE MARK AND EXT. VERB DISPLAY SYS.
	TC	SETXTACT	
ALM/END	TC	FALTON	# TURN ON OPERATOR ERROR LIGHT
GOPIN	TC	POSTJUMP	
	CADR	PINBRNCH	
OC24100	OCT	24100	
# Page 239			
# VBZERO	VERB 40	DESCRIPTION	
#			
# 1.		REQUIRE NOUN 20 (ICDU ANGLES)	
# 2.		REQUIRE AVAILABILITY OF EXT VERB DISPLAY SYSTEM	
# 3.		IF EITHER OF ABOVE CONDITIONS NOT PRESENT, TURN ON OPERATOR ERROR LIGHT AND GO	
# 4.		SET EXT VERB DISPLAY ACTIVE FLAG.	
# 5.		EXECUTE IMUZERO (ZERO IMU CDU ANGLES).	
# 6.		EXECUTE IMUSTALL (ALLOW TIME FOR DATA TRANSFER).	
# 7.		RELEASE EXT. VERB DISPLAY SYSTEM.	
VBZERO	TC	OP/INERT	
	TC	IMUZEROK	# RETURN HERE IF NOUN = ICDU(20)
	TC	ALM/END	# RETURN HERE IF NOUN = OCDU(91)
			# (NOT IN USE YET)
IMUZEROK	TC	CKMODCAD	# KEYBOARD REQUEST FOR ISS CDUZERO
	TC	BANKCALL	
	CADR	IMUZERO	
	TC	BANKCALL	# STALL
	CADR	IMUSTALL	
	TC	+1	
	TC	GOPIN	
OP/INERT	CS	OCT24	
	AD	NOUNREG	
	EXTEND		

```

BZF      XACTOQ      # IF = 20.

INCR     Q
AD       OPIMDIFF    # -71
EXTEND
BZF      XACTOQ

TC       ALM/END      # ILLEGAL.

OPIMDIFF DEC      -71

```

Page 240

VBCOARK VERB 41 DESCRIPTION

COARSE ALIGN IMU OR RADAR

#

1. REQUIRE NOUN 20 OR NOUN 91 OR TURN ON OPERATOR ERROR.

2. REQUIRE EXT VERB DISPLAY SYS AVAILABLE OR TURN ON OPERATOR ERROR LIGHT

#

CASE 1 NOUN 20 (ICDU ANGLES)

3. SET EXT VERB DISPLAY ACTIVE FLAG.

4. DISPLAY FLASHING V25,N22 (LOAD NEW ICDU ANGLES).

#

RESPONSES

A. TERMINATE

1. RELEASE EXT VERB DISPLAY SYSTEM.

B. PROCEED

1. DISPLAY FLASHING V25,N23 (LOAD DELTA ICDU ANGLES).

RESPONSES:

A. TERMINATE

1. RELEASE EXT VERB DISPLAY SYSTEM.

B. PROCEED

1. EXECUTE ICORK2.

C. ENTER

1. INCREMENT CDU ANGLES

2. EXECUTE ICORK2

C. ENTER

1. EXECUTE ICORK2

ICORK2

1. RE-DISPLAY VERB 41.

2. EXECUTE IMUCCARS (IMU COARSE ALIGN).

3. EXECUTE IMUSTALL (ALLOW TIME FOR DATA TRANSFER).

4. RELEASE EXT VERB DISPLAY SYSTEM.

#

CASE 2 NOUN 91 (OCDU ANGLES)

5. (REQUIRE OPTICS SWITCH TO BE AT COMUTER OR TURN ON OPERATOR ERROR AND

OPTICS AVAILABLE AND DISPLAY FLASHING V24,N92....LOAD NEW OPTICS ANGLE

AND RELEASE EXT VERB DISPLAY SYSTEM).

July 12, 2016

521

```
#      6.      RESPONSES TO V29,N92
#      A.      TERMINATE
#              RELEASE EXT VERB DISPLAY SYS.
#      B.      PROCEED OR ENTER
#              RE-DISPLAY VERB 41,      SET SWITCH TO INDICATE COURSE ALIGN OPTICS WORK
#              RELEASE EXT VERB DISPLAY SYSTEM.
```

```
VBCOARK      TC      OP/INERT
              TC      IMUCOARK      # RETURN HERE IF NOUN = ICDU (20)
              TC      OPTCOARK      # RETURN HERE IF NOUN = OCDU (91)
```

RETURNS TO L+1 IF NOUN 20 -- TO L+2 IF NOUN 91.

```
IMUCOARK      TC      CKMODCAD      # COARSE ALIGN FROM KEYBOARD
              TC      TESTXACT
              CAF      VNLODCDU      # CALL FOR THETAD LOAD
              TC      BANKCALL
              CADR      GOXDSPF
              TC      TERMEXTV
              TCF      +1
```

Page 241

```
ICORK2      CAF      IMUCOARV      # RE-DISPLAY COARSE ALIGN VERB.
              TC      BANKCALL
              CADR      EXDSPRET

              TC      BANKCALL      # CALL MODE SWITCHING PROG
              CADR      IMUCOARS

              TC      BANKCALL      # STALL
              CADR      IMUSTALL
              TC      ENDEXTVB
              TC      ENDEXTVB
```

```
VNLODCDU      VN      2522
IMUCOARV      VN      4100
```

Page 242

TEMPORARY ROUTINE TO RUN THE OPTICS CDUS FROM THE KEYBOARD

```
OPTCOARK      CA      OPTCADR
              TC      CKMODCAD +1
              TC      TESTXACT
              CAF      EBANK5
              TS      EBANK
```

	CCS	SWSAMPLE	# SEE IF SWITCH AT COMPUTER
	TC	+5	# SWITCH AT COMPUTER
	TC	+1	# NOT ON COMPUTER
	TC	FALTON	# TURN ON OPERATOR ERR
	TC	ALARM	# AND ALARM
	OCT	00115	
	CCS	OPTIND	# SEE IF OPTICS AVAILABLE
	TC	OPTC1	# IN USE
	TC	OPTC1	# IN USE
	TC	OPTC1	# IN USE
	TC	ALARM	# OPTICS RESERVED (OPTIND=-0)
	OCT	00117	
	TC	ENDEXT	
OPTC1	CAF	VNLDOCDU	# VERB-NOUN TO LOAD OPTICS CDUS
	TC	BANKCALL	
	CADR	GOXDSPF	
	TC	TERMEXTV	
	TC	+1	# PROCEED
	CA	SAC	
	TS	DESOPTS	
	CA	PAC	
	TS	DESOPPT	
	CAF	OPTCOARV	# RE-DISPLAY OUR OWN VERB
	TC	BANKCALL	
	CADR	EXDSPRET	
	CAF	ONE	
	TS	OPTIND	# SET COARS WORKING
	TC	ENDEXTVB	
	TC	ENDEXTVB	
VNLDOCDU	VN	2492	
OPTCOARV	EQUALS	IMUCOARV	# DIFFERENT NOUNS.
# Page 243			
# IMUFINEK	VERB 42	DESCRIPTION	
#	FINE ALIGN IMU		
#			
#	1.	REQUIRE EXT VERB DISPLAY AVAILABLE AND SET BUSY FLAG OR TURN ON OPER	
#	2.	DISPLAY FLASHING V25,N93....LOAD DELTA GYRO ANGLES....	
#		RESPONSES	

```

#           A.      TERMINATE
#           1.      RELEASE EXT VERB DISPLAY SYSTEM.
#           B.      PROCEED OR ENTER
#           1.      RE-DISPLAY VERB 42
#           2.      EXECUTE IMUFINE (IMU FIVE ALIGN MODE SWITCHING).
#           3.      EXECUTE IMUSTALL (ALLOW FOR DATA TRANSFER)
#                   A.      FAILED
#                           1.      RELEASE EXT VERB DISPLAY SYSTEM.
#                           B.      GOOD
#                           1.      EXECUTE IMUPULSE (TORQUE IRIGS).
#                           2.      EXECUTE IMUSTALL AND RELEASE EXT VERB DISPLAY S
#
IMUFINEK    TC      CKMODCAD      # FINE ALIGN WITH GYRO TORQUING.
            TC      TESTXACT
            CAF     VNLODGYR      # CALL FOR LOAD OF GYRO COMMANDS
            TC      BANKCALL
            CADR    GOXDSPF
            TC      TERMEXTV
            TC      +1            # PROCEED WITHOUT A LOAD
            CAF     IMUFINEV      # RE-DISPLAY OUR OWN VERB
            TC      BANKCALL
            CADR    EXDSPRET
            TC      BANKCALL      # CALL MODE SWITCH PROG
            CADR    IMUFINE
            TC      BANKCALL      # HIBERNATION
            CADR    IMUSTALL
            TC      ENDEXTVB
#
FINEK2      CAF     LGYROBIN      # PINBALL LEFT COMMANDS IN OGC REGISTERS
            TC      BANKCALL
            CADR    IMUPULSE
            TC      BANKCALL      # WAIT FOR PULSES TO GET OUT.
            CADR    IMUSTALL
            TC      ENDEXTVB
            TC      ENDEXTVB
#
LGYROBIN    ECADR   OGC
VNLODGYR    VN      2593
IMUFINEV     VN      4200      # FINE ALIGN VERB
CKMODCAD    CA      MODECADR
# Page 244

```

```

                                EXTEND
                                BZF    TCQ
                                TC      ALM/END                                # SOMEBODY IS USING MODECADR SO EXIT

# GOLOADLV      VERB 50      DESCRIPTION
#      AND OTHER PLEASE
#      DO SOMETHING VERBS
#
# PLEASE PERFORM, MARK, CALIBRATE, ETC.
#
#      1.      PRESSING ENTER ON DSKY INDICATES REQUESTED ACTION HAS BEEN PERFORMED.
#              SAME RECALL AS A COMPLETED LOAD.
#      2.      THE EXECUTION OF A VERB 33 (PROCEED WITHOUT DATA) INDICATES THE REQUIREMENT

GOLOADLV      TC      FLASHOFF
              CAF      PINSUPBT
              EXTEND
              WRITE    SUPERBNK      # TURN ON FE7
              TC      POSTJUMP
              SBANK=    PINSUPER
              CADR      LOADLV1

# V60      VERB 60
V60            EXTEND                                # SET ASTRONAUT TOTAL ATTITUDE (N17) EQUAL
              DCA      CDUX                                # TO PRESENT ATTITUDE
              DXCH     CPHIX
              CA      CDUZ
              TS      CPHIX    +2
              TC      GOPIN

# V61      VERB 61
V61            TC      DOWNFLAG      # SET NEEDLFLG TO 0 (FLAGWRD0,BIT9), PHASE
              ADRES    NEEDLFLG      # PLANE A/P FOLLOWING ERROR DISPLAYED
              TC      GOPIN

# V62      VERB 62
V62            TC      UPFLAG      # SET NEEDLFLG TO 1 (FLAGWRD0,BIT9),
              ADRES    NEEDLFLG      # TOTAL ATTITUDE ERROR DISPLAYED

              TC      UPFLAG      # SET N22ORN17 TO 1 (FLAGWRD9,BIT6),
              ADRES    N22ORN17      # COMPUTE TOTAL ATTITUDE ERROR WRT N22
              TC      GOPIN

# V63      VERB 63
V63            TC      UPFLAG      # SET NEEDLFLG TO 1 (FLAGWRD0,BIT9),
              ADRES    NEEDLFLG      # TOTAL ATTITUDE ERROR DISPLAYED

```

July 12, 2016

525

Page 245

TC	DOWNFLAG	# SET N22ORN17 TO 0 (FLAGWRD9,BIT6),
ADRES	N22ORN17	# COMPUTE TOAL ASTRONAUT ATTITUDE ERROR
TC	GOPIN	

Page 246

ALINTIME

VERB 55	DESCRIPTION
#	
# 1.	SET EXT VERB DISPLAY BUSY FLAG.
# 2.	DISPLAY FLASHING V25,N24 (LOAD DELTA TIME FOR AGC CLOCK.
# 3.	REQUIRE EXECUTION OF VERB 23.
# 4.	ADD DELTA TIME, RECEIVED FROM INPUT REGISTER, TO THE COMPUTER TIME.
# 5.	RELEASE EXT VERB DISPLAY SYSTEM.

COUNT 04/R33

ALINTIME

TC	TESTXACT	
CAF	VNLODDT	
TC	BANKCALL	
CADR	GOMARKF	
TC	ENDEXT	# TERMINATE
TC	ENDEXT	# PROCEED
CS	DEC23	# DATA IN OR RESEQUENCE (UNLIKELY)
AD	MPAC	# RECALL LEFT VERB IN MPAC
EXTEND		
BZF	UPDATIME	# GO AHEAD WITH UPDATE ONLY IF RECALL
TC	ENDEXT	# WITH V23 (DATA IN).

UPDATIME

INHINT		# DELTA TIME IS IN DSPTM1, +1.
CAF	ZERO	
TS	MPAC +2	# NEEDED FOR TP AGREE
TS	L	# ZERO T1 + 2 WHILE ALIGNING.
DXCH	TIME2	
DXCH	MPAC	
DXCH	DSPTM2 +1	# INCREMENT
DAS	MPAC	
TC	TPAGREE	# FORCE SIGN AGREEMENT.
DXCH	MPAC	# NEW CLOCK.
DAS	TIME2	
RELINT		
UPDTMEND	TC	ENDEXT
DEC23	DEC	23 # V 23

VNLODDT VN 2524 # V25N24 FOR LOAD DELTA TIME

Page 247

SYSTEST VERB 92 DESCRIPTION

OPERATE SELECTED SYSTEM TEST.

#

1. REQUIRE P00 OR P00- OR TURN ON OPERATOR ERROR.

2. TURN OFF DAP IF IT IS ON.

3. DISPLAY FLASHING V21,N01 (LOAD TEST NUMBER 1 THRU 17).

4. UPON ENTRY OF TEST NUMBER, SCHEDULE TSELECT WITH PRIORITY 20.

#

TSELECT

1. IF LOADED TEST NUMBER IS VALID, GO TO THAT TEST ROUTINE, OTHERWISE T
REPEAT LOAD REQUEST DISPLAY. (NO. 3 ABOVE)

EBANK= QPLACE

COUNT 04/EXTVB

SYSTEST

TC CHKPOOH

CA FLAGWRD1

IS NODOP01 FLAGBIT ON? (SET BY P11)

MASK NOP01BIT

EXTEND

BZF V92CONT

IF IT'S NOT YET SET, CONTINUE

TC P00D00

IT'S ON. SEND NODO ALARM FOR P07

OCT 1521

V92CONT

TC EXDAPOFF

TURN DAP OFF IF IT'S ON

CAF PRI020

TC FINDVAC

EBANK= QPLACE

SBANK= IMUSUPER

2CADR REDO

TC GOPIN

REDO AND TSELECT ARE NOW IN SYSTEM TEST.

COUNT* \$\$/EXTVB

CKOPTVB VERB 65 DESCRIPTION

OPTICAL VERIFICATION FOR PRELAUNCH.

1. SCHEDULE GCOMPVER, OPTICAL VERIFICATION SUBPROGRAM, WITH PRIORITY 17

CKOPTVB

TC CHECKMM

MM 02

I WONDER IF PRELAUNCH IS RUNNING

TC ALM/END

NOT RUNNING OPERATOR ERROR

INHINT

July 12, 2016

527

```
CAF      PRI016          # PRELAUNCH OPTICAL VERIFICATION
TC       FINDVAC
EBANK=   QPLACE
2CADR    COMVER          # STANDARD LEADIN TO GCOMPVER.

TC       GOPIN
```

Page 248

V 78

TO CHANGE GYROCOMPASS AZIMUTH

```
CHAZFOGC TC      CHECKMM      # IS IT PRELAUNCH
          MM       02
          TC       ALM/END      # NO -- OPERATOR ERROR
```

```
CAF      PRI016          # PRELAUNCH AZIMUTH CHANGE
TC       FINDVAC
EBANK=   XSM
2CADR    AZMTHCG1

TC       PHASCHNG
OCT      00174
TC       GOPIN
```

Page 249

IMUATTCK VERB 43 DESCRIPTION

LOAD IMU ATTITUDE ERROR METERS

#

- # 1. REQUIRE PROGRAM 00 ACTIVE, COARSE ALIGN ENABLE BIT OFF AND ZERO ICDU BIT OFF.
- # 2. IF GUID REF RELEASE OR LIFTOFF HAS OCCURRED REQUIRE EXT VERB DISPLAY AVAILABLE
- # FLAG, OTHERWISE ALLOW CURRENT EXT VERB DISPLAY TO BE OVER-RIDDEN.
- # 3. REMOVE COARSE ALIGN ENABLE AND IMU ERROR COUNTER ENABLE.
- # 4. DISPLAY FLASHING V25,N22 (LOAD NEW ICDU ANGLES).
- # 5. UPON PROCEED OR ENTER RESPONSE, INITIALIZE CURRENT DAC AND COMMAND VALUES, ENAB
- # TRANSFER LOADED VALUES TO REGISTERS, AND SEND COMMANDS.
- # 6. IF BUSY FLAG SET, RESET IT TO RELEASE EXT VERB DISPLAY.

```
IMUATTCK TC      CHKPOOH

          CAF      OCTAL30      # SEE IF IMU ZERO AND IMU COARSE ARE ON
          EXTEND
          RAND      CHAN12
          CCS       A
          TCF       ALM/END      # NOT ALLOWED IF IMU COARSE OR IMU ZERO ON

          TC       CKLFTBTS      # IS IT BEFORE OR AFTER LIFTOFF
          TC       TESTXACT      # AFTER
          CS       OCT50         # REMOVE COARSE AND ECTR ENABLE
```

```

EXTEND
WAND      CHAN12

CAF       VNLODCDU
TC        BANKCALL
CADR      GOXDSPF
TCF       TRMATTCK
TC        +1
CAF       EBANK6
TS        EBANK      # SET E6 FOR NEEDLES.

EBANK=    AK

TC        BANKCALL      # INITIALIZE CURRENT DAC AND
CADR      NEEDLE11      # COMMAND VALUES.

TC        BANKCALL      # ENABLE ERROR COUNTERS.
CADR      NEEDLER2

CAF       TWO           # 4 MS MIN.
TC        WAITLIST
EBANK=    AK
2CADR     ATTCK1

TRMATTCK  TC          CKLFTBTS      # IS IT BEFORE OR AFTER LIFTOFF
          TCF         ENDEXT      # AFTER
          TC          GOPIN

# Page 250
ATTCK1    EXTEND      # TRANSFER LOADED VALUES TO DESIRED REQS.
          DCA         THETAD
          DXCH        AK
          CAE         THETAD +2
          TS          AK          +2

          TC          IBNKCALL      # SENDS COMMANDS LIMITED TO +,- 384 PULSES
          CADR        NEEDLES      # AND LEAVES ERROR COUNTERS ENABLED.

          TC          TASKOVER

CKLFTBTS  CAF         GRRBKBIT      # HAS LIFTOFF OCCURRED
          MASK        FLAGWRD5
          CCS         A
          TC          Q            # YES
          CAF         BIT5
          EXTEND
          RAND        CHAN30

```


	CCS	A	
	TCF	Q+1	
XACTOQ	TC	Q	# YES
OCTAL30	OCT	30	
VB64	TC	CHKPOOH	# DEMAND PROGRAM 00.
	TC	TESTXACT	# IF DISPLAY SYS. NOT BUSY, MAKE IT BUSY.
	INHINT		
	CAF	PRI04	
	TC	FINDVAC	
	EBANK=	RHOSB	
	2CADR	SBANDANT	# CALC.,DISPLAY S-BAND ANTENNA ANGLES.
	TC	ENDOFJOB	

ENATMA VERB 58 DESCRIPTION
 # ENABLE AUTOMATIC ATTITUDE MANEUVER
 #
 # VERB58 RESETS STIKFLAG TO ENABLE R61 TO PERFORM AUTOMATIC TRACKING MANEUVERS, AFTER INTERRUPT
 # ACTIVITY.

ENATMA	TC	DOWNFLAG	# RESET STIKFLAG.
	ADRES	STIKFLAG	# BIT 14 FLAG 1
	TC	GOPIN	

Page 251

STROKON VERB 68 DESCRIPTION
 # STROKE TEST SETUP/ENABLE
 # 1. SET EXT VERB DISPLAY BUSY FLAG
 # 2. SCHEDULE STRKTST1 WITH PRIORITY 30.
 # 3. RELEASE EXT VERB DISPLAY.

	EBANK=	T5TVCDT	
STROKON	CS	FLAGWRD6	# V68 PERMITTED ONLY DURING TVC
	MASK	OCT60000	
	EXTEND		
	BZMF	ALM/END	# NOT TVC....FLASH OP ERROR LIGHT
	CAF	PRI030	# JOB REQUEST, TO SET UP STROKE TEST,
	TC	NOVAC	# INCLUDING INITIALIZATIONS
	SBANK=	PINSUPER	# Added RSB 2009.
	EBANK=	STROKER	
	2CADR	STRKTSTI	
	TC	GOPIN	

STABLISH VERB 46 DESCRIPTION

```

#      ESTABLISH G AND N AUTOPILOT CONTROL
#      1.      SETS UP EITHER RCS, ENTRY, OR SATURN
#      2.      IF TVC IS ON, SETS UP CSM/LM SWITCH-OVER
#              FROM HIGH BW TO LOW BW

```

```

STABLISH      CAF      EBANK6      # V46 - SET EBANK TO E6
              TS      EBANK

              CS      FLAGWRD6      # TEST FOR TVC
              MASK    OCT60000
              EXTEND
              BZMF    +8

              CAE      DAPDATR1      # TET FOR CSM/LM
              MASK    BIT14
              EXTEND
              BZMF    +3

              TC      POSTJUMP      # CSM/LM, SO PERFORM HB TO LB SWITCH-OVER
              CADR    PRESWTCH

+3            TC      ALM/END

+8            TC      POSTJUMP      # SET UP RCS, ENTRY, OR SATURN-STICK DAP
              CADR    DAPFIG

```

Page 252

```

# CREMANU      VERB 49      DESCRIPTION
#      START AUTOMATIC ATTITUDE MANEUVER
#

```

```

#      1.      REQUIRE PROGRAM 00 ACTIVE.
#      2.      SET EXT VERB DISPLAY BUSY FLAG.
#      3.      SCHEDULE R62DISP WITH PRIORITY 10.
#      4.      RELEASE EXT VERB DISPLAY.
#

```

R62DISP

```

#      1.      DISPLAY FLASHING V06,N22 (DECIMAL DISPLAY NEW ICDU ANGLES).  UPON IM
#              4 FOR RESTART OF DISPLAY SEQUENCE.
#      RESPONSES
#      A.      TERMINATE
#              1.      GOTOPOOH
#      B.      PROCEED
#              1.      SET 3AXISFLG TO INDICATE MANEUVER IS SPECIFIED BY 3 A
#              2.      EXECUTE R60CSM (ATTITUDE MANEUVER).
#              3.      ZERO GROUP 4 (END R62).
#      C.      ENTER

```

1. REPEAT FLASHING V06,N22.

CREWMANU TC CHKPOOH # DEMAND P00

TC TESTXACT

CAF PRIO10

TC FINDVAC

EBANK= CPHI

2CADR R62DISP

TC ENDOFJOB

Page 253

DAPDISP VERB 48 DESCRIPTION

LOAD AUTOPILOT DATA (ROUTINE R03)

#

0. CHECKFAIL AND RETURN IF TVC.

1. REQUIRE EXT VERB DISPLAY AVAILABLE AND SET BUSY FLAG.

2. LOWER PRIORITY TO 10.

3. DISPLAY FLASHING V04,N46 (DISPLAY AUTOPILOT CONFIGURATION).

4. UPON PROCEED RESPONSE, EXECUTE S41.2.

5. DISPLAY FLASHING V06,N47 (DISPLAY CSM WGT., LEM WGT.)

6. UPON PROCEED RESPONSE EXECUTE S40.14.

7. DISPLAY FLASHING V06,N48 (DISPLAY PITCH TRIM, YAW TRIM)

8. UPON PROCEED RESPONSE, RELEASE EXTENDED VERB DISPLAY SYSTEM.

COUNT* \$\$/EXTVB

DAPDISP CS FLAGWRD6

MASK OCT60000

EXTEND

BZMF +2 # TVC = 10, CS YIELDS 01, BZMF TO CONTINUE

TC ALM/END # RETURN IF TVC

TC TESTXACT

TC BANKCALL

CADR DAPDISP1

BANK 42

SETLOC EXTVBS

BANK

COUNT 24/R03

DAPDISP1 CAF EBANK6

TS EBANK

	CAF	PRI010				
	TC	PRI0CHNG				
DONOUN46	CAF	V04N46	#	R1	R2	
	TC	BANKCALL	#	DAPDATR1	DAPDATR2	
	CADR	GOXDSPF	#	GOXDSP ROUTINES USED FOR EXTENDED VERBS.		
	TC	ENDEXT	#	EXT. VBS GO TO ENDEXT, NOT ENDOFJOB.		
	TC	+2				
	TC	DONOUN46				
	CA	DAPDATR1				
	MASK	BIT4				
	CCS	A				
	TCF	MAXIN				
	TC	DOWNFLAG				
	ADRES	MAXDBFLG				
MAXOUT	TC	BANKCALL				
	CADR	S41.2				
DONOUN47	CAF	V06N47	#	R1	R2	R3
# Page 254	TC	BANKCALL	#	CSM WGT.	LEM WGT.	BLANK
	CADR	GOXDSPF				
	TC	ENDEXT				
	TC	+2				
	TC	DONOUN47				
	CAE	DAPDATR1	#	DO MASS PROPERTIES CALCULATION ONLY IF		
	MASK	PRI030	#	CONFIG = 1(CSM), 2 (CSM/LM), 6(CSM/LMA)		
	EXTEND					
	BZF	DONOUN48	#	SKIP IF 0,4		
	COM					
	MASK	PRI030				
	EXTEND					
	BZF	DONOUN48	#	SKIP IF 3,7		
	INHINT					
	TC	IBNKCALL				
	CADR	MASSPROP	#	UPDATE IXX, IAVG, IAVG/TLX		
	RELINT					
	TC	BANKCALL				
	CADR	S40.14	#	COMPUTE RCS DAP STUFF		
DONOUN48	CAF	V0648	#	R1	R2	R3
	TC	BANKCALL	#	PTRIM	YTRIM	BLANK

July 12, 2016

533

	CADR	GOXDSPF
	TC	ENDEXT
	TC	ENDEXT
	TC	DONOUN48
MAXIN	TC	UPFLAG
	ADRES	MAXDBFLG
	TC	MAXOUT
V0648	VN	0648
V06N47	VN	0647
V04N46	VN	0446
	BANK	43
	SETLOC	EXTVERBS
	BANK	
	COUNT*	\$\$/EXTVB

```
# V82PERF      VERB82      DESCRIPTION
#      REQUEST ORBIT PARAMETERS DISPLAY (R30)
#
#      1.      IF AVERAGE G IS OFF:
#                  FLASH DISPLAY V04N06.  R2 INDICATES WHICH SHIP'S STATE VECTOR IS
#                  TO BE UPDATED.  INITIAL CHOICE IS THIS SHIP (R2=1).  ASTRONAUT
#                  CAN CHANGE TO OTHER SHIP BY V22EXE, WHERE X NOT EQ I.
#                  SELECTED STATE VECTOR UPDATED BY THISPREC (OTHPREC).
#                  CALLS SR30.1 (WHICH CALLS TFFCONMU + TFFRP/RA) TO CALCULATE
# Page 255
#                  RPER (PERIGEE RADIUS), RAPO (APOGEE RADIUS), HPER (PERIGEE
#                  HEIGHT ABOVE LAUNCH PAD OR LUNAR LANDING SITE), HAPO (APOGEE
#                  HEIGHT AS ABOVE), TPER (TIME TO PERIGEE), TFF (TIME TO
#                  INTERSECT 300 KFT ABOVE PAD OR 35KFT ABOVE LANDING SITE).
#                  FLASH MONITOR V16N44 (HAPO, HPER, TFF).  TFF IS -59M59S IF IT WAS
#                  NOT COMPUTABLE, OTHERWISE IT INCREMENTS ONCE PER SECOND.
#                  ASTRONAUT HAS OPTION TO MONITOR TPER BY KEYING IN N 32 E.
#                  DISPLAY IS IN HMS, IS NEGATIVE (AS WAS TFF), AND INCREMENTS
#                  ONCE PER SECOND ONLY IF TFF DISPLAY WAS -59M59S.
#
#      2.      IF AVERAGE G IS ON:
#                  CALLS SR30.1 APPROX EVERY TWO SECS.  STATE VECTOR IS ALWAYS
#                  FOR THIS VEHICLE.  V82 DOES NOT DISTURB STATE VECTOR.  RESULTS
#                  OF SR30.1 ARE RAPO, RPER, HAPO, HPER, TPER, TFF.
#                  FLASH MONITOR V16N44 (HAPO, HPER, TFF).
#                  IF MODE IS P11, THEN CALL DELRSPL SO ASTRONAUT CAN MONITOR
#                  RESULTS BY N50E.  SPLASH COMPUTATION DONE ONCE PER TWO SECS.
```

```

#
# ADDENDUM:      HAPO AND HPER SHOULD BE CHANGED TO READ HAPOX AND HPERX IN THE
#                ABOVE REMARKS.

V82PERF          TC          TESTXACT

                  CAF        PRI07
                  TC         PRI0CHNG
                  TC         POSTJUMP
                  CADR        V82CALL          # ***** V82CALL MUST NOT BE A FINDVAC JOB.

# VB83PERF        VERB 83          DESCRIPTION
#                REQUEST RENDEZVOUS PARAMETER DISPLAY (R31)
#
#                1.      SET EXT VERB DISPLAY BUSY FLAG.
#                2.      SCHEDULE V83CALL WITH PRIORITY 10.
#                A.      DISPLAY
#                        R1      RANGE
#                        R2      RANGE RATE
#                        R3      THETA

V83PERF          TC          TESTXACT
                  INHINT
                  CS         FLAGWRD9          # SET R31 FLAG-BIT 4 FLAGWRD9
                  MASK       R31FLBIT
                  ADS        FLAGWRD9
                  CAF        PRI05
                  TC         NOVAC
                  SBANK=     LOWSUPER          # Added by RSB 2009
                  EBANK=     SUBEXIT
                  2CADR      R31CALL

                  TC         ENDOFJOB

# Page 256
V85PERF          TC          TESTXACT
                  INHINT
                  CS         R31FLBIT          # RESET R31 FLAG TO INDICATE R34
                  MASK       FLAGWRD9
                  TS         FLAGWRD9
                  TC         V83PERF +5

# Page 257
#                GOTOR21        VERB 57
#                GOTOR23-       VERB 54          DESCRIPTION
# SET UP MARKING FOR R22 (REND TRACK DATA PROC)
# 1.      SET EXT VERB DISPLAY BUSY FLAG

```

- # 2. IF REND (P20 RUNNING) + TRACK (TRACKING ALLOWED) FLAGS ARE SET,
 # SCHEDULE R21 OR R23 WITH PRIORITY 16, OTHERWISE TURN ON ALARM 406
 # 3. RELEASE EXT VERB DISPLAY SYSTEM

```

GOTOR21      TC      DOWNFLAG      # CLEAR R23FLG
              ADRES    R23FLG      # BIT 9 FLAG 1
              TC      +3
GOTOR23      TC      UPFLAG        # SET R23FLG
              ADRES    R23FLG      # BIT 9 FLAG 1
              TC      TESTXACT
              CA      FLAGWRDO      # VB 57 UNACCEPTABLE UNLESS BOTH
              MASK     RNDVZBIT      # RENDEZVOUS AND TRACK FLAGS ON
              EXTEND
              BZF      R22ALARM

              CA      FLAGWRD1
              MASK     TRACKBIT
              EXTEND
              BZF      R22ALARM

              CA      FLAGWRD1      # TEST R23FLG
              MASK     R23BIT
              EXTEND
              BZF      REGR21      # R21
              CAF      PRI016
              TC      NOVAC
              EBANK=    MRKBUF1
              2CADR     R23CSM

REGR21      TC      ENDOFJOB
             CAF      PRI016
             TC      NOVAC
             EBANK=    MRKBUF1
             2CADR     R21CSM

R22ALARM     TC      ENDOFJOB
             TC      ALARM          # VERB 57 WAS SELECTED AND NEITHER REND
             OCT      00406        # NOR TRACK FLAG WERE ON.
             TC      ENDEXT

```

Page 258

VERB 86 DESCRIPTION

V86 IS TO R23 AS MARK REJECT IS TO R21

V86 IS THE MARK REJECT FOR R23 (THE BACKUP MARKING ROUTINE)

EBANK= MRKBUF1

V86PERF	CAF	EBANK7	# BACKUP MARK REJECT (R23)
	XCH	EBANK	
	CA	NEGONE	
	TS	MRKBUF1	
	TC	GOPIN	

Page 259

TRACKTRM VERB 56 DESCRIPTION

TERMINATE TRACKING (P20)

1. KNOCK DOWN RENDEZVOUS, TRACK, AND UPDATE FLAGS.

2. REQUIRE P20 NOT RUNNING ALONE OR GO TO GOTOPOOH (REQUEST PROGRAM 00)

3. REQUIRE R22 RUNNING OR GO TO PINBRNCH.

4. IF INTEGRATION RUNNING, STALL UNTIL IT IS COMPLETED, THEN ZERO GROUPS

3. KNOCK DOWN RENDEZFOUS, R22, R21, TRACK, UPDATE, AND TARG1 FLAGS.

4. GO TO ENEMA (SOFTWARE RESTART).

REFERENCE

P20 RENDEZVOUS NAVIGATION

R21 RENDEZVOUS TRACKING SIGHTING MARK.

R22 RENDEZVOUS TRACKING DATA PROCESSING.

TRACKTRM CA RNDVZBIT # IS REND FLAG ON

MASK FLAGWRDO

EXTEND

BZF GOPIN # NO

TC DOWNFLAG

ADRES RNDVZFLG

CA TRACKBIT # IS TRACK FLAG ON

MASK FLAGWRD1

EXTEND

BZF GOPIN # NO

TC DOWNFLAG

ADRES TRACKFLG

TC DOWNFLAG

ADRES UPDATFLG

TC DOWNFLAG

ADRES IMUSE

CAF EBANK6

TS EBANK

INHINT


```

TC      STOPRATE

CAF     NEGONE
TS      OPTIND

TC      INTPRET
CALL
        INTSTALL      # DON'T INTERRUPT INTEGRATION
EXIT

TC      2PHSCHNG

# Page 260
OCT     2              # KILL GROUP 2 TO HALT P20 ACTIVITY
OCT     1              # ALSO KILL GROUP 1

CLEANOUT INHINT
TC      POSTJUMP
CADR    ENEMA          # CAUSE RESTART

# LEMVEC      VERB 80      DESCRIPTION
#      UPDATE LEM STATE VECTOR
#      RESET VEHUPFLG TO 0

LEMVEC   TC      DOWNFLAG
        ADRES    VEHUPFLG      # VEHUPFLG DOWN INDICATES LEM

        TCF      GOPIN

# CSMVEC      VERB 81      DESCRIPTION
#      UPDATE CSM STATE VECTOR
#      SET VEHUPFLG TO 1

CSMVEC   TC      UPFLAG
        ADRES    VEHUPFLG      # VEHUPFLG UP INDICATES CM

        TCF      GOPIN

# DNEDUMP     VERB 74      DESCRIPTION
#      INITIALZE DOWN-TELEMETRY PROGRAM FOR ERASABLE MEMORY DUMP.
#
#      1.      SET EXT VERB DISPLAY BUSY FLAG.
#      2.      REPLACE CURRENT DOWNLIST WITH ERASABLE MEMORY.
#      3.      RELEASE EXT VERB DISPLAY.

        EBANK=   10
DNEDUMP  CAF      LDNDUMPI

```

	TS	DNTMGOTO	
	TC	GOPIN	
V74	EQUALS	DNEDUMP	
LDNDUMPI	REMADR	DNDUMPI	
# LFTFLGON	VERB 75	DESCRIPTION	
#	SET LIFT-OFF FLAG		
#	1.	SETUP GRRBKFLG, GUIDANCE REFERENCE RELEASE BACK-UP FLAG.	
#	2.	RETURN VIA PINBRNCH	
LFTFLGON	TC	UPFLAG	# VB 75 -- SET LIFTOFF FLAG BIT
	ADRES	GRRBKFLG	# BIT 5 FLAG 5
	TC	GOPIN	

Page 261

CHKPOOH	CA	MODREG
	EXTEND	
	BZF	TCQ
	TCF	ALM/END

EXDAPOFF	EXTEND	
	DCA	IDLECADR # SET T5 TO IDLE.
	DXCH	T5LOC
	CS	OCT60000
	MASK	FLAGWRD6 # RESET DAPBITS 1 AND 2.
	TS	FLAGWRD6
	TC	Q
	SBANK=	PINSUPER # Added RSB 2009
	EBANK=	PACTOFF
IDLECADR	2CADR	T5IDLOC

Page 262

# VERB 89	DESCRIPTION	RENDEZVOUS FINAL ATTITUDE ROUTINE (R63)
#		
#	# CALLED BY VERB 89 ENTER DURING P00. PRIO 10 IS USED. CALCULATES AND	
#	# DISPLAYS FINAL GIMBAL ANGLES TO POINT CSM +X AXIS OR PREFERRED AXIS	
#	# (UNIT(Z)COS55 DEG + UNIT(X)SIN55 DEG) AT LM.	
#		
#	# 1. KEY IN V 89 E ONLY IF IN PROG 00. IF NOT IN P00, OPERATOR ERROR AND	
#	# EXIT R63, OTHERWISE CONTINUE.	
#		
#	# 2. IF IN P00, DO IMU STATUS CHECK ROUTINE (R02BOTH). IF IMU ON AND ITS	
#	# ORIENTATION KNOWN TO CGC, CONTINUE.	
#		

```

# 3. FLASH DISPLAY V 04 N 06.  R2 INDICATES WHICH SPACECRAFT AXIS IS TO
# BE POINTED AT LM.  INITIAL CHOICE IS PREFERRED AXIS (R2=1).
# ASTRONAUT CAN CHANGE TO (+X) AXIS (R2 NOT = 1) BY V 22 E 2 E.  CONTINUE
# AFTER KEYING IN PROCEED.
#
# 4. SET PREFERRED ATTITUDE FLAG ACCORDING TO OPTION DESIRED.  SET FLAG
# FOR PREFERRED AXIS.  RESET FLAG FOR X AXIS.
#
# 5. CURRENT TIME IS STORED AND R63COMP IS CALLED
#
#       R63COMP JOB:
#
#           UPDATES CSM AND LM STATE VECTORS USING CONIC EQUATIONS.
#
#           CALCULATES BOTH PREFERRED AND X AXIS TRACKING ATT FROM CSM TO LM.
#
#           DESIRED GIMBAL ANGLES AS INDICATED BY PREFERRED ATTITUDE FLAG
#           ARE STORED FOR LATER R60CSM CALL.
#
# 6.  FLASH DISPLAY V 06 N18 AND AWAIT RESPONSE.
#
# 7.  RECYCLE:  RETURN TO STEP 5.
#       TERMINATE:  EXIT R63 ROUTINE
#       PROCEED:  RESET 3AXISFLG AND CALL R60CSM FOR ATTITUDE MANEUVER.

```

```

V89PERF      TC      CHKPOOH      # DEMAND P00
              TC      TESTXACT
              INHINT
              CAF      PRI010
              TC      FINDVAC
              SBANK=   LOWSUPER      # Added by RSB 2009.
              EBANK=   P21TIME
              2CADR    V89CALL

```

```

              TCF      ENDOFJOB

```

```

WMATRXNG     TC      DOWNFLAG      # RESET RENDWFLAG
              ADRES    RENDWFLG

```

```

# Page 263

```

```

              TC      DOWNFLAG      # RESET ORBWFLAG
              ADRES    ORBWFLAG
              TC      GOPIN

```

```

GOSHOSUM     EQUALS  SHOWSUM

```

SHOWSUM	TC	CHKPOOH	
	TC	TESTXACT	# *
	CAF	S+1	# *
	TS	SKEEP6	# * SHOWSUM OPTION
	CAF	S+ZERO	# *
	TS	SMODE	# * TURN OFF SELF-CHECK
	CA	SELFADRS	# *
	TS	SELFRET	# *
	TC	STSHOSUM	# * ENTER ROPECHK
SDISPLAY	LXCH	SKEEP2	# * BNK NO FOR DSP
	LXCH	SKEEP3	# * BUGGER WORD FOR DSP
NOKILL	CA	ADRS1	# *
	TS	MPAC +2	# *
	CA	VNCON	# * 0501
	TC	BANKCALL	# *
	CADR	GOXDSPF	# *
	TC	+3	# *
	TC	NXTBNK	# *
	TC	NOKILL	# *
	CA	SELFADRS	
	TS	SKEEP1	
	TC	ENDEXT	# *
VNCON	VN	501	# *
ENDSUMS	CA	SKEEP6	# *
	EXTEND		# *
	BZF	SELFCHK	# * ROPECHK, START SELFCHK AGAIN.
	TC	STSHOSUM	# * START SHOWSUM AGAIN.

VB 76 --- SET PREFERRED ATTITUDE FLAG --- DRIVE TO PREFERRED.

SETPRFLG	TC	UPFLAG	
	ADRES	PRFTRKAT	# BIT 10 FLAG 5
	TC	GOPIN	

VB 77 --- RESET PREFERRED ATTITUDE FLAG --- DRIVE TO +X-AXIS ATT.

RESETPRF	TC	DOWNFLAG	
	ADRES	PRFTRKAT	# BIT 10 FLAG 5
	TC	GOPIN	

VB 87 --- SET VHF RANGE FLAG --- ALLOWS R22 TO ACCEPT RANGE DATA.

SETVHFLG	TC	INTPRET	
	SET	EXIT	
		VHFRFLAG	
	TC	GOPIN	

VB 88 --- RESET VHF RANGE FLAG --- STOPS ACCEPTANCE OF RANGE DATA.

RESETVHF	TC	INTPRET	
	CLEAR	EXIT	
		VHFRFLAG	
	TC	TRFAILOF	# TRACKER FAIL LIGHT
	TC	GOPIN	

VERB 66. VEHICLES ARE ATTACHED. --- MOVE THIS VEHICLE STATE VECTOR TO
OTHER VEHICLE STATE VECTOR.

USE SUBROUTINE GENTRAN.

	EBANK=	RRECTHIS	
ATTACHED	CAF	PRI010	
	TC	FINDVAC	
	EBANK=	RRECTHIS	
	2CADR	ATTACHIT	

TC	ENDOFJOB
----	----------

ATTACHIT	TC	INTPRET	
	CALL		
		INTSTALL	
	SET	BON	
		MOONOTH	
		MOONTHIS	
		+3	
	CLEAR		
		MOONOTH	
	EXIT		
	CAF	OCT51	
	TC	GENTRAN	
	ADRES	RRECTHIS	# OUR STATE VECTOR INTO OTHER VIA GENTRAN
	ADRES	RRECTOTH	

TACHEXIT	RELINT	
	TC	INTPRET

```

                                CALL          # UPDATE RN, VN, R-OTHER, V-OTHER
                                PTOACSM
# Page 265
                                LXA,2        CALL
                                PBODY
                                SVDWN1
                                CALL          SVDWN2
                                EXIT
                                CAF          TCPINAD
                                INDEX        FIXLOC
                                TS           QPRET
                                TC           POSTJUMP
                                CADR         INTWAKE

TCPIN                          RTB
                                PINBRNCH

OCT51                          OCT          51
TCPINAD                        CADR         TCPIN

# VERB 47.  MOVE LM STATE VECTOR INTO CSM STATE VECTOR

LMTOCMSV                      CAF          PRI010
                                TC           FINDVAC
                                EBANK=      RRECTHIS
                                2CADR       LMTOCM

                                TC           ENDOFJOB

LMTOCM                         TC           INTPRET
                                CALL
                                INTSTALL
                                SET          BON
                                MOONTHIS
                                MOONOTH
                                +3
                                CLEAR
                                MOONTHIS
                                EXIT

                                CAF          OCT51
                                TC           GENTRAN
                                ADRES        RRECTOTH
                                ADRES        RRECTHIS
                                # LM STATE VECTOR INTO CM VIA GENTRAN

```

```

TCF      TACHEXIT

# VERB 94 --- DO R64 VIA ENEMA TO PICK UP IN P23.

VERB94    CAF      V94FLBIT
          MASK     FLAGWRD9      # IS V94FLAG SET
# Page 266
          EXTEND
          BZF      ALM/END      # NO --- OPERATOR ERROR

          TC       DOWNFLAG
          ADRES    V94FLAG

          TC       CHECKMM      # IS IT P23
          MM       23
          TC       ALM/END      # NO -- OPERATOR ERROR
          TC       PHASCHNG
          OCT      112          # SET GROUP 2 TO DO R64

          TC       CLEANOUT     # CAUSE RESTART

# V90PERF  VERB 90      DESCRIPTION
#          REQUEST RENDEZVOUS OUT-OF-PLANE DISPLAY (R36)
#
#          1.      SET EXT VERB DISPLAY BUSY FLAG.
#          2.      SCHEDULE R36 CALL WITH PRIORITY 10
#          A.      DISPLAY
#                   TIME OF EVENT -- HOURS, MINUTES, SECONDS
#                   Y          OUT-OF-PLANE POSITION -- NAUTICAL MILES
#                   YDOT      OUT-OF-PLANE VELOCITY -- FEET/SECOND
#                   PSI        ANGLE BTW LINE OF SIGHT AND FORWARD
#                   DIRECTION VECTOR IN HORIZONTAL PLANE -- DEGREES

V90PERF    TC       TESTXACT
          CAF      PRI07      # R36,V90
          TC       FINDVAC
          SBANK=    PINSUPER   # Added RSB 2009
          EBANK=    RPASS36
          2CADR     R36

          TCF      ENDOFJOB

# VERB 96  SET QUITFLAG TO STOP INTEGRATION

VERB96     TC       UPFLAG      # QUITFLAG WILL CAUSE INTEGRATION TO EXIT

```

	ADRES	QUITFLAG	#	AT NEXT TIMESTEP
	TC	UPFLAG		
	ADRES	V96ONFLG		
	CAF	ZERO		
	TC	POSTJUMP		
	CADR	V37	#	GO TO P00
	EBANK=	LANDMARK		
V52	TC	CHECKMM	#	IS P22 OPERATING
	MM	22		
	TC	ALM/END	#	NO
	CAF	LANDBANK		
	TS	EBANK		
# Page 267				
	CS	PRI07	# YES	SET BITS 12,11,10 OF LANDMARK =
	MASK	LANDMARK	#	BITS 14,13,12 OF MARKSTAT AFTER
	TS	LANDMARK	#	SUBT. THEM FROM 5 TO GET OFFSET
	CA	MARKSTAT	#	MARK NO.
	TS	SR		
	CA	SR		
	CA	SR		
	MASK	PRI07		
	CS	A		
	AD	PRI05		
	ADS	LANDMARK		
	TC	GOPIN		
LANDBANK	ECADR	LANDMARK		
# VERB 67 ASTRONAUT DISPLAY OF W MATRIX				
V67	TC	TESTXACT		
	CAF	PRI05		
	TC	FINDVAC		
	EBANK=	W		
	2CADR	V67CALL		
	TC	ENDOFJOB		
# VB 44. SET SURFACE FLAG.				
SETSURF	TC	UPFLAG		
	ADRES	SURFFLAG		
	TCF	GOPIN		


```
# VB 45. RESET SURFACE FLAG
```

```
RESTSRF      TC      DOWNFLAG  
              ADRES   SURFFLAG  
              TCF     GOPIN
```

This code is written to file `src/EXTENDED-VERBS.s`.

B.31 FINDCDUW–GUIDAP INTERFACE

```

546  <src/FINDCDUW-GUIDAP-INTERFACE.s 546>≡
# Copyright:      Public domain.
# Filename:       FINDCDUW--GUIDAP_INTERFACE.agc
# Purpose:        Part of the source code for Luminary 1A build 099.
#                It is part of the source code for the Lunar Module's (LM)
#                Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:     yaYUL
# Contact:        Hartmuth Gutsche <hgutsche@explornet.com>.
# Website:        www.ibiblio.org/apollo.
# Pages:          908-925
# Mod history:    2009-05-28 HG    Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum. The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum. Many thanks to both. The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo. If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
#
# Page 908
# PROGRAM NAME:   FINDCDUW
# MOD NUMBER:     1           68-07-15
# MOD AUTHOR:     KLUMPP
#
# OBJECTS OF MOD:
#               1.      TO SUPPLY COMMANDED GIMBAL ANGLES FOR NOUN 22.
#               2.      TO MAINTAIN CORRECT AND CURRENT THRUST
#                       DIRECTION DATA IN ALL MODES. THIS IS DONE BY
#                       FETCHING FOR THE THRUST DIRECTION FILTER THE
#                       CDUD'S IN PNGCS-AUTO, THE CDU'S IN ALL OTHER
#                       MODES.
#               3.      TO SUBSTITUTE A STOPRATE FOR THE NORMAL
#                       AUTOPILOT COMMANDS WHENEVER
#                       1) NOT IN PNGCS-AUTO, OR
#                       2) ENGINE IS OFF.
#
# FUNCTIONAL DESCRIPTION:

```

```

#
# FINDCDUW PROVIDES THE INTERFACES BETWEEN THE VARIOUS POWERED FLITE GUIDANCE PROGRAMS
# AND THE DIGITAL AUTOPILOT.  THE INPUTS TO FINDCDUW ARE THE THRUST COMMAND VECTOR
# AND THE WINDOW COMMAND VECTOR, AND THE OUTPUTS ARE THE GIMBAL ANGLE
# INCRMENTS, THE COMMANDED ATTITUDE ANGLE RATES, AND THE COMMANDED
# ATTITUDE LAG ANGLES (WHICH ACCOUNT FOR THE ANGLES BY WHICH THE BODY WILL
# LAG BEHIND A RAMP COMMAND IN ATTITUDE ANGLE DUE TO THE FINITE ANGULAR
# ACCELERATIONS AVAILABLE).
#
# FINDCDUW ALIGNS THE ESTIMATED THRUST VECTOR FROM THE THRUST DIRECTION
# FILTER WITH THE THRUST COMMAND VECTOR, AND, WHEN XDVINHIB SET,
# ALIGNS THE +Z HALF OF THE LM ZX PLANE WITH THE WINDOW COMMAND VECTOR.
#
# Page 909
# SPECIFICATIONS:
#
# INITIALIZATION:      A SINGLE INTERPRETIVE CALL TO INITCDUW IS REQUIRED
#                      BEFORE EACH GUIDED MANEUVER USING FINDCDUW.
#
# CALL:                INTERPRETIVE CALL TO FINDCDUW WITH THE THRUST COMMAND
#                      VECTOR IN MPAC.  INTERPRETIVE CALL TO FINDCDUW -2 WITH
#                      THE THRUST COMMAND VECTOR IN UNFC/2 AND NOT IN MPAC.
#
# RETURNS:             NORMAL INTERPRETIVE IN ALL CASES
#                      1.  NORMALLY ALL AUTOPILOT CMDS ARE ISSUED.
#                      2.  IF NOT PNGCS AUTO, DO STOPRATE AND RETURN
#                          WITHOUT ISSUING AUTOPILOT CMDS.
#                      3.  IF ENGINE OFF, DO STOPRATE AND RETURN WITHOUT
#                          ISSUING AUTOPILOT CMDS.
#
# ALARMS:              00401  IF INPUTS DETERMINE AN ATTITUDE IN GIMBAL LOCK.
#                      FINDCDUW DRIVES CDUXD AND CDUYD TO THE RQD VALUES,
#                      BUT DRIVES CDUZD ONLY TO THE GIMBAL LOCK CONE.
#
#                      00402  IF UNFC/2 OR UNWC/2 PRODUCE OVERFLOW WHEN
#                      UNITIZED USING NORMUNIT.  FINDCDUW ISSUES
#                      STOPRATE AS ONLY INPUT TO AUTOPILOT.
#
# INPUTS:              UNFC/2      THRUST COMMAND VECTOR, NEED NOT BE SEMI-UNIT.
#                      UNWC/2      WINDOW COMMAND VECTOR, NEED NOT BE SEMI-UNIT.
#                      OGABIAS     POSSIBLE BIAS FOR OUTER GIMBAL ANGLE (ZEROED IN INITCDU
#                      XOVINHIB    FLAG DENOTING X AXIS OVERRIDE INHIBITED.
#                      CSMDOCKD    FLAG DENOTING CSM DOCKED.
#                      STEERSW     FLAG DENOTING INSUFF THRUST FOR THRUST DIR FLTR.
#
# OUTPUTS:             DELCDUX,Y,Z

```

```

# OMEGAPD,+1,+2
# DELPEROR,+1,+2
# CPHI,+1,+2 FOR NOUN22
#
# DEBRIS: FINDCDUW DESTROYS SINCDUX,Y,Z AND COSCDUX,Y,Z BY
# WRITING INTO THESE LOCATIONS THE SINES AND COSINES
# OF THE CDUD'S IN PNGCS-AUTO, OF THE CDU'S OTHERWISE.

```

```

# Page 910

```

```

# INITIALIZATION FOR FINDCDUW

```

```

BANK 30
SETLOC FCDUW
BANK
EBANK= ECDUW
COUNT* $$/FCDUW

```

```

INITCDUW VLOAD
UNITX
STORE UNFV/2
STORE UNWC/2
RVQ

```

```

# FINDCDUW PRELIMINARIES

```

```

FINDCDUW VLOAD # FINDCDUW -2: ENTRY WHEN UNFC/2 PRE-STORD
UNFC/2 # INPUT VECTORS NEED NOT BE SEMI-UNIT
BOV SETPD # FINDCDUW: ENTRY WHEN UNFC/2 IN MPAC
FINDCDUW # INTERPRETER NOW INITIALIZED
22 # LOCS 0 THRU 21 FOR DIRECTION COSINE MAT
STQ EXIT
QCDUWUSR # SAVE RETURN ADDRESS

```

```

# MORE HAUSKEEPING

```

```

CA ECDUWL
XCH EBANK # SET EBANK
TS ECDUWUSR # SAVE USER'S EBANK

```

```

CA DAPBOOLS
MASK CSMDOCKD # CSMDOCKD MUST NOT BE BIT15
CCS A
CA ONE # INDEX IF CSM DOCKED
TS NDXCDUW

```

```

CA XOVINHIB # XOVINHIB MUST NOT BE BIT15

```

July 12, 2016

549

```

                TS      FLPAUTNO      # SET TO POS-NON-ZERO FLAG PNGCS AUTO NOT

                MASK     DAPBOOLS
                TS      FLAGOODW      # FLAGOODW = ANY PNZ NUMBER IF XOY INHIBTD

# Page 911
# FETCH BASIC DATA
                INHINT      # RELINT AT PAUTNO (TC INTPRET)

                CA      CDUX      # FETCH CDUX,CDUY,CDUZ IN ALL CASES, BUT
                TS      CDUSPOTX  #      REPLACE BELOW IF PNGCS AUTO
                CA      CDUY
                TS      CDUSPOTY
                CA      CDUZ
                TS      CDUSPOTZ

                CA      BIT10      # PNGCS CONTROL BIT
                EXTEND
                RAND     CHAN30
                CCS      A
                TCF      PAUTNO      # NOT PNGCS (BITS INVERTED)

                CA      BIT14      # AUTO MODE BIT
                EXTEND
                RAND     CHAN31
                CCS      A
                TCF      PAUTNO      # NOT AUTO (BITS INVERTED)

                TS      FLPAUTNO      # RESET FLAG PNGCS AUTO NOT

                CA      CDUXD      # PNGCS AUTO:  FETCH CDUXD,CDUYD,CDUZD
                TS      CDUSPOTX
                CA      CDUYD
                TS      CDUSPOTY
                CA      CDUZD
                TS      CDUSPOTZ

# Page 912
# FETCH INPUTS
PAUTNO        TC      INTPRET      # ENTERING THRUST CMD STILL IN MPAC
                RTB
                NORMUNIT
                STOVL    UNX/2      # SEMI-UNIT THRUST CMD AS INITIAL UNX/2
                UNWC/2
                RTB      RTB
                NORMUNIT
```

```

                                QUICTRIG      # ALWAYS RQD TO OBTAIN TRIGS OF CDUD'S
STOVL      UNZ/2                # SEMI-UNIT WINDOW CMD AS INITIAL UNZ/2
                                DELV
BOVB      UNIT
                                NOATTCNT      # AT LEAST ONE ENTERING CMD VCT ZERO
BOV      CALL
                                AFTRFLTR      # IF UNIT DELV OVERFLOWS SKIP FILTER
                                *SMNB*        # YIELDS UNIT(DELV) IN VEH COORDS FOR FLTR

# THRUST DIRECTION FILTER

                                EXIT

                                CA      UNFVY/2      # FOR RESTARTS, UNFV/2 ALWAYS INTACT, MPAC
                                LXCH    MPAC   +3      #      RENEWD AFTER RETURN FROM CALLER,
                                TC      FLTRSUB      #      TWO FILTER UPDATES MAY BE DONE.
                                TS      UNFVY/2      # UNFV/2 NEED NOT BE EXACTLY SEMI-UNIT.

                                CA      UNFVZ/2
                                LXCH    MPAC +5
                                TC      FLTRSUB
                                TS      UNFVZ/2

                                TC      INTERPRET      # COMPLETES FILTER

# Page 913
# FIND A SUITABLE WINDOW POINTING VECTOR

AFTRFLTR      SLOAD      BHIZ      # IF XOY NOT INHIBITED, GO FETCH ZNB
                                FLAGOODW
                                FETCHZNB
                                VLOAD      CALL
                                UNZ/2
                                UNWCTEST

FETCHZNB      VLOAD
                                ZNBPIP
                                STCALL    UNZ/2
                                UNWCTEST

                                VLOAD      VCOMP      # Z AND -X CAN'T BOTH PARALLEL UNFC/2
                                XNBPIP
                                STORE      UNZ/2

# COMPUTE THE REQUIRED DIRECTION COSINE MATRIX

```

```

DCMCL      VLOAD  VXV
              UNZ/2
              UNX/2
UNIT      PUSH      # UNY/2 FIRST ITERATION
VXV      VSL1
              UNX/2
STORE    UNZ/2      # -UNZ/2 FIRST ITERATION
VXSC     PDVL      # EXCHANGE -UNFVZ/2 UNZ/2 FOR UNY/2
              UNFVZ/2 # MUST BE SMALL
VXSC     BVSU      # YIELDS -UNFVY/2 UNY/2-UNFVZ/2 UNZ/2
              UNFVY/2 # MUST BE SMALL
VSL1     VAD
              UNX/2
UNIT      # TOTALLY ELIMINATES THRUST POINTING ERROR
STORE    UNX/2      # UNX/2
VXV      VSL1
              UNZ/2      # -UNZ/2 WAS STORED HERE REMEMBER
STORE    UNY/2      # UNY/2
VCOMP    VXV
              UNX/2
VSL1
STORE    UNZ/2      # UNZ/2

# Page 914
# COMPUTES THE REQUIRED GIMBAL ANGLES

CALL
          NB2CDUSP    # YIELDS THE RQD GIMBAL ANGLES, 2'S, PI
EXIT

# LIMIT THE MIDDLE GIMBAL ANGLE & COMPUTE THE UNLIMITED GIMBAL ANGLE CHGS

CA      MPAC +2      # LIMIT THE MGA
TS      L            # CAN'T LXCH: NEED UNLIMITED MGA FOR ALARM
CA      CDUZDLIM
TC      LIMITSUB     # YIELDS LIMITED MGA. 1 BIT ERROR POSSIBLE
XCH     MPAC +2      # BECAUSE USING 2'S COMP. WHO CARES?
EXTEND
MSU     MPAC +2      # THIS BETTER YIELD ZERO
EXTEND
BZF     +2
TCF     ALARMGA

MGARET   INHINT      # RELINT AT TC INTPRET AFTER TCQCUDW

ZL

```

```

                                CA      TWO
DELGMBLP                      TS      TEM2

                                CA      L
                                EXTEND
                                SQUARE
                                AD      HI5
                                EXTEND
                                BZMF    +3
                                CA      ZERO
                                TS      FLAGOODW

                                INDEX   TEM2
                                CA      MPAC
                                INDEX   TEM2
                                TS      CPHI
                                EXTEND
                                INDEX   TEM2
                                MSU      CDUXD
                                COM
                                INDEX   TEM2
                                TS      -DELGMB
                                TS      L
                                CCS      TEM2
                                TCF      DELGMBLP

```

Page 915

BRANCHES TO NOATTCNT

```

                                CCS      FLPAUTNO
                                TCF      NOATTCNT +2
                                CA      FLAGWRD5
                                MASK     ENGONBIT
                                EXTEND
                                BZF      NOATTCNT +2

```

Page 916

LIMIT THE ATTITUDE ANGLE CHANGES

#

```

# THIS SECTION LIMITS THE ATTITUDE ANGLE CHANGES ABOUT A SET OF ORTHOGONAL VEHICLE AXES
# THESE AXES COINCIDE WITH THE COMMANDED VEHICLE AXES IF AND ONLY IF CDUXD IS ZERO.
# THE COMMANDED VEHICLE SYSTEM ROTATED ABOUT THE X AXIS TO BRING THE Z AXIS INTO ALIGNMENT
# AXIS. ATTITUDE ANGLE CHANGES IN THE PRIME SYSTEM ARE RELATED TO SMALL GIMBAL ANGLES
#
# [ -DELATTX ] [ 1 SIN(CDUZD) 0 ] [ -DELGBX ]
# [          ] [          ] [          ]

```


July 12, 2016

553

```

#      [ -DELATTYPRIME ] = [ 0      COS(CDUZD)      0 ] [ -DELGMBY ]
#      [                  ] [                  ] [                  ]
#      [ -DELATTZPRIME ] [ 0      0      1 ] [ -DELGMBZ ]

      LXCH      -DELGMB +2      # SAME AS -DELATTZPRIME UNLIMITED
      INDEX      NDXCDUW
      CA          DAZMAX
      TC          LIMITSUB
      TS          -DELGMB +2      # -DELGMBZ

      CA          -DELGMB +1
      EXTEND
      MP          COSCDUZ      # YIELDS -DELATTYPRIME/2 UNLIMITED
      TS          L
      INDEX      NDXCDUW
      CA          DAY/2MAX
      TC          LIMITSUB
      EXTEND
      DV          COSCDUZ
      XCH          -DELGMB +1      # -DELGMBY, FETCHING UNLIMITED VALUE

      EXTEND
      MP          SINCDUZ
      DDOUBL
      COM
      EXTEND      # YIELDS +DELATTX UNLIMITD, MAG < 180 DEG.
      MSU          -DELGMB      #      BASED ON UNLIMITED DELGMBV.
      TS          L      #      ONE BIT ERROR IF OPERANDS IN MSU
      INDEX      NDXCDUW      #      OF MIXED SIGNS.  WHO CARES?
      CA          DAXMAX
      TC          LIMITSUB
      TS          -DELGMB      # SAVE LIMITED +DELATTX
      CCS          FLAGOODW
      CS          -DELGMB      # FETCH IT BACK CHGING SIGN IF WINDOW GOOD
      TS          -DELGMB      # OTHERWISE USE ZERO FOR -DELATTX
      CS          -DELGMB +1
      EXTEND
      MP          SINCDUZ
      DDOUBL      # YIELDS -CNTRIB TO -DELATTX FROM -DELGMBY
      ADS          -DELGMB      # -DELGMBX.  NO OVERFLOW SINCE LIMITED TO
                                # 20DEG(1+SIN(70DEG)/COS(70DEG)) < 180DEG

# Page 917
# COMPUTE COMMANDED ATTITUDE RATES
#
#      [ OMEGAPD ] [ -2      -4 SINCDUZ      +0      ] [ -DELGMBZ ]

```

```

#      [      ] [      ] [      ] [      ]
#      [ OMEGAQD ] = [ +0      -8 COSCDUZ COSCDUX      -4 SINCDUX ] [ -DELGMBY ]
#      [      ] [      ] [      ] [      ]
#      [ OMEGARD ] [ +0      +8 COSCDUZ SINCDUX      -4 COSCDUX ] [ -DELGMBZ ]
#
# ATTITUDE ANGLE RATES IN UNITS OF PI/4 RAD/SEC = K TRIG FCNS IN UNITS OF 2 X GIMBAL
# PI/2 RAD/SEC.  THE CONSTANTS ARE BASED ON DELGMB BEING THE GIMBAL ANGLE CHANGES IN
# AND 2 SECONDS BEING THE COMPUTATION PERIOD (THE PERIOD BETWEEN SUCCESSIVE PASSES T

```

```

CS      -DELGMB
TS      OMEGAPD
CS      -DELGMB +1
EXTEND
MP      SINCDUZ
DDOUBL
ADS     OMEGAPD
ADS     OMEGAPD

CS      -DELGMB +1
EXTEND
MP      COSCDUX
DDOUBL
EXTEND
MP      COSCDUZ
TS      OMEGAQD
CS      -DELGMB +2
EXTEND
MP      SINCDUX
ADS     OMEGAQD
ADS     OMEGAQD
ADS     OMEGAQD

CA      -DELGMB +1
EXTEND
MP      SINCDUX
DDOUBL
EXTEND
MP      COSCDUZ
TS      OMEGARD
CS      -DELGMB +2
EXTEND
MP      COSCDUX
ADS     OMEGARD
ADS     OMEGARD
ADS     OMEGARD

```

July 12, 2016

555

Page 918

FINAL TRANSFER

```
CDUWXFR      CA      TWO
              TS      TEM2
              INDEX    TEM2
              CA      -DELGMB
              EXTEND
              MP      DT/DELT      # RATIO OF DAP INTERVAL TO CDUW INTERVAL
              TC      ONESTO2S
              INDEX    TEM2
              TS      DELCDUX      # ANGLE INTERFACE

              INDEX    TEM2
              CCS      OMEGAPD
              AD      ONE
              TCF      +2
              AD      ONE
              EXTEND      # WE NOW HAVE ABS(OMEGAPD,QD,RD)
              INDEX    TEM2
              MP      OMEGAPD
              EXTEND
              MP      BIT11      # 1/16
              EXTEND
              INDEX    TEM2      #
              DV      1JACC      # UNITS PI/4 RAD/SEC
              TS      L
              CA      DELERLIM
              TC      LIMITSUB
              INDEX    TEM2
              TS      DELPEROR      # LAG ANGLE = OMEGA ABS(OMEGA)/2 ACCEL
              CCS      TEM2
              TCF      CDUWXFR
```

HAUSKEEPING AND RETURN

```
TCQCUDW      CA      ECDUWUSR
              TS      EBANK      # RETURN USER'S EBANK

              TC      INTERPRET
              SETPD    GOTO
                      0
                      QCUDWUSR      # NORMAL AND ABNORMAL RETURN TO USER
```

Page 919

THRUST VECTOR FILTER SUBROUTINE

```

FLTRSUB      EXTEND
              QXCH    TEM2
              TS      TEM3          # SAVE ORIGINAL OFFSET
              COM      # ONE MCT, NO WDS, CAN BE SAVED IF NEG OF
              AD      L          # ORIG OFFSET ARRIVES IN A, BUT IT'S
              EXTEND    # NOT WORTH THE INCREASED OBSCURITY.
              INDEX    NDXCDUW
              MP      GAINFLTR
              TS      L          # INCR TO OFFSET, UNLIMITED
              CA      DUNFVLIM    # SAME LIMIT FOR Y AND Z
              TC      LIMITSUB    # YIELDS INCR TO OFFSET, LIMITED
              AD      TEM3        # ORIGINAL OFFSET
              TS      L          # TOTAL OFFSET, UNLIMITED
              CA      UNFVLIM     # SAME LIMIT FOR Y AND Z
              TC      LIMITSUB    # YIELDS TOTAL OFFSET, LIMITED
              TC      TEM2

```

SUBR TO TEST THE ANGLE BETWEEN THE PROPOSED WINDOW AND THRUST CMD VCTS

```

UNWCTEST      DOT      DSQ
                UNX/2
              DSU      BMN
                DOTSWFMX
                DCMCL
              SSP      RVQ          # RVQ FOR ALT CHOICE IF DOT MAGN TOO LARGE
                FLAGOODW          # ZEROING WINDOW GOOD FLAG
                0

```

Page 920

NB2CDUSP RETURNS THE 2'S COMPLEMENT, PI, SP CDU ANGLES X,Y,Z IN MPAC,+1,+2 GIVEN T
 # ARE THE SEMI-UNIT NAV BASE VECTORS X,Y,X EXPRESSED IN STABLE MEMBER COORDINATES, L
 # NB2CDUSP USES THE ARCTRGSP WHICH HAS A MAXIMUM ERROR OF +-4 BITS.

```

NB2CDUSP      DLOAD    DSQ
                2
              BDSU     BPL
                DP1/4TH
                +3
              DLOAD
                ZEROVECS          # IN CASE SIN WAS SLIGHTLY > 1/2
              SQRT     EXIT        # YIELDS COS(CDUZ) IN UNITS OF 2
              EXTEND
              DCA      MPAC
              DDOUBL

```

July 12, 2016

557

```
TS      TEM5
TCF     +3
CA      POSMAX      # OVERFLOW.  FETCH POSMAX, MPAC ALWAYS POS
TS      TEM5      # COS(CDUZ) IN TEM5, UNITS 1
```

```
INDEX   FIXLOC
CA       2
LXCH    MPAC
TC      ARCTRGSP
TS      MPAC +2      # CDUZ
```

```
CA      ZERO
TC      DVBYCOSM
CA      FOUR
TC      DVBYCOSM
CS      TEM1
TC      ARCTRGSP
TS      MPAC +1      # CDUY
```

```
CA      BIT4
TC      DVBYCOSM
CA      16OCT
TC      DVBYCOSM
CS      TEM1
TC      ARCTRGSP
TS      MPAC      # CDUX
```

```
TC      INTERPRET
RVQ
```

16OCT OCT 16

Page 921

```
# THE ELEMENTS OF THE NAV BASE MATRIX WHICH WE MUST DIVIDE BY COS(MGA)
# ALREADY CONTAIN COS(MGA)/2 AS A FACTOR. THEREFORE THE QUOTIENT SHOULD
# ORDINARILY NEVER EXCEED 1/2 IN MAGNITUDE. BUT IF THE MGA IS NEAR PI/2
# THEN COS(MGA) IS NEAR ZERO, AND THERE MAY BE SOME CHAFF IN THE OTHER
# ELEMENTS OF THE MATRIX WHICH WOULD PRODUCE CHAOS UNDER DIVISION.
# BEFORE DIVIDING WE MAKE SURE COS(MGA) IS AT LEAST ONE BIT LARGER
# THAN THE MAGNITUDE OF THE HIGH ORDER PART OF THE OPERAND.
#
# IF ONE OR MORE DIVIDES CANNOT BE PERFORMED, THIS MEANS THAT THE
# REQUIRED MGA IS VERY NEARLY +-PI/2 AND THEREFORE THE OTHER GIMBAL
# ANGLES ARE INDETERMINATE. THE INNER AND OUTER GIMBAL ANGLES RETURNED
# IN THIS CASE WILL BE RANDOM MULTIPLES OF PI/2.
```

DVBYCOSM	AD	FIXLOC	
	TS	ADDRWD	# ADRES OF OPERAND
	INDEX	ADDRWD	# FETCH NEG ABS OF OPERAND, AD TEM5, AND
	CA	0	# SKIP DIVIDE IF RESULT NEG OR ZERO
	EXTEND		
	BZMF	+2	
	COM		
	AD	TEM5	# C(A) ZERO OR NEG, C(TEM5) ZERO OR POS
	EXTEND		
	BZMF	TSL&TCQ	# DIFFERENCE ALWAYS SMALL IF BRANCH
	EXTEND		# TEM5 EXCEEDS ABS HIGH ORDER PART OF
	INDEX	ADDRWD	# OPERAND BY AT LEAST ONE BIT.
	DCA	0	# THEREFORE IT EXCEEDS THE DP OPERAND
	EXTEND		# AND DIVISION WILL ALWAYS SUCCEED.
TSL&TCQ	DV	TEM5	
	TS	L	
	LXCH	TEM1	
	TC	Q	

Page 922

ARCTRGSP RETURNS THE 2'S COMPLEMENT, PI, SP ANGLE IN THE A REGISTER GIVEN ITS SINE
 # UNITS OF 2. THE RESULT IS AN UNAMBIGUOUS ANGLE ANYWHERE IN THE CIRCLE, WITH A MAX
 # THE ERROR IS PRODUCED BY THE SUBROUTINE SPARCSIN WHICH IS USED ONLY IN THE REGION -

ARCTRGSP	EXTEND		
	BZF	SINZERO	# TO AVOID DIVIDING BY ZERO
	EXTEND		
	QXCH	TEM4	
	TS	TEM2	
	CA	L	
	TS	TEM3	
	CA	ZERO	
	EXTEND		
	DV	TEM2	
	EXTEND		
	BZF	USECOS	
	CCS	TEM3	# SIN IS SMALLER OR EQUAL
	CA	ZERO	
	TCF	+4	
	CS	TEM2	# IF COS NEG, REVERSE SIGN OF SIN,
	TS	TEM2	# ANGLE = PI-ARCSIN(SIN)
	CA	NEGMAX	# PICK UP PI, 2'S COMPLEMENT

```

          TS      TEM3          # WE NO LONGER NEED COS
          CA      TEM2
          TC      SPARCSIN -1
          TC      ONESTO2S
          EXTEND
          MSU      TEM3
1T02&TCQ  TC      ONESTO2S
          TC      TEM4

USECOS    CS      TEM3          # COS IS SMALLER
          TC      SPARCSIN -1    # ANGLE = SIGN(SIN)(FI/2-ARCSIN(COS))
          AD      HALF
          TS      TEM3          # WE NO LONGER NEED COS
          CCS      TEM2
          CA      TEM3
          TCF      1T02&TCQ
          CS      TEM3
          TCF      1T02&TCQ

SINZERO   CCS      L
          CA      ZERO
          TC      Q
          CA      NEGMAX        # PI, 2'S COMP
          TC      Q

```

Page 923

SPARCSIN TAKES AN ARGUMENT SCALED UNITY IN A AND RETURNS AN ANGLE SCALED
 # 180 DEGREES IN A. IT HAS BEEN UNIT TESTED IN THE REGION $\pm .94$ (± 70
 # DEGREES) AND THE MAXIMUM ERROR IS ± 5 BITS WITH AN AVERAGE TIME OF
 # 450 MICROSECONDS. SPARCSIN -1 TAKES THE ARGUMENT SCALED TWO. (BOB CRISP)

```

          DOUBLE
SPARCSIN  TS      SR
          TCF      +4
          INDEX    A
          CS      LIMITS
          TS      SR
          EXTEND
          MP      A
          TS      TEM1
          EXTEND
          MP      DPL9
          AD      DPL7
          EXTEND
          MP      TEM1
          AD      DPL5

```

```

                                EXTEND
                                MP      TEM1
                                AD      DPL3
                                EXTEND
                                MP      TEM1
                                AD      DPL1
                                EXTEND
                                MP      SR
                                TC      Q
DPL1      DEC      10502
DPL3      DEC      432
DPL5      DEC      7300
DPL7      DEC      -11803
DPL9      DEC      8397

```

Page 924

LIMITSUB LIMITS THE MAGNITUDE OF THE POSITIVE OR NEGATIVE VARIABLE

ARRIVING IN L TO THE POSITIVE LIMIT ARRIVING IN A.

THE SIGNED LIMITED VARIABLE IS RETURNED IN A.

#

VERSION COUTESY HUGH BLAIR-SMITH

```

LIMITSUB      TS      TEM1
              CA      ZERO
              EXTEND
              DV      TEM1
              CCS      A
              LXCH     TEM1
              TCF      +2
              TCF      +3
              CA      L
              TC      Q
              CS      TEM1
              TC      Q

```

SUBROUTINE TO CONVERT 1'S COMP SP TO 2'S COMP

```

ONESTO2S      CCS      A
              AD      ONE
              TC      Q
              CS      A
              TC      Q

```

NO ATTITUDE CONTROL

```

NOATTCNT      TC      ALARM

```


July 12, 2016

561

```
OCT      00402      # NO ATTITUDE CONTROL

+2      INHINT      # COME HERE FOR NOATTCNT WITHOUT ALARM
        TC      IBNKCALL      # RELINT AT TC INTPRET AFTER TCQCDUW
        FCADR      STOPRATE
        TCF      TCQCDUW      # RETURN TO USER SKIPPING AUTOPILOT CMDS

# MIDDLE GIMBAL ANGLE ALARM

ALARMGA      TC      ALARM
            OCT      00401
            TCF      MGARET

# Page 925
*****
# CONSTANTS
*****

# ADDRESS CONSTANTS

ECDUWL      ECADR      ECDUW

# THRUST DIRECTION FILTER CONSTANTS

GAINFLTR      DEC      .2      # GAIN FILTER SANS CSM
            DEC      .1      # GAIN FILTER WITH CSM

DUNFVLIM      DEC      .007 B-1      # 7 MR MAX CHG IN F DIR IN VEH IN 2 SECS.
            # THIS DOES NOT ALLOW FOR S/C ROT RATE.

UNFVLIM      DEC      .129 B-1      # 129 MR MAX THRUST OFFSET. 105 MR TRAVEL
            # +10MR DEFL+5MR MECH MOUNT+9MR ABLATION.

# CONSTANT RELATED TO GIMBAL ANGLE COMPUTATIONS

DOTSWFMX      DEC      .93302 B-4      # LIM COLNRTY OF UNWC/2 & UNFC/2 TO 85 DEG
            # LOWER PART COMES FROM NEXT CONSTANT

DAXMAX      DEC      .1111111111      # DELATTX LIM TO 20 DEG IN 2 SECS, 1'S, PI
            DEC      .0111111111      # 2 DEG WHEN CSM DOCKED

DAY/2MAX      DEC      .0555555555      # LIKEWISE FOR DELATTY
            DEC      .0055555555

DAZMAX      =      DAXMAX      # LIKEWISE FOR DELATTZ
```

```
CDUZDLIM      DEC      .3888888888      # 70 DEG LIMIT FOR MGA, 1'S, PI

# CONSTANTS FOR DATA TRANSFER

DT/DELT      DEC      .05      # .1 SEC/2 SEC WHICH IS THE AUTOPILOT
                                # CONTROL SAMPLE PERIOD/COMPUTATION PERIOD

DELERLIM      =      DAY/2MAX      #      0 DEG LIMIT FOR LAG ANGLES, 1'S, PI

# *** END OF FLY      .132 ***
This code is written to file src/FINDCDUW--GUIDAP-INTERFACE.s.
```

B.32 FIXED FIXED CONSTANT POOL

[illegible]

BIT TABLE

BIT15	OCT	40000
BIT14	OCT	20000
BIT13	OCT	10000
BIT12	OCT	04000
BIT11	OCT	02000
BIT10	OCT	01000
BIT9	OCT	00400
BIT8	OCT	00200
BIT7	OCT	00100
BIT6	OCT	00040
BIT5	OCT	00020
BIT4	OCT	00010
BIT3	OCT	00004
BIT2	OCT	00002
BIT1	OCT	00001

DO NOT DESTROY THIS COMBINATION, SINCE IT IS USED IN DOUBLE PRECISION INSTRUCTIONS

NEGO	OCT	-0	# MUST PRECEDE ZERO
ZERO	OCT	0	# MUST FOLLOW NEGO
# BIT1	OCT	00001	
# NO.WDS	OCT	2	# INTERPRETER
# OCTAL3	OCT	3	# INTERPRETER
# R3D1	OCT	4	# PINBALL
FIVE	OCT	5	
# REVCNT	OCT	6	# INTERPRETER
SEVEN	OCT	7	
# BIT4	OCT	00010	
# R2D1	OCT	11	# PINBALL
OCT11	=	R2D1	# P20S
# BINCON	DEC	10	# PINBALL (OCTAL 12)
ELEVEN	DEC	11	
# OCT14	OCT	14	# ALARM AND ABORT (FILLER)
OCT15	OCT	15	
# R1D1	OCT	16	# PINBALL
# Page 1201			
LOW4	OCT	17	
# BIT5	OCT	00020	
# ND1	OCT	21	# PINBALL
# VD1	OCT	23	# PINBALL
# OCT24	OCT	24	# SERVICE ROUTINES
# MD1	OCT	25	# PINBALL
BITS4&5	OCT	30	
# OCT31	OCT	31	# SERVICE ROUTINES
CALLCODE	OCT	00032	

July 12, 2016

565

# LOW5	OCT	37	# PINBALL
# 33DEC	DEC	33	# PINBALL (OCTAL 41)
# 34DEC	DEC	34	# PINBALL (OCTAL 42)
TBUILDFX	DEC	37	# BUILDUP FOR CONVENIENCE IN DAPTESTING
TDECAYFX	DEC	38	# CONVENIENCE FOR DAPTESTING
# BIT6	OCT	00040	
OCT50	OCT	50	
DEC45	DEC	45	
SUPER011	OCT	60	# BITS FOR SUPERBNK SETTING 011.
.5SEC	DEC	50	
# BIT7	OCT	00100	
SUPER100	=	BIT7	# BITS FOR SUPERBNK SETTING 100
			# (LAST 4K OF ROPE)
SUPER101	OCT	120	# BITS FOR SUPERBNK SETTING 101
# OCT121	OCT	121	# SERVICE ROUTINES
			# (FIRST 8K OF ACM)
SUPER110	OCT	140	# BITS FOR SUPERBNK SETTING 110.
			# (LAST BK OF ACM)
1SEC	DEC	100	
# LOW7	OCT	177	# INTERPRETER
# BIT8	OCT	00200	
# OT215	OCT	215	# ALARM AND ABORT
# 8,5	OCT	00220	# P20-P25 SUNDANCE
2SECS	DEC	200	
# LOW8	OCT	377	# PINBALL
# BIT9	OCT	00400	
GN/CCODE	OCT	00401	# SET S/C CONTROL SWITCH TO G/N
3SECS	DEC	300	
4SECS	DEC	400	
LOW9	OCT	777	
# BIT10	OCT	01000	
# 5.5DEGS	DEC	.03056	# P20-P25 SUNDANCE (OCTAL 00765)
# OCT1103	OCT	1103	# ALARM AND ABORT
C5/2	DEC	.0363551	# (OCTAL 01124)
V05N09	VN	0509	# (SAME AS OCTAL 1211)
OCT1400	OCT	01400	
V06N22	VN	0622	
# MID5	OCT	1740	# PINBALL
BITS2-10	OCT	1776	
LOW10	OCT	1777	
# Page 1202			
# BIT11	OCT	02000	
# 2K+3	OCT	2003	# PINBALL
LOW7+2K	OCT	2177	# OP CODE MASK + BANK 1 FBANK SETTING
EBANK5	OCT	02400	

PRI03	OCT	03000	
EBANK7	OCT	03400	
# LOW11	OCT	3777	# PINBALL
# BIT12	OCT	04000	
# RELTAB	OCT	04025	# T4RUPT
PRI05	OCT	05000	
PRI06	OCT	06000	
PRI07	OCT	07000	
# BIT13	OCT	10000	
#	OCT	10003	# T4RUPT RELTAB +1D
# 13,7,2	OCT	10102	# P20-P25 SUNDANCE
PRI011	OCT	11000	
# PRI012	OCT	12000	# BANKCALL
PRI013	OCT	13000	
PRI014	OCT	14000	
#	OCT	14031	# T4RUPT RELTAB +2D
PRI015	OCT	15000	
PRI016	OCT	16000	
# 85DEGS	DEC	.45556	# P20-P25 SUNDANCE (OCTAL 16450)
PRI017	OCT	17000	
OCT17770	OCT	17770	
# BIT14	OCT	20000	
#	OCT	20033	# T4RUPT RELTAB +3D
PRI021	OCT	21000	
	BLOCK	03	
	COUNT	03/FCONS	
PRI022	OCT	22000	# SERVICE ROUTINES
PRI023	OCT	23000	
PRI024	OCT	24000	
# 5/8+1	OCT	24001	# SINGLE PRECISION SUBROUTINES
#	OCT	24017	# T4RUPT RELTAB +4D
PRI025	OCT	25000	
PRI026	OCT	26000	
PRI027	OCT	27000	
# CHRPRIO	OCT	30000	# PINBALL
#	OCT	30036	# T4RUPT RELTAB +5D
PRI031	OCT	31000	
C1/2	DEC	.7853134	# (OCTAL 31103)
PRI032	OCT	32000	
PRI033	OCT	33000	
PRI034	OCT	34000	
#	OCT	34034	# T4RUPT RELTAB +6D
PRI035	OCT	35000	
PRI036	OCT	36000	

July 12, 2016

567

Page 1203

PRI037	OCT	37000		
63/64+1	OCT	37401		
# MID7	OCT	37600	# PINBALL	
OCT37766	OCT	37766		
OCT37774	OCT	37774		
OCT37776	OCT	37776		
# DPOSMAX	OCT	37777		
# BIT15	OCT	40000		
# OCT40001	OCT	40001	# INTERPRETER (CS 1 INSTRUCTION)	
DLOADCOD	OCT	40014		
DLOAD*	OCT	40015		
#	OCT	40023	# T4RUPT	RELTAB +7D
BIT15+6	OCT	40040		
OCT40200	OCT	40200		
#	OCT	44035	# T4RUPT	RELTAB +8D
#	OCT	50037	# T4RUPT	RELTAB +9D
#	OCT	54000	# T4RUPT	RELTAB +10D
-BIT14	OCT	57777		
# RELTAB11	OCT	60000	# T4RUPT	
C3/2	DEC	-.3216147	#	(OCTAL 65552)
13,14,15	OCT	70000		
-1/8	OCT	73777		
HIGH4	OCT	74000		
-ENDERAS	DEC	-2001	#	(OCTAL 74056)
# HI5	OCT	76000	# PINBALL	
HIGH9	OCT	77700		
# -ENDVAC	DEC	-45	# INTERPRETER	(OCTAL 77722)
# -OCT10	OCT	-10	#	(OCTAL 77767)
# NEG4	DEC	-4	#	(OCTAL 77773)
NEG3	DEC	-3		
NEG2	OCT	77775		
NEGONE	DEC	-1		

Page 1204

DEFINED BY EQUALS

IT WOULD BE TO THE USERS ADVANTAGE TO OCCASIONALLY CHECK ANY OF THESE SYMBOLS IN ORDER TO PRE
ACCIDENTAL DEFINITION CHANGES.

MINUS1	=	NEG1
NEG1	=	NEGONE
ONE	=	BIT1
TWO	=	BIT2
THREE	=	OCTAL3

LOW2	=	THREE	
FOUR	=	BIT3	
SIX	=	REVCNT	
LOW3	=	SEVEN	
EIGHT	=	BIT4	
NINE	=	R2D1	
TEN	=	BINCON	
NOUTCON	=	ELEVEN	
OCT23	=	VD1	
OCT25	=	MD1	
PRI01	=	BIT10	
EBANK3	=	OCT1400	
PRI02	=	BIT11	
OCT120	=	SUPER101	
OCT140	=	SUPER110	
2K	=	BIT11	
EBANK4	=	BIT11	
PRI04	=	BIT12	
EBANK6	=	PRI03	
QUARTER	=	BIT13	
PRI010	=	BIT13	
OCT10001	=	CCSL	
POS1/2	=	HALF	
PRI020	=	BIT14	
HALF	=	BIT14	
PRI030	=	CHRPRI0	
BIT13-14	=	PRI030	# INTERPRETER USES IN PROCESSING STORECODE
OCT30002	=	TLOAD +1	
B12T14	=	PRI034	
NEGMAX	=	BIT15	
VLOADCOD	=	BIT15	
VLOAD*	=	OCT40001	
OCT60000	=	RELTAB11	
BANKMASK	=	HI5	

This code is written to file `src/FIXED-FIXED-CONSTANT-POOL.s`.

B.33 FLAGWORD ASSIGNMENTS

569

<src/FLAGWORD-ASSIGNMENTS.s 569>≡

```
# Copyright:    Public domain.
# Filename:     FLAGWORD_ASSIGNMENTS.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#              It is part of the source code for the Lunar Module's (LM)
#              Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:   yaYUL
# Contact:      Onno Hommes <ohommes@cmu.edu>.
# Website:      www.ibiblio.org/apollo.
# Pages:        0061-0089
# Mod history:  2009-05-15 OH   Transcribed from page images.
#              2009-05-17 RSB   Extended to (blank) p. 89.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum. The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum. Many thanks to both. The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo. If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
```

Page 61

```
# FLAGWORDS 0-11      ARE DOWNLINKED AND CAN BE SET AND CLEARED BY UP-FLAG AND DOWN-FLAG INST
#                      INTERPRETER. THESE WERE PREVIOUSLY LISTED UNDER "INTERPRETIVE SWITCH E
#                      THE ERASABLE LOG SECTION. FLAGWORDS 12 & 13 WERE PREVIOUSLY RADMODES A
#                      ARE STILL DOWNLINKED UNDER THOSE NAMES.
```

ALPHABETICAL LIST OF FLAGWORDS

# FLAGWORD	DEC. NUMBER	BIT AND FLAG	BIT NAME
# ACCOKFLG	207	BIT 3 FLAG 13	ACCSOKAY
# ACC4-2FL	199	BIT 11 FLAG 13	ACC4OR2X
# ACMODFLG	032	BIT 13 FLAG 2	ACMODBIT
# ALTSCALE	186	BIT 9 FLAG 12	ALTSCBIT

# ANTENFLG	183	BIT 12 FLAG 12	ANTENBIT
# AORBSFLG	205	BIT 5 FLAG 13	AORBSYST
# AORBTFLG	200	BIT 10 FLAG 13	AORBTRAN
# APSESW	130	BIT 5 FLAG 8	APSESBIT
# APSFLAG	152	BIT 13 FLAG 10	APSFLBIT
# ASTNFLAG	108	BIT 12 FLAG 7	ASTNBIT
# ATTFLAG	104	BIT 1 FLAG 6	ATTFLBIT
# AUTOMODE	193	BIT 2 FLAG 12	AUTOMBIT
# AUTR1FLG	209	BIT 1 FLAG 13	AUTRATE1
# AUTR2FLG	208	BIT 2 FLAG 13	AUTRATE2
# AUXFLAG	103	BIT 2 FLAG 6	AUXFLBIT
# AVEGFLAG	115	BIT 5 FLAG 7	AVEGFBIT
# AVEMIDSW	149	BIT 1 FLAG 9	AVEMDBIT
# AVFLAG	040	BIT 5 FLAG 2	AVFLBIT
# CALCMAN2	043	BIT 2 FLAG 2	CALC2BIT
# CALCMAN3	042	BIT 3 FLAG 2	CALC3BIT
# CDESFLAG	180	BIT 15 FLAG 12	CDESBIT
# CMOONFLG	123	BIT 12 FLAG 8	CMOONBIT
# COGAFLAG	131	BIT 4 FLAG 8	COGAFBIT
# CSMDKFLG	197	BIT 13 FLAG 13	CSMDOCKD
# CULTFLAG	053	BIT 7 FLAG 3	CULTBIT
# DAPBOOLS		FLGWRD13	
# DBSELFLG	206	BIT 4 FLAG 13	DBSELECT
# DESIGFLG	185	BIT 10 FLAG 12	DESIGBIT
# DIDFLAG	016	BIT 14 FLAG	DIDFLBIT
# DIMOFLAG	059	BIT 1 FLAG 3	DIMOBIT
# DMENFLG	081	BIT 9 FLAG 5	DMENFBIT
# DRIFTDFL	202	BIT 8 FLAG 13	DRIFTBIT
# DRIFTFLG	030	BIT 15 FLAG 2	DRFTBIT
# DSKYFLAG	075	BIT 15 FLAG 5	DSKYFBIT
# Page 62			
# D6OR9FLG	058	BIT 2 FLAG 3	D6OR9BIT
# ENGONFLG	083	BIT 7 FLAG 5	ENGONBIT
# ERADFLAG	017	BIT 13 FLAG 1	ERADFBIT
# ETPIFLAG	038	BIT 7 FLAG 2	ETPIBIT
# FINALFLG	039	BIT 6 FLAG 2	FINALBIT
# FLAGWRD0	(000-014)	(STATE +0)	
# FLAGWRD1	(015-029)	(STATE +1)	
# FLAGWRD2	(030-044)	(STATE +2)	
# FLAGWRD3	(045-059)	(STATE +3)	
# FLAGWRD4	(060-074)	(STATE +4)	
# FLAGWRD5	(075-089)	(STATE +5)	
# FLAGWRD6	(090-104)	(STATE +6)	
# FLAGWRD7	(105-119)	(STATE +7)	
# FLAGWRD8	(120-134)	(STATE +8D)	
# FLAGWRD9	(135-149)	(STATE +9D)	

EQUIVALENT FI

# FLAP	142	BIT 8 FLAG 9	FLAPBIT
# FLGWRD10	(150-164)	(STATE +10D)	
# FLGWRD11	(165-179)	(STATE +11D)	
# FLGWRD12	(180-194)	(STATE +12D)	
# FLGWRD13	(195-209)	(STATE +13D)	
# FLPC	138	BIT 12 FLAG 9	FLPCBIT
# FLPI	139	BIT 11 FLAG 9	FLPIBIT
# FLRCS	149	BIT 10 FLAG 9	FLRCSBIT
# FLUNDISP	125	BIT 10 FLAG 8	FLUNDBIT
# FLVR	136	BIT 14 FLAG 9	FLVRBIT
# FREEFLAG	012	BIT 3 FLAG 0	FREEFBIT
# FSPASFLG	005	BIT 10 FLAG 0	FSPASBIT
# GLOKFAIL	046	BIT 14 FLAG 3	GLOKFBIT
# GMBDRVSW	095	BIT 10 FLAG 6	GMBDRBIT
# GUESSW	028	BIT 2 FLAG 1	GUESSBIT
# HFLSHFLG	179	BIT 1 FLAG 11	HFLSHBIT
# IDLEFLAG	113	BIT 7 FLAG 7	IDLEFBIT
# IGNFLAG	107	BIT 13 FLAG 7	IGNFLBIT
# IMPULSW	036	BIT 9 FLAG 2	IMPULBIT
# IMUSE	007	BIT 8 FLAG 0	IMUSEBIT
# INFINFLG	128	BIT 7 FLAG 8	INFINBIT
# INITALGN	133	BIT 2 FLAG 8	INITABIT
# INTFLAG	151	BIT 14 FLAG 10	INTFLBIT
# INTYPFLG	056	BIT 4 FLAG 3	INTYPBIT
# ITSWICH	105	BIT 15 FLAG 7	ITSWBIT
# JSWITCH	001	BIT 14 FLAG 0	JSWCHBIT
# LETABORT	141	BIT 9 FLAG 9	LETABBIT
# LMOONFLG	124	BIT 11 FLAG 8	LMOONBIT
# LOKONSW	010	BIT 5 FLAG 0	LOKONBIT
# LOSCMFLG	033	BIT 12 FLAG 2	LOSCMBIT
# LRALTFLG	190	BIT 5 FLAG 12	LRALTBIT
# LRBYPASS	165	BIT 15 FLAG 11	LRBYBIT
# LRINH	172	BIT 8 FLAG 11	LRINHBIT
# LRPOSFLG	189	BIT 6 FLAG 12	LRPOSBIT
# LRVELFLG	187	BIT 8 FLAG 12	LRVELBIT
# Page63			
# LUNAFLAG	048	BIT 12 FLAG 3	LUNABIT
# MANUFLAG	106	BIT 14 FLAG 7	MANUFBIT
# MGLVFLAG	088	BIT 2 FLAG 5	MGLVFBIT
# MIDAVFLG	148	BIT 2 FLAG 9	MIDAVBIT
# MIDFLAG	002	BIT 13 FLAG 0	MIDFLBIT
# MID1FLAG	147	BIT 3 FLAG 9	MID1BIT
# MKOVFLAG	072	BIT 3 FLAG 4	MKOVBIT
# MOONFLAG	003	BIT 12 FLAG 0	MOONBIT
# MRKIDFLG	060	BIT 15 FLAG 4	MRKIDBIT
# MRKNVFLG	066	BIT 9 FLAG 4	MRKNVBIT

# MRUPTFLG	070	BIT 5 FLAG 4	MRUPTBIT
# MUNFLAG	097	BIT 8 FLAG 6	MUNFLBIT
# MWAITFLG	064	BIT 11 FLAG 4	MWAITBIT
# NEEDLFLG	011	BIT 4 FLAG 0	NEEDLBIT
# NEWIFLG	122	BIT 13 FLAG 8	NEWIBIT
# NJETSFLG	015	BIT 15 FLAG	NJETSBIT
# NODOFLAG	044	BIT 1 FLAG 2	NODOBIT
# NOLRRREAD	170	BIT 10 FLAG 11	NOLRRBIT
# NORMSW	110	BIT 10 FLAG 7	NORMSBIT
# NORRMON	086	BIT 4 FLAG 5	NORRMBIT
# NOR29FLG	049	BIT 11 FLAG 3	NR29FBIT
# NOTHROTL	078	BIT 12 FLAG 5	NOTHRBIT
# NOUPFLAG	024	BIT 6 FLAG 1	NOUPFBIT
# NRMNVFLG	067	BIT 8 FLAG 4	NRMNVBIT
# NRMIDFLG	062	BIT 13 FLAG 4	NRMIDBIT
# NRUPTFLG	071	BIT 4 FLAG 4	NRUPTBIT
# NTARGFLG	102	BIT 3 FLAG 6	NTARGBIT
# NWAITFLG	065	BIT 10 FLAG 4	NWAITBIT
# OLDESFLG	014	BIT 1 FLAG 0	OLDESBIT
# OPTNSW	038	BIT 7 FLAG 2	OPTNBIT
# ORBWFLAG	054	BIT 6 FLAG 3	ORBWFBIT
# ORDERSW	129	BIT 6 FLAG 8	ORDERBIT
# OURRCFLG	198	BIT 12 FLAG 13	OURRCBIT
# PDSPFLAG	063	BIT 12 FLAG 4	PDSPFBIT
# PFRATFLG	041	BIT 4 FLAG 2	PFRATBIT
# PINBRFLG	069	BIT 6 FLAG 4	PINBRBIT
# PRECIFLG	052	BIT 8 FLAG 3	PRECIBIT
# PRIODFLG	061	BIT 14 FLAG 1	PRIODBIT
# PRONVFLG	068	BIT 7 FLAG 4	PRONVBIT
# PSTHIGAT	169	BIT 11 FLAG 11	PSTHIBIT
# PULSEFLG	195	BIT 15 FLAG 13	PULSES
# P21FLAG	004	BIT 11 FLAG 0	P21FLBIT
# P25FLAG	006	BIT 9 FLAG 0	P25FLBIT
# P39/79SW	126	BIT 9 FLAG 8	P39SWBIT
# QUITFLAG	145	BIT 5 FLAG 9	QUITBIT
# RADMODES		FLGWDR12	
# RASFLAG		FLGWDR10	
# RCDUFALL	188	BIT 7 FLAG 12	RCDUFBIT
# RCDUOFLG	182	BIT 13 FLAG 12	RCDUOBIT
# READLR	174	BIT 6 FLAG 11	READLBIT
# Page 64			
# READRFLG	051	BIT 9 FLAG 3	READRBIT
# READVEL	175	BIT 5 FLAG 11	READVBIT
# REDFLAG	099	BIT 6 FLAG 6	REDFLBIT
# REFSMFLG	047	BIT 13 FLAG 3	REFSMBIT
# REINTFLG	158	BIT 7 FLAG 10	REINTBIT

EQUIVALENT FI

EQUIVALENT FI

# REMODFLG	181	BIT 14 FLAG 12	REMODBIT
# RENDWFLG	089	BIT 1 FLAG 5	RENDWBIT
# REPOS MON	184	BIT 11 FLAG 12	REPOSBIT
# RHCSCFLG	203	BIT 7 FLAG 13	RHCSCALE
# RNDVZFLG	008	BIT 7 FLAG 0	RNDVZBIT
# RNGEDATA	176	BIT 4 FLAG 11	RNGEDBIT
# RNGSCFLG	080	BIT 10 FLAG 5	RNGSCBIT
# RODFLAG	018	BIT 12 FLAG 1	RODFLBIT
# ROTFLAG	144	BIT 6 FLAG 9	ROTFLBIT
# RPQFLAG	120	BIT 15 FLAG 8	RPQFLBIT
# RRDATAFL	191	BIT 4 FLAG 12	RRDATA BT
# RRNBSW	009	BIT 6 FLAG 0	RRNBBIT
# RRRSFLAG	192	BIT 3 FLAG 12	RRRSBIT
# RVSW	111	BIT 9 FLAG 7	RVSWBIT
# R04FLAG	051	BIT 9 FLAG 3	R04FLBIT
# R10FLAG	013	BIT 2 FLAG 0	R10FLBIT
# R61FLAG	020	BIT 10 FLAG 1	R61FLBIT
# R77FLAG	079	BIT 11 FLAG 5	R77FLBIT
# SCALBAD	177	BIT 3 FLAG 11	SCABBIT
# SLOPESW	027	BIT 3 FLAG 1	SLOPEBIT
# SNUFFER	077	BIT 13 FLAG 5	SNUFFBIT
# SOLNSW	087	BIT 3 FLAG 5	SOLNSBIT
# SRCHOPTN	031	BIT 14 FLAG 2	SRCHOBIT
# STATEFLG	055	BIT 5 FLAG 3	STATEBIT
# STEERSW	034	BIT 11 FLAG 2	STEERBIT
# SURFFLAG	127	BIT 8 FLAG 8	SURFFBIT
# SWANDISP	109	BIT 11 FLAG 7	SWANDBIT
# S32.1F1	090	BIT 15 FLAG 6	S32BIT1
# S32.1F2	091	BIT 14 FLAG 6	S32BIT2
# S32.1F3A	092	BIT 13 FLAG 6	S32BIT3A
# S32.1F3B	093	BIT 12 FLAG 6	S32BIT3B
# TFFSW	119	BIT 1 FLAG 7	TFFSWBIT
# TRACKFLG	025	BIT 5 FLAG 1	TRACKBIT
# TURNONFL	194	BIT 1 FLAG 12	TURNONBT
# ULLAGFLG	204	BIT 6 FLAG 13	ULLAGER
# UPDATFLG	023	BIT 7 FLAG 1	UPDATBIT
# UPLOCKFL	116	BIT 4 FLAG 7	UPLOCBIT
# USEQRFLG	196	BIT 14 FLAG 13	USEQRJTS
# VEHUPFLG	022	BIT 8 FLAG 1	VEHUPBIT
# VELDATA	173	BIT 7 FLAG 11	VELDABIT
# VERIFLAG	117	BIT 3 FLAG 7	VERIFBIT
# VFLAG	050	BIT 10 FLAG 3	VFLAGBIT
# VFLSHFLG	178	BIT 2 FLAG 11	VFLSHBIT
# VINTFLAG	057	BIT 3 FLAG 3	VINTFBIT
# VXINH	168	BIT 12 FLAG 11	VXINHBIT
# Page 65			

EQUIVALENT FLAG NAME:

# V37FLAG	114	BIT 6 FLAG 7	V37FLBIT
# V67FLAG	112	BIT 8 FLAG 7	V67FLBIT
# V82EMFLG	118	BIT 2 FLAG 7	V82EMBIT
# XDELVFLG	037	BIT 8 FLAG 2	XDELVBIT
# XDSPFLAG	074	BIT 1 FLAG 4	XDSPBIT
# XORFLG	171	BIT 9 FLAG 11	XORFLBIT
# XOVINFLG	201	BIT 9 FLAG 13	XOVINHIB
# 3AXISFLG	084	BIT 6 FLAG 5	3AXISBIT
# 360SW	134	BIT 1 FLAG 8	360SWBIT

ASSIGNMENT AND DESCRIPTION OF FLAGWORDS

FLAGWRDO = STATE +0 # (000-014)

(SET) (RES)

BIT 15 FLAG 0 (S)

= 000D
= BIT15

BIT 14 FLAG 0 (S)

JSWITCH = 001D
JSWCHBIT = BIT14# INTEGRATION OF W INTE
MATRIX VECTO

BIT 13 FLAG 0 (S)

MIDFLAG = 002D
MIDFLBIT = BIT13# INTEGRATION WITH INTE
SECONDARY BODY AND SOLAR
SOLAR PERTURBATIONS

BIT 12 FLAG 0 (L)

MOONFLAG = 003D
MOONBIT = BIT12# MOON IS SPHERE OF EART
INFLUENCE INFLU

BIT 11 FLAG 0

P21FLAG = 004D
P21FLBIT = BIT11# USE BASE VECTORS 1ST P
ALREADY CALCULATED ULATI

BIT 10 FLAG 0

FSPASFLG = 005D
FSPASBIT = BIT10# FIRST PASS THROUGH NOT P
REPOSITION ROUTINE REPOS

Page 66

BIT 9 FLAG 0 (S)

P25FLAG = 006D
P25FLBIT = BIT9

P25 OPERATING P25 M

July 12, 2016

575

# BIT 8 FLAG 0 (S)				
IMUSE	=	007D	#	IMU IN USE
IMUSEBIT	=	BIT8		IMU NOT IN USE
# BIT 7 FLAG 0 (S)				
RNDVZFLG	=	008D	#	P20 RUNNING (RADAR
RNDVZBIT	=	BIT7	#	IN USE)
# BIT 6 FLAG 0 (S)				
RRNBSW	=	009D	#	RADAR TARGET IN
RRNBBIT	=	BIT6	#	NB COORDINATES
# BIT 5 FLAG 0 (S)				
LOKONSW	=	010D	#	RADAR LOCK-ON
LOKONBIT	=	BIT5	#	DESIRED
# BIT 4 FLAG 0 (S)				
NEEDLFLG	=	011D	#	TOTAL ATTITUDE
NEEDLBIT	=	BIT4	#	ERROR DISPLAYED
# BIT 3 FLAG 0				
FREEFLAG	=	012D	#	(USED BY P51-53 TEMP IN MANY DIFFERENT
FREEFBIT	=	BIT3	#	ROUTINES & BY LUNAR + SOLAR EPHEMERIDES)
# BIT 2 FLAG 0				
R10FLAG	=	013D	#	R10 OUTPUTS DATA TO
R10FLBIT	=	BIT2	#	ALTITUDE & ALTITUDE
			#	RATE METERS ONLY
			#	
# BIT 1 FLAG 0 (L)				
OLDESFLG	=	014D	#	R29 GYRO CMD LOOP
OLDESBIT	=	BIT1	#	REQUESTED
FLAGWRD1	=	STATE +1	#	(015-029)
# Page 67			#	(SET)
				(RESET)
# BIT 15 FLAG 1 (S)				
NJETSFLG	=	015D	#	TWO JET RCS BURN
NJETSBIT	=	BIT15		FOUR JET RCS BU
# BIT 14 FLAG 1 (L)				

DIDFLAG	=	016D	#	INERTIAL DATA IS	PERFO
DIDFLBIT	=	BIT14	#	AVAILABLE	INITI
# BIT 13 FLAG 1 (S)					
ERADFLAG	=	017D	#	COMPUTE REARTH	USE C
ERADFBIT	=	BIT13	#	FISCHER ELLIPSOID	PAD R
# BIT 12 FLAG 1					
RODFLAG	=	018D	#	IF IN P66, NORMAL	IF IN
RODFLBIT	=	BIT12	#	OPERATION CONTINUES.	IALIZ
			#	RESTART CLEARS FLAG	FORM
# BIT 11 FLAG 1					
	=	019D			
	=	BIT11			
# BIT 10 FLAG 1 (L)					
R61FLAG	=	020D	#	RUN R61 LEM	RUN R
R61FLBIT	=	BIT10			
# BIT 9 FLAG 1					
	=	021D			
	=	BIT9			
# BIT 8 FLAG 1 (S)					
VEHUPFLG	=	022D	#	CSM STATE-VECTOR	LEM S
VEHUPBIT	=	BIT8	#	BEING UPDATED	BEING
# BIT 7 FLAG 1 (S)					
UPDATFLG	=	023D	#	UPDATING BY MARKS	UPDAT
UPDATBIT	=	BIT7	#	ALLOWED	NOT A
# BIT 6 FLAG 1 (S)					
NOUPFLAG	=	024D	#	NEITHER CSM	EITH
			#	NOR LM STATE VECTOR	VECT
NOUPFBIT	=	BIT6	#	MAY BE UPDATED	UPDAT
# Page 68					
# BIT 5 FLAG 1 (S)					
TRACKFLG	=	025D	#	TRACKING ALLOWED	TRAC
TRACKBIT	=	BIT5			
# BIT 4 FLAG 1					
	=	026D			
	=	BIT4			

July 12, 2016

577

BIT 3 FLAG 1 (S)

SLOPESW = 027D

SLOPEBIT = BIT3

BIT 2 FLAG 1 (S)

GUESSW = 028D

GUESSBIT = BIT2

BIT 1 FLAG 1

= 029D

= BIT1

FLAGWRD2 = STATE +2

ITERATE WITH BIAS
METHOD IN ITERATOR
#

ITERATE WITH RE
FALSI METHOD IN
ITERATOR

NO STARTING VALUE
FOR ITERATION

STARTING VALUE
ITERATION EXIST

OH 2009-05-15 Scan does not have this line

(030-044)

(SET) (RESET)

BIT 15 FLAG 2 (S)

DRIFTFLG = 030D

DRFTBIT = BIT15

T3RUPT CALLS GYRO
COMPENSATION

T3RUPT DOES NO
COMPENSATION

BIT 14 FLAG 2 (S)

SRCHOPTN = 031D

SRCHOBIT = BIT14

RADAR IN AUTOMATIC
SEARCH OPTION (R24)

RADAR NOT IN AU
MATIC SEARCH OF

BIT 13 FLAG 2 (S)

ACMODFLG = 032D

ACMODBIT = BIT13

MANUAL ACQUISITION
BY RENDEZVOUS RADAR

AUTO ACQUISITIO
BY RENDEZVOUS R

BIT 12 FLAG 2 (S)

LOSCMFLG = 033D

LOSCMBIT = BIT12

LINE OF SIGHT BEING
COMPUTED (R21)

LINE OF SIGHT M
BEING COMPUTED

Page 69

BIT 11 FLAG 2 (S)

STEERSW = 034D

STEERBIT = BIT11

SUFFICIENT THRUST
IS PRESENT

INSUFFICIENT TH
IS PRESENT

BIT 10 FLAG 2 (S)

= 035D

= BIT10

OH 2009-05-15 These two line don't appear in

BIT 9 FLAG 2 (S)

IMPULSW = 036D

MINIMUM IMPULSE
BURN (CUTOFF TIME

STEERING BURN (C
CUTOFF TIME YET

IMPULBIT	=	BIT9	#	SPECIFIED)	AVAIL
# BIT 8 FLAG 2 (S)					
XDELVFLG	=	037D	#	EXTERNAL DELTAV VG	LAMBE
XDELVBIT	=	BIT8	#	COMPUTATION	VG CO
# BIT 7 FLAG 2 (S)					
ETPIFLAG	=	038D	#	ELEVATION ANGLE	TPI 7
			#	SUPPLIED FOR	FOR P
ETPIBIT	=	BIT7	#	P34,74	ELEVA
# BIT 7 FLAG 2 (L)					
OPTNSW	=	ETPIFLAG	#	SOI PHASE OF P38/78	SOR P
OPTNBIT	=	BIT7			
# BIT 6 FLAG 2 (S)					
FINALFLG	=	039D	#	LAST PASS THROUGH	INTER
			#	RENDEZVOUS PROGRAM	RENDE
FINALBIT	=	BIT6	#	COMPUTATIONS	COMPU
# BIT 5 FLAG 2 (S)					
AVFLAG	=	040D	#	LEM IS ACTIVE	CSM 1
AVFLBIT	=	BIT5	#	VEHICLE	VEHIC
# BIT 4 FLAG 2 (S)					
PFRATFLG	=	041D	#	PREFERRED ATTITUDE	PREFE
PFRATBIT	=	BIT4	#	COMPUTED	NOT C
# BIT 3 FLAG 2 (S)					
# Page 70					
CALCMAN3	=	042D	#	NO FINAL ROLL	FINAL
CALC3BIT	=	BIT3	#		NECES
# BIT 2 FLAG 2 (S)					
CALCMAN2	=	043D	#	PERFORM MANEUVER	BYPAS
CALC2BIT	=	BIT2	#	STARTING PROCEDURE	PROCI
# BIT 1 FLAG 2 (S)					
NODOFLAG	=	044D	#	V37 NOT PERMITTED	V37 P
NODOBIT	=	BIT1			
FLAGWRD3	=	STATE +3	#	(045-059)	
			#	(SET)	(RESI

July 12, 2016

579

# BIT 15 FLAG 3	=	045D	#		
	=	BIT15	#	OH 2009-05-15 This line is not in scans	
# BIT 14 FLAG 3 (S)					
GLOKFAIL	=	046D	#	GIMBAL LOCK HAS	NOT IN GIMBAL L
GLOKFBIT	=	BIT14	#	OCCURRED	
# BIT 13 FLAG 3 *** PROTECTED FROM FRESH START ***					
REFSMFLG	=	047D	#	REFSMMAT GOOD	REFSMMAT NO GOC
REFSMBIT	=	BIT13			
# BIT 12 FLAG 3 (S)					
LUNAFLAG	=	048D	#	LUNAR LAT-LONG	EARTH LAT-LONG
LUNABIT	=	BIT12			
# BIT 11 FLAG 3 (L)					
NOR29FLG	=	049D	#	R29 NOT ALLOWED	R29 ALLOWED (RR
NR29FBIT	=	BIT11	#		IGNATED POWERED
# BIT 10 FLAG 3 (S)					
VFLAG	=	050D	#	LESS THAN TWO STARS	TWO STARS IN FI
VFLAGBIT	=	BIT10	#	IN FIELD OF VIEW	OF VIEW
# BIT 9 FLAG 3 (S)					
R04FLAG	=	051D	#	ALARM 521	ALARM 521 ALLOW
			#	SUPPRESSED	
# Page 71					
R04FLBIT	=	BIT9			
# BIT 9 FLAG 3 (L)					
READRFLG	=	R04FLAG	#	READING RR DATA	NOT READING RR
READRBIT	=	BIT9	#	PURSUANT TO R29	PURSUANT TO R29
# BIT 8 FLAG 3 (S)					
PRECIFLG	=	052D	#	NORMAL INTEGRATION	ENGAGES 4-TIME
			#	IN P00	(P00) LOGIC IN
PRECIBIT	=	BIT8	#		GRATION
# BIT 7 FLAG 3 (S)					
CULTFLAG	=	053D	#	STAR OCCULTED	STAR NOT OCCULT
CULTBIT	=	BIT7			
# BIT 6 FLAG 3 (S)					
ORBWFLAG	=	054D	#	W MATRIX VALID FOR	W MATRIX INVALID
ORBWFBIT	=	BIT6	#	ORBITAL NAVIGATION	ORBITAL NAVIGAT

# BIT 5 FLAG 3 (S)				
STATEFLG	=	055D	#	PERMANENT STATE
STATEBIT	=	BIT5	#	VECTOR UPDATED
				PERMA
# BIT 4 FLAG 3 (S)				
INTYPFLG	=	056D	#	CONIC INTEGRATION
INTYPBIT	=	BIT4		ENCKE
# BIT 3 FLAG 3 (S)				
VINTFLAG	=	057D	#	CSM STATE VECTOR
VINTFBIT	=	BIT3	#	BEING INTEGRATED
				LEM S
# BIT 2 FLAG 3 (S)				
D6OR9FLG	=	058D	#	DIMENSION OF W IS 9
D6OR9BIT	=	BIT2	#	FOR INTEGRATION
				DIMEN
# BIT 1 FLAG 3 (S)				
DIM0FLAG	=	059D	#	W MATRIX IS TO BE
DIM0BIT	=	BIT1	#	USED
				W MAT
FLAGWRD4	=	STATE +4	#	(060-074)
				USED
# Page 72				
			#	(SET)
				(RESI
# BIT 15 FLAG 4 (S)				
MRKIDFLG	=	060D	#	MARK DISPLAY IN
MRKIDBIT	=	BIT15	#	ENDIDLE
				NO MA
# BIT 14 FLAG 4 (S)				
PRIODFLG	=	061D	#	PRIORITY DISPLAY IN
PRIODBIT	=	BIT14	#	ENDIDLE
				NO PR
# BIT 13 FLAG 4 (S)				
NRMIDFLG	=	062D	#	NORMAL DISPLAY IN
NRMIDBIT	=	BIT13	#	ENDIDLE
				NO NO
# BIT 12 FLAG 4 (S)				
PDSPFLAG	=	063D	#	P20 SETS SO AS TO
			#	TURN A NORMAL DIS-
PDSPFBIT	=	BIT12	#	PLAY INTO A PRIORITY
			#	DISPLAY IN R60
				LEAVI
# BIT 11 FLAG 4 (S)				
MWAITFLG	=	064D	#	HIGHER PRIORITY
				NO H

July 12, 2016

581

MWAITBIT	=	BIT11	#	DISPLAY OPERATING	DISPLAY OPERATING
			#	WHEN MARK	WHEN MARK DISPL
			#	DISPLAY INITIATED	INITIATED
# BIT 10 FLAG 4 (S)					
NWAITFLG	=	065D	#	HIGHER PRIORITY	NO HIGHER PRIOR
			#	DISPLAY OPERATING	DISPLAY OPERATI
NWAITBIT	=	BIT10	#	WHEN NORMAL	WHEN NORMAL DIS
			#	DISPLAY INITIATED	INITIATED
# BIT 9 FLAG 4 (S)					
MRKNVFLG	=	066D	#	ASTRONAUT USING	ASTRONAUT NOT U
			#	KEYBOARD WHEN MARK	KEYBOARD WHEN M
MRKNVBIT	=	BIT9	#	DISPLAY INITIATED	DISPLAY INITIAT
# BIT 8 FLAG 4 (S)					
NRMNVFLG	=	067D	#	ASTRONAUT USING	ASTRONAUT NOT U
			#	KEYBOARD WHEN	KEYBOARD WHEN
NRMNVBIT	=	BIT8	#	NORMAL DISPLAY	NORMAL DISPLAY
			#	INITIATED	INITIATED
# BIT 7 FLAG 4 (S)					
PRONVFLG	=	068D	#	ASTRONAUT USING	ASTRONAUT NOT U
# Page 73					
PRONVBIT	=	BIT7	#	KEYBOARD WHEN	KEYBOARD WHEN
			#	PRIORITY DISPLAY	PRIORITY DISPLA
			#	INITIATED	INITIATED
# BIT 6 FLAG 4 (S)					
PINBRFLG	=	069D	#	ASTRONAUT HAS	ASTRONAUT HAS M
			#	INTERFERED WITH	INTERFERED WITH
PINBRBIT	=	BIT6	#	EXISTING DISPLAY	EXISTING DISPLA
# BIT 5 FLAG 4 (S)					
MRUPTFLG	=	070D	#	MARK DISPLAY	MARK DISPLAY NO
			#	INTERRUPTED BY	INTERRUPTED BY
MRUPTBIT	=	BIT5	#	PRIORITY DISPLAY	PRIORITY DISPLA
# BIT 4 FLAG 4 (S)					
NRUPTFLG	=	071D	#	NORMAL DISPLAY	NORMAL DISPLAY
			#	INTERRUPTED BY	INTERRUPTED BY
NRUPTBIT	=	BIT4	#	PRIORITY OR MARK	PRIORITY OR MAR
			#	DISPLAY	DISPLAY
# BIT 3 FLAG 4 (S)					

MKOVFLAG	=	072D	#	MARK DISPLAY OVER	NO MA
MKOVBIT	=	BIT3	#	NORMAL	NORMA
# BIT 2 FLAG 4					
	=	073D			
	=	BIT2		# OH 2009-05-15 Not in scan.	
# BIT 1 FLAG 4 (S)					
XDSPFLAG	=	074D	#	MARK DISPLAY NOT	NO SE
XDSPBIT	=	BIT1	#	TO BE INTERRUPTED	INFOR
FLAGWRD5	=	STATE +5	#	(075-089)	
			#	(SET)	(RESI
# BIT 15 FLAG 5 (S)					
DSKYFLAG	=	075D	#	DISPLAYS SENT TO	NO D
DSKYFBIT	=	BIT15	#	DSKY	
# BIT 14 FLAG 5					
	=	076D			
	=	BIT14			
# Page 74					
# BIT 13 FLAG 5 (S,L)					
SNUFFER	=	077D	#	U,V JETS DISABLED	U,V J
			#	DURING DPS	DURIN
SNUFFBIT	=	BIT13	#	BURNS (V65)	BURNS
# BIT 12 FLAG 5 (S)					
NOTHROTL	=	078D	#	INHIBIT FULL	PERM
NOTHRBIT	=	BIT12	#	THROTTLE	
# BIT 11 FLAG 5 (S,L)					
R77FLAG	=	079D	#	R77 IS ON,	R77
			#	SUPPRESS ALL RADAR	
			#	ALARMS AND TRACKER	
R77FLBIT	=	BIT11	#	FAILS	
# BIT 10 FLAG 5 (S)					
RNGSCFLG	=	080D	#	SCALE CHANGE HAS	NO SC
			#	OCCURRED DURING	OCCUR
RNGSCBIT	=	BIT10	#	RR READING	RR RE
# BIT 9 FLAG 5 (S)					

July 12, 2016

583

DMENFLG	=	081D	#	DIMENSION OF W IS 9	DIMENSION OF W
DMENFBIT	=	BIT9	#	FOR INCORPORATION	FOR INCORPORATI
# BIT 8 FLAG 5	(S)				
	=	082D			
	=	BIT8			
# BIT 7 FLAG 5	(S)				
ENGONFLG	=	083D	#	ENGINE TURNED ON	ENGINE TURNED C
ENGONBIT	=	BIT7	#		
# BIT 6 FLAG 5	(S)				
3AXISFLG	=	084D	#	MANEUVER SPECIFIED	MANEUVER SPECIF
			#	BY THREE AXES	BY ONE AXIS; R6
3AXISBIT	=	BIT6	#		CALLS VECPOINT.
# BIT 5 FLAG 5					
	=	085D			
	=	BIT5		# OH 2009-05-15 Not in scan	
# BIT 4 FLAG 5	(S)				
# Page 75					
NORRMON	=	086D	#	BYPASS RR GIMBAL	PERFORM
NORRMBIT	=	BIT4	#	MONITOR	RR GIMBAL MONIT
# BIT 3 FLAG 5	(S)				
SOLNSW	=	087D	#	LAMBERT DOES NOT	LAMBERT CONVERG
			#	CONVERGE, OR TIME-RAD	TIME-RADIUS NON
SOLNSBIT	=	BIT3	#	NEARLY CIRCULAR	CIRCULAR
# BIT 2 FLAG 5	(S)				
MGLVFLAG	=	088D	#	LOCAL VERTICAL	MIDDLE GIMBAL A
			#	COORDINATES	COMPUTED
MGLVFBIT	=	BIT2	#	COMPUTED	
# BIT 1 FLAG 5	(S)				
RENDWFLG	=	089D	#	W MATRIX VALID	W MATRIX INVALI
			#	FOR RENDEZVOUS	FOR RENDEZVOUS
RENDWBIT	=	BIT1	#	NAVIGATION	NAVIGATION
FLAGWRD6	=	STATE +6	#	(090-104)	
			#	(SET)	(RESET)

BIT 15 FLAG 6 (S)

S32.1F1 = 090D
S32BIT1 = BIT15

DELTA V AT CSI TIME DVT1
ONE EXCEEDS MAX

BIT 14 FLAG 6 (S)

S32.1F2 = 091D
S32BIT2 = BIT14

FIRST PASS OF REITH
NEWTON ITERATION NEWTO

BIT 13 FLAG 6 (S)

S32.1F3A = 092D
S32BIT3A = BIT13

BIT 13 AND BIT 12 FUNCTION AS AN OR
PAIR (13,12) INDICATING THE POSSIBL
CURRENCE OF 2 NEWTON ITERATIONS FOR
IN THE PROGRAM IN THE FOLLOWING OR
(0,1) (I.E. BIT 13 RESET, BIT 12 SE
= FIRST NEWTON ITERATION BEING
(0,0)= FIRST PASS OF SECOND NEWTON
(1,1)= 50 FT/SEC STAGE OF SECOND NE
(1,0)= REMAINDER OF SECOND NEWTON I

BIT 12 FLAG 6 (S)

S32.1F3B = 093D
S32BIT3B = BIT12

BIT 11 FLAG 6 (S)

= 094D
= BIT11

#

Page 76

BIT 10 FLAG 6 (S)

GMBDRVSW = 095D
GMBDRBIT = BIT10

TRIMGIMB OVER TRIMC
#

BIT 9 FLAG 6

= 096D
= BIT9

#

BIT 8 FLAG 6 (S)

MUNFLAG = 097D
MUNFLBIT = BIT8

SERVICER CALLS SERV
MUNRVG CALC

BIT 7 FLAG 6 (L)

= 098D
= BIT7

#

BIT 6 FLAG 6 (L)

REDFLAG = 099D
REDFLBIT = BIT6

LANDING SITE LAND
REDESIGNATION REDES
PERMITTED PERM

BIT 5 FLAG 6

= 100D
= BIT5

OH 2009-05-15 Not in scan

July 12, 2016

585

# BIT 4 FLAG 6			#		
	=	101D			
	=	BIT4	#	OH 2009-05-15 Not in scan	
# BIT 3 FLAG 6 (S)					
NTARGFLG	=	102D	#	ASTRONAUT DID	ASTRONAUT DID M
			#	OVERWRITE DELTA	OVERWRITE DELTA
NTARGBIT	=	BIT3	#	VELOCITY AT TPI	VELOCITY
			#	OR TPM (P34,35)	
# BIT 2 FLAG 6					
AUXFLAG	=	103D	#	PROVIDING IDLEFLAG	SERVICER WILL S
AUXFLBIT	=	BIT2	#	IS NOT SET, SERV-	DVMON ON ITS NE
			#	ICER WILL EXERCISE	PASS EVEN IF TH
			#	DVMON ON ITS NEXT	IDLEFLAG IS NOT
			#	PASS.	IT WILL THEN SE
			#		AUXFLAG.
# BIT 1 FLAG 6 (L)					
ATTFLAG	=	104D	#	LEM ATTITUDE EXISTS	NO LEM ATTITUDE
			#	IN MOON-FIXED	AVAILABLE IN MO
# Page 77					
ATTFLBIT	=	BIT1	#	COORDINATES	FIXED COORDINAT
FLAGWRD7	=	STATE +7	#	(105-119)	
			#	(SET)	(RESET)
# BIT 15 FLAG 7 (S)					
ITSWICH	=	105D	#	R34;TPI TIME TO BE	TPI HAS BEEN
ITSWBIT	=	BIT15	#	COMPUTED	COMPUTED
# BIT 14 FLAG 7 (S)					
MANUFLAG	=	106D	#	ATTITUDE MANEUVER	NO ATTITUDE MAN
			#	GOING DURING RR	DURING RR SEAR
MANUFBIT	=	BIT14	#	SEARCH	
# BIT 13 FLAG 7 (S)					
IGNFLAG	=	107D	#	TIG HAS ARRIVED	TIG HAS NOT ARR
IGNFLBIT	=	BIT13	#		
# BIT 12 FLAG 7 (S)					
ASTNFLAG	=	108D	#	ASTRONAUT HAS	ASTRONAUT HAS M
ASTNBIT	=	BIT12	#	OKAYED IGNITION	OKAYED IGNITION

# BIT 11 FLAG 7 (L)				
SWANDISP	=	109D	#	LANDING ANALOG
SWANDBIT	=	BIT11	#	DISPLAYS ENABLED
# BIT 10 FLAG 7 (S)				
NORMSW	=	110D	#	UNIT NORMAL INPUT
NORMSBIT	=	BIT10	#	TO LAMBERT
# BIT 9 FLAG 7 (S)				
RVS	=	111D	#	DO NOT COMPUTE
RVS	=	111D	#	FINAL STATE VECTOR
RVS	=	111D	#	IN TIME-DELTA
# BIT 8 FLAG 7 (S)				
V67FLAG	=	112D	#	ASTRONAUT OVERWRITE
V67FLBIT	=	BIT8	#	W-MATRIX INITIAL
			#	VALUES
# Page 78				
# BIT 7 FLAG 7 (S)				
IDLEFLAG	=	113D	#	NO DV MONITOR
IDLEFBIT	=	BIT7	#	
# BIT 6 FLAG 7 (S)				
V37FLAG	=	114D	#	AVERAGEG (SERVICER)
V37FLBIT	=	BIT6	#	RUNNING
# BIT 5 FLAG 7 (S)				
AVEGFLAG	=	115D	#	AVERAGEG (SERVICER)
AVEGFBIT	=	BIT5	#	DESIRED
# BIT 4 FLAG 7 (S)				
UPLOCKFL	=	116D	#	K-KBAR-K FAIL
UPLOCBIT	=	BIT4	#	
# BIT 3 FLAG 7 (S)				
VERIFLAG	=	117D	#	CHANGED WHEN V33E OCCURS AT END OF
VERIFBIT	=	BIT3	#	
# BIT 2 FLAG 7 (L,C)				
V82EMFLG	=	118D	#	MOON VICINITY
V82EMBIT	=	BIT2	#	
# BIT 1 FLAG 7 (S)				
TFFSW	=	119D	#	CALCULATE TPERIGEE

July 12, 2016

587

TFFSWBIT	=	BIT1	#		
FLAGWRD8	=	STATE +8D	#	(120-134)	
			#	(SET)	(RESET)
# BIT 15 FLAG 8 (S)					
RPQFLAG	=	120D	#	RPQ NOT COMPUTED	RPQ COMPUTED
			#	(RPQ = VECTOR BE-	
RPQFLBIT	=	BIT15	#	TWEEN SECONDARY BODY	
			#	AND PRIMARY BODY)	
# BIT 14 FLAG 8					
	=	121D	#		
	=	BIT14	#		
# Page 79					
# BIT 13 FLAG 8 (S)					
NEWIFLG	=	122D	#	FIRST PASS THROUGH	SUCCEEDING ITER
NEWIBIT	=	BIT13	#	INTEGRATION	OF INTEGRATION
# BIT 12 FLAG 8 *** PROTECTED FROM FRESH START ***					
CMOONFLG	=	123D	#	PERMANENT CSM STATE	PERMANENT CSM S
CMOONBIT	=	BIT12	#	IN LUNAR SPHERE	IN EARTH SPHERE
# BIT 11 FLAG 8 *** PROTECTED FROM FRESH START ***					
LMOONFLG	=	124D	#	PERMANENT LM STATE	PERMANENT LM ST
LMOONBIT	=	BIT11	#	IN LUNAR SPHERE	IN EARTH SPHERE
# BIT 10 FLAG 8 (L)					
FLUNDISP	=	125D	#	CURRENT GUIDANCE	CURRENT GUIDANC
FLUNDBIT	=	BIT10	#	DISPLAYS INHIBITED	DISPLAYS PERMIT
# BIT 9 FLAG 8 (L)					
P39/79SW	=	126D	#	P39/79 OPERATING	P38/78 OPERATIN
P39SWBIT	=	BIT9	#		
# BIT 8 FLAG 8 *** PROTECTED FROM FRESH START ***					
SURFFLAG	=	127D	#	LM ON LUNAR SURFACE	LM NOT ON LUNAR
SURFFBIT	=	BIT8	#		SURFACE
# BIT 7 FLAG 8 (S)					
INFINFLG	=	128D	#	NO CONIC SOLUTION	CONIC SOLUTION
			#	(CLOSURE THROUGH	EXISTS
INFINBIT	=	BIT7	#	INFINITY REQUIRED)	

# BIT 6 FLAG 8 (S)				
ORDERSW	=	129D	#	ITERATOR USES 2ND
ORDERBIT	=	BIT6	#	ORDER MINIMUM MODE
				ITERA
# BIT 5 FLAG 8 (S)				
APSESW	=	130D	#	RDESIRED OUTSIDE
			#	PERICENTER-APOCENTER
APSESBIT	=	BIT5	#	RANGE IN TIME-RADIUS
				RANGE
# BIT 4 FLAG 8 (S)				
COGAFLAG	=	131D	#	NO CONIC SOLUTION --
			#	TOO CLOSE TO RECTI-
				CONIC
				EXIST
# Page 80				
COGAFBIT	=	BIT4	#	LINEAR (COGA OVERFLWS)
				OVER
# BIT 3 FLAG 8 (S)				
	=	132D	#	
	=	BIT3	#	OH 2009-05-15 Line not in scan
# BIT 2 FLAG 8 (L)				
INITALGN	=	133D	#	INITIAL PASS THRU
INITABIT	=	BIT2	#	P57
				SECO
				(CHE
# BIT 1 FLAG 8 (S)				
360SW	=	134D	#	TRANSFER ANGLE NEAR
360SWBIT	=	BIT1	#	360 DEGREES
				TRANS
				NEAR
FLAGWRD9	=	STATE +9D	#	(135-149)
			#	(SET)
				(RES
# BIT 15 FLAG 9				
	=	135D	#	
	=	BIT15	#	
# BIT 14 FLAG 9 (L)				
FLVR	=	136D	#	VERTICAL RISE
FLVRBIT	=	BIT14	#	(ASCENT GUIDANCE)
				NON-V
# BIT 13 FLAG 9				
	=	137D	#	
	=	BIT13	#	OH 2009-05-15 Line not in scan

July 12, 2016

589

# BIT 12 FLAG 9 (L)				
FLPC	=	138D	#	NO POSITION CONTROL
FLPCBIT	=	BIT12	#	(ASCENT GUIDANCE)
				POSITION CONTROL
# BIT 11 FLAG 9 (L)				
FLPI	=	139D	#	PRE-IGNITION PHASE
FLPIBIT	=	BIT11	#	(ASCENT GUIDANCE)
				REGULAR GUIDANCE
# BIT 10 FLAG 9 (L)				
FLRCS	=	140D	#	RCS INJECTION MODE
FLRCSBIT	=	BIT10	#	(ASCENT GUIDANCE)
				MAIN ENGINE MODE
# BIT 9 FLAG 9 (L)				
# Page 81				
LETABORT	=	141D	#	ABORT PROGRAMS
LETABBIT	=	BIT9	#	ARE ENABLED
				ABORT PROGRAMS
				ARE NOT ENABLED
# BIT 8 FLAG 9 (L)				
FLAP	=	142D	#	APS CONTINUED ABORT
			#	AFTER DPS STAGING
FLAPBIT	=	BIT8	#	(ASCENT GUIDANCE)
				APS ABORT IS NO
				CONTINUATION
# BIT 7 FLAG 9 (L)				
	=	143D		
	=	BIT7		
				# OH 2009-05-15 Line not in scan
# BIT 6 FLAG 9 (L)				
ROTFLAG	=	144D	#	P70 AND P71 WILL
ROTFLBIT	=	BIT6	#	FORCE VEHICLE
			#	ROTATION IN THE
			#	PREFERRED DIRECTION
				P70 AND P71 WILL
				FORCE VEHICLE
				ROTATION IN THE
				PREFERRED DIREC
# BIT 5 FLAG 9 (S)				
QUITFLAG	=	145D	#	DISCONTINUE INTEGR.
QUITBIT	=	BIT5	#	
				CONTINUE INTEGR
# BIT 4 FLAG 9				
	=	146D	#	
	=	BIT4	#	
# BIT 3 FLAG 9 (L)				
MID1FLAG	=	147D	#	INTEGRAT TO TDEC
MID1FBIT	=	BIT3	#	
				INTEGRATE TO TH
				THEN-PRESENT TI

# BIT 2 FLAG 9 (L)					
MIDAVFLG	=	148D	#	INTEGRATION ENTERED	INTEG
			#	FROM ONE OF MIDTOAV	NOT F
MIDAVBIT	=	BIT2	#	PORTALS	MIDTO
# BIT 1 FLAG 9 (S)					
AVEMIDSW	=	149D	#	AVETOMID CALLING	NO AV
			#	FOR W.MATRIX INTEGR	ALLOW
AVEMDBIT	=	BIT1	#	DON'T WRITE OVER RN,	PIPT
			#	VN,PIPTIME	
RASFLAG	EQUALS	FLGWRD10	#	WAS ONLY AN INSTALL-ERASTALL FLAG	
# Page 82					
FLGWRD10	=	STATE +10D	#	(150-164)	
			#	(SET)	(RES)
# BIT 15 FLAG 10 (S)					
	=	150D	#		
	=	BIT15	#	OH 2009-05-15 Line not in scan	
# BIT 14 FLAG 10 (L,C)					
INTFLAG	=	151D	#	INTEGRATION IN	INTEG
INTFLBIT	=	BIT14	#	PROGRESS	PROGR
# BIT 13 FLAG 10 (S,L)					
APSFLAG	=	152D	#	ASCENT STAGE	DESCR
APSFLBIT	=	BIT13	#	*** PROTECTED FROM FRESH ST	
# BIT 12 FLAG 10					
	=	153D	#		
	=	BIT12	#	OH 2009-05-15 Line not in scan	
# BIT 11 FLAG 10					
	=	154D	#		
	=	BIT11	#	OH 2009-05-15 Line not in scan	
# BIT 10 FLAG 10					
	=	155D	#		
	=	BIT10	#	OH 2009-05-15 Line not in scan	
# BIT 9 FLAG 10					
	=	156D	#		
	=	BIT9	#	OH 2009-05-15 Line not in scan	

July 12, 2016

591

# BIT 8 FLAG 10			#	
	=	157D	#	OH 2009-05-15 Line not in scan
	=	BIT8		
# BIT 7 FLAG 10 (L,C)				
REINTFLG	=	158D	#	INTEGRATION ROUTINE
REINTBIT	=	BIT7	#	TO BE RESTARTED
				INTEGRATION ROUTINE
				NOT TO BE RESTARTED
# BIT 6 FLAG 10			#	
	=	159D	#	OH 2009-05-15 Line not in scan
	=	BIT6		
# BIT 5 FLAG 10			#	
	=	160D	#	OH 2009-05-15 Line not in scan
	=	BIT5		
# Page 83				
# BIT 4 FLAG 10			#	
	=	161D	#	OH 2009-05-15 Line not in scan
	=	BIT4		
# BIT 3 FLAG 10			#	
	=	162D	#	OH 2009-05-15 Line not in scan
	=	BIT3		
# BIT 2 FLAG 10			#	
	=	163D	#	OH 2009-05-15 Line not in scan
	=	BIT2		
# BIT 1 FLAG 10			#	
	=	164D	#	OH 2009-05-15 Line not in scan
	=	BIT1		
FLGWRD11	=	STATE +11D	#	(165-179)
			#	(SET) (RESET)
# BIT 15 FLAG 11 (L)(R12)				
LRBYPASS	=	165D	#	BYPASS ALL LANDING
LRBYBIT	=	BIT15	#	RADAR UPDATES
				DO NOT BYPASS LANDING
				UPDATES
# BIT 14 FLAG 11			#	
	=	166D		

	=	BIT14	#		
# BIT 13 FLAG 11					
	=	167D	#		
	=	BIT13	#		
# BIT 12 FLAG 11 (L)(R12)					
VXINH	=	168D	#	IF Z VELOCITY DATA	UPDAT
			#	UNREASONABLE,	VELO
VXINHBIT	=	BIT12	#	BYPASS X VELOCITY	
			#	UPDATE ON NEXT PASS	
# BIT 11 FLAG 11 (L)(R12)					
PSTHIGAT	=	169D	#	PAST HIGATE	PREH
PSTHIBIT	=	BIT11	#		
# BIT 10 FLAG 11 (L)(R12)					
# Page 84					
NOLRREAD	=	170D	#	LANDING RADAR	LR NO
			#	REPOSITIONING;	
NOLRRBIT	=	BIT10	#	BYPASS UPDATE	
# BIT 9 FLAG 11 (L)(R12)					
XORFLG	=	171D	#	BELOW LIMIT	ABOV
			#	INHIBIT X AXIS	NOT
XORFLBIT	=	BIT9	#	OVERRIDE	
# BIT 8 FLAG 11					
LRINH	=	172D	#	LANDING RADAR UP-	LR UP
LRINHBIT	=	BIT8	#	DATES PERMITTED	BY AS
			#	BY ASTRONAUT	
# BIT 7 FLAG 11 (L)(R12)					
VELDATA	=	173D	#	LR VELOCITY	LR VI
VELDABIT	=	BIT7	#	MEASUREMENT MADE	NOT M
# BIT 6 FLAG 11 (L)(R12)					
READLR	=	174D	#	OK TO READ LR	DO NO
READLBIT	=	BIT6	#	RANGE DATA	DATA
# BIT 5 FLAG 11 (L)(R12)					
READVEL	=	175D	#	OK TO READ LR	DO NO
READVBIT	=	BIT5	#	VELOCITY DATA	VELO
# BIT 4 FLAG 11 (L)(R12)					

July 12, 2016

593

RNGEDATA	=	176D	#	LR ALTITUDE	LR ALTITUDE MEA
RNGEDBIT	=	BIT4	#	MEASUREMENT MADE	NOT MADE
# BIT 3 FLAG 11					
SCALBAD	=	177D	#	LR LOW SCALE DISP-	LS SCALE DISCRE
SCABBIT	=	BIT3	#	CRETE NOT PRESENT	APPEARS OK
			#	WHEN IT SHOULD	
# BIT 2 FLAG 11 (L)(R12)					
VFLSHFLG	=	178D	#	LR VELOCITY FAIL	LR VEL FAIL LAM
			#	LAMP SHOULD BE	SHOULDN'T FLASH
VFLSHBIT	=	BIT2	#	FLASHING	
# BIT 1 FLAG 11 (L)(R12)					
# Page 85					
HFLSHFLG	=	179D	#	LR ALTITUDE FAIL	LR ALTITUDE FAI
HFLSHBIT	=	BIT1	#	LAMP SHOULD BE	LAMP SHOULD NOT
			#	FLASHING	FLASHING
RADMODES	EQUALS	FLGWRD12	#	RADAR FLAG WORD	
FLGWRD12	=	STATE +12D	#	(180-194)	WAS RADMODES
			#	(SET)	(RESET)
# BIT 15 FLAG 12					
CDESFLAG	=	180D	#	CONTINUOUS DESIG-	LGC CHECKS FOR
CDESBIT	=	BIT15	#	NATE, LGC COMMANDS	ON WHEN ANTENNA
			#	RR REGARDLESS OF	BEING DESIGNATE
			#	LOCK-ON	
# BIT 14 FLAG 12					
REMODFLG	=	181D	#	CHANGE IN ANTENNA	NO REMODE REQUE
REMODBIT	=	BIT14	#	MODE BEEN REQUESTED	OR OCCURRING
			#	I.E., REMODE	
# BIT 13 FLAG 12					
RCDUOFLG	=	182D	#	RR CDU'S BEING	RR CDU'S NOT BE
RCDUOBIT	=	BIT13	#	ZEROED	ZEROED
# BIT 12 FLAG 12					
ANTENFLG	=	183D	#	RR ANTENNA MODE IS	RR ANTENNA IN M
ANTENBIT	=	BIT12	#	MODE 2	
# BIT 11 FLAG 12					
REPOSOMON	=	184D	#	REPOSITION MONITOR.	NO REPOSITION T

REPOSBIT	=	BIT11	#	RR REPOSITION IS	PLAC
			#	TAKING PLACE	
# BIT 10 FLAG 12					
DESIGFLG	=	185D	#	RR DESIGNATE	RR DE
DESIGBIT	=	BIT10	#	REQUESTED OR IN	REQU
			#	PROGRESS	PROGE
# BIT 9 FLAG 12					
ALTSCALE	=	186D	#	LR ALTITUDE READING	LR AI
ALTSCBIT	=	BIT9	#	IS ON HIGH SCALE	IS ON
# Page 86					
# BIT 8 FLAG 12					
LRVELFLG	=	187D	#	LR VELOCITY DATA	NO LR
LRVELBIT	=	BIT8	#	FAIL	FAIL
# BIT 7 FLAG 12					
RCDUFAIL	=	188D	#	RR CDU FAIL HAS	RR CI
RCDUFBIT	=	BIT7	#	NOT OCCURRED	
# BIT 6 FLAG 12					
LRPOSFLG	=	189D	#	LANDING RADAR	LR PO
LRPOSBIT	=	BIT6	#	POSITION 2	
# BIT 5 FLAG 12					
LRALTFLG	=	190D	#	LR ALTITUDE DATA	NO LR
LRALTBIT	=	BIT5	#	FAIL. COULD NOT BE	FAIL
			#	READ SUCCESSFULLY.	
# BIT 4 FLAG 12					
RRDATAFL	=	191D	#	RR DATA FAIL.	NO RE
RRDATAFT	=	BIT4	#	DATA COULD NOT BE	
			#	READ SUCCESSFULLY	
# BIT 3 FLAG 12					
RRRSFLAG	=	192D	#	RR RANGE READING	RR RA
RRRSBIT	=	BIT3	#	ON THE HIGH SCALE	THE I
# BIT 2 FLAG 12					
AUTOMODE	=	193D	#	RR NOT IN AUTO MODE.	RR IN
AUTOMBIT	=	BIT2	#	AUTO MODE DISCRETE	
			#	IS NOT PRESENT	
# BIT 1 FLAG 12					
TURNONFL	=	194D	#	RR TURN-ON SEQUENCE	NO RE

July 12, 2016

595

TURNONBT	=	BIT1	#	IN PROGRESS. (ZERO	SEQUENCE IN PRO
			#	CDU'S, FIX ANTENNA	
			#	MODE)	
DAPBOOLS	EQUALS	FLGWRD13	#	DIGITAL AUTOPILOT FLAGWORD	
# Page 87					
FLGWRD13	=	STATE +13D	#	(195-209) WAS DAPBOOLS	
			#	(SET)	(RESET)
# BIT 15 FLAG 13					
PULSEFLG	=	195D	#	MINIMUM IMPUSE	NOT IN MINIMUM
PULSES	=	BIT15	#	COMMAND MODE IN	IMPULSE COMMAND
			#	"ATT HOLD" (V76)	(V77)
# BIT 14 FLAG 13					
USEQRFLG	=	196D	#	GIMBAL UNUSABLE.	TRIM GIMBAL MAY
USEQRJTS	=	BIT14	#	USE JETS ONLY.	USED.
# BIT 13 FLAG 13					
CSMDKFLG	=	197D	#	CSM DOCKED. USE	CSM NOT DOCKED
CSMDOCKD	=	BIT13	#	BACKUP DAP	
# BIT 12 FLAG 13					
OURRCFLG	=	198D	#	CURRENT DAP PASS	CURRENT DAP PAS
OURRCBIT	=	BIT12	#	IS RATE COMMAND	NOT RATE COMMAND
# BIT 11 FLAG 13					
ACC4-2FL	=	199D	#	4 JET X-AXIS TRANS-	2 JET X-AXIS TR
ACC4OR2X	=	BIT11	#	LATION REQUESTED	LATION REQUESTED
# BIT 10 FLAG 13					
AORBFLG	=	200D	#	B SYSTEM FOR X-	A SYSTEM FOR X-
AORBTRAN	=	BIT10	#	TRANSLATION	TRANSLATION PRE
# BIT 9 FLAG 13					
XOVINFLG	=	201D	#	X-AXIS OVERRIDE	X-AXIS OVERRIDE
XOVINHIB	=	BIT9	#	LOCKED OUT	
# BIT 8 FLAG 13					
DRIFTDFL	=	202D	#	ASSUME 0 OFFSET	USE OFFSET ACCE
DRIFTBIT	=	BIT8	#	DRIFTING FLIGHT	ION ESTIMATE
# BIT 7 FLAG 13					
RHCSCFLG	=	203D	#	NORMAL RHC SCALING	FINE RHC SCALIN

RHCSCALE	=	BIT7	#	REQUESTED	REQU
# Page 88					
# BIT 6 FLAG 13					
ULLAGFLG	=	204D	#	ULLAGE REQUEST BY	NO ID
ULLAGER	=	BIT6	#	MISSION PROGRAM	REQU
# BIT 5 FLAG 13					
AORBSFLG	=	205D	#	P-AXIS COUPLES 7.15	P-AXI
AORBSYST	=	BIT5	#	AND 8.16 PREFERRED	AND 3
# BIT 4 FLAG 13					
DBSELFLG	=	206D	#	MAX DB SELECTED	MIN I
DBSELECT	=	BIT4	#	BY CREW (5 DEG)	CREW
# BIT 3 FLAG 13					
ACCOKFLG	=	207D	#	CONTROL AUTHORITY	RESTA
ACCSOKAY	=	BIT3	#	VALUES FROM 1/ACCS	SINCE
			#	USABLE	OUTPU
# BIT 2 FLAG 13					
AUTR2FLG	=	208D	#	# THESE FLAGS ARE USED TOGETHER TO ID	
AUTRATE2	=	BIT2	#	# ASTRONAUT-CHOSEN KALCMANU MANEUVER	
# BIT 1 FLAG 13					
AUTR1FLG	=	209D	#	(0,0)=(BIT2,BIT1)=	0.2 DEG/SEC
AUTRATE1	=	BIT1	#	(0,1)=	0.5 DEG/SEC
			#	(1,0)=	2.0 DEG/SEC
			#	(1,1)=	10.0 DEG/SEC

Page 89 (nothing on this page)

This code is written to file `src/FLAGWORD-ASSIGNMENTS.s`.

B.34 FRESH START AND RESTART

```

597  <src/FRESH-START-AND-RESTART.s 597>≡
# Copyright:    Public domain.
# Filename:     FRESH_START_AND_RESTART.agc
# Purpose:      Part of the source code for Comanche, build 055. It
#               is part of the source code for the Command Module's
#               (CM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:   yaYUL
# Reference:    pp. 181-210
# Contact:      Ron Burkey <info@sandroid.org>,
#               Fabrizio Bernardini <fabrizio@spacecraft.it>
# Website:      http://www.ibiblio.org/apollo.
# Mod history:  2009-05-16 FB   Transcription Batch 2 Assignment.
#               2009-05-20 RSB   Removed an extraneous "TC STARTSUB".
#               2009-05-21 RSB   Changed a "TC BANKCALL" to "TC STOPRATE"
#                               in INITSUB.
#
# The contents of the "Comanche055" files, in general, are transcribed
# from scanned documents.
#
# Assemble revision 055 of AGC program Comanche by NASA
# 2021113-051. April 1, 1969.
#
# This AGC program shall also be referred to as Colossus 2A
#
# Prepared by
#               Massachussets Institute of Technology
#               75 Cambridge Parkway
#               Cambridge, Massachusetts
#
# under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information. Please report any errors to info@sandroid.org.
#
# Page 181
# PROGRAM DESCRIPTION
#
# FUNCTIONAL DESCRIPTION
#
# SLAP1          MAN INITIATED FRESH START
# 1. EXECUTE STARTSUB
# 2. TURN OFF DSKY DISCRETE-LAMPS
# 3. CLEAR FAIL REGISTERS, SELF-CHECK ERROR COUNTER AND RESTART
#    COUNTER
#
#               8 APRIL, 1967
#               SUNDISK REV 120

```

```

#      4.      EXECUTE DOFSTART
#
# DOFSTART    MACHINE INITIATED FRESH START
#
#      1.      CLEAR SELF-CHECK REGISTERS, MODE REGISTER AND CDUZ REGISTER
#      2.      CLEAR PHASE TABLE
#      3.      INITIALIZE IMU FLAGS
#      4.      INITIALIZE FLAGWORDS
#      5.      TRANSFER CONTROL TO IDLE LOOP IN DUMMYJOB
#
# GOPROG      HARDWARE RESTART
#
#      0.      EXECUTE STARTSUB
#      1.      TRANSFER CONTROL TO DOFSTART IF ANY OF THE FOLLOWING CONDITIONS
#                EXIST.
#                A.      RESTART OCCURRED DURING EXECUTION OF ERASCHK.
#                B.      BOTH OSCILLATOR FAIL AND AGC WARNING ARE ON.
#                C.      MARK REJECT AND EITHER NAV OR MAIN DSKY ERROR LIGHT RESET
#                        ARE ON.
#      2.      SCHEDULE A T5RUPT PROGRAM FOR THE DAP
#      3.      SET FLAGWRD5 BITS FOR INTWAKE ROUTINE
#      4.      EXTINGUISH ALL DSKY LAMPS, EXCEPT FOR PROGRAM ALARM, GIMBAL LOCK, AND
#                NO ATT
#      5.      INITIALIZE IMU FLAGS
#      6.      IF ENGINE COMMAND IS ON (FLAGWRD5, BIT 7), SET ENGINE ON (CHANNEL
#                11, BIT 13).
#      7.      TRANSFER CONTROL TO GOPROG3
#
# ENEMA       SOFTWARE RESTART -- INITIATED BY MAJOR MODE CHANGE
#
#      1.      EXECUTE STARTSB2
#      2.      KILL PROGRAMS THAT WERE INTEGRATING OR WAITING FOR INTEGRATION
#                ROUTINE
#      3.      TRANSFER CONTROL TO GOPROG3
#
# GOPROG3     SUBROUTINE COMMON TO GOPROG AND ENEMA
#
#      1.      TEST PHASE TABLES -- IF INCORRECT, DISPLAY ALARM 1107 AND
#                TRANSFER CONTROL TO DOFSTART
#      2.      DISPLAY MAJOR MODE
#      3.      IF ANY GROUPS WERE ACTIVE UPON RESTART, TRANSFER CONTROL TO THE
#
# Page 182    RESTARTS SUBROUTINE TO RESCHEDULE PENDING TASKS, LONGCALLS, AND
#                JOBS (P20 IS RESTARTED VIA FINDVAC)
#      4.      IF NO GROUPS WERE ACTIVE UPON RESTART, DISPLAY ALARM CODE
#                1110 (RESTART WITH NO ACTIVE GROUPS)

```

```

#      5.      TRANSFER CONTROL TO IDLE LOOP IN DUMMYJOB
#
# STARTSUB      SUBROUTINE COMMON TO SLAP1 AND GOPROG
#
#      1.      CLEAR OUTBIT CHANNELS 5 AND 6
#      2.      INITIALIZE TIMES5, TIME4, TIME3
#      3.      TRANSFER CONTROL TO STARTSB2
#
# STARTSB2      SUBROUTINE COMMON TO STARTSUB AND ENEMA
#
#      1.      INITIALIZE OUTBIT CHANNELS 11,12,13, AND 14
#      2.      REPLACE ALL TASKS ON WAITLIST WITH ENDTASK
#      3.      MAKE ALL EXECUTEVE REGISTERS AVAILABLE
#      4.      MAKE ALL VAC AREAS AVAILABLE
#      5.      CLEAR DSKY REGISTERS
#      6.      ZERO NUMEROUS SWITCHES
#      7.      INITIALIZE OPTICS FLAGS
#      8.      INITIALIZE PIPA AND TELEMETRY FAIL FLAGS
#      9.      INITIALIZE DOWN TELEMETRY
#
# INPUT/OUTPUT INITIALIZATION
#
#      A.      CALLING SEQUENCE
#
#              SLAP1 --          TC POSTJUMP          OR          VERB 36,ENTER
#                               CADR SLAP1
#
#              ENEMA --         TC POSTJUMP          *** DO NOT CALL ENEMA WITHOUT ***
#                               CADR ENEMA          *** CONSULTING POOH PEOPLE ***
#
#      B.      OUTPUT
#
#              ERASABLE MEMORY INITIALIZATION
#
# PROGRAM ANALYSIS
#
#      A.      SUBROUTINES CALLED
#
#              MR.KLEAN, WAITLIST, DSPMM, ALARM, RESTARTS, FINDVAC
#
#      B.      ALARMS
#
#              1107    PHASE TABLE ERROR
#              1110    RESTART WITH NO ACTIVE GROUPS

```

	BANK	10	
	SETLOC	FRANDRES	
	BANK		
	EBANK=	LST1	
	COUNT	05/START	
SLAP1	INHINT		# FRESH START. COMES HERE FROM PINBALL.
	TC	STARTSUB	# SUBROUTINE DOES MOST OF THE WORK.
STARTSW	TCF	SKIPSIM	# PATCH...TCF STARTSIM...FOR SIMULATION
STARTSIM	CAF	BIT14	
	TC	FINDVAC	
SIM2CADR	OCT	77777	# PATCH 2CADR (AND EBANK DESIGNATION) OF
	OCT	77777	# SIMULATION START ADDRESS.
SKIPSIM	CA	DSPTAB +11D	
	MASK	BITS4&6	
	AD	BIT15	
	TS	DSPTAB +11D	# REQUESTED FRESH START.
	CA	ZERO	# SAME STORY ON ZEROING FAILREG.
	TS	ERCOUNT	
	TS	FAILREG	
	TS	FAILREG +1	
	TS	FAILREG +2	
	TS	REDOCTR	
	CS	PRI012	
	TS	DSRUPTSW	
DOFSTART	CAF	ZERO	# DO A FRESH START.
	TS	ERESTORE	# ***** MUST NOT BE REMOVED FROM DOFSTART
	TS	SMODE	# ***** MUST NOT BE REMOVED FROM DOFSTART
	TS	UPSVFLAG	# UPDATE STATE VECTOR REQUEST FLAGWORD
	EXTEND		
	WRITE	CHAN5	# TURN OFF RCS JETS
	EXTEND		
	WRITE	CHAN6	# TURN OFF RCS JETS
	EXTEND		
	WRITE	DSALMOUT	# ZERO CHANNEL 11
	EXTEND		
	WRITE	CHAN12	# ZERO CHANNEL 12
	EXTEND		
	WRITE	CHAN13	# ZERO CHANNEL 13

July 12, 2016

601

Page 184

```
EXTEND
WRITE  CHAN14      # ZERO CHANNEL 14
TS     WTOPTION
TS     DNLSTCOD

TS     NVSAVE
TS     EBANKTEM
TS     RATEINDX
TS     TRKMKCNT
TS     VHFCNT
TS     EXTVBACT

CS     DSPTAB +11D
MASK   BITS4&6
CCS    A
TC     +4
CA     BITS4&6
EXTEND
WOR    CHAN12      # THE IMU WAS IN COARSE ALIGN IN GIMBAL
TC     MR.KLEAN    # LOCK, SO PUT IT BACK INTO COARSE ALIGN.

CS     ZERO
TS     MODREG

CAF    PRI030
TS     RESTREG

CAF    IM30INIF     # FRESH START IMU INITIALIZATION.
TS     IMODES30

CAF    NEGONE
TS     OPTIND       # KILL COARSE OPTICS

CAF    OPTINITF
TS     OPTMODES

CAF    IM33INIT
TS     IMODES33

EXTEND
DCA    T5IDLER      # LET T5 IDLE.
DXCH   T5LOC

CA     SWINIT
TS     STATE
```

```

CA      FLAGWRD1
MASK    NOP01BIT      # LEAVE NODOP01 FLAG UNTOUCHED
AD      SWINIT +1
TS      FLAGWRD1

CA      SWINIT +2
TS      STATE +2

CA      FLAGWRD3

# Page 185
MASK    BIT13          # REFSMMAT FLAG
AD      SWINIT +3
TS      FLAGWRD3

EXTEND
DCA     SWINIT +4
DXCH    STATE +4
EXTEND
DCA     SWINIT +6
DXCH    STATE +6
CA      FLAGWRD8
MASK    OCT6200        # CMOONFLG, LMOONFLG, AND SUFFLAG
AD      SWINIT +8D
TS      FLAGWRD8

CA      SWINIT +9D
TS      STATE +9D

EXTEND
DCA     SWINIT +10D
DXCH    STATE +10D

ENDRSTRT TC      POSTJUMP
        CADR    DUMMYJOB +2      # PICKS UP AT RELINT.  (IN A SWITCHED BANK.)

MR.KLEAN INHINT
        EXTEND
        DCA     NEG0
        DXCH    -PHASE2

POOKLEAN EXTEND
        DCA     NEG0
        DXCH    -PHASE4
        EXTEND
        DCA     NEG0
        DXCH    -PHASE1

V37KLEAN EXTEND

```

July 12, 2016

603

```
DCA      NEG0
DXCH     -PHASE3
EXTEND
DCA      NEG0
DXCH     -PHASE5
EXTEND
DCA      NEG0
DXCH     -PHASE6
TC       Q
```

OCT6200 OCT 6200

Page 186

COMES HERE FROM LOCATION 4000, GOJAM, RESTART ANY PROGRAMS WHICH MAY HAVE BEEN RUNNING AT THE

```
GOPROG      INCR      REDOCTR      # ADVANCE RESTART COUNTER.

          LXCH      Q
          EXTEND
          ROR       SUPERBNK
          DXCH      RSBBQ
          TC       BANKCALL      # STORE ERASABLES FOR DEBUGGING PURPOSES.
          CADR      VAC5STOR
          CA       BIT15      # TEST OSC FAIL BIT TO SEE IF WE HAVE
          EXTEND      # HAD A POWER TRANSIENT. IF SO, ATTEMPT
          WAND      CHAN33      # A RESTART. IF NOT, CHECK THE PRESENT
          EXTEND      # STATE OF AGC WARNING BIT.
          BZF       BUTTONS

          CA       BIT14      # IF AGC WARNING ON (BIT = 0), DO A
          EXTEND      # FRESH START ON THE ASSUMPTION THAT
          RAND      CHAN33      # WE ARE IN A RESTART LOOP.
          EXTEND
          BZF       NONAVKEY +1

BUTTONS      TC       LIGHTSET      # MAKE FRESH START CHECKS BEFORE ERESTORE.

# ERASCHK TEMPORARILY STORES THE CONTENTS OF TWO ERASABLE LOCATIONS, X
# AND X+1 INTO SKEEP5 AND SKEEP6. IT ALSO STORES X INTO SKEEP7 AND
# ERESTORE. IF ERASCHK IS INTERRUPTED BY A RESTART, C(ERESTORE) SHOULD
# EQUAL C(SKEEP7), AND SHOULD BE A + NUMBER LESS THAN 2000 OCT. OTHERWISE
# C(ERESTORE) SHOULD EQUAL +0.
```

```
CAF      HI5
MASK     ERESTORE
EXTEND
```

```

BZF      +2          # IF ERESTORE NOT = +0 OR +N LESS THAN 2K,
TCF      NONAVKEY +1 # DOUBT E MEMORY AND DO A FRESH START
CS        ERESTORE
EXTEND
BZF      ELRSKIP -1
AD        SKEEP7
EXTEND
BZF      +2          # = SKEEP7, RESTORE E MEMORY.
TCF      NONAVKEY +1 # NOT = SKEEP7, DOUBT E MEM, DO FRESH START
CA        SKEEP4
TS        EBANK      # EBANK OF E MEMORY THAT WAS UNDER TEST.
EXTEND    # (NOT DXCH SINCE THIS MIGHT HAPPEN AGAIN)
DCA      SKEEP5
INDEX    SKEEP7
DXCH     0000        # E MEMORY RESTORED
CA        ZERO
TS        ERESTORE

# Page 187
ELRSKIP
TC        STARTSUB  # DO INITIALIZATION AFTER ERASE RESTORE.
CA        FLAGWRD6  # RESTART AUTOPILOTS
EXTEND
MP        BIT3      # BITS 15,14    00      T5IDLOC
MASK      SIX       #              01      REDORCS
EXTEND    #              10      REDOTVC
INDEX     A         #              11      REDOSAT
DCA      T5IDLER
DXCH     T5LOC

CS        INTFLBIT
MASK      RASFLAG
TS        RASFLAG

CA        OPTMODES
MASK      OPTINITR
AD        BIT7
TS        OPTMODES

CAF       BIT6
MASK      IMODES33
AD        IM33INIT
TS        IMODES33

CA        9,6,4     # LEAVE PROG ALARM, GIMBAL LOCK, NO ATT
MASK      DSPTAB +11D # LAMPS INTACT ON HARDWARE RESTART
AD        BIT15
XCH       DSPTAB +11D

```

July 12, 2016

605

	MASK	BIT4	# IF NO ATT LAMP WAS ON, LEAVE ISS IN
	EXTEND		# COARSE ALIGN
	BZF	NOCOARSE	
	TC	IBNKCALL	# IF NO ATT LAMP ON, RETURN ISS TO
	CADR	SETCOARS	# COARSE ALIGN.
	CAF	SIX	
	TC	WAITLIST	
	EBANK=	CDUIND	
	2CADR	CA+ECE	
NOCOARSE	CAF	IFAILINH	# LEAVE FAILURE INHIBITS INTACT ON
	MASK	IMODES30	# HARDWARE RESTART. RESET ALL
	AD	IM30INIR	# FAILURE CODES.
	TS	IMODES30	
	CS	FLAGWRD5	
	MASK	ENGONBIT	
	CCS	A	
	TCF	GOPROG3	
	CAF	BIT13	
	EXTEND		
# Page 188	WOR	DSALMOUT	# TURN ENGINE ON
	TCF	GOPROG3	
ENEMA	INHINT		
	TC	LIGHTSET	# EXIT TO DOFSTART IF ERROR RESET AND
	TC	STARTSB2	# MARK REJECT DEPRESSED SIMULTANEOUSLY
	CS	INTMASK	# RESET INTEGRATION BITS
	MASK	RASFLAG	
	TS	RASFLAG	
	CS	FLAGWRD6	# IS TVC ON
	MASK	OCT60000	
	EXTEND		
	BZMF	GOPROG3	# NO
	CAF	.5SEC	# YES, CALL TVCEXEC TASK WHICH WAS KILLED
	TC	WAITLIST	# IN STARTSB2
	EBANK=	BZERO	
	2CADR	TVCEXEC	
GOPROG3	CAF	NUMGRPS	# VERIFY PHASE TABLE AGREEMENTS
PCLLOOP	TS	MPAC +5	
	DOUBLE		

```

EXTEND
INDEX    A
DCA      -PHASE1      # COMPLEMENT INTO A, DIRECT INTO L.
EXTEND
RXOR     LCHAN        # RESULT MUST BE -0 FOR AGREEMENT.
CCS      A
TCF      PTBAD        # RESTART FAILURE.
TCF      PTBAD
TCF      PTBAD

CCS      MPAC +5      # PROCESS ALL RESTART GROUPS.
TCF      PCLOOP

TS       MPAC +6      # SET TO +0.
TC       MMDSPRAY    # DISPLAY MAJOR MODE

INHINT                                # RELINT DONE IN MMDSPRAY

CAE      FLAGWRD6     # IS RCS DAP RUNNING (BITS 15 14 OF
MASK     OCT60000     # FLAGWORD6 = 01)
EXTEND                                # YES, DO STOPRATE
BZMF     NXTRST -1    # NO, SKIP TO NXTRST -1
CAF      EBANK6       # STOPRATE IS DONE IN EBANK 6
TS       EBANK
TC       STOPRATE     # ZERO DELCDUS, WBODYS, AND BIASES THUS
                                # STOPPING AUTOMATIC MANEUVERING

CAF      EBANK3
TS       EBANK

# Page 189
NXTRST   CAF          NUMGRPS      # SEE IF ANY GROUPS RUNNING
          TS          MPAC +5
          DOUBLE
          INDEX       A
          CCS         PHASE1
          TCF         PACTIVE      # PNZ -- GROUP ACTIVE.
          TCF         PINACT       # +0 -- GROUP NOT RUNNING.

PACTIVE  TS          MPAC
          INCR        MPAC        # ABS OF PHASE.
          INCR        MPAC +6     # INDICATE GROUP DEMANDS PRESENT.
          CA          RACTCADR
          TC          SWCALL      # MUST RETURN TO SWRETURN.

PINACT   CCS         MPAC +5      # PROCESS ALL RESTART GROUPS.
          TCF         NXTRST

```

```

                CCS      MPAC +6          # NO, CHECK PHASE ACTIVITY FLAG
                TCF      ENDRSTRT        # PHASE ACTIVE
                CAF      BIT15          # IS MODE -0
                MASK      MODREG
                EXTEND
                BZF      GOTOPOOH        # NO
                TCF      ENDRSTRT        # YES
PTBAD          TC       ALARM          # SET ALARM TO SHOW PHASE TABLE FAILURE.
                OCT      1107

                TCF      DOFSTART        # IN R21.

# *****

# DO NOT USE GOPROG2 OR ENEMA WITHOUT CONSULTING POOH PEOPLE.

GOPROG2        EQUALS  ENEMA
OCT10000       =      BIT13
OCT30000       =      PRI030
OCT7777        OCT    7777
RACTCADR       CADR   RESTARTS

LIGHTSET       CAF     BIT7            # DOFSTART IF MARK REJECT AND EITHER
                EXTEND                    # ERROR LIGHT RESET BUTTONS ARE DEPRESSED
                RAND    NAVKEYIN
                EXTEND
                BZF     NONAVKEY        # NO MARK REJECT
                CAF     OCT37
                EXTEND
                RAND    NAVKEYIN        # NAV DSKY KEYCODES, MARK, MARK REJECT
                AD      -ELR
                EXTEND
                BZF     NONAVKEY +1
                EXTEND

# Page 190
                READ    MNKEYIN        # MAIN DSKY KEYCODES
                AD      -ELR
                EXTEND
                BZF     +2

NONAVKEY       TC      Q

                TC      STARTSUB
                TCF     DOFSTART
STARTSUB       CAF     LDNPAS1        # SET POINTER SO NEXT 20MS DOWNRUPT WILL

```

	TS	DNTMGOTO	# CAUSE THE CURRENT DOWNLIST TO BE # INTERRUPTED AND START SENDING FROM THE # BEGINNING OF THE CURRENT DOWNLIST.
	CAF	POSMAX	
	TS	TIME3	# 37777 TO TIME3.
	AD	MINUS2	
	TS	TIME4	# 37775 TO TIME4.
	AD	NEGONE	
	TS	TIME5	# 37774 TO TIME5.
STARTSB2	CAF	OCT77603	# TURN OFF UPLINK ACTY, TEMP CAUTION, KR,
	EXTEND		# FLASH, OP. ERROR, LEAVE OTHERS UNCHANGED.
	WAND	DSALMOUT	
	CAF	OCT74777	# TURN OFF TEST ALARMS, STANDBY ENABLE.
	EXTEND		
	WAND	CHAN13	
	CS	PRI025	# CLEAR R21MARK, P21FLAG, AND SKIPVHF BIT.
	MASK	FLAGWRD2	
	AD	SKIPVBIT	# NOW SET SKIPVHF FLAG.
	TS	FLAGWRD2	
	EBANK=	LST1	
	CAF	STARTEB	
	TS	EBANK	# SET FOR E3
	CAF	NEG1/2	# INITIALIZE WAITLIST DELTA-TS.
	TS	LST1 +7	
	TS	LST1 +6	
	TS	LST1 +5	
	TS	LST1 +4	
	TS	LST1 +3	
	TS	LST1 +2	
	TS	LST1 +1	
	TS	LST1	
	CS	ENDTASK	
	TS	LST2	
	TS	LST2 +2	
	TS	LST2 +4	
# Page 191	TS	LST2 +6	
	TS	LST2 +8D	
	TS	LST2 +10D	
	TS	LST2 +12D	
	TS	LST2 +14D	
	TS	LST2 +16D	

	CS	ENDTASK +1	
	TS	LST2 +1	
	TS	LST2 +3	
	TS	LST2 +5	
	TS	LST2 +7	
	TS	LST2 +9D	
	TS	LST2 +11D	
	TS	LST2 +13D	
	TS	LST2 +15D	
	TS	LST2 +17D	
	CS	ZERO	# MAKE ALL EXECUTIVE REGISTER SETS
	TS	PRIORITY	# AVAILABLE.
	TS	PRIORITY +12D	
	TS	PRIORITY +24D	
	TS	PRIORITY +36D	
	TS	PRIORITY +48D	
	TS	PRIORITY +60D	
	TS	PRIORITY +72D	
	TS	DSRUPTSW	
	TS	NEWJOB	# SHOWS NO ACTIVE JOBS.
	CAF	VAC1ADRC	# MAKE ALL VAC AREAS AVAILABLE.
	TS	VAC1USE	
	AD	LTHVACA	
	TS	VAC2USE	
	AD	LTHVACA	
	TS	VAC3USE	
	AD	LTHVACA	
	TS	VAC4USE	
	AD	LTHVACA	
	TS	VAC5USE	
	CAF	TEN	# BLANK DSKY REGISTERS (PROGRAM, VERB, NOUN, # R1, R2, R3)
DSPOFF	TS	MPAC	
	CS	BIT12	
	INDEX	MPAC	
	TS	DSPTAB	
	CCS	MPAC	
	TCF	DSPOFF	
	TS	DELAYLOC	
# Page 192	TS	DELAYLOC +1	

	TS	DELAYLOC +2	
	TS	DELAYLOC +3	
	TS	R1SAVE	
	TS	INLINK	
	TS	DSPCNT	
	TS	CADRSTOR	
	TS	REQRET	
	TS	CLPASS	
	TS	DSPLOCK	
	TS	MONSAVE	# KILL MONITOR
	TS	MONSAVE1	
	TS	VERBREG	
	TS	NOUNREG	
	TS	DSPLIST	
	TS	MARKSTAT	
	TS	IMUCADR	
	TS	OPTCADR	
	TS	RADCADR	
	TS	ATTCADR	
	TS	LGYRO	
	TS	FLAGWRD4	# KILL INTERFACE DISPLAYS
	CAF	NOUTCON	
	TS	NOUT	
	CAF	BIT14	
	MASK	EXTVBACT	
	TS	EXTVBACT	
	CAF	LECHK	# SELF CHECK GO-TO REGISTER.
	TS	SELFRET	
	CS	VD1	
	TS	DSPCOUNT	
	TC	Q	
T5IDLOC	CA	L	# T5RUPT COMES HERE EVERY 163.84 SECS
	TCF	NOQRSM +1	# WHEN NOBODY IS USING IT.
T5IDLER	EBANK=	OGANOW	
	2CADR	T5IDLOC	
	EBANK=	OGANOW	
	2CADR	REDORCS	
	EBANK=	OGANOW	
	2CADR	REDOTVC	

July 12, 2016

611

EBANK= OGANOW
2CADR REDOSAT
Page 193

IFAILINH OCT 435
LDNPHAS1 GENADR DNPHASE1
LESCHK GENADR SELFCHK
VAC1ADRC ADRES VAC1USE
LTHVACA DEC 44

INTMASK OCT 20100
OCT77603 OCT 77603
OCT74777 OCT 74777
STARTEB ECADR LST1
NUMGRPS EQUALS FIVE
-ELR OCT -22
IM30INIF OCT 37411
IM30INIR OCT 37000
IM33INIT = PRI016
9,6,4 OCT 450
OPTINITF OCT 130
OPTINITR OCT 430
SWINIT OCT 0
OCT 0
OCT 0
OCT 0

-ERROR LIGHT RESET KEY CODE.
INHIBITS IMU FAIL FOR 5 SEC AND PIP ISSW
NO PIP OR TM FAIL SIGNALS.

OCT 0
OCT 00200
OCT 0
OCT 0
OCT 0
OCT 0
OCT 0
OCT 0
OCT 0

Page 194
PROGRAM NAME GOTOP00H ASSEMBLY SUNDISK
LOG SECTION FRESH START AND RESTART

FUNCTIONAL DESCRIPTION

1. DISPLAY MAJOR MODE NUMBER 00 IN DSKY REGISTER R1 AND R3. FLASH V50N07 ON DSKY.
2. PERMIT A CURRENT PENDING REQUEST (FLASH ON DSKY) TO BE REPLACED (WITHOUT AN ABC
CHANGE REQUEST.
#

```

# INPUT/OUTPUT INFORMATION
#
#       A. CALLING SEQUENCE           TC GOTOP00H
#       B. ERASABLE INITIALIZATION   NONE
#       C. OUTPUT                     FLASH V 50 NOUN 07 ON DSKY
#       D. DEBRIS                     L
#
# PROGRAM ANALYSIS
#
#       A. SUBROUTINES CALLED         GOPERF3, LINUS
#       B. NORMAL EXIT                TCF ENDOFJOB
#       C. ALARM AND ABORT EXITS      NONE
#
#       BLOCK      02
#       SETLOC     FFTAG10
#       BANK
#
#       COUNT     02/P00
#
# GOTOP00H        TC      PHASCHNG      # RESTART GOTOP00H
#                  OCT     14
#
#                  TC      POSTJUMP
#                  CADR     GOP00FIX
#                  BANK     10
#                  SETLOC   VERB37
#                  BANK
#
#                  COUNT   04/P00
#
# GOP00FIX        TC      INITSUB
#                  TC      CLEARMRK +2
#                  CAF     V37N99
#                  TC      BANKCALL
#                  CADR     GOFLASH
#                  TCF     -3
#
# # Page 195
#                  TCF     -4
#                  TCF     -5
#
# V37N99          VN      3799
#
# # Page 196
# # PROGRAM NAME      V37              ASSEMBLY SUNDISK
#
# # LOG SECTION       FRESH START AND RESTART
#

```

FUNCTIONAL DESCRIPTION

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

1. CHECK IF NEW PROGRAM ALLOWED. IF BIT 1 OF FLAGWRD2 (NODOFLAG) IS SET, AN ALARM 1520
2. CHECK FOR VALIDITY OF PROGRAM SELECTED. IF AN INVALID PROGRAM IS SELECTED, THE OPER
- SET AND CURRENT ACTIVITY, IF ANY, CONTINUE.
3. SERVICER IS TERMINATED IF IT HAS BEEN RUNNING.
4. INSTALL IS EXECUTED TO AVOID INTERRUPTING INTEGRATION.
5. THE ENGINE IS TURNED OFF AND THE DAP IS INITIALIZED FOR COAST.
6. TRACK, UPDATE, AND TARG1 FLAGS ARE SET TO ZERO.
7. DISPLAY SYSTEM IS RELEASED.
8. THE FOLLOWING ARE PERFORMED FOR EACH OF THE THREE CASES.
 - A. PROGRAM SELECTED IS P00.
 1. RENDEZVOUS FLAG IS RESET. (KILL P20)
 2. STATINT1 IS SCHEDULED BY SETTING RESTART GROUP 2.
 3. MAJOR MODE 00 IS STORED IN THE MODE REGISTER (MODREG).
 4. SUPERBANK 3 IS SELECTED.
 5. NODOFLAG IS RESET.
 6. ALL RESTART GROUPS EXCEPT GROUP2 ARE CLEARED. CONTROL IS TRANSFERRED
 - WHICH CAUSES ALL CURRENT ACTIVITY TO BE DISCONTINUED AND A 9 MINUTE
 - INITIATED.
 - B. PROGRAM SELECTES IS P20.
 1. IF THE CURRENT MAJOR MODE IS THE SAME AS THE SELECTED NEWPROGRAM. T
 - VIA V37XEQ, ALL RESTART GROUPS, EXCEPT GROUP 4 ARE CLEARED.
 2. IF THE CURRENT MAJOR MODE IS NOT EQUAL TO THE NEW REQUEST, A CHECK I
 - ED MAJOR MODE HAS BEEN RUNNING THE BACKGROUND,
 - AND IF IT HAS, NO NEW PROGRAM IS SCHEDULED, THE EXISTING
 - P20 IS RESTARTED TO CONTINUE, AND ITS MAJORE MODE IS SET.
 3. CONTROL IS TRANSFERRED TO GOPROG2.
 - C. PROGRAM SELECTED IS NEITHER P00 NOR P20
 1. V37XEQ IS SCHEDULED (AS A JOB) BY SETTING RESTART GROUP 4
 2. ALL CURRENT ACTIVITY EXCEPT RENDEZVOUS AND TRACKING IS DISCONTINUED
 - GROUPS. GROUP 2 IS CLEARED. IF THE RENDEZVOUS FLAG IS ON P20 IS REST
 - TO CONTINUE.

INPUT/OUTPUT INFORMATION

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

A. CALLING SEQUENCE

CONTROL IS DIRECTED TO V37 BY THE VERBFAN ROUTINE.

VERBFAN GOES TO C(VERBTAB+C(VERBREG)). VERB 37 = MMCHANG.

MMCHANG EXECUTES A TC POSTJUMP, CADR V37.

B. ERASABLE INITIALIZATION

NONE

C. OUTPUT

MAJOR MODE CHANGE

```

#
#      D. DEBRIS
#          MMNUMBER, MPAC +1, MINDEX, BASETEMP +C(MINDEX), FLAGWRD0, FLAGWRD1, I
#          GOLOC, GOLOC +1, GOLOC +2, BASETEMP, -PHASE2, PHASE2, -PHASE4
#
# PROGRAM ANALYSIS
#
#      A. SUBROUTINES CALLED
#          ALARM, RELDSP, PINBRNCH, INTSTALL, ENGINOF2, ALLCOAST, V37KLEAN, GOF
#          DSPMM
#
#      B. NORMAL EXIT                                TC ENDOFJOB
#
#      C. ALARMS                                1520 (MAJOR MODE CHANGE NOT PERMITTED)

          BLOCK    02
          SETLOC   FFTAG10
          BANK

          COUNT    02/V37

OCT24      MM      20
OCT31      MM      25
          BANK     27
          SETLOC   VERB37
          BANK

          COUNT    04/V37

V37        TS      MMNUMBER      # SAVE MAJOR MODE
          CAF      PRI030        # RESTART AT PINBALL PRIORITY
          TS      RESTREG

          CA      IMODES30      # IS IMU BEING INITIALIZED
          MASK    BIT6
          CCS      A
          TCF     CANTROO

          CAF      BIT13        # IS ENGINE ON
          EXTEND
          RAND     DSALMOUT
          CCS      A
          TCF     ROOTOP00      # YES, SET UP FOR P00

          CS      FLAGWRD6      # NO, IS TVC DAP ON
          MASK    OCT60000

```

July 12, 2016

615

```
EXTEND
BZMF  ISITP00          # NO, CONTINUE WITH ROO

ROOTOP00  INHINT
CAF      EBANK6

# Page 198

TS      EBANK
EBANK=   DAPDATR1
CAE      CSMMASS
TS      MASSTMP
TC      IBNKCALL
CADR     SPSOFF
TC      IBNKCALL
CADR     MASSPROP
CAF      3.1SEC
TC      IBNKCALL
CADR     RCSDAPON +1

TC      IBNKCALL
CADR     TVCZAP          # DISABLE TVC
CAF      ZERO
TS      MMNUMBER
RELINT
CAF      FIVE
TC      BANKCALL
CADR     DELAYJOB
CAF      ZERO
EXTEND
WRITE    5
EXTEND
WRITE    6
ISITP00  CA      MMNUMBER
EXTEND
BZF      ISSERVON        # YES, CHECK SERVICER STATUS

CS      FLAGWRD2          # NO, IS NODO V37 FLAG SET
MASK     NODOBIT
CCS      A
TCF      CHECKTAB        # NO
CANTR00  TC      ALARM
OCT      1520

V37BAD   TC      RELDSP          # RELEASES DISPLAY FROM ASTRONAUT

TC      POSTJUMP          # BRING BACK LAST NORMAL DISPLAY IF THERE
CADR     PINBRNCH        # WAS ONE.  OY
```

CHECKTAB	CA	NOV37MM	# THE NO. OF MM
AGAINMM	TS	MPAC +1	
	NDX	MPAC +1	
	CA	PREMM1	# OBTAIN WHICH MM THIS IS FOR
	MASK	LOW7	
	COM		
	AD	MMNUMBER	
	CCS	A	
	CCS	MPAC +1	# IF GR, SEE IF ANY MORE IN LIST
# Page 199	TCF	AGAINMM	# YES, GET NEXT ONE
	TCF	V37NONO	# LAST TIME OR PASSED MM
	CA	MPAC +1	
	TS	MINDEX	# SAVE INDEX FOR LATER
ISSERVON	CS	FLAGWRD7	# V37 FLAG SET -- I.E., IS SERVICER C
	MASK	V37FLBIT	
	CCS	A	
	TCF	CANV37	# NO
	INHINT		
	CS	AVEGBIT	# YES TURN OFF AVERAGE G FLAG AND WA
	MASK	FLAGWRD1	# FOR SERVICER TO RETURN TO CANV37
	TS	FLAGWRD1	
	TCF	ENDOFJOB	
CANV37	CAF	R00AD	
	TS	TEMPFLSH	
	TC	PHASCHNG	
	OCT	14	
ROC	TC	INTPRET	
	CALL		# WAIT FOR INTEGRATION TO FINISH
		INTSTALL	
DUMMYAD	EXIT		
	CS	OCT1400	# CLEAR CAUTION RESET
	EXTEND		# AND TEST CONNECTOR OUTBIT
	WAND	11	
	CAF	OCT44571	# CLEAR ENABLE OPTICS ERROR COUNTER,

July 12, 2016

617

	EXTEND		# TRAKERS ON BIT, TVC ENABLE, ZERO OPTICS,
	WAND	12	# DISENGAGE OPTICS DAP, SIVB IN J SEQUENCE
			# START, AND SIVB CUTOFF BIT.
	CS	OCT600	# CLEAR UNUSED BITS
	EXTEND		
	WAND	13	
	TC	INITSUB	
	TC	CLEARMRK	
	TC	DOWNFLAG	
	ADRES	STIKFLAG	
# Page 200	TC	BANKCALL	
	CADR	UPACTOFF	# TURN OFF UPLINK ACTIV LIGHT
	TC	DOWNFLAG	
	ADRES	VHFRFLAG	
	TC	DOWNFLAG	
	ADRES	R21MARK	
	CCS	MMNUMBER	# IS THIS A POOH REQUEST
	TCF	NOUVEAU	# NO, PICK UP NEW PROGRAM
	COUNT	04/P00	
POOH	TC	RELDSP	# RELEASE DISPLAY SYSTEM
	CAF	PRI05	# SET VARIABLE RESTART REGISTER FOR P00.
	TS	PHSPRDT2	
	INHINT		
	CS	NODOBIT	# TURN OFF NODOFLAG.
	MASK	FLAGWRD2	
	TS	FLAGWRD2	
	CA	FIVE	# SET 2.5 RESTART FOR STATEINT1
	TS	L	
	COM		
	DXCH	-PHASE2	
	CS	BIT7-8	# RESET IMUSE + KILL P20 BY TURNING OFF
	MASK	FLAGWRD0	

	TS	FLAGWRDO	#	RENDFLG
	CAF	DNLADPOO		
	COUNT	04/V37		
SEUDOP00	TS	DNLSTCOD	#	SET UP APPROPRIATE DOWNLIST CODE
			#	(OLD ONE WILL BE FINISHED FIRST)
	CS	OCT01120	#	TURN OFF TRACK, TARG1, UPDATE FLAG
	TS	EBANKTEM		
	MASK	FLAGWRD1		
	TS	FLAGWRD1		
GROUPKIL	TC	IBNKCALL	#	KILL GROUPS 3,5,6
	CADR	V37KLEAN		
	CCS	MMNUMBER	#	IS IT POOH
	TCF	RENDV00	#	NO
# Page 201	TC	IBNKCALL		
	CADR	POOKLEAN	#	REDUNDANT EXCEPT FOR GROUP 4.
GOMOD	CA	MMNUMBER		
	TS	MODREG		
GOGOPROG	TC	POSTJUMP		
	CADR	GOPROG2		
RENDV00	CS	MMNUMBER	#	IS NEW PROG = 20
	AD	OCT24	#	20
	EXTEND			
	BZF	RENDNOO	#	YES
	TCF	POOFIZZ		
RENDNOO	CS	MMNUMBER		
	AD	MODREG		
	EXTEND			
	BZF	KILL20		
	CA	FLAGWRDO	#	IS RENDZV00 FLAG SET
	MASK	RNDVZBIT		
	CCS	A		
	TCF	STATQUO		
POOFIZZ	CAF	RNDVZBIT		
	MASK	FLAGWRDO		

July 12, 2016

619

KILL20	CCS	A	
	TCF	REV37	
	EXTEND		# NO, KILL GROUPS 1 + 2
	DCA	NEGO	
	DXCH	-PHASE1	
	EXTEND		
	DCA	NEGO	
	DXCH	-PHASE2	
REV37	CAF	V37QCAD	# SET RESTART POINT
	TS	TEMPFLSH	
	TCF	GOGOPROG	
STATQUO	CS	FLAGWRD1	# SET TRACKFLAG AND UPDATE FLAG
	MASK	OCT120	
	ADS	FLAGWRD1	
	EXTEND		# KILL GROUP 4
	DCA	NEGO	
	DXCH	-PHASE4	
# Page 202	TCF	GOMOD	
NOUVEAU	CAF	RNDVZBIT	
	MASK	FLAGWRDO	
	CCS	A	
	TCF	+3	
	TC	DOWNFLAG	# NO, RESET IMUINUSE FLAG
	ADRES	IMUSE	# BIT 8 FLAG 0
+3	INDEX	MINDEX	
	CAF	DNLADMM1	# OBTAIN NEW DOWNLIST ADDRESS
	INHINT		
	TCF	SEUDOP00	
V37NONO	TC	FALTON	# COME HERE IF MM REQUESTED DOESN'T EXIST
	TCF	V37BAD	
OCT00010	EQUALS	BIT4	
V37XEQ	INHINT		
	INDEX	MINDEX	
	CAF	PREMM1	

	TS	MMTEMP	# OBTAIN PRIORITY BITS 15-11
	TS	CYR	# SHIFT RIGHT TO BITS 14-10
	CA	CYR	
	MASK	PRI037	
	TS	PHSPRDT4	# PRESET GROUP 4 RESTART PRIORITY
	TS	NEWPRIO	# STORE PRIO FOR SPVAC
	CA	MMTEMP	# OBTAIN EBANK -- BITS 8, 9, 10 OF M
	EXTEND		
	MP	BIT8	
	MASK	LOW3	
	TS	L	
	INDEX	MINDEX	
	CAF	FCADRMM1	
	TS	BASETEMP	
	MASK	HI5	
	ADS	L	
	CA	BASETEMP	# OBTAIN GENADR PORTION OF 2CADR.
	MASK	LOW10	
	AD	BIT11	
	TC	SPVAC	
V37XEQC	CA	MMTEMP	# UPON RETURN FROM FINDVAC PLACE THE
	MASK	LOW7	# NEW MM IN MODREG (THE LOW 7 BITS OF
	TC	NEWMODEA	# PHSBRDT1)
# Page 203			
# FOR SUNDISK ONLY			
	TC	RELDSP	# RELEASE DISPLAY
	TC	ENDOFJOB	# AND EXIT
INITSUB	EXTEND		
	QXCH	MPAC +1	
	CAF	EBANK6	# SET E6 FOR DEADBAND CODING
	TS	EBANK	# WILL BE RESET IN STARTSB2
	INHINT		
	TC	STOPRATE	
	CA	FLAGWRD9	# RESTORE DEADBAND
	MASK	MAXDBBIT	
	CCS	A	
	TCF	SETMAXER	# MAX DE SELECTED

July 12, 2016

621

	TC	BANKCALL	# MIN DE SELECTED
	CADR	SETMINDB	
	TCF	RAKE	
SETMAXER	TC	BANKCALL	
	CADR	SETMAXDB	
RAKE	CAF	ELEVEN	# THIS PART CLEARS FLAGWORD BITS.
+1	TS	MPAC	# LOOP COMES HERE
	INDEX	MPAC	
	CS	FLAGTABL	
	INDEX	MPAC	
	MASK	FLAGWRDO	
	INDEX	MPAC	# PUT REVISED FLAGWORD BACK.
	TS	FLAGWRDO	
	CCS	MPAC	
	TCF	RAKE +1	# GET THE NEXT FLAGWORD
	RELINT		
	TC	UPFLAG	# NOW SET IMPULSW
	ADRES	IMPULSW	
	CA	NEGONE	
	TS	OPTIND	
	TC	MPAC +1	# RETURN FROM INITSUB
FLAGTABL	OCT	0	
	OCT	00040	# IDLEFAIL
	OCT	06000	# P21FLAG, STEERSW
	OCT	0	
	OCT	0	
	OCT	04140	# V59FLAG, ENGONFLG, 3AXISFLG
	OCT	10000	# STRULLSW
	OCT	16000	
	OCT	0	
# Page 204	OCT	42000	# SWTOVER, V94FLAG
	OCT	0	
	OCT	0	
	SETLOC	VAC5LOC	
	BANK		
VAC5STOR	CA	ZERO	# INITIALIZE INDEX REGISTERS
	TS	ITEMP1	
	TS	ITEMP2	
V5LOOP1	EXTEND		# LOOP TO STORE LOCS, BANKSETS, AND PRIOS.

	INDEX	ITEMP1	
	DCA	LOC	
	INDEX	ITEMP2	
	DXCH	VAC5	
	INDEX	ITEMP1	
	CA	PRIORITY	
	INDEX	ITEMP2	
	TS	VAC5 +2	
	CS	ITEMP2	# HAVE WE STORED THEM ALL?
	AD	EIGHTEEN	
	EXTEND		
	BZF	V5OUT1	# YES, GET PHASE INFORMATION.
	CA	TWELVE	# NO, INCREMENT INDEXES AND LOOP.
	ADS	ITEMP1	
	CA	THREE	
	ADS	ITEMP2	
	TCF	V5LOOP1	
V5OUT1	EBANK=	PHSNAME1	
	CA	EBANK3	# PHSNAME REGISTERS ARE IN EBANK3.
	TS	EBANK	
	CA	ELEVEN	# GET PHASE 2CADRS.
	TC	GENTRAN	
	ADRES	PHSNAME1	
	ADRES	VAC5 +21D	
	CA	ZERO	# NOW INITIALIZE INDEXES AGAIN.
	TS	ITEMP1	
	TS	ITEMP2	
V5LOOP2	INDEX	ITEMP1	# LOOP TO GET PHASE TABLES.
	CA	PHASE1	
	INDEX	ITEMP2	
	TS	VAC5 +33D	
# Page 205			
	CS	ITEMP2	# DO WE HAVE THEM ALL?
	AD	FIVE	
	EXTEND		
	BZF	V5OUT2	# YES, GO FINISH UP.
	CA	TWO	# NO, INCREMENT INDEXES AND LOOP.

July 12, 2016

623

	ADS	ITEMP1	
	INCR	ITEMP2	
	TCF	V5LOOP2	
V5OUT2	CA	MPAC +3	
	TS	VAC5 +39D	
	EXTEND		
	DCA	NEWLOC	
	DXCH	VAC5 +40D	
	CA	NEWJOB	
	TS	VAC5 +22D	
	CA	NEWPRIO	
	TS	VAC5 +26D	
	TC	SWRETURN	
EIGHTEEN	OCT	22	
	SETLOC	VERB37	
	BANK		
NEG7	EQUALS	OCT77770	
OCT44571	OCT	44571	# CONSTANTS TO CLEAR CHANNEL BITS IN V37
OCT600	OCT	600	
	EBANK=	PACTOFF	
POODAPAD	2CADR	T5IDLOC	
MMTEMP	EQUALS	PHSPRDT3	
BASETEMP	EQUALS	TBASE4	
BIT7-8	OCT	300	
OCT01120	OCT	01120	
V37QCAD	CADR	V37XEQ +3	
R00AD	CADR	DUMMYAD	
	EBANK=	DAPDATR1	
RCSADDR4	2CADR	RCSATT	
3.1SEC	OCT	37312	# 2.5 + 0.6 SEC

FOR VERB 37 TWO TABLES ARE MAINTAINED. EACH TABLE HAS AN ENTRY FOR EACH
MAJOR MODE THAT CAN BE STARTED FROM THE KEYBOARD. THE ENTRIES ARE PUT
INTO THE TABLE WITH THE ENTRY FOR THE HIGHEST MAJOR MODE COMING FIRST,
Page 206

TO THE LOWEST MAJOR MODE WHICH IS THE LAST ENTRY IN EACH TABLE.

#

THE FCADRM TABLE CONTAINS THE FCADR OF THE STARTING JOB OF
THE MAJOR MODE. FOR EXAMPLE,

#

#	FCADRM1	FCADR	P79	# START OF P 79
#		FCADR	PROG18	# START OF P 18
#		FCADR	P01	# START OF P 01

FCADRM1

EQUALS

FCADR	P79
FCADR	P78
FCADR	P77
FCADR	P76
FCADR	P75
FCADR	P74
FCADR	P73
FCADR	P72
FCADR	P62
FCADR	P61
FCADR	P54
FCADR	P53
FCADR	PROG52
FCADR	P51
FCADR	P47CSM
FCADR	P41CSM
FCADR	P40CSM
FCADR	P39
FCADR	P38
FCADR	P37
FCADR	P35
FCADR	P34
FCADR	P33
FCADR	P32
FCADR	P31
FCADR	P30
FCADR	P23
FCADR	PROG22
FCADR	PROG21
FCADR	PROG20
FCADR	P17
FCADR	P06
FCADR	GTSCPSS1

GYROCOMPASS STANDARD LEAD 1

THE PREMM TABLE CONTAINS THE E-BANK, MAJOR MODE, AND PRIORITY
INFORMATION, IT IS IN THE FOLLOWING FORM,


```

#
#       PPP PPE EEM MMM MMM
#
# Page 207
#       WHERE THE       7 M BITS CONTAIN THE MAJOR MODE NUMBER
#                       3 E BITS CONTAIN THE E-BANK NUMBER
#                       5 P BITS CONTAIN THE PRIORITY AT WHICH THE JOB IS
#                       TO BE STARTED
#
#       FOR EXAMPLE,
#
#           PREMM1           OCT       67213           # PRIORITY       33
#                               # E-BANK       5
#                               # MAJOR MODE    11
#                               OCT       25437           # PRIORITY       12
#                               # E-BANK       6
#                               # MAJOR MODE    31

PREMM1      EQUALS
OCT      27117           # MM 79           EBANK 4           PRIO 13
OCT      27116           # MM 78           EBANK 4           PRIO 13 (original says
OCT      27115           # MM 77           EBANK 4           PRIO 13
OCT      27714           # MM 76           EBANK 4           PRIO 13
OCT      27113           # MM 75           EBANK 4           PRIO 13
OCT      27112           # MM 74           EBANK 4           PRIO 13
OCT      27111           # MM 73           EBANK 4           PRIO 13
OCT      27110           # MM 72           EBANK 4           PRIO 13
OCT      27476           # MM 62           EBANK 6           PRIO 13
OCT      27475           # MM 61           EBANK 6           PRIO 13
OCT      27266           # MM 54           EBANK 5           PRIO 13
OCT      27265           # MM 53           EBANK 5           PRIO 13
OCT      27264           # MM 52           EBANK 5           PRIO 13
OCT      27263           # MM 51           EBANK 5           PRIO 13
OCT      27657           # MM 47           EBANK 7           PRIO 13
OCT      27451           # MM 41           EBANK 6           PRIO 13
OCT      27450           # MM 40           EBANK 6           PRIO 13
OCT      27047           # MM 39           EBANK 4           PRIO 13
OCT      27046           # MM 38           EBANK 4           PRIO 13
OCT      27645           # MM 37           EBANK 7           PRIO 13
OCT      27043           # MM 35           EBANK 4           PRIO 13
OCT      27042           # MM 34           EBANK 4           PRIO 13
OCT      27041           # MM 33           EBANK 4           PRIO 13
OCT      27040           # MM 32           EBANK 4           PRIO 13
OCT      27637           # MM 31           EBANK 7           PRIO 13
OCT      27636           # MM 30           EBANK 7           PRIO 13
OCT      27227           # MM 23           EBANK 5           PRIO 13

```

OCT	27226	# MM 22	EBANK 5	PRI0 13
OCT	27025	# MM 21	EBANK 4	PRI0 13
OCT	27424	# MM 20	EBANK 6	PRI0 13
OCT	27021	# MM 17	EBANK 6	PRI0 13
OCT	27006	# MM 06	EBANK 4	PRI0 13
OCT	41201	# MM 01	EBANK 5	PRI0 20

Page 208

THE FOLLOWING LIST IS FOR THE PURPOSE OF VERIFYING THAT THE EBA

EBANK= TIG	# EBANK SETTING REQUIRED BY MM 76
EBANK= KT	# EBANK SETTING REQUIRED BY MM 75
EBANK= SUBEXIT	# EBANK SETTING REQUIRED BY MM 74
EBANK= SUBEXIT	# EBANK SETTING REQUIRED BY MM 73
EBANK= SUBEXIT	# EBANK SETTING REQUIRED BY MM 72
EBANK= AOG	# EBANK SETTING REQUIRED BY MM 62
EBANK= AOG	# EBANK SETTING REQUIRED BY MM 61
EBANK= BESTI	# EBANK SETTING REQUIRED BY MM 54
EBANK= STARIND	# EBANK SETTING REQUIRED BY MM 53
EBANK= BESTI	# EBANK SETTING REQUIRED BY MM 52
EBANK= STARIND	# EBANK SETTING REQUIRED BY MM 51
EBANK= P40TMP	# EBANK SETTING REQUIRED BY MM 47
EBANK= DAPDATR1	# EBANK SETTING REQUIRED BY MM 41
EBANK= KMPAC	# EBANK SETTING REQUIRED BY MM 40
EBANK= KT	# EBANK SETTING REQUIRED BY MM 35
EBANK= SUBEXIT	# EBANK SETTING REQUIRED BY MM 34
EBANK= SUBEXIT	# EBANK SETTING REQUIRED BY MM 33
EBANK= SUBEXIT	# EBANK SETTING REQUIRED BY MM 32
EBANK= +MGA	# EBANK SETTING REQUIRED BY MM 30
EBANK= LANDMARK	# EBANK SETTING REQUIRED BY MM 23
EBANK= MARKINDX	# EBANK SETTING REQUIRED BY MM 22
EBANK= WHOCARES	# EBANK SETTING REQUIRED BY MM 21
EBANK= ESTROKER	# EBANK SETTING REQUIRED BY MM 20
EBANK= TIME2SAV	# EBANK SETTING REQUIRED BY MM 06
EBANK= QPLACE	# EBANK SETTING REQUIRED BY MM 01

NOTE: THE FOLLOWING CONSTANT IS THE NUMBER OF ENTRIES IN EACH OF
 # ----- THE ABOVE LISTS-1 (I.E., THE NUMBER OF MAJOR MODES (EXCEPT P00)
 # THAT CAN BE CALLED FROM THE KEYBOARD MINUS ONE)

EPREMM1	EQUALS	# END OF PREMM1 TABLE
	SETLOC PREMM1	# THIS CODING WILL AUTOMATICALLY CHANGE
NO.MMS	=MINUS EPREMM1	# THE "NOV37MM" CONSTANT AS ENTRIES ARE
	SETLOC VERB37	# INSERTED(IN) OR DELETED(FROM) THE
	BANK	# "PREMM1" TABLE.

July 12, 2016

627

NOV37MM ADRES NO.MMS -1 # ITEMS IN "PREMM1" TABLE - 1. *DON'T MOVE*

DNLADMM1 EQUALS
 ADRES RENDEZVU # P79
 ADRES RENDEZVU # P78
 ADRES RENDEZVU # P77
 ADRES RENDEZVU # P76
 ADRES RENDEZVU # P75
 ADRES RENDEZVU # P74
 ADRES RENDEZVU # P73

Page 209

 ADRES RENDEZVU # P72
 ADRES ENTRYUPD # P62
 ADRES POWERED # P61
 ADRES COSTALIN
 ADRES COSTALIN
 ADRES COSTALIN # P52
 ADRES COSTALIN # P51
 ADRES POWERED # P47
 ADRES POWERED # P41
 ADRES POWERED # P40
 ADRES RENDEZVU # P39
 ADRES RENDEZVU # P38
 ADRES RENDEZVU # P37
 ADRES RENDEZVU # P35
 ADRES RENDEZVU # P34
 ADRES RENDEZVU # P33
 ADRES RENDEZVU # P32
 ADRES RENDEZVU
 ADRES RENDEZVU # P30
 ADRES RENDEZVU # P23
 ADRES P22DNLST # P22
 ADRES RENDEZVU # P21
 ADRES RENDEZVU # P20
 ADRES RENDEZVU # P17
 ADRES COSTALIN # P06
 ADRES COSTALIN # P01

DNLADPOO = ZERO
COSTALIN = 0
ENTRYUPD = 1
RENDEZVU = 2
POWERED = 3

P22DNLST = 4

ORBITAL INTEGRATION CONSTANTS

THESE CONSTANTS ARE USED IN COMPUTING THE SETTING OF MIDFLAG.

RMM 2DEC 2538.09 E3 B-27 # 800 KM ABOVE LUNAR SURFACE

RME 2DEC 7178165 B-29 # 800 KM ABOVE EQ. RADIUS

BANK 13
SETLOC INTINIT
BANK

COUNT* \$\$/INTIN

EBANK= RRECTCSM

STATEUP SET BOF # EXTRAPOLATE CM STATE VECTOR
 VINTFLAG

Page 210

 ORBFLAG # ALSO 6X6 W-MATRIX IF VALID
 +3 # FOR ORBITAL NAVIGATION

SET DIMOFLAG

CLEAR CALL
 PRECIFLG
 INTEGRV

BON DLOAD
 SURFFLAG
 STATEND
 TETCSM

STCALL TDEC1
 INTSTALL

CLEAR CALL # EXTRAPOLATE LM STATE VECTOR
 VINTFLAG

 SETIFLGS # AND 6X6 W-MATRIX IF VALID

BOF SET # FOR RENDEZVOUS NAVIGATION
 RENWFLG
 +2

 DIMOFLAG
SET CALL
 PRECIFLG
 INTEGRV

STATEND CLRGO
 NODOFLAG
 ENDINT

July 12, 2016

629

```
# THIS VINT IS CALLED BY MIDTOAV1 AND 2
```

```
THISVINT      SET      RVQ  
                VINTFLAG
```

This code is written to file `src/FRESH-START-AND-RESTART.s`.

B.35 GIMBAL LOCK AVOIDANCE

```

630  <src/GIMBAL-LOCK-AVOIDANCE.s 630>≡
    # Copyright:    Public domain.
    # Filename:     GIMBAL_LOCK_AVOIDANCE.agc
    # Purpose:      Part of the source code for Comanche, build 055.
    #               It is part of the source code for the Command Module's (CM)
    #               Apollo Guidance Computer (AGC), Apollo 11.
    # Assembler:    yaYUL
    # Reference:     pp. 412-413
    # Contact:       Onno Hommes <ohommes@cmu.edu>.
    # Website:       www.ibiblio.org/apollo.
    # Mod history:   05/07/09 OH      Transcription Batch 1 Assignment
    #
    # The contents of the "Comanche055" files, in general, are transcribed
    # from scanned documents.
    #
    #       Assemble revision 055 of AGC program Comanche by NASA
    #       2021113-051.  April 1, 1969.
    #
    #       This AGC program shall also be referred to as Colossus 2A
    #
    #       Prepared by
    #
    #               Massachussets Institute of Technology
    #               75 Cambridge Parkway
    #               Cambridge, Massachusetts
    #
    #       under NASA contract NAS 9-4065.
    #
    # Refer directly to the online document mentioned above for further information.
    # Please report any errors to info@sandroid.org.

    # Page 412

                BANK      15
                SETLOC    KALCMON1
                BANK

                EBANK=     BCDU

    # DETECTING GIMBAL LOCK
    LOCKSKIRT      EQUALS  WCALC
    WCALC          LXC,1   DLOAD*
                                RATEINDX
                                ARATE,1
                SR4      CALL          # COMPUTE THE INCREMENTAL ROTATION MATRIX

```

July 12, 2016

631

		DELCOMP	# DEL CORRESPONDING TO A 1 SEC ROTATION
			# ABOUT COF
DLOAD*		VXSC	
		ARATE,1	
		COF	
MXV			
		QUADROT	
STODL		BRATE	
		AM	
DMP		DDV*	
		ANGLTIME	
		ARATE,1	
SR			
		5	
STOVL		TM	
		BRATE	
VXSC			
		BIASCALE	
STORE		BIASTEMP	# ATTITUDE ERROR BIAS TO PREVENT OVERSHOOT
			# IN SYSTEM
SETGO			# STATE SWITCH CALCMAN2 (43D)
		CALCMAN2	# 0(OFF) = BYPASS STARTING PROCEDURE
		NEWANGL +1	# 1(ON) = START MANEUVER
ARATE	2DEC	.0022222222	# = .05 DEG/SEC
	2DEC	.0088888889	# = .2 DEG/SEC
	2DEC	.0222222222	# = .5 DEG/SEC
	2DEC	.0888888889	# = 2 DEG/SEC \$22.5 DEG/SEC
ANGLTIME	2DEC	.000190735	# = 100B - 19
			# MANEUVER ANGLE TO MANEUVER TIME
QUADROT	2DEC	.1	# ROTATION MATRIX FROM S/C AXES TO CONTROL
# Page 413			
	2DEC	0	# AXES (X ROT = -7.25 DEG)
	2DEC	0	
	2DEC	0	
	2DEC	.099200	# = (.1)COS7.25

```
2DEC    -.012620    # =-(.1)SIN7.25
2DEC    0
2DEC    .012620    # (.1)SIN7.25
2DEC    .099200    # (.1)COS7.25
BIASCALE    2DEC    .0002543132    # = (450/180)(1/0.6)(1/16384)
This code is written to file src/GIMBAL-LOCK-AVOIDANCE.s.
```


B.36 GROUND TRACKING DETERMINATION PROGRAM

```

633  <src/GROUND-TRACKING-DETERMINATION-PROGRAM.s 633>≡
    # Copyright:    Public domain.
    # Filename:     GROUND_TRACKING_DETERMINATION_PROGRAM.agc
    # Purpose:      Part of the source code for Comanche, build 055.
    #               It is part of the source code for the Command Module's (CM)
    #               Apollo Guidance Computer (AGC), Apollo 11.
    # Assembler:   yaYUL
    # Reference:    pp. 456-459
    # Contact:      Onno Hommes <ohommes@cmu.edu>.
    # Website:      www.ibiblio.org/apollo.
    # Mod history:  2009-05-07 OH   Transcription Batch 1 Assignment
    #               2009-05-20 RSB   Corrected a couple of DIMOFLAG to DIMOFLAG.
    #
    # The contents of the "Comanche055" files, in general, are transcribed
    # from scanned documents.
    #
    #       Assemble revision 055 of AGC program Comanche by NASA
    #       2021113-051.  April 1, 1969.
    #
    #       This AGC program shall also be referred to as Colossus 2A
    #
    #       Prepared by
    #                   Massachussets Institute of Technology
    #                   75 Cambridge Parkway
    #                   Cambridge, Massachusetts
    #
    #       under NASA contract NAS 9-4065.
    #
    # Refer directly to the online document mentioned above for further information.
    # Please report any errors to info@sandroid.org.
    #
    # Page 456
    # GROUND TRACKING DETERMINATION PROGRAM -- P21
    #
    # PROGRAM DESCRIPTION
    #       MOD NO -- 1
    #       MOD BY -- N. M. NEVILLE
    #
    # FUNCTIONAL DESCRIPTION --
    #       TO PROVIDE THE ASTRONAUT DETAILS OF THE LM OR CSM GROUND TRACK WITHOUT
    #       THE NEED FOR GROUND COMMUNICATION (REQUESTED BY DSKY).
    #

```

```

# CALLING SEQUENCE --
#   ASTRONAUT REQUEST THROUGH DSKY V37E21E
#
# SUBROUTINES CALLED --
#   GOPERF4
#   GOFLASH
#   THISPREC
#   OTHPREC
#   LAT-LONG
#
# NORMAL EXIT MODES --
#   ASTRONAUT REQUEST THROUGH DSKY TO TERMINATE PROGRAM V34E
#
# ALARM OR ABORT EXIT MODES --
#   NONE
#
# OUTPUT --
#   OCTAL DISPLAY OF OPTION CODE AND VEHICLE WHOSE GROUND TRACK IS TO BE
#   COMPUTED
#       OPTION CODE      00002
#       THIS              00001
#       OTHER             00002
#   DECIMAL DISPLAY OF TIME TO BE INTEGRATED TO HOURS , MINUTES , SECONDS
#   DECIMAL DISPLAY OF LAT, LONG, ALT
#
# ERASABLE INITIALIZATION REQUIRED
#   AX0      2DEC    4.652459653 E-5   RADIANS      "68-69 CONSTANTS"
#   -AY0     2DEC    2.147535898 E-5   RADIANS
#   AZ0      2DEC    .7753206164      REVOLUTIONS
#   FOR LUNAR ORBITS 504LM VECTOR IS NEEDED
#   504LM    2DEC    -2.700340600 E-5   RADIANS
#   504LM _2 2DEC    -7.514128400 E-4   RADIANS
#   504LM _4 2DEC    _2.553198641 E-4   RADIANS
#
#   NONE
#
# DEBRIS

# Page 457
#   CENTRALS -- A,Q,L
#   OTHER -- THOSE USED BY THE ABOVE LISTED SUBROUTINES
#   SEE LEMPREC, LAT-LONG

```

SBANK= LOWSUPER # FOR LOW 2CADR'S.

BANK 33

July 12, 2016

635

```

SETLOC  P20S
BANK

EBANK=  P21TIME
COUNT 24/P21

PROG21  CAF    ONE
        TS     OPTION2      # ASSUMED VEHICLE IS LM, R2 = 00001
        CAF    BIT2         #  OPTION 2
        TC     BANKCALL
        CADR    GOPERF4
        TC     GOTOP00H      # TERMINATE
        TC     +2            # PROCEED VALUE OF ASSUMED VEHICLE OK
        TC     -5            # R2 LOADED THROUGH DSKY
P21PROG1 CAF    V6N34        # LOAD DESIRED TIME OF LAT-LONG.
        TC     BANKCALL
        CADR    GOFLASH
        TC     GOTOP00H      # TERM
        TC     +2            # PROCEED VALUES OK
        TC     -5            # TIME LOADED THROUGH DSKY
        TC     INTERPRET
        DLOAD
        DSPTM1
STCALL  TDEC1                # INTEG TO TIME SPECIFIED IN TDEC
        INTSTALL
BON      SET
        P21FLAG
        P21CONT              # ON...RECYCLE USING BASE VECTOR
        VINTFLAG             # OFF...1ST PASS CALC BASE VECTOR
SLOAD    SR1
        OPTION2
BHIZ     CLEAR
        +2                  # ZERO...THIS VEHICLE (CM)
        VINTFLAG             # ONE...OTHER VEHICLE(LM)
CLEAR    CLEAR
        DIMOFLAG
        INTYPFLG            # PRECISION
CALL      INTEGRV            # CALCULATE
GOTO      .AND               # .AND
        P21VSAVE             # ..SAVE BASE VECTOR
P21CONT  VLOAD               # RECYCLE...INTEG FROM BASE VECTOR
        P21BASER

# Page 458
STOVL    RCV                 # ..POS
        P21BASEV
```

	STODL	VCV	# ..VEL
		P21TIME	
	STORE	TET	# ..TIME
	CLEAR	CLEAR	
		DIMOFFLAG	
		MOONFLAG	
	SLOAD	BZE	
		P21ORIG	
		+3	# ZERO = EARTH
	SET		# ...2 = MOON
		MOONFLAG	
	CALL		
		INTEGRVS	
P21VSAVE	DLOAD		# SAVE CURRENT BASE VECTOR
		TAT	
	STOVL	P21TIME	# ..TIME
		RATT1	
	STOVL	P21BASER	# ..POS B-29 OR B-27
		VATT1	
	STORE	P21BASEV	# ..VEL B-7 OR B-5
	ABVAL	SL*	
		0,2	
	STOVL	P21VEL	# /VEL/ FOR N73 DSP
		RATT	
	UNIT	DOT	
		VATT	# U(R).(V)
	DDV	ASIN	# U(R).U(V)
		P21VEL	
	STORE	P21GAM	# SIN-1 U(R).U(V), -90 TO +90
	SXA,2	SET	
		P21ORIG	# 0 = EARTH 2 = MOON
		P21FLAG	
P21DSP	CLEAR	SLOAD	# GENERATE DISPLAY DATA
		LUNAFLAG	
		X2	
	BZE	SET	
		+2	# 0 = EARTH
		LUNAFLAG	
	VLOAD		
		RATT	
	STODL	ALPHAV	
		TAT	
	CLEAR	CALL	
		ERADFLAG	
		LAT-LONG	
	DMP		# MPAC = ALT, METERS B-29

July 12, 2016

637

```
# Page 459

STORE      K.01
P21ALT      # ALT/100 FOR N73 DSP

EXIT
CAF         V06N43      # DISPLAY LAT, LONG, ALT
TC          BANKCALL    # LAT, LONG = REVS B0 BOTH EARTH/MOON
CADR        GOFLASH     # ALT = METERS B-29 BOTH EARTH/MOON
TC          GOTOP00H     # TERM
TC          GOTOP00H
TC          INTPRET      # V32E RECYCLE
DLOAD      DAD
           P21TIME
           600SEC      # 600 SECONDS OR 10 MIN
STORE      DSPTEM1
RTB
           P21PROG1

600SEC      2DEC      60000      # 10 MIN

P21ONENN    OCT      00001      # NEEDED TO DETERMINE VEHICLE
           OCT      00000      # TO BE INTEGRATED
V06N43      VN        00643
V6N34       VN        00634
K.01        2DEC      .01
```

This code is written to file `src/GROUND-TRACKING-DETERMINATION-PROGRAM.s`.

B.37 IMU CALIBRATION AND ALIGNMENT

```

638  <src/IMU-CALIBRATION-AND-ALIGNMENT.s 638>≡
# Copyright:    Public domain.
# Filename:     IMU_CALIBRATION_AND_ALIGNMENT.agc
# Purpose:     Part of the source code for Comanche, build 055. It
#              is part of the source code for the Command Module's
#              (CM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:   yaYUL
# Reference:    pp. 423-455
# Contact:     Onno Hommes <ohommes@cmu.edu>
# Website:     http://www.ibiblio.org/apollo.
# Mod history: 2009-05-10 OH   Batch 1 Assignment Comanche Transcription
#              2009-05-20 RSB   Corrections: P00D00H -> P00D00H, definition
#                               of 25DECML fixed.
#              2009-05-23 RSB   At SPECSTS, corrected to PRI022.
#
# The contents of the "Comanche055" files, in general, are transcribed
# from scanned documents.
#
# Assemble revision 055 of AGC program Comanche by NASA
# 2021113-051. April 1, 1969.
#
# This AGC program shall also be referred to as Colossus 2A
#
# Prepared by
#
#           Massachussets Institute of Technology
#           75 Cambridge Parkway
#           Cambridge, Massachusetts
#
# under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information. Please report any errors to info@sandroid.org.
#
# Page 423
# NAME --      IMU PERFORMANCE TESTS 2
#
# DATE --      MARCH 20, 1967
#
# BY --        SYSTEM TEST GROUP 864-6900 EXT. 1274
#
# MODNO. --    ZERO
#
# FUNCTIONAL DESCRIPTION
#

```

POSITIONING ROUTINES FOR THE IMU PERFORMANCE TESTS AS WELL AS SOME OF
 # THE TESTS THEMSELVES. FOR A DESCRIPTION OF THESE SUBROUTINES AND THE
 # OPERATING PROCEDURES (TYPICALLY) SEE STG MEMO 685. THEORETICAL REF. E-1973

	BANK	33	
	SETLOC	IMUCAL	
	BANK		
IMUTEST	EBANK=	POSITON	
	CA	ZERO	
	TS	DRIFTT	
	TS	GEOCOMP1	
	CAF	TESTTIME	
	TS	LENGTHOT	
	TC	COAALIGN	# TAKE CARE OF DRIFT FLAG
	CAF	1SECX	
	TS	1SECXT1	
	CA	OC14400	
	TS	1/PIPADT	
GUESS	TC	INTPRET	# CALCULATE -COS LATITUDE AND SIN LATITUDE
	CALL		
		LATAZCHK	
	COS	DCOMP	
	SL1		
	STODL	WANGI	
		LATITUDE	
	SIN	SL1	
	STORE	WANGO	
	EXIT		
GEOIMUTT	TC	BANKCALL	# GYROCOMPASS COMES IN HERE
	CADR	IMUZERO	
	TC	IMUSTLLG	
IMUBACK	CA	ZERO	
	TS	NDXCTR	
	TS	TORQNDX	
	TS	TORQNDX +1	
NBPOSPL	CA	DEC17	
	TS	ZERONDX1	
	CA	XNBADR	
# Page 424			
	TC	ZEROING	
	CA	HALF	
	TS	XNB	
	TC	INTPRET	
	DLOAD	SIN	

		AZIMUTH	
	STORE	YNB	+2
	STODL	ZNB	+4
		AZIMUTH	
	COS		
	STORE	YNB	+4
	DCOMP		
	STORE	ZNB	+2
	EXIT		
	TC	CHECKMM	
	MM	03	# SEE IF IN OPTICAL VERIFICATION
	TCF	+2	# NO
	TCF	SETNBPOS	+1 # YES
	TC	INTPRET	
	CALL		
		CALCGA	
	EXIT		
	TC	BANKCALL	
	CADR	IMUCOARS	
	CAF	GLOKFBIT	# IF GLOKFAIL SET, GIMBAL LOCK
	MASK	FLAGWRD3	
	EXTEND		
	BZF	+2	
	INCR	NDXCTR	# +1 IF IN GIMBAL LOCK, OTHERWISE 0
	TC	DOWNFLAG	# RESET GIMBAL LOCK FLAG
	ADRES	GLOKFAIL	# BIT 14 FLAG 3
	TC	IMUSTLLG	
	CCS	NDXCTR	# IF ONE GO AND DO A PIPA TEST ONLY
	TC	PIPACHK	# ALIGN AND MEARSUE VERTICAL PIPA RA
	TC	BANKCALL	
	CADR	IMUFINE	
	TC	IMUSTLLG	
	EXTEND		
	DCA	PERFDLAY	
	TC	LONGCALL	
	SBANK=	LOWSUPER	# RSB 2004, OH 2009: Ask Ron
	EBANK=	POSITON	
	2CADR	GOESTIMS	
	CA	ESTICADR	
	TC	JOBSLEEP	
GOESTIMS	CA	ESTICADR	
	TC	JOBWAKE	
	TC	TASKOVER	
ESTICADR	CADR	ESTIMS	

TORQUE	CA	ZERO	
	TS	DSPTM2	
	CA	DRIFTI	
	TS	DSPTM2 +1	
	INDEX	POSITON	
	TS	SOUTHDR -1	
	TC	SHOW	
PIPACHK	INDEX	NDXCTR	# PIPA TEST
	TC	+1	
	TC	EARTH*E	
	CA	DEC57	
	TS	LENGTHOT	
	CA	ONE	
	TS	RESULTCT	
	CA	ZERO	
	INDEX	PIPINDEX	
	TS	PIPA	
	TS	DATAPL	
	TS	DATAPL +4	
	TC	CHECKG	# PIP PULSE CATCHING ROUTINE
	INHINT		
	CAF	TWO	
	TC	TWIDDLE	
	EBANK=	XSM	
	ADRES	PIPATASK	
	TC	ENDOFJOB	
PIPATASK	EXTEND		
	DIM	LENGTHOT	
	CA	LENGTHOT	
	EXTEND		
	BZMF	STARTPIP	
	CAF	BIT10	
	TC	TWIDDLE	
	EBANK=	XSM	
	ADRES	PIPATASK	
STARTPIP	CAF	PRI020	
	TC	FINDVAC	
	EBANK=	XSM	
	2CADR	PIPJOB	
	TC	TASKOVER	
PIPJOB	INDEX	NDXCTR	
	TC	+1	
	TC	EARTH*E	

```

# Page 426
CA      LENGTHOT
EXTEND
BZMF    +2
TC      ENDOFJOB
CA      FIVE

TS      RESULTCT
TC      CHECKG
EXTEND
DCS     DATAPL
DAS     DATAPL  +4

TC      INTPRET
DLOAD   DSU
        DATAPL  +6
        DATAPL  +2
BPL     CALL
        AINGOTN
        OVERFFIX
AINGOTN PDDL   DDV
        DATAPL  +4
SL4     DMPR
        DEC585      # DEC585 HAS BEEN REDEVINED FOR LEM
RTB
        SGNAGREE
STORE   DSPTM2
EXIT
CCS     NDXCTR
TC      COAALIGN      # TAKE PLATFORM OUT OF GIMBAL LOCK
TC      SHOW
VERTDRFT CA    3990DEC      # ABOUT 1 HOUR VERTICAL DRIFT TEST
TS      LENGTHOT
INDEX   POSITON
CS      SOUTHDR -2
TS      DRIFTT
CA      XSM      +4      # 0 IF POSN 4
EXTEND
BZF     PON2
PON4    CS      BIT5      # OFFSET PLATFORM
ADS     ERCOMP1 +2
CA      BIT5
ADS     ERCOMP1
TCF     PONG
PON2    CS      BIT5
ADS     ERCOMP1 +2
CA      BIT5

```

	ADS	ERCOMP1 +4	
PONG	TC	EARTH*	
	CA	ZERO	# ALLOW ONLY SOUTH GYRO EARTH RATE COMPENS
	TS	ERVECTOR	
	TS	ERVECTOR +1	
GUESS1	CAF	POSMAX	
	TS	TORQNDX	
	TS	TORQNDX +1	
	CA	CDUX	
	TS	LOSVEC	
# Page 427			
	TC	ESTIMS	
VALMIS	CA	DRIFT0	
	TS	DSPTM2 +1	
	CA	ZERO	
	TS	DSPTM2	
	TC	SHOW	
ENDTEST1	TC	DOWNFLAG	# IMU NOT IN USE
	ADRES	IMUSE	# BIT 8 FLAG 0
	CS	ZERO	
	TC	NEWMODEX +3	
	TC	BANKCALL	
	CADR	MKRELEAS	
	TC	ENDEXT	
# Page 428			
OVERFFIX	DAD	DAD	
		DPPOSMAX	
		ONEDPP	
	RVQ		
COAALIGN	EXTEND		# COARSE ALIGN SUBROUTINE
	QXCH	QPLACE	
	CA	ZERO	
	TS	THETAD	
	TS	THETAD +1	
	TS	THETAD +2	
	TC	BANKCALL	
	CADR	IMUCOARS	
	TC	BANKCALL	
	CADR	IMUSTALL	
	TC	SOMERR2	
	TC	QPLACE	
IMUSTLLG	EXTEND		

	QXCH	QPLACE	
	TC	COAALIGN	+10
CHECKG	EXTEND		# PIP PULSE CATCHING ROUTINE
	QXCH	QPLACE	
	TC	+6	
CHECKG1	RELINT		
	CA	NEWJOB	
	EXTEND		
	BZMF	+6	
	TC	CHANG1	
	INHINT		
	INDEX	PIPINDEX	
	CS	PIPAX	
	TS	ZERONDX	
	INHINT		
	INDEX	PIPINDEX	
	CA	PIPAX	
	AD	ZERONDX	
	EXTEND		
	BZF	CHECKG1	
	INDEX	PIPINDEX	
	CA	PIPAX	
	INDEX	RESULTCT	
	TS	DATAPL	
	TC	FINETIME	
	INDEX	RESULTCT	
	TS	DATAPL	+1
# Page 429	INDEX	RESULTCT	
	LXCH	DATAPL	+2
	RELINT		
ENDCHKG	TC	QPLACE	
ZEROING	TS	L	
	TCF	+2	
ZEROING1	TS	ZERONDX1	
	CAF	ZERO	
	INDEX	L	
	TS	0	
	INCR	L	
	CCS	ZERONDX1	
	TCF	ZEROING1	
	TC	Q	

July 12, 2016

645

```

      SETLOC  IMUCAL3
      BANK
ERTHRVSE  DLOAD  PDDL
          SCHZEROS      # PD24 = (SIN  (missing)  -COS (missing) 0)(OMEG/MS)
          LATITUDE
      COS      DCOMP
      PDDL     SIN
          LATITUDE
      VDEF     VXSC
          OMEG/MS
      STORE    ERVECTOR
      RTB
          LOADTIME
      STOVL    TMARK
          SCHZEROS
      STORE    ERCOMP1
      RVQ
      SETLOC  IMUCAL
      BANK
EARTHRR   ITA     RTB      # CALCULATES AND COMPENSATES EARTH RATE
          S2
          LOADTIME
      STORE    TEMPTIME
      DSU      BPL
          TMARK
          ERTHR
      CALL
          OVERFFIX
ERTHR      SL      VXSC
          9D
          ERVECTOR
      MXV     VAD
          XSM
          ERCOMP1
      STODL   ERCOMP1
          TEMPTIME
      STORE    TMARK
      AXT,1    RTB
      ECADR    ERCOMP1
          PULSEIMU
      GOTO
          S2

EARTHRR*  EXTEND
      QXCH     QPLACES
      TC       INTPRET
```

	CALL	EARTHRR	
PROUT	EXIT		
	TC	IMUSTLLG	
	TC	QPLACES	
# Page 431			
SHOW	EXTEND		
	QXCH	QPLACE	
SHOW1	CA	POSITON	
	TS	DSPTM2 +2	
	CA	VB06N98	
	TC	BANKCALL	
	CADR	GOFLASH	
	TC	ENDTEST1	# V34
	TC	QPLACE	# V33
	TCF	SHOW1	
OC14400	OCT	14400	
3990DEC	=	OMEG/MS	
VB06N98	VN	0698	
TESTTIME	OCT	01602	
DEC17	=	ND1	
OGCPL	ECADR	OGC	
1SECX	=	1SEC	
DEC57	=	VD1	
XNBADR	GENADR	XNB	
XSMADR	GENADR	XSM	
OMEG/MS	2DEC	.24339048	
P11OUT	TC	BANKCALL	
	CADR	MATRXJOB	# RETURN TO P11
	COUNT	02/COMST	
	BLOCK	2	
FINETIME	INHINT		# RETURNS WITH INTERRUPT INHIBITED
	EXTEND		
	READ	LOSCALAR	
	TS	L	
	EXTEND		
	RXOR	LOSCALAR	
	EXTEND		
	BZF	+4	
	EXTEND		
	READ	LOSCALAR	

```

      TS      L
+4    CS      POSMAX
      AD      L
      EXTEND
      BZF      FINETIME +1
      EXTEND
      READ     HISCALAR
      TC Q

```

```

# Page 432
# PROGRAM NAME:  OPTIMUM PRELAUNCH ALIGNMENT CALIBRATION
# DATE:  NOVEMBER 2 1966
# BY:  GEORGE SCHMIDT IL 7-146 EXT. 126
# MOD NO 3
#
# FUNCTIONAL DESCRIPTION
#
# THIS SECTON CONSISTS OF PRELAUNCH ALIGNMENT AND GYRO DRIFT TESTS
# INTEGRATED TOGETHER TO SAVE WORDS.  COMPASS IS COMPLETELY RESTART
# PROOFED EXCEPT FOR THE FIRST 30 SECONDS OR SO.  PERFORMANCE TESTS OF
# THE IRIGS IS RESTART PROOFED ENOUGH TO GIVE 75 PERCENT CONFIDENCE THAT
# IF A RESTART OCCURS THE DATA WILL STILL BE GOOD.  GOOD PRACTICE TO RECYCL
# WHEN A RESTART OCCURS UNLESS IT HAPPENS NEAR THE END OF A TEST -- THEN WAIT
# FOR THE DATA TO FLASH.
#
# A RESTART IN GYROCOMPASS DURING GYRO TORQUING CAUSES PULSES TO BE LOST.
# THE PRELAUNCH ALIGNMENT TECHNIQUE IS BASICALLY THE SAME AS IN BLOCK 1
# EXCEPT THAT IT HAS BEEN SIMPLIFIED IN THE SENSE THAT SMALL ANGLE APPROX.
# HAVE BEEN USED.  THE DRIFT TESTS USE A UNIQUE IMPLEMENTATION OF THE
# OPTIMUM STATISTICAL FILTER.  FOR A DESCRIPTION SEE E-1973.  BOTH OF THESE
# ROUTINES USE STANDARD SYSTEM TEST LEADIN PROCEDURES.  THE INITIALIZATION
# PROCEDURE THE DRIFT TESTS IS IN THE JDC'S.  THE INITIALIZATION METHOD
# FOR GYROCOMPASS IS AN ERAS LOAD THEN A MISSION PHASE CALL.
# THE COMPASS ALIGNS TO Z DOWN, X DOWNRANGE, HAS THE CAPABILITY
# CHANGE AZIMUTH WHILE RUNNING, IS COMPENSATED FOR
# COMPONENT ERRORS, IS CAPABLE OF OPTICAL VERIFICATION (CSM ONLY).
#
# COMPASS ERASABLE LOAD REQUIRED
#
# 1.      LAUNCHAZ-DP AZIMUTH IN REV FROM NORTH OF XSM DESIRED      (NOM=.2)
# 2.      LATITUDE-DP-OF LAUNCH PAD
# 3.      AZIMUTH-DP-OF ZNB OF VEHICLE
# 4.      IMU COMPENSATION PARAMETERS
# 5.      AZ AND ELEVATION OF TARGETS 1,2      *****OPTIONAL*****
#
# TO PERFORM AS PART OF COMPASS

```

```

#
#      1.      OPTICAL VERIFICATION: V 65 E
#      2.      AXIMUTH CHANGE: V 78 E
#
# SUBROUTINES CALLED
#
# DURING OPTICAL VERIFICATION (CSM ONLY) ESSENTIALLY ALL OF INFLIGHT ALIGN
# IS CALLED IN ONE WAY OR ANOTHER.  SEE THE LISTING.
#
# NORMAL EXIT
#
# DRIFT TESTS.  LENGTHOT GOES TO ZERO-RETURN TO IMU PERF TEST2 CONTROL
# GYROCOMPASS-MANY, SEE THE LISTING.
#
# ALARMS
#
# 1600  OVERFLOW IN DRIFT TEST
# Page 433
# 1601  BAD IMU TORQUE ABORT
# 1602  BAD OPTICS DURING VERIFICATION-RETURN TO COMPASS      CSM ONLY
#
# OUTPUT
#
# DRIFT TESTS:  FLASHING DISPLAYS OF RESULTS-CONTROLLED IN IMU PERF TESTS 2
# COMPASS-PROGRAM MODE LIGHTS TELL YOU WHAT PHAS OF PROGRAM YOU ARE IN
#      01      INITIALIZING THE PLATFORM POSITION AND ERASABLE
#      02      GYROCOMPASSING
#      03      DOING OPTICAL VERIFICATION (CSM)
#
# DEBRIS
#
# ALL CENTRALS, ALL OF EBANK XSM

# Page 434
# MOST OF THE ROUTINES COMMON TO ALIGNMENT AND CALIBRATION APPEAR
# ON THE NEXT FEW PAGES.

COUNT  33/P02

EBANK=   XSM
BANK     33
SETLOC   IMUCAL
BANK

ESTIMS   TC      2PHSCHNG      # COMES HERE FROM IMU2
          OCT     00075

```


July 12, 2016

649

```
RSTGTS1      OCT      00004      # TURN OFF GROUP 4 IF ON
              INHINT      # COMES HERE PHASE1 RESTART
              CA          TIME1
              TS          GTSWTLT1
              CAF         ZERO      # ZERO THE PIPAS
              TS          PIPAX
              TS          PIPAY
              TS          PIPAZ
              RELINT
              CA          77DECML    # ZERO ALL NECESSARY LOCATIONS
              TS          ZERONDX1
              CA          ALXXXZ
              TC          ZEROING
              TC          INTPRET
              SLOAD
              SCHZEROS
              STOVL      GCOMPSW -1
              INTVAL +2      # LOAD SOME INITIAL DRIFT GAINS
              STOVL      ALX1S
              SCHZEROS
              STORE      GCOMP
              STORE      DELVX      # GCOMPZER SUBROUTINE NO LONGER NEEDED
              EXIT

              CCS          GEOCOMP1  # NON ZERO IF COMPASS.
              TC          +2
              TC          SLEEPIE +1
              TC          INTPRET
              CALL
              ERTHRVSE
              EXIT
              CA          LENGHOT    # TIMES FIVE IS THE NUM OF SEC ERECTING
              TS          ERECTIME

              TC          NEWMODEX
              MM          02
              TC          BANKCALL   # SET UP PIPA FAIL TO CAUSE ISS ALARM
# Page 435
              CADR      PIPUSE      # COMPASS NEVER TURNS THIS OFF
              TC          ANNNNNN    # END OF FIRST TIME THROUGH
```

Page 436

COMES HERE AT THE END OF EVERY ITERATION THROUGH DRIFT TEST OR COMPASS

SET UP WAITLIST SECTION

```
SLEEPIE      TS          LENGHOT    # TEST NOT OVER-DECREMENT LENGHOT
```

```

          TC      PHASCHNG      # CHANGE PHASE
          OCT      00135
          CCS      TORQNDX      # ARE WE DOING VERTDRIFT
          TC      EARTH*      # TRUE TORQUE SOUTH GYRO
WTLISTNT  TC      CHKCOMED      # SEE IF COMPASS OVER
          TC      SETGWLST
          TC      ENDOFJOB

SETGWLST  EXTEND
          QXCH     MPAC          # CALLED EVERY WAITLIST OR AZIMUTH CHANGE
          INHINT
          CS      TIME1
          AD      GTSWTLT1
          EXTEND
          BZMF     +2
          AD      NEGMAX          # 10 MS ERROR OK
          AD      1SECXT1        # 1 SEC FOR CALIBRATION, .5 SEC IN COMPASS
          EXTEND
          BZMF     RIGHTGTS
WTGTSMPL  TC      TWIDDLE
          EBANK=   ALTIM
          ADRES    ALLOOP
          TC      MPAC
RIGHTGTS  CAF      FOUR          # SET UP NEXT WAITLIST-ALLOW SOME TIME
          TC      WTGTSMPL      # END OF WAITLIST SECTION

# STORE AND LOAD DATA SECTIONS FOR RESTART PROOFING

25DECML   EQUALS   OCT31
STOREDTA  CAF      25DECML
          TS      MPAC
          INDEX   MPAC
          CAE     THETAX1
          INDEX   MPAC
          TS      RESTARPT
          CCS     MPAC
          TCF     STOREDTA +1
          TC      Q

LOADSTD   CAF      25DECML
          TS      MPAC
          INDEX   MPAC
          CA      RESTARPT
          INDEX   MPAC

```

July 12, 2016

651

TS THETAX1
CCS MPAC
TCF LOADSTDT +1
TC Q

COMES HERE EVERY ITERATION BY A WAITLIST CALL SET IN SLEEPIE

ALLOOP CA TIME1
TS GTSWTLT1 # STORE TIME TO SET UP NEXT WAITLIST.
ALLOOP3 CA ALTIM
TS GEOSAVE1
TC PHASCHNG
OCT 00115
ALLOOP1 CAE GEOSAVE1
TS ALTIM
CCS A
CA A # SHOULD NEVER HIT THIS LOCATION
TS ALTIMS
CS A
TS ALTIM
CAF ZERO
XCH PIPAX
TS DELVX
CAF ZERO
XCH PIPAY
TS DELVY
CAF ZERO
XCH PIPAZ
TS DELVZ
CAF 19DECML # 23 OCT
TC NEWPHASE
OCT 00005
SPECSTS CAF PRI022
TC FINDVAC
EBANK= GEOSAVE1
2CADR ALFLT # START THE JOB
TC TASKOVER

Page 438

THIS IS PART OF THE JOB DONE EVERY ITERATION

ALFLT TC STOREDTA # STORE DATA IN CASE OF RESTART IN JOB
TC PHASCHNG # THIS IS THE JOB DONE EVERY ITERATION
OCT 00215
TCF +2

```

ALFLT1      TC      LOADSTD      # COMES HERE ON RESTART

            CCS      GEOCOMP1
            TC        +2
            TC        NORMLOP
            TC        CHKCOMED      # SEE IF PRELAUNCH OVER
            TC        BANKCALL      # COMPENSATION IF IN COMPASS
            CADR      1/PIPA
NORMLOP     TC        INTPRET
            DLOAD
            INTVAL
            STOVL      S1
            DELVX
            VXM        VSL1
            XSM
            DLOAD      DCOMP
            MPAC +3
            STODL      DPIPAY
            MPAC +5
            STORE      DPIPZ

            SETPD      AXT,1
            0
            8D
            SLOAD      DCOMP
            GEOCOMP1
            BMN
            ALWAYSG      # DO A QUICK COMPASS

# Page 439
# NOW WE HAVE JUST THE CALIBRATION PARTS OF THE PROGRAM-NEXT PAGES

            COUNT      33/COMST

ALCGKK      SLOAD      BMN
            ALTIMS
            ALFLT3      # NO NEW GAINS NEEDED
ALKCG       AXT,2      LXA,1      # LOADS SLOPES AND TIME CONSTANTS AT RQST
            12D
            ALX1S
ALKCG2      DLOAD*     INCR,1
            ALFDK +144D,1
            DEC        -2
            STORE      ALDK +10D,2
            TIX,2      SXA,1
            ALKCG2

```

ALX1S

ALFLT3	AXT,1	8D	# MEASUREMENT INCORPORATION ROUTINES
			# AND GAIN UPDATES
DEMLP	DLOAD*	DMP	
		DPIPAY +8D,1	
		PIPASC	
	SLR	BDSU*	
		9D	
		INTY +8D,1	
	STORE	INTY +8D,1	
	PDDL	DMP*	
		VELSC	
		VLAUN +8D,1	
	SL2R		
	DSU	STADR	
	STORE	DELM +8D,1	
	STORE	DELM +10D,1	
	TIX,1	AXT,2	
		DEMLP	
		4	
ALILP	DLOAD*	DMPR*	
		ALK +4,2	
		ALDK +4,2	
	STORE	ALK +4,2	
	TIX,2	AXT,2	
		ALILP	
		8D	
ALKLP	LXC,1	SXA,1	
		CMPX1	
		CMPX1	
	DLOAD*	DMPR*	
		ALK +1,1	
		DELM +8D,2	
# Page 440	DAD*		
		INTY +8D,2	
	STORE	INTY +8D,2	
	DLOAD*	DAD*	
		ALK +12D,2	
		ALDK +12D,2	
	STORE	ALK +12D,2	
	DMPR*	DAD*	
		DELM +8D,2	
		INTY +16D,2	
	STORE	INTY +16D,2	

```

DLOAD* DMP*
      ALSK +1,1
      DELM +8D,2
SL1R   DAD*
      VLAUN +8D,2
STORE  VLAUN +8D,2
TIX,2  AXT,1
      ALKLP
      8D

LOOSE  DLOAD* PDDL*          # EXTRAPOLATE SWAY VARIABLES
      ACCWD +8D,1
      VLAUN +8D,1
PDDL*  VDEF
      POSNV +8D,1
MXV    VSL1
      TRANSM1
DLOAD  MPAC
STORE  POSNV +8D,1
DLOAD  MPAC +3
STORE  VLAUN +8D,1
DLOAD  MPAC +5
STORE  ACCWD +8D,1
TIX,1  LOOSE

      AXT,2  AXT,1          # EVALUATE SINES AND COSINES
      6
      2
BOOP   DLOAD* DMPR
      ANGX +2,1
      GEORGEJ
      SR2R
      PUSH   SIN
      SL3R   XAD,1
      X1
      STORE  16D,2
      DLOAD
      COS
      STORE  22D,2          # COSINES
      TIX,2
      BOOP

```

July 12, 2016

655

```
PERFERAS      EXIT
               CA      EBANK7
               EBANK=   LAT(SPL)
               TS      EBANK
               TC      LAT(SPL)      # GOTO ERASABLE ONLY TO RETURN

# CAUTION
#
# THE ERASABLE PROGRAM THAT DOES THE CALCULATIONS MUST BE LOADED
# BEFORE ANY ATTEMPT IS MAKE TO RUN THE IMU PERFORMANCE TEST

ONCEMORE      EBANK=   LENGTHOT
               CCS     LENGTHOT
               TC      SLEEPIE      # TEST NOT OVER SET UP NEXT WAITLIST
               CCS     TORQNDX
               TCF     +2
               TC      SETUPER1
               CA      CDUX
               TS      LOSVEC +1     # FOR TROUBLESHOOTING POSNS 2$4 VD
SETUPER1      TC      INTPRET      # DRIFT TEST OVER
               DLOAD   PDDL          # ANGLES FROM DRIFT TEST ONLY
               ANGZ
               ANGY
               PDDL    VDEF
               ANGX
               VCOMP   VXSC
               GEORGEJ
               MXV     VSR1
               XSM
               STORE   OGC
               EXIT

TORQINCH      TC      PHASCHNG
               OCT     00005
               CA      OGCPL
               TC      BANKCALL
               CADR     IMUPULSE
               TC      IMUSTLLG
               CCS     TORQNDX      # + IF IN VERTICAL DRIFT TEST
               TC      VALMIS      # VERT DRIFT TEST OVER
               TC      INTPRET

# Page 442
               CALL
               ERTHRVS
               EXIT
```

	TC	TORQUE	# GO TO IMU2 FOR A PIPA TEST AND DISPLAY
SOMEERRR	TC	ALARM	
	OCT	1600	
	TC	+3	
SOMERR2	TC	ALARM	
	OCT	1601	
	TC	PHASCHNG	
	OCT	00005	
	TC	ENDTEST1	

THE FAMOUS MAGIC NUMBERS OF SCHMIDT ARE NOW PART OF AN ERASABLE LOAD

DEC585	OCT	02222	# 1170 B+14 ORDER IS NOW IMPORTANT
SCHZEROS	2DEC	.00000000	
	2DEC	.00000000	
	OCT	00000	
ONEDPP	OCT	00000	
	OCT	00001	# ABOVE ORDER IS IMPORTANT
INTVAL	OCT	4	
	OCT	2	
	DEC	144	
	DEC	-1	
SOUPLY	2DEC	.93505870	# INITIAL GAINS FOR PIP OUTPUTS
	2DEC	.26266423	# INITIAL GAINS/4 FOR ERECTION ANGLES
77DECML	DEC	77	
ALXXXZ	GENADR	ALX1S -1	

GYROCOMPASS PORTIONS FINISH THIS LOG SECTION

COUNT 33/P01

INITIALIZATION SECTION

GTSCPSS	CA	FLAGWRD1	# CALLED BY V37
	MASK	NOP01BIT	
# Page 443			
	EXTEND		
	BZF	GTSCPSSA	
	TC	P00D00	
	OCT	1521	# NODO ALARM FOR P01 - P11 ALREADY DONE
GTSCPSSA	CAF	ONE	
	TS	GEOCOMP1	# THIS IS THE LEAD IN FOR COMPASS

July 12, 2016

657

```

                                CA      1/PIPGT
                                TS      1/PIPADT
NXXTENN                        CA      BIT8
                                TS      LENGTHOT
                                CAF     1/2SECX      # COMPASS IS A .5 SEC LOOP
                                TS      1SECXT1
                                CAF     ONE
                                TS      PREMTRX1
                                TS      PERFDLAY +1
                                CAF     ZERO
                                TS      PERFDLAY
                                EXTEND
                                DCA     LUNCHAZ1
                                DXCH    NEWAZ1
                                EXTEND
                                DCA     LUNCHAZ1
                                DXCH    OLDAZMTH
SETUPGC                        CA      DEC17
                                TS      ZERONDX1
                                CA      XSMADR
                                TC      ZEROING
                                TC      POSN17C
                                TC      GEOIMUTT      # GO TO IMU2 FOR FURTHER INITIALIZATION

POSN17C                        EXTEND      # COMPASS POSITION Z DOWN, X DOWNRANGE
                                QXCH    QPLACE      # FROM NORTH IN REVOLUTIONS + CLOCKWISE
                                CS      HALF      # ALL THIS TO INITIALIZE MATRIX
                                TS      ZSM
                                TC      INTPRET
                                DLOAD    PUSH
                                NEWAZ1
                                SIN
                                STORE   XSM      +4
                                STODL   YSM      +2
                                COS
                                STORE   YSM      +4
                                DCOMP
                                STORE   XSM      +2
                                EXIT
                                TC      QPLACE
```

Page 444

JOB DONE EVERY ITERATION THROUGH COMPASS PROGRAM. SET BY TASK ALLOOP

COUNT 33/P02

ALWAYSG	DLOAD*	DSU*	# COMPASS AND ERECT
		DPIPAY +8D,1	
		FILDELV1 +8D,1	
	DMPR	DAD*	
		GEOCONS1	
		FILDELV1 +8D,1	
	STORE	FILDELV1 +8D,1	
	DAD*		
		INTVEC1 +8D,1	
	STORE	INTVEC1 +8D,1	
	DMPR	DAD*	
		GEOCONS2	
		FILDELV1 +8D,1	
	DMPR	PUSH	
		GEOCONS5	
	TIX,1	SLOAD	
		ALWAYSG	
		ERECTIM1	
	BZE	DLOAD	
		COMPGS	
		THETAN1 +2	
	DSU	STADR	
	STODL	THETAN1 +2	# ERECTION ONLY.
	BDSU		
		THETAN1 +4	
	STORE	THETAN1 +4	
	GOTO		
		ADDINDRF	
COMPGS	DLOAD	DAD	# COMPASS
		THETAN1	
		FILDELV1	
	STODL	THETAN1	
		FILDELV1	
	DMPR	BDSU	
		GEOCONS3	
		THETAN1 +4	
	STODL	THETAN1 +4	
		FILDELV1 +4	
	DMPR	BDSU	
		GEOCONS3	
		THETAN1 +2	
	PDDL	DMPR	
		INTVEC1 +4	
		GEOCONS4	
	BDSU	STADR	

July 12, 2016

659

```
# Page 445
ADDINDRF      STORE  THETAN1 +2
               EXIT

ENDGTSAL      CCS    LENGTHOT      # IS 5 SEC OVER-THE TIME TO TORQ PLATFORM
               TC     SLEEPIE      # NO-SET UP NEXT WAITLIST CALL FOR .5 SEC
               TC     CHKCOMED
               CCS    LGYRO        # YES BUT ARE GYROS BUSY
               TCF    SLEEPIE +1   # BUSY-GET THEM .5 SECONDS FROM NOW

LASTGTS       TC     INTPRET
               VLOAD
               ERCOMP1
               STODL  THETAX1
               TMARK
               STORE  ALK
               EXIT      # PREVIOUS SECTION WAS FOR RESTARTS

RESTAIER      TC     PHASCHNG
               OCT    00275
               TC     INTPRET      # ADD COMPASS COMMANDS INTO ERATE
               VLOAD  MXV
               THETAN1
               XSM
               VSL1   VAD
               THETAX1
               STODL  ERCOMP1
               ALK
               STORE  TMARK
               EXIT
               TC     EARTH*      # TORQUE IT ALL IN
               CAE    ERECTIM1
               TS     GEOSAVE1
               TC     PHASCHNG
               OCT    00155
               TC     INTPRET
               VLOAD
               SCHZEROS
               STORE  THETAN1
               EXIT
               CCS    PREMTRXC
               TC     NOCHORLD
               TC     PHASCHNG
               OCT    00255
               TC     INTPRET
RESTEST3
```

	DLOAD	LAUNCHAZ	
	DSU	BZE	
		OLDAZMTH	
		NOAZCHGE	
	STORE	OD	
# Page 446			
	SLOAD	DAD	
		ONEDPP +1	
		PREMTRXC	# DOES NOT CHANGE LAUNCHAZ
	STODL	PREMTRXC	
		LAUNCHAZ	
	STODL	NEWAZMTH	
		OD	
ADERCOMP	STORE	ERCOMP +4	
	EXIT		
	TC	POSN17C	
	TC	PHASCHNG	
	OCT	00335	
RESCHNG	EXTEND		
	DCA	NEWAZMTH	
	DXCH	OLDAZMTH	
	CA	BIT7	# SPEND 320 SEC ERECTING
	TS	LENGTHOT	
	TC	PHASCHNG	
	OCT	00075	
SPITGYRO	CA	ERCOMPPL	
	TC	BANKCALL	
	CADR	IMUPULSE	
	TC	BANKCALL	
	CADR	IMUSTALL	
	TC	SOMERR2	
	TC	ESTIMS	# RE-INITIALIZE
NOAZCHGE	EXIT		
	CA	ONE	
	TS	PREMTRXC	
NOCHORLD	CCS	GEOSAVE1	
	TS	ERECTIM1	# COUNTS DOWN FOR ERECTION.
ANNNNNN	CAF	NINE	
	TS	LENGTHOT	
	TC	SLEEPIE +1	

July 12, 2016

661

```
CHKCOMED      INHINT
               CS      MODREG      # CHECK FOR MM 07 FIRST
               AD      SEVEN
               EXTEND
               BZF      GOBKCALB    # IF MM 07 RETURN TO PERF TEST
               CS      ZERO
               EXTEND
               RXOR     CHAN30      # READ AND INVERT BITS IN CHANNEL 30
               MASK     BIT5        # LIFTOFF BIT
               CCS      A
               TCF      PRELTERM    # LIFTOFF HAS OCCURRED

# Page 447
               CA      GRRBKBIT    # CHECK FOR BACKUP LIFTOFF
               MASK     FLAGWRD5    # BIT5 FLAGWRD5
               CCS      A
               TCF      PRELTERM    # BACKUP RECEIVED

               RELINT
GOBKCALB       TC      Q

PRELTERM       CA      PRIO22      # PRELAUNCH DONE -- SET UP P11
               TC      PRIOCHNG    # INCREASE PRIORITY HIGHER THAN SERVICER
               INHINT
               TC      POSTJUMP
               CADR      P11

ERCOMPPL       ECADR    ERCOMP

GEOCONS5       EQUALS   HIDPHALF
1/PIPAGT       OCT      06200
17DECML        =       ND1        # OCT 21
19DECML        =       VD1        # OCT 23
1/2SECX        =       .5SEC

# Page 448
GEOSTR4        EQUALS   ENDOFJOB

# Page 449
# OPTICAL VERIFICATION ROUTINES FOR GYROCOMPASS

               COUNT    33/P03

GCOMPVER       TC      PHASCHNG    # OPTICAL VERIFICATION ROUTINE
```

	OCT	00154	
	TC	NEWMODEX	# ENTERED BY VERB 65 ENTER
	MM	03	
SETNBPOS	TC	NBPOSPL	
	TC	BANKCALL	
	CADR	MKRELEAS	
OPTDATA	CAF	BIT1	# CALLS FOR AZIMUTH AND ELEVATION OF TARGET
	ZL		# 1, THEN TARGET 2
	LXCH	RUN	# AZIMUTH CLOCKWISE FROM NORTH TO TARGET
	TS	DSPTM1 +2	# ELEVATION MEASURED FROM HORIZONTAL
	EXTEND		
	INDEX	RUN	
	DCA	TAZEL1	
	DXCH	DSPTM1	
	CAF	V05N30E	
	TC	BANKCALL	
	CADR	GODSPRET	
	CAF	VN0641	
	TC	BANKCALL	
	CADR	GOFLASH	
	TC	GCOMP5	
	TC	+3	
	TC	-8D	
VN0641	VN	0641	
	DXCH	DSPTM1	# TAZEL1 TARGET 1 AZIMUTH
	INDEX	RUN	
	DXCH	TAZEL1	# TAZEL1 +2 TARGET 2 AZIMUTH
	CCS	RUN	
	TCF	+4	
	CAF	TWO	
	TS	L	
	TCF	OPTDATA +2	# MPAC 1ST PASS=0 2ND PASS=2
	TC	CONTIN33	
V05N30E	VN	0530	
	TC	INTPRET	# UNDYNAMIC ASSEMBLER
TAR/EREF	AXT,1	AXT,2	# TARGET VECTOR
		2	# SIN(EL) -COS(AZ)COS(EL) SIN(AZ)COS(EL)
		12D	
	SSP	SETPD	
		S2	
		6	
# Page 450		0	

July 12, 2016

663

TAR1	SLOAD*	SR2	# X1=2 X2=12 S2=6 X1=0 X2=6 S2=6
		TAZEL1 +3,1	
	STORE	0	# PD00 ELEVATION PD00
	SIN		
	STORE	18D,2	# PD06 *** SIN(EL) ***PD12
	DLOAD		
		0	
	COS	PUSH	# PD00 COS(EL) PD00
	SLOAD*	RTB	
		TAZEL1 +2,1	
		CDULOGIC	
	STORE	2	# PD02 AZIMUTH PD02
	SIN	DMP	
		0	
	SL1		
	STORE	22D,2	# PD10 *** SIN(AZ)COS(EL) ***PD16
	DLOAD	COS	
		2	
	DMP	SL1	
	DCOMP	AXT,1	
		0	
	STORE	20D,2	# PD08 *** -COS(AZ)COS(EL) ***PD14
	TIX,2	RVQ	
		TAR1	
	BANK	33	
	SETLOC	IMUCAL	
	BANK		
	COUNT*	\$\$/P03	
CONTIN33	CA	ONE	
	TS	STARCODE	
	CA	ZERO	
	TC	TARGDRVE	
	TC	INTPRET	
	CALL		
		TAR/EREF	
NEXTBKNS	VLOAD	MXV	
		6D	
		XSM	
	VSL1		
	STOVL	STARAD	
		12D	
	MXV	VSL1	
		XSM	

	STCALL	STARAD +6
		LITTLSUB
	STORE	LOSVEC
# Page 451	EXIT	
	TC	BANKCALL
	CADR	MKRELEAS
NEXBNKSS	CAF	TWO
	TS	STARCODE
	CAF	SIX
	TC	TARGDRVE
	TC	INTPRET
	CALL	
		LITTLSUB
	STOVL	12D
		LOSVEC
	STCALL	06D
		AXISGEN
	CALL	
		CALCGTA
	EXIT	
GCOMP4	CAF	V06N93S
	TC	BANKCALL
	CADR	GOFLASH
	TC	GCOMP5
	TCF	+2
	TCF	GCOMP4
	TC	INTPRET
	VLOAD	VAD
		OGC
		ERCOMP1
	STORE	ERCOMP1
	EXIT	
GCOMP5	TC	BANKCALL
	CADR	MKRELEAS
	TC	DOWNFLAG
	ADRES	TRM03FLG
	TC	NEWMODEX
	MM	02
	TC	PHASCHNG
	OCT	00004
	TC	ENDOFJOB
V06N93S	VN	0693

July 12, 2016

665

GTSOPTCS TC ALARM
GTSOPTSS OCT 01602
TC GCOMP5

BANK 34
SETLOC IMUCAL1
BANK

Page 452

COUNT 34/COMST

LATAZCHK DLOAD SL2 # CALLS FOR AZIMUTH AND LATITUDE

LATITUDE
STODL DSPTEM1 +1
AZIMUTH

RTB EXIT
1ST02S

XCH MPAC
TS DSPTEM1
TC BANKCALL
CADR CLEANDSP
CAF VNG0641
TC BANKCALL
CADR GOFLASH

NOT ALLOWED

TC +2
TC +2

TC -5
TC INTPRET

SLOAD RTB
DSPTEM1
CDULOGIC

STORE AZIMUTH

SLOAD SR2
DSPTEM1 +1

STORE LATITUDE

RVQ

VNG0641 VN 0641

BANK 33
SETLOC IMUCAL
BANK

TARGDRVE COUNT* \$\$/P03

EXTEND

QXCH QPLAC

TS TARG1/2
 TC INTPRET
 CALL
 LXC,1 TAR/EREF
 VLOAD*
 TARG1/2
 6D,1
 STCALL STAR
 SXTANG
 EXIT
 CA SAC
 TS DESOPTS

Page 453

RETARG

CA PAC
 TS DESOPTT
 CAF ZERO
 TS OPTIND
 CAF ONE
 TC BANKCALL
 CADR SXTMARK
 TC BANKCALL
 CADR OPTSTALL
 TC GTSOPTCS
 CAE FLAGWRD1
 MASK TRM03BIT
 CCS A
 TC GCOMP5
 INDEX MARKSTAT
 CA QPRET
 EXTEND
 BZF RETARG1
 TC QPLAC

RETARG1

CA ZERO
 XCH MARKSTAT
 CCS A
 INDEX A
 TS A
 TCF RETARG
 BANK 33
 SETLOC IMUCAL
 BANK
 COUNT* \$\$/P03

RELEASE PREVIOUSLY GRABBED VAC AREA

GO DO SXTMARK AGAIN

July 12, 2016

667

PIPASC	2DEC	.76376833
VELSC	2DEC	-.52223476
ALSK	2DEC	.17329931
	2DEC	-.00835370
GEORGEJ	2DEC	.63661977
GEOCONS1	2DEC	.1
GEOCONS2	2DEC	.005
GEOCONS3	2DEC	.062
GEOCONS4	2DEC	.0003

Page 454

LITTLSUB	COUNT	33/P02
	STQ	
		QPLAC
	LXC,1	VLOAD*
		MARKSTAT
		2,1
	STCALL	CDUSPOT
		SXTNB
	CALL	
		TRG*NBSM
	GOTO	
		QPLAC

AZMTHCG1	EXIT	
	TC	INTPRET
	DLOAD	RTB
		NEWAZMTH
		1ST02S
	EXIT	
	XCH	MPAC
	TS	DSPTM1
	TC	BANKCALL
	CADR	CLEANDSP
	CAF	VN0629
	TC	BANKCALL

CADR	GOFLASH
TCF	+2
TCF	+2
TCF	-5
TC	INTPRET
SLOAD	RTB
	DSPTM1
	CDULOGIC
STORE	LAUNHAZ
EXIT	
CA	ZERO
TS	PREMTRXC
TC	PHASCHNG
OCT	00004
TC	POSTJUMP
CADR	PINBRNCH

VN0629	VN	0629
--------	----	------

Page 455

*** END OF COMAID .029 ***

This code is written to file `src/IMU-CALIBRATION-AND-ALIGNMENT.s`.

B.38 IMU COMPENSATION PACKAGE

```

669  <src/IMU-COMPENSATION-PACKAGE.s 669>≡
# Copyright:    Public domain.
# Filename:     IMU_COMPENSATION_PACKAGE.agc
# Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:   yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        297-306
# Mod history:  2009-05-08 RSB   Adapted from the Colossus249/ file of the
#               same name, using Comanche055 page images/
#               2009-05-21 RSB   In IRIGZ, PRI017 corrected to PRI021.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#               Colossus 2A

# Page 297

BANK      7
SETLOC    IMUCOMP
BANK
EBANK=    NBDX

COUNT    06/ICOMP

1/PIPA    CAF      LGCOMP      # SAVE EBANK OF CALLING PROGRAM
          XCH      EBANK
          TS        MODE

          CCS      GCOMPSW      # BYPASS IF GCOMPSW NEGATIVE

```

	TCF	+3	
	TCF	+2	
	TCF	IRIG1	# RETURN
	INHINT		# ASSURE COMPLETE COMPENSATION OF DELV'S # FOR DOWNLINK.
1/PIPA1	CAF	FOUR	# PIPAZ, PIPAY, PIPAX
	TS	BUF +2	
	INDEX	BUF +2	
	CA	PIPASCF	# (P.P.M.) X 2(-9)
	EXTEND		
	INDEX	BUF +2	
	MP	DELVX	# (PP) X 2(+14) NOW (PIPA PULSES) X 2(+5)
	TS	Q	# SAVE MAJOR PART
	CA	L	# MINOR PART
	EXTEND		
	MP	BIT6	# SCALE 2(+9) SHIFT RIGHT 9
	INDEX	BUF +2	
	TS	DELVX +1	# FRACTIONAL PIPA PULSES SCALED 2(+14)
	CA	Q	# MAJOR PART
	EXTEND		
	MP	BIT6	# SCALE 2(+9) SHIFT RIGHT 9
	INDEX	BUF +2	
	DAS	DELVX	# (PIPAI) + (PIPAI)(SF)
	INDEX	BUF +2	
	CS	PIPABIAS	# (PIPA PULSES)/(CS) X 2(-8)
	EXTEND		
	MP	1/PIPADT	# (CS) X 2(+8) NOW (PIPA PULSES) X 2(+0)
	EXTEND		
	MP	BIT1	# SCALE 2(+14) SHIFT RIGHT 14
	INDEX	BUF +2	
	DAS	DELVX	# (PIPAI) + (PIPAI)(SFE) - (BIAS)(DELTAT)
	CCS	BUF +2	# PIPAZ, PIPAY, PIPAX
# Page 298	AD	NEG1	
	TCF	1/PIPA1 +1	
	NOOP		# LESS THAN ZERO IMPOSSIBLE
	RELINT		
# Page 299			

July 12, 2016

671

IRIGCOMP	TS	GCOMP SW	# INDICATE COMMANDS 2 PULSES OR LESS.
	TS	BUF	# INDEX COUNTER. IRIGX, IRIGY, IRIGZ.
IRIGX	EXTEND		
	DCS	DEL VX	# (PIPA PULSES) X 2(+14)
	DXCH	MPAC	
	CA	ADIA X	# (GYRO PULSES)/(PIPA PULSE) X 2(-3) *
	TC	GCOMP SUB	# -(ADIA X)(PIPA X) (GYRO PULSES) X 2(+14)
	EXTEND		#
	DCS	DEL VY	# (PIPA PULSES) X 2(+14)
	DXCH	MPAC	#
	CS	ADSRAX	# (GYRO PULSES)/(PIPA PULSE) X 2(-3) *
	TC	GCOMP SUB	# -(ADSRAX)(PIPA Y) (GYRO PULSES) X 2(+14)
#	EXTEND		# ***
#	DCS	DEL VY	# *** (PIPA PULSES) X 2(+14)
#	DXCH	MPAC	# ***
#	CA	ADOAX	# *** (GYRO PULSES)/(PIPA PULSE) X 2(-3) *
#	TC	GCOMP SUB	# *** -(ADOAX)(PIPA Z) (GYRO PULSES) X 2(+14)
	CS	NBDX	# (GYRO PULSES)/(CS) X 2(-3)
	TC	DRIFT SUB	# -(NBDX)(DELTAT) (GYRO PULSES) X 2(+14)
IRIGY	EXTEND		
	DCS	DEL VY	# (PIPA PULSES) X 2(+14)
	DXCH	MPAC	
	CA	ADIA Y	# (GYRO PULSES)/(PIPA PULSE) X 2(-3) *
	TC	GCOMP SUB	# -(ADIA Y)(PIPA Y) (GYRO PULSES) X 2(+14)
	EXTEND		
	DCS	DEL VZ	# (PIPA PULSES) X 2(+14)
	DXCH	MPAC	
	CS	ADSRAY	# (GYRO PULSES)/(PIPA PULSE) X 2(-3) *
	TC	GCOMP SUB	# +(ADSRAY)(PIPA Z) (GYRO PULSES) X 2(+14)
#	EXTEND		# ***
#	DCS	DEL VX	# *** (PIPA PULSES) X 2(+14)
#	DXCH	MPAC	# ***
#	CA	ADOAY	# *** (GYRO PULSES)/(PIPA PULS) X 2(-3) *
#	TC	GCOMP SUB	# *** -(ADOAY)(/PIPA X) (GYRO PULSES) X 2(+14)
	CS	NBDY	# (GYRO PULSES)/(CS) X 2(-5)
	TC	DRIFT SUB	# -(NBDY)(DELTAT) (GYRO PULSES) X 2(+14)
IRIGZ	EXTEND		

```

# Page 300
DCS      DELVY      # (PIPA PULSES) X 2(-14)
DXCH     MPAC
CA       ADSRAZ     # (GYRO PULSES)/(PIPA PULSE) X 2(-3)

TC       GCOMPSUB   # -(ADSRAZ)(PIPAY)      (GYRO PULSES) X 2(+14)

EXTEND
DCS      DELVZ      # (PIPA PULSES) X 2(+14)
DXCH     MPAC
CA       ADIAZ      # (GYRO PULSES)/(PIPA PULSE) X 2(-3)
TC       GCOMPSUB   # -(ADIAZ)(PIPAZ)      (GYRO PULSES) X 2(+14)

# EXTEND
# DCS      DELVX      # ***      (PIPA PULSE) X 2(+14)
# DXCH     MPAC      # ***
# CS       ADOAZ      # ***      (GYRO PULSES)/(PIPA PULSE) X 2(-3)
# TC       GCOMPSUB   # ***      +(ADOAZ)(PIPAZ) (GYRO PULSES) X 2(+14)

CA       NBDZ       #      (GYRO PULSES)/(CS) X 2(-5)
TC       DRIFTSUB   #      +(NBDZ)(DELTAT) (GYRO PULSES) X 2(+14)

# Page 301
CCS      GCOMPSW    # ARE GYRO COMMANDS GREATER THAN 2 PULSES
TCF      +2         # YES
TCF      IRIG1      # NO

CA       PRI021     # HIGHER THAN SERVICER -- LESS THAN PRELAUNCH
TC       NOVAC
EBANK=   NBDX
2CADR    1/GYRO

RELINT
IRIG1    CA       MODE      # SET EBANK FOR RETURN
         TS       EBANK
         TCF      SWRETURN

GCOMPSUB XCH      MPAC      # ADIA OR ADSRA COEFFICIENT ARRIVES IN A
EXTEND   MPAC      # C(MPAC) = (PIPA PULSES) X 2(+14)
MP       MPAC      # (GYRO PULSES)/(PIPA PULSE) X 2(-3)
DXCH     VBUF      # NOW = (GYRO PULSES) X 2(+11)

CA       MPAC +1    # MINOR PART OF PIPA PULSES
EXTEND
MP       MPAC      # ADIA OR ADSRA
TS       L
CAF      ZERO

```


July 12, 2016

673

```
DAS      VBUF      # NOW = (GYRO PULSES) X 2(+11)      *

CA      VBUF      # PARTIAL RESULT -- MAJOR
EXTEND
MP      BIT12     # SCALE 2(+3)  SHIFT RIGHT 3      *
INDEX   BUF       # RESULT = (GYRO PULSES) X 2(+14)
DAS     GCOMP     # HI(ADIA)(PIPAI) OR HI(ADSRA)(PIPAI)

CA      VBUF +1   # PARTIAL RESULT -- MINOR
EXTEND
MP      BIT12     # SCALE 2(+3)  SHIFT RIGHT 3      *
TS      L
CAF     ZERO
INDEX   BUF       # RESULT = (GYRO PULSES) X 2(+14)
DAS     GCOMP     # (ADIA)(PIPAI) OR (ADSRA)(PIPAI)

TC      Q

# Page 302
DRFTSUB  EXTEND
QXCH    BUF +1

EXTEND      # C(A) = NBD      (GYRO PULSES)/(CS) X 2(-5)
MP          # (CS) X 2(+8) NO (GYRO PULSES) X 2(+3)
LXCH       MPAC +1      # SAVE FOR FRACTIONAL COMPENSATION
EXTEND
MP          # SCALE 2(+11)  SHIFT RIGHT 11
INDEX      BUF
DAS        GCOMP      # HI(NBD)(DELTAT)      (GYRO PULSES) X 2(+14)

CA          # NOW MINOR PART
EXTEND
MP          # SCALE 2(+11)      SHIFT RIGHT 11
TS          L
CAF         ZERO
INDEX      BUF      # ADD IN FRACTIONAL COMPENSATION
DAS        GCOMP    # (NBD)(DELTAT)      (GYRO PULSES) X 2(+14)

DRFTSUB2  CAF      TWO      # PIPAX, PIPAY, PIPAZ
AD        BUF
XCH       BUF
INDEX     A
CCS       GCOMP      # ARE GYRO COMMANDS 1 PULSE OR GREATER
TCF       +2        # YES
TC        BUF +1    # NO
```

	MASK	NEGONE	
	CCS	A	# ARE GYRO COMMANDS GREATER THAN 2 PULSES
	TS	GCOMPSW	# YES -- SET GCOMPSW POSITIVE
	TC	BUF +1	# NO
# Page 303			
1/GYRO	CAF	FOUR	# PIPAZ, PIPAY, PIPAX
	TS	BUF	
	INDEX	BUF	# SCALE GYRO COMMANDS FOR IMUPULSE
	CA	GCOMP +1	# FRACTIONAL PULSES
	EXTEND		
	MP	BIT8	# SHIFT RIGHT 7
	INDEX	BUF	
	TS	GCOMP +1	# FRACTIONAL PULSES SCALED
	CAF	ZERO	# SET GCOMP = 0 FOR DAS INSTRUCTION
	INDEX	BUF	
	XCH	GCOMP	# GYRO PULSES
	EXTEND		
	MP	BIT8	# SHIFT RIGHT 7
	INDEX	BUF	
	DAS	GCOMP	# ADD THESE TO FRACTIONAL PULSES ABOVE
	CCS	BUF	# PIPAZ, PIPAY, PIPAX
	AD	NEG1	
	TCF	1/GYRO +1	
LGCOMP	ECADR	GCOMP	# LESS THAN ZERO IMPOSSIBLE
	CAF	LGCOMP	
	TC	BANKCALL	
	CADR	IMUPULSE	# CALL GYRO TORQUING ROUTINE
	TC	BANKCALL	
	CADR	IMUSTALL	# WAIT FOR PULSES TO GET OUT
	TCF	ENDOFJOB	# TEMPORARY
GCOMP1	CAF	FOUR	# PIPAZ, PIPAY, PIPAX
	TS	BUF	
	INDEX	BUF	# RESCALE
	CA	GCOMP +1	
	EXTEND		
	MP	BIT8	# SHIFT MINOR PART LEFT 7 -- MAJOR PART = 0
	INDEX	BUF	
	LXCH	GCOMP +1	# BITS 8-14 OF MINOR PART WERE = 0

July 12, 2016

675

	CCS	BUF	# PIPAZ, PIPAY, PIPAX
	AD	NEG1	
	TCF	GCOMP1 +1	
V06N30S	VN	0630	
	TCF	ENDOFJOB	
# Page 304			
NBDONLY	CCS	GCOMP5W	# BYPASS IF GCOMP5W NEGATIVE
	TCF	+3	
	TCF	+2	
	TCF	ENDOFJOB	
	INHINT		
	CCS	FLAGWRD2	# PREREAD T3RUPT MAY COINCIDE
	TCF	ENDOFJOB	
	TCF	ENDOFJOB	
	TCF	+1	
	CA	TIME1	# (CS) X 2(+14)
	XCH	1/PIPADT	# PREVIOUS TIME
	RELINT		
	COM		
	AD	1/PIPADT	
NBD2	CCS	A	# CALCULATE ELAPSED TIME.
	AD	ONE	# NO TIME1 OVERFLOW
	TCF	NBD3	# RESTORE TIME DIFFERENCE AND JUMP
	TCF	+2	# TIME1 OVERFLOW
	TCF	ENDOFJOB	# IF ELAPSED TIME = 0 (DIFFERENCE = -0)
	COM		# CALCULATE ABSOLUTE DIFFERENCE
	AD	POSMAX	
NBD3	EXTEND		# C(A) = DELTAT (CS) X 2(+14)
	MP	BIT10	# SHIFT RIGHT 5
	DXCH	VBUF	
	EXTEND		
	DCA	VBUF	
	DXCH	MPAC	# DELTAT NOW SCALED (CS) X 2(+19)
	CAF	ZERO	
	TS	GCOMP5W	# INDICATE COMMANDS 2 PULSES OR LESS.
	TS	BUF	# INDEX X, Y, Z.
	CS	NBDX	# (GYRO PULSES)/(CS) X 2(-5)
	TC	FBIASSUB	# -(NBOX)(DELTAT) (GYRO PULSES) X 2(+14)

	EXTEND		
	DCS	VBUF	
	DXCH	MPAC	# DELTAT SCALED (CS) X 2(+19)
	CA	NBDY	# (GYRO PULSES)/(CS) X 2(-5)
	TC	FBIASSUB	# -(NBDY)(DELTAT) (GYRO PULSES) X 2(+19)
	EXTEND		
	DCS	VBUF	
	DXCH	MPAC	# DELTAT SCALED (CS) X 2(+19)
	CS	NBDZ	# (GYRO PULSES)/(CS) X 2(-5)
# Page 305	TC	FBIASSUB	# +(NBDZ)(DELTAT) (GYRO PULSES) X 2(+19)
	CCS	GCOMPSW	# ARE GYRO COMMANDS GREATER THAN 2 PULSES
	TCF	1/GYRO	# YES
	TCF	ENDOFJOB	# NO
# Page 303			
FBIASSUB	XCH	Q	
	TS	BUF +1	
	CA	Q	# NBD SCALED (GYRO PULSES)/(CS) X 2(-5)
	EXTEND		
	MP	MPAC	# DELTAT SCALED (CS) X 2(+19)
	INDEX	BUF	
	DAS	GCOMP	# HI(NBD)(DELTAT) (GYRO PULSES) X 2(+19)
	CA	Q	# NO FRACTIONAL PART
	EXTEND		
	MP	MPAC +1	
	TS	L	
	CAF	ZERO	
	INDEX	BUF	
	DAS	GCOMP	# (NBD)(DELTAT) (GYRO PULSES) X 2(+19)
	TCF	DRFTSUB2	# CHECK MAGNITUDE OF COMPENSATION
LASTBIAS	TC	BANKCALL	
	CADR	PIPUSE	
	CCS	GCOMPSW	# BYPASS IF GCOMPSW NEGATIVE
	TCF	+3	
	TCF	+2	
	TCF	ENDOFJOB	
	CAF	PRI031	# 2 SECONDS SCALED (CS) X 2(+8)

July 12, 2016

677

```
XCH      1/PIPADT
COM
AD       PIPTIME1 +1      # TIME AT PIPA1 =0
TCF      NBD2
```

This code is written to file `src/IMU-COMPENSATION-PACKAGE.s`.

B.39 IMU MODE SWITCHING ROUTINES

```

678  <src/IMU-MODE-SWITCHING-ROUTINES.s 678>≡
# Copyright:      Public domain.
# Filename:       IMU_MODE_SWITCHING_ROUTINES.agc
# Purpose:       Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Ron Burkey <info@sandroid.org>.
# Website:       www.ibiblio.org/apollo.
# Pages:         1420-1448
# Mod history:   2009-05-10 SN      (Sergio Navarro). Started adapting from
#               the Colossus249/ file of the same name,
#               using Comanche055 page images.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#   Assemble revision 055 of AGC program Comanche by NASA
#   2021113-051.  10:28 APR. 1, 1969
#
#   This AGC program shall also be referred to as
#   Colossus 2A

# Page 1420

BLOCK      02
SETLOC     FFTAG3
BANK

EBANK=     COMMAND

# FIXED-FIXED ROUTINES

COUNT     02/IMODE

ZEROICDU   CAF      ZERO      # ZERO ICDU COUNTERS.
           TS       CDUX

```

July 12, 2016

679

```

        TS      CDUY
        TS      CDUZ
        TC      Q

SPSCODE      =      BIT9

# Page 1421
# IMU ZEROING ROUTINE.

        BANK    11
        SETLOC  MODESW
        BANK

        COUNT   07/IMODE

IMUZERO      INHINT
              CS      DSPTAB +11D
              MASK    BITS4&6
              CCS      A
              TCF      IMUZEROA

              TC      ALARM
              OCT      00206
              # IF SO.

              TCF      CAGETSTJ +4
              # IMMEDIATE FAILURE.

IMUZEROA     TC      CAGETSTJ

# DO ALL THE WORK.

              CS      IMODES33
              MASK    SUPER011
              ADS      IMODES33
              # DISABLE DAP AUTO AND HOLD MODES
              #      BIT5 FOR GROUND

              CS      IMODES30
              MASK    BITS3&4
              ADS      IMODES30
              # INHIBIT ICDUFAIL AND IMUFAIL (IN CASE WE
              # JUST CAME OUT OF COARSE ALIGN).

              CS      BITS4&6
              EXTEND
              WAND     CHAN12
              # SEND ZERO ENCODE WITH COARSE AND ERROR
              # COUNTER DISABLED.

              TC      NOATTOFF
              # TURN OFF NO ATT LAMP.

              CAF      BIT5
              EXTEND
```

```

                                WOR      CHAN12

                                TC        ZEROICDU
                                CAF        BIT6      # WAIT 320 MS TO GIVE AGS ADEQUATE TIME TO
                                TC        WAITLIST    # RECEIVE ITS PULSE TRAIN.
                                EBANK=     CDUIND
                                2CADR     IMUZERO2

                                CS        IMODES30    # SEE IF IMU OPERATING AND ALARM IF NOT.
                                MASK       BIT9
                                CCS        A
                                TCF        MODEEXIT

# Page 1422
                                TC        ALARM
                                OCT       210

MODEEXIT      RELINT            # GENERAL MODE-SWITCHING EXIT.
              TCF        SWRETURN

IMUZERO2      TC        CAGETEST
              TC        ZEROICDU      # ZERO CDUX, CDUY, CDUZ

              CS        BIT5          # REMOVE ZERO DISCRETE.
              EXTEND
              WAND       CHAN12

              CAF        BIT11        # WAIT 10 SECS FOR CTRS TO FIND GIMBALS
              TC        VARDELAY

IMUZERO3      TC        CAGETEST
              CS        BITS3&4      # REMOVE IMUFAIL AND ICDUFAIL INHIBIT.
              MASK      IMODES30
              TS        IMODES30

              CS        SUPER011     # ENABLE DAP AUTO AND HOLD MODES
              MASK      IMODES33     #          BIT5 FOR GROUND
              TS        IMODES33

              TC        IBNKCALL     # SET ISS WARNING IF EITHER OF ABOVE ARE
              CADR      SETISSW      # PRESENT.

              TCF        ENDIMU

```

Page 1423

IMU COARSE ALIGN MODE.

July 12, 2016

681

```
IMUCOARS      INHINT
               TC      CAGETSTJ
               TC      SETCOARS

               CAF      SIX
               TC      WAITLIST
               EBANK=   CDUIND
               2CADR   COARS

               TCF      MODEEXIT

COARS          TC      CAGETEST
               CAF      BIT6          # ENABLE ALL THREE ISS CDU ERROR COUNTERS
               EXTEND
               WOR      CHAN12

COARS1         CAF      TWO          # SET CDU INDICATOR
               TS      CDUIND

               INDEX   CDUIND        # COMPUTE THETAD -- THETAA IN 1'S
               CA      THETAD        #      COMPLEMENT FORM
               EXTEND
               INDEX   CDUIND
               MSU     CDUX
               EXTEND
               MP      BIT13          # SHIFT RIGHT 2
               XCH     L              # ROUND
               DOUBLE
               TS      ITEMP1
               TCF     +2
               ADS     L

               INDEX   CDUIND        # DIFFERENCE TO BE COMPUTED
               LXCH    COMMAND
               CCS     CDUIND
               TC      COARS1

               CAF      TWO          # MINIMUM OF 4 MS WAIT
               TC      VARDELAY

# Page 1424
COARS2         TC      CAGETEST      # DON'T CONTINUE IF CAGED.
               TS      ITEMP1        # SET TO +0.
               CAF      TWO          # SET CDU INDICATOR.
               TS      CDUIND
               +3
```

	INDEX	CDUIND	
	CCS	COMMAND	# NUMBER OF PULSES REQUIRED
	TC	COMPOS	# GREATER THAN MAX ALLOWED
	TC	NEXTCDU +1	
	TC	COMNEG	
	TC	NEXTCDU +1	
COMPOS	AD	-COMMAX	# COMMAX = MAX NUMBER OF PULSES ALLOWED
	EXTEND		# MINUS ONE
	BZMF	COMZERO	
	INDEX	CDUIND	
	TS	COMMAND	# REDUCE COMMAND BY MAX NUMBER OF PULSES
	CS	-COMMAX-	# ALLOWED
NEXTCDU	INCR	ITEMP1	
	AD	NEGO	
	INDEX	CDUIND	
	TS	CDUXCMD	# SET UP COMMAND REGISTER.
	CCS	CDUIND	
	TC	COARS2 +3	
	CCS	ITEMP1	# SEE IF ANY PULSES TO GO OUT.
	TCF	SENDPULS	
	TC	FIXDELAY	# WAIT FOR GIMBALS TO SETTLE.
	DEC	150	
CHKCORS	CAF	TWO	# AT END OF COMMAND, CHECK TO SEE THAT
	TS	ITEMP1	# GIMBALS ARE WITHIN 2 DEGREES OF THETAD.
	INDEX	A	
	CA	CDUX	
	EXTEND		
	INDEX	ITEMP1	
	MSU	THETAD	
	CCS	A	
	TCF	COARSERR	
	TCF	CORSCHK2	
	TCF	COARSERR	
# Page 1425			
CORSCHK2	CCS	ITEMP1	
	TCF	CHKCORS	
	TCF	ENDIMU	# END OF COARSE ALIGNMENT
COARSERR	AD	COARSTOL	# 2 DEGREES.

```

EXTEND
BZMF    CORSCHK2

TC      ALARM      # COARSE ALIGN ERROR.
OCT     211

TCF     IMUBAD

COARSTOL DEC    -.01111      # 2 DEGREES SCALED AT HALF-REVOLUTIONS

COMNEG  AD      -COMMAX
EXTEND
BZMF    COMZERO
COM
INDEX   CDUIND
TS      COMMAND
CA      -COMMAX-
TC      NEXTCDU

COMZERO CAF      ZERO
INDEX   CDUIND
XCH     COMMAND
TC      NEXTCDU

SENDPULS CAF     13,14,15
EXTEND
WOR     CHAN14
CAF     600MS
TCF     COARS2 -1      # THEN TO VARDELAY

CA+ECE  CAF      BIT6      # ENABLE ALL THREE ISS CDU ERROR COUNTERS
EXTEND
WOR     CHAN12
TC      TASKOVER

# Page 1426
SETCOARS CAF     BIT4      # BYPASS IF ALREADY IN COARSE ALIGN
EXTEND
RAND    CHAN12
CCS     A
TC      Q

CS      BIT6      # CLEAR ISS ERROR COUNTERS
EXTEND
WAND    CHAN12

```

	CS	BIT10	# KNOCK DOWN GYRO ACTIVITY
	EXTEND		
	WAND	CHAN14	
	CS	ZERO	
	TS	GYROCMD	
	CAF	BIT4	# PUT ISS IN COARSE ALIGN
	EXTEND		
	WOR	CHAN12	
	CS	DSPTAB +11D	# TURN ON NO ATT LAMP
	MASK	OCT40010	
	ADS	DSPTAB +11D	
	CS	IMODES33	# DISABLE DAP AUTO AND HOLD MODES
	MASK	BIT6	
	ADS	IMODES33	
	CS	IMODES30	# DISABLE IMUFAIL
	MASK	BIT4	
	ADS	IMODES30	
RNDREFDR	CS	BIT5	# KNOCK DOWN TRACK FLAG
	MASK	FLAGWRD1	
	TS	FLAGWRD1	
	CS	BIT15	# KNOCK DOWN DRIFT FLAG
	MASK	FLAGWRD2	
	TS	FLAGWRD2	
	CS	BIT13	# KNOCK DOWN REFSMMAT FLAG
	MASK	FLAGWRD3	
	TS	FLAGWRD3	
	TC	Q	
OCT40010	OCT	40010	
# Page 1427			
# IMU FINE ALIGN MODE SWITCH.			
IMUFINE	INHINT		
	TC	CAGETSTJ	# SEE IF IMU BEING CAGED.
	CS	BITS4-5	# RESET ZERO AND COARSE
	EXTEND		

July 12, 2016

685

	WAND	CHAN12	
	CS	BIT6	# INSURE DAP AUTO AND HOLD MODES ENABLED
	MASK	IMODES33	
	TS	IMODES33	
	TC	NOATTOFF	
	CAF	BIT10	# IMU FAIL WAS INHIBITED DURING THE
	TC	WAITLIST	# PRESUMABLY PRECEDING COARSE ALIGN. LEAVE
	EBANK=	CDUIND	
	2CADR	IFAILOK	# IT ON FOR THE FIRST 5 SECS OF FINE ALIGN
	CAF	2SECS	
	TC	WAITLIST	
	EBANK=	CDUIND	
	2CADR	IMUFINED	
	TCF	MODEEXIT	
IMUFINED	TC	CAGETEST	# SEE THAT NO ONE HAS CAGED THE IMU.
	TCF	ENDIMU	
# Page 1428			
IFAILOK	TC	CAGETSTQ	# ENABLE IMU FAIL UNLESS IMU BEING CAGED.
	TCF	TASKOVER	# IT IS.
	CAF	BIT4	# DON'T RESET IMU FAIL INHIBIT IF SOMEONE
	EXTEND		# HAS GONE INTO COARSE ALIGN.
	RAND	CHAN12	
	CCS	A	
	TCF	TASKOVER	
	CS	IMODES30	# RESET IMUFAIL.
	MASK	BIT13	
	ADS	IMODES30	
	CS	BIT4	
PFAILOK2	MASK	IMODES30	
	TS	IMODES30	
	TC	IBNKCALL	# THE ISS WARNING LIGHT MAY COME ON NOW
	CADR	SETISSW	# THAT THE INHIBIT WAS BEEN REMOVED.
	TCF	TASKOVER	
PFAILOK	TC	CAGETSTQ	# ENABLE PIP FAIL PROG ALARM.
	TCF	TASKOVER	

	CS	IMODES30	# RESET IMU AND PIPA FAIL BITS.
	MASK	BIT10	
	ADS	IMODES30	
	CS	IMODES33	
	MASK	BIT13	
	ADS	IMODES33	
	CS	BIT5	
	TCF	PFAILOK2	
NOATTOFF	CS	OCT40010	# SUBROUTINE TO TURN OFF NO ATT LAMP.
	MASK	DSPTAB +11D	
	AD	BIT15	
	TS	DSPTAB +11D	
	TC	Q	
# Page 1429			
# ROUTINES TO INITIATE AND TERMINATE PROGRAM USE OF THE PIPAS. NO IMUSTALL REQUIRED			
PIPUSE	CS	ZERO	
	TS	PIPAX	
	TS	PIPAY	
	TS	PIPAZ	
PIPUSE1	TC	CAGETSTQ	# DO NOT ENABLE PIPA FAIL IF IMU IS CAGED
	TCF	SWRETURN	
	INHINT		
	CS	BIT1	# IF PIPA FAILS FROM NOW ON (UNTIL
	MASK	IMODES30	# PIPFREE), LIGHT ISS WARNING.
	TS	IMODES30	
PIPFREE2	TC	IBNKCALL	# ISS WARNING MIGHT COME ON NOW.
	CADR	SETISSW	# (OR GO OFF ON PIPFREE).
	TCF	MODEEXIT	
PIPFREE	INHINT		# PROGRAM DONE WITH PIPAS. DON'T LIGHT
	CS	IMODES30	# ISS WARNING.
	MASK	BIT1	
	ADS	IMODES30	
	MASK	BIT10	# IF PIP FAIL ON, DO PROG ALARM AND RESET
	CCS	A	# ISS WARNING.
	TCF	MODEEXIT	

July 12, 2016

687

TC ALARM
OCT 212

INHINT

TCF PIPFREE2

Page 1430

THE FOLLOWING ROUTINE TORQUES THE IRIGS ACCORDING TO DOUBLE PRECISION INPUTS IN THE SIX REGIS
BEGINNING AT THE ECADR ARRIVING IN A. THE MINIMUM SIZE OF ANY PULSE TRAIN IS 16 PULSES (.25
UNSENT PORTION OF THE COMMAND IS LEFT INTACT AT TEH INPUT COMMAND REGISTERS.

EBANK= 1400 # VARIABLE, ACTUALLY.

IMUPULSE TS MPAC +5 # SAVE ARRIVING ECADR.
 TC CAGETSTJ # DON'T PROCEED IF IMU BEING CAGED.

CCS LGYRO # SEE IF GYROS BUSY.
TC GYROBUSY # SLEEP.

TS MPAC +2
CAF BIT6 # ENABLE THE POWER SUPPLY.
EXTEND
WOR CHAN14

GWAKE2 CAF FOUR
 TC WAITLIST # (IF A JOB WAS PUT TO SLEEP, THE POWER
EBANK= CDUIND # SUPPLY IS LEFT ON BY THE WAKING JOB).
2CADR STRITGYRO

CA MPAC +5 # SET UP EBANK, SAVING CALLER'S EBANK FOR
XCH EBANK # RESTORATION ON RETURN.
XCH MPAC +5
TS LGYRO # RESERVES GYROS.
MASK LOW8
TS ITEMP1

GYROAGRE CAF TWO # FORCE SIGN AGREEMENT ON INPUTS.
 TS MPAC +3
DOUBLE
AD ITEMP1
TS MPAC +4
EXTEND
INDEX A
DCA 1400

DXCH MPAC
 TC TPAGREE
 DXCH MPAC
 INDEX MPAC +4
 DXCH 1400

CCS MPAC +3
 TCF GYROAGRE

CA MPAC +5 # RESTORE CALLER'S EBANK.
 TS EBANK
 TCF MODEEXIT

Page 1431

ROUTINES TO ALLOW TORQUING ONLY ONE JOB AT A TIME.

GYROBUSY EXTEND # SAVE RETURN 2FCADR.

DCA BUF2

DXCH MPAC

REGSLEEP CAF LGWAKE

TCF JOBSLEEP

GWAKE CCS LGYRO # WHEN AWAKENED, SEE IF GYROS STILL BUSY.
 TCF REGSLEEP # IF SO, SLEEP SOME MORE.

TS MPAC +2

EXTEND

DCA MPAC

DXCH BUF2

CAF ONE

TCF GWAKE2

RESTORE SWRETURN INFO.

LGWAKE CADR GWAKE

Page 1432

GYRO-TORQUING WAITLIST TASKS.

STRITGYRO CS GDESELCT # DE-SELECT LAST GYRO.

EXTEND

WAND CHAN14

TC CAGETEST

STRITGYR2 CA LGYRO # JUMP ON PHASE COUNTER IN BITS 13-14.

EXTEND

MP BIT4

	INDEX	A	
	TCF	+1	
	TC	GSELECT	# =0. DO Y GYRO.
	OCT	00202	
	TC	GSELECT	# =1. DO Z GYRO.
	OCT	00302	
	TC	GSELECT -2	# =2. DO X GYRO.
	OCT	00100	
	CAF	ZERO	# =3. DONE
	TS	LGYRO	
	CAF	LGWAKE	# WAKE A POSSIBLE SLEEPING JOB.
	TC	JOBWAKE	
NORESET	TCF	IMUFINED	# DO NOT RESET POWER SUPPLY.
# Page 1433			
-2	CS	FOUR	# SPECIAL ENTRY TO REGRESS LGYRO FOR X.
	ADS	LGYRO	
GSELECT	INDEX	Q	# SELECT GYRO.
	CAF	0	# PACKED WORD CONTAINS GYRO SELECT BITS
	TS	ITEMP4	# AND INCREMENT TO LGYRO.
	MASK	SEVEN	
	AD	BIT13	
	ADS	LGYRO	
	TS	EBANK	
	MASK	LOW8	
	TS	ITEMP1	
	CS	SEVEN	
	MASK	ITEMP4	
	TS	ITEMP4	
	EXTEND		# MOVE DP COMMAND TO RUPTREGS FOR TESTING.
	INDEX	ITEMP1	
	DCA	1400	
	DXCH	RUPTREG1	
	CCS	RUPTREG1	
	TCF	MAJ+	
	TCF	+2	
	TCF	MAJ-	
	CCS	RUPTREG2	

	TCF	MIN+	
	TCF	STRTGYR2	
	TCF	MIN-	
	TCF	STRTGYR2	
# Page 1434			
MIN+	AD	-GYROMIN	# SMALL POSITIVE COMMAND. SEE IF AT LEAST
	EXTEND		# 16 GYRO PULSES.
	BZMF	STRTGYR2	
MAJ+	EXTEND		# DEFINITE POSITIVE OUTPUT.
	DCA	GYROFRAC	
	DAS	RUPTREG1	
	CA	ITEMP4	# SELECT POSITIVE TORQUING FOR THIS GYRO.
	EXTEND		
	WOR	CHAN14	
	CAF	LOW7	# LEAVE NUMBER OF POSSIBLE 8192 AUGMENTS
	MASK	RUPTREG2	# TO INITIAL COMMAND IN MAJOR PART OF LONG
	XCH	RUPTREG2	# TERM STORAGE AND TRUNCATED FRACTION
GMERGE	EXTEND		# IN MINOR PART. THE MAJOR PART WILL BE
	MP	BIT8	# COUNTED DOWN TO ZERO IN THE COURSE OF
	TS	ITEMP2	# PUTTING OUT THE ENTIRE COMMAND.
	CA	RUPTREG1	
	EXTEND		
	MP	BIT9	
	TS	RUPTREG1	
	CA	L	
	EXTEND		
	MP	BIT14	
	ADS	ITEMP2	# INITIAL COMMAND.
	EXTEND		# SEE IF MORE THAN ONE PULSE TRAIN NEEDED
	DCA	RUPTREG1	# (MORE THAN 16383 PULSES).
	AD	MINUS1	
	CCS	A	
	TCF	LONGGYRO	
-GYROMIN	OCT	-176	# MAY BE ADJUSTED TO SPECIFY MINIMUM CMD
	TCF	+4	
	CAF	BIT14	
	ADS	ITEMP2	
	CAF	ZERO	
+4	INDEX	ITEMP1	

July 12, 2016

691

```
# Page 1435
LASTSEG      DXCH      1400
              CA        ITEMP2      # ENTIRE COMMAND.
              TS        GYROCMD
              EXTEND
              MP        BIT10      # WAITLIST DT
              AD        THREE      # TRUNCATION AND PHASE UNCERTAINTIES.
              TC        WAITLIST
              EBANK=    CDUIND
              2CADR     STRTGYRO

GYROEXIT     CAF        BIT10
              EXTEND
              WOR       CHAN14
              TCF       TASKOVER

LONGGYRO     INDEX     ITEMP1
              DXCH      1400      # INITIAL COMMAND OUT PLUS N AUGMENTS OF
              CAF       BIT14      # 8192.  INITIAL COMMAND IS AT LEAST 8192.
              AD        ITEMP2
              TS        GYROCMD

AUG3         EXTEND
              MP        BIT10      # GET WAITLIST DT TO TIME WHEN TRAIN IS
              AD        NEG3      # ALMOST OUT.
              TC        WAITLIST
              EBANK=    CDUIND
              2CADR     8192AUG

              TCF       GYROEXIT

8192AUG      TC        CAGETEST

              CAF       BIT4
              EXTEND
              RAND      CHAN12
              CCS       A
              TCF       IMUBAD
              CA        LGYRO      # ADD 8192 PULSES TO GYROCMD
              TS        EBANK
              MASK      LOW8
              TS        ITEMP1

              INDEX     ITEMP1      # SEE IF THIS IS THE LAST AUG.
              CCS       1400
              TCF       AUG2      # MORE TO COME.
```

	CAF	BIT14	
	ADS	GYROCMD	
	TCF	LASTSEG +1	
# Page 1436			
AUG2	INDEX	ITEMP1	
	TS	1400	
	CAF	BIT14	
	ADS	GYROCMD	
	TCF	AUG3	# COMPUTE DT.
# Page 1437			
MIN-	AD	-GYROMIN	# POSSIBLE NEGATIVE OUTPUT.
	EXTEND		
	BZMF	STRGTYR2	
MAJ-	EXTEND		# DEFINITE NEGATIVE OUTPUT.
	DCS	GYROFRAC	
	DAS	RUPTREG1	
	CA	ITEMP4	# SELECT NEGATIVE TORQUING FOR THIS GYRO.
	AD	BIT9	
	EXTEND		
	WOR	CHAN14	
	CS	RUPTREG1	# SET UP RUPTREGS TO FALL INTO GMERGE.
	TS	RUPTREG1	# ALL NUMBERS PUT INTO GYROCMD ARE
	CS	RUPTREG2	# POSITIVE -- BIT9 OF CHAN 14 DETERMINES
	MASK	LOW7	# THE SIGN OF THE COMMAND.
	COM		
	XCH	RUPTREG2	
	COM		
	TCF	GMERGE	
GDESELECT	OCT	1700	# TURN OFF SELECT AND ACTIVITY BITS.
GYROFRAC	2DEC	.215 B-21	
# Page 1438			
# IMU MODE SWITCHING ROUTINES COME HERE WHEN ACTION COMPLETE.			
ENDIMU	EXTEND		# MODE IS BAD IF CAGE HAS OCCURRED OR IF
	READ	DSALMOUT	# ISS WARNING IS ON.
	MASK	BIT1	
	CCS	A	

	TCF	IMUBAD	
IMUGOOD	TCF	GOODEND	# WITH C(A) = 0.
IMUBAD	CAF	ZERO	
	TCF	BADEND	
CAGETEST	CAF	BIT6	# SUBROUTINE TO TERMINATE IMU MODE
	MASK	IMODES30	# SWITCH IF IMU HAS BEEN CAGED.
	CCS	A	
	TCF	IMUBAD	# DIRECTLY.
	TC	Q	# WITH C(A) = +0.
CAGETSTQ	CS	IMODES30	# SKIP IF IMU NOT BEING CAGED.
	MASK	BIT6	
	CCS	A	
	INCR	Q	
	TC	Q	
CAGETSTJ	CS	IMODES30	# IF DURING MODE SWITCH INITIALIZATION.
	MASK	BIT6	# IT IS FOUND THAT THE IMU IS BEING CAGED.
	CCS	A	# SET IMUCADR TO -0 TO INDICATE OPERATION
	TC	Q	# COMPLETE BUT FAILED. RETURN IMMEDIATELY
	CS	ZERO	# TO SWRETURN.
	TS	IMUCADR	
	TCF	MODEEXIT	

Page 1439

GENERALIZED MODE SWITCHING TERMINATION. ENTER AT GOODEND FOR SUCCESSFUL COMPLETION OF AN I/O
 # OR AT BADEND FOR AN UNSUCCESSFUL ONE. C(A) OR ARRIVAL =0 FOR IMU, 1 FOR OPTICS.

BADEND	TS	RUPTREG2	# DEVICE INDEX.
	CS	ZERO	# FOR FAILURE.
	TCF	GOODEND +2	
GOODEND	TS	RUPTREG2	
	CS	ONE	# FOR SUCCESS.
	TS	RUPTREG3	
	INDEX	RUPTREG2	# SEE IF USING PROGRAM ASLEEP.
	CCS	MODECADR	
	TCF	+2	# YES -- WAKE IT UP.
	TCF	ENDMODE	# IF 0, PROGRAM NOT IN YET.
	CAF	ZERO	# WAKE SLEEPING PROGRAM.
	INDEX	RUPTREG2	

```

XCH      MODECADR
TC        JOBWAKE

CS        RUPTREG3      # ADVANCE LOC IF SUCCESSFUL.
INDEX     LOCCTR
ADS       LOC

TCF       TASKOVER

ENDMODE   CA        RUPTREG3      # -0 INDICATES OPERATION COMPLETE BUT
INDEX     RUPTREG2      # UNSUCCESSFUL: -1 INDICATES COMPLETE AND
TS        MODECADR      # SUCCESSFUL.
TCF       TASKOVER

# Page 1440
# GENERAL STALLING ROUTINE.  USING PROGRAMS COME HERE TO WAIT FOR I/O COMPLETION.
#
# PROGRAM DESCRIPTION                                DATE -- 21 FEB 1967
#                                                    LOG SECTION IMU MODE SWITCHING
# MOD BY -- R. MELANSON TO ADD DOCUMENTATION      ASSEMBLY SUNDISK REV. 82
#
# FUNCTIONAL DESCRIPTION --
#   TO DELAY FURTHER EXECUTION OF THE CALLING ROUTINE UNTIL ITS SELECTED
#   I/O FUNCTION IS COMPLETE.  THE FOLLOWING CHECKS ON THE CALLING ROUTINES
#   MODEECADR ARE MADE AND ACTED UPON.
#   1) +0 INDICATES INCOMPLETE I/O OPERATION.  CALLING ROUTINE IS PUT TO
#   SLEEP.
#   2) -1 INDICATES COMPLETED I/O OPERATION.  STALL BYPASSES JOBSLEEP
#   CALL AND RETURNS TO CALLING ROUTINE AT L+3.
#   3) -0 INDICATES COMPLETED I/O WITH FAILURE.  STALL CLEARS MODECADR
#   AND RETURNS TO CALLING ROUTINE AT L+2.
#   4) VALUE GREATER THAN 0 INDICATES TWO ROUTINES CALLING FOR USE OF
#   SAME DEVICE.  STALL EXITS TO ABORT WHICH EXECUTES A PROGRAM
#   RESTART WHICH IN TURN CLEARS ALL MODECADR REGISTERS.
#
# CALLING SEQUENCE --
#   L      TC        BANKCALL
#   L+1    CADR      (ONE OF 5 STALL ADDRESSES.  I.E., IMUSTALL, OPTSTALL, RADSTALL,
#                   AOTSTALL, OR ATTSTALL)
#
# NORMAL-EXIT MODE --
#   TCF JOBSLEEP    OR      TCF MODEEXIT
#
# ALARM OR ABORT EXIT MODE --
#   TC      ABORT
#

```

```

# OUTPUT --
#     MODECADR=CADR    IF JOBSLEEP
#     MODECADR=+0      IF I/O COMPLETE
#     BUF2=L+3         IF I/O COMPLETE AND GOOD.
#     BUF2=L+2         IF I/O COMPLETE BUT FAILED.
#
# ERASABLE INITIALIZATION --
#     BUF2 CONTAINS RETURN ADDRESS PLUS 1,(L+2)
#     BUF2+1 CONTAINS FBANK VALUE OF CALLING ROUTINE.
#     MODECADR OF CALLING ROUTINE CONTAINS +0,-1,-0 OR CADR RETURN ADDRESS.
#
# DEBRIS --
#     RUPTREG2 AND CALLING ROUTINE MODECADR.

AOTSTALL      CAF      ONE          # AOT.
               TC       STALL

RADSTALL      CAF      TWO
               TCF      STALL

# Page 1441
OPTSTALL      EQUALS   AOTSTALL

IMUSTALL      CAF      ZERO         # IMU.

STALL         INHINT
               TS       RUPTREG2    # SAVE DEVICE INDEX.
               INDEX    A           # SEE IF OPERATION COMPLETE.
               CCS      MODECADR
               TCF      MODABORT     # ALLOWABLE STATES ARE +0, -1, AND -0.
               TCF      MODESLP      # OPERATION INCOMPLETE.
               TCF      MODEGOOD     # COMPLETE AND GOOD IF = -1.

MG2           INDEX    RUPTREG2    # COMPLETE FAILED IF -0.  RESET TO +0.
               TS       MODECADR    # RETURN TO CALLER.
               TCF      MODEEXIT

MODEGOOD      CCS      A           # MAKE SURE INITIAL STATE -1.
               TCF      MODABORT

               INCR     BUF2        # IF SO, INCREMENT RETURN ADDRESS AND
               TCF      MG2         # RETURN IMMEDIATELY, SETTIN CADR = +0.

MODESLP       TC       MAKECADR    # CALL FROM SWITCHABLE FIXED ONLY.
               INDEX    RUPTREG2
               TS       MODECADR

```

```

                                TCF      JOBSLEEP

MODABORT      TC      POOD00      # TWO PROGRAMS USING THE SAME DEVICE.
                                OCT      1210

```

```
# Page 1442
```

```
# CONSTANTS FOR MODE SWITCHING ROUTINES
```

```

BITS3&4      =      OCT14
BITS4&6      =      OCT50
BITS4-5      EQUALS  BITS4&5
IMUSEFLG     EQUALS  BIT8      # INTERPRETER SWITCH 7.
-COMMAX      DEC      -191
-COMMAX-     DEC      -192
600MS        DEC      60
IMUFIN20     =      IMUFINE
GOMANUR      CA      ATTCADR    # IS KALCMANU FREE
                                EXTEND
                                BZF      +3
                                TC      POOD00    # NO
                                OCT      1210    # 2 TRYING TO USE SAME DEVICE

+3           EXTEND
                                DCA      BUF2
                                DXCH     ATTCADR    # SAVE FINAL RETURN FOR KALCMAN3

                                CA      BBANK
                                MASK     SEVEN
                                ADS      ATTCADR +1

                                CA      PRIORITY
                                MASK     PRI037
                                TS      ATTPRIO    # SAVE USER'S PRIO

                                CAF      KALEBCON   # SET EBANK FOR KALCMAN3
                                TS      EBANK
                                TC      POSTJUMP
                                CADR     KALCMAN3
KALEBCON     ECADR     BCDU

```

```
# Page 1443
```

```
# PROGRAM DESCRIPTION
```

```
# IMU STATUS CHECK ROUTINE R02 (SUBROUTINE UTILITY)
```

```
# MOD NO -- 1
```

```
# MOD BY -- N.BRODEUR
```

```
#
```



```

# FUNCTIONAL DESCRIPTION
#     TO CHECK WHETHER IMU IS ON AND IF ON WHETHER IT IS ALIGNED TO AN
#     ORIENTATION KNOWN BY THE CMC. TO REQUEST SELECTION OF THE APPROPRIATE
#     PROGRAM IF THE IMU IS OFF OR NOT ALIGNED TO AN ORIENTATION KNOWN BY THE
#     CMC. CALLED THROUGH BANKCALL
#
# CALLING SEQUENCE --
#     L      TC      BANKCALL
#     L+1    CADR    R02BOTH
#
# SUBROUTINES CALLED
#     VARALARM
#     FLAGUP
#
# NORMAL EXIT MODES
#     AT L+2 OF CALLING SEQUENCE
#
# ALARM OR ABORT EXIT MODES
#     GOTOPPOH, WITH ALARM
#
# ERASABLE INITIALIZATION REQUIRED
#     NONE
#
# DEBRIS
#     CENTRALS -- A,Q,L

BANK      34
SETLOC    R02
BANK
COUNT    04/R02      # COUNT*

DEC51     DEC      51
R02BOTH   CAF      BIT13
          MASK     STATE +3      # REFSMFLG
          CCS      A
          TC       R02ZERO      # ZERO IMUS

          CA       IMODES30
          MASK     BIT9          # IS ISS INITIALIZED
          EXTEND
          BZF      +2
          CS       BIT4          # SEND IMU ALARM CODE 210
          AD       OCT220        # SEND REFSMM ALARM
          TC       VARALARM

          TC       GOTOPPOH

```

Page 1444

R02ZERO	TC	UPFLAG
	ADRES	IMUSE
	TCF	SWRETURN
OCT220	OCT	220

Page 1445

PROGRAM DESCRIPTION: P06 10 FEB 67

#

TRANSFER THE ISS/CMC FROM THE OPERATE TO THE STANDBY CONDITION.

#

THE NORMAL CONDITION OF READINESS OF THE GNCS WHEN NOT IN USE IS STANDBY. IN THIS
 # HEATER POWER IS ON. THE IMU OPERATE POWER IS OFF. THE COMPUTER POWER IS ON. THE
 # CMC STANDBY ON THE MAIN AND LEB DISKYS IS ON.

#

CALLING SEQUENCE:

ASTRONAUT REQUEST THROUGH DSKY V37E 06E.

#

SUBROUTINES CALLED:

#	GOPERF1
#	BANKCALL
#	FLAGDOWN

#

Page 1446

PRESTAND PREPARES FOR STANDBY BY SNAPSHOTTING THE SCALER AND TIME1 TIME2.
 # THE LOW 5 BITS OF THE SCALER ARE INSPECTED TO INSURE COMPATIBILITY
 # BETWEEN THE SCALER READING AND THE TIME1 TIME2 READING.

SETLOC	P05P06
BANK	

EBANK=	TIME2SAV
COUNT*	\$\$/P06

P06	TC	UPFLAG	# SET NODOV37 BIT
	ADRES	NODOFLAG	

PRESTAND	INHINT		
	EXTEND		
	DCA	TIME2	# SNAPSHOT TIME1 TIME2
	DXCH	TIME2SAV	
	TC	SCALPREP	
	TC	PRESTAND	# T1,T2,SCALER NOT COMPATIBLE
	DXCH	MPAC	# T1,T2 AND SCALER OK
	DXCH	SCALSAVE	# STORE SCALER

```

INHINT
TC      BANKCALL
CADR    RNDREFDR      # REFSMM, DRIFT, TRACK FLAGS DOWN

TC      DOWNFLAG
ADRES   IMUSE          # IMUSE DOWN
TC      DOWNFLAG
ADRES   RNDVZFLG      # RNDVZFLG DOWN

CAF      BIT11
EXTEND
WOR      CHAN13        # SET STANDBY ENABLE BIT

TC      PHASCHNG      # SET RESTART TO POSTAND WHEN STANDBY
OCT      07024        # RECOVERS
OCT      20000
EBANK=   SCALSAVE
2CADR    POSTAND

CAF      OCT62
TC      BANKCALL
CADR     GOPERF1
TCF      -3
TCF      -4
TCF      -5

```

```

OCT62      EQUALS .5SEC      # DEC 50 = OCT 62

```

```

# THE LOW 5 BITS OF THE SCALER READS 10000 FOR THE FIRST INTERVAL AFTER A
# Page 1447.
# T1 INCREMENT. IF SCALPREP DETECTS THIS INTERVAL THE T1,T2 AND SCALER
# DATA ARE NOT COMPATIBLE AND RETURN IS TO L+1 FOR ANOTHER READING OF THE
# DATA. OTHERWISE, THE RETURN IS TO L+2 TO PROCEED. ROUTINE ALSO PREPARES
# THE SCALER READING FOR COMPUTATION OF THE INCREMENT TO UPDATE T1T2. (THE
# 10 MS BIT (BIT 6) OF THE SCALER IS INCREMENTED 5 MS OUT OF PHASE FROM
# T1.) ADDITION OF 5 MS (BIT 5) TO THE SCALER READING HAS THE EFFECT OF
# ADJUSTING BIT 6 IN THE SCALER TO BE IN PHASE WITH BIT 1 OF T1. THE LOW 5
# BITS OF THE SCALER READING ARE THEN SET TO ZERO, TO TRUNCATE THE SCALER
# DATA TO 10 MS. RESULTS ARE STORED IN MPAC, +1.

```

```

SCALPREP      EXTEND
                QXCH      MPAC +2
                TC        FINETIME +1
                RELINT
                DXCH      MPAC
                CA        BIT5      # ADD 5 MS TO THE SCALER READING.

```

```

TS      L
CA      ZERO
DAS     MPAC
CS      LOW5      # SET LOW 5 BITS OF (SCALER+5MS) TO ZERO
MASK    MPAC +1   # AND STORE RESULTS IN MPAC,+1.
XCH     MPAC +1
MASK    LOW5      # TEST LOW 5 BITS OF SCALER FOR THE FIRST
                  # INTERVAL AFTER THE T1 INCREMENT
                  # (NOW = 00000, SINCE BIT 5 ADDED).
                  # IS IT 1ST INTERVAL AFTER T1 INCREMENT
CCS     A          # NO
INCR    MPAC +2    # YES
TC      MPAC +2

# POSTAND RECOVERS TIME AFTER STANDBY.  THE SCALER IS SNAPSHOTTED AND THE
# TIME1 TIME2 COUNTER IS SET TO ZERO.  THE LOW 5 BITS OF THE SCALER ARE
# INSPECTED TO INSURE COMPATIBILITY BETWEEN THE SCALER READING AND THE
# CLEARING OF THE TIME COUNTER.  IT THEN COMPUTES THE DIFFERENCE IN SCALER
# VALUES (IN DP) AND ADDS THIS TO THE PREVIOUSLY SNAPSHOTTED VALUES OF
# TIME1 TIME2 AND PLACES THIS NEW TIME INTO THE TIME1 TIME2 COUNTER.

COUNT*  $$/P05

POSTAND  CS      BIT11      # RECOVER TIME AFTER STANDBY
EXTEND
WAND     CHAN13      # CLEAR STANDBY ENABLE BIT
INHINT
CA      ZERO
TS      L
DXCH     TIME2      # CLEAR TIME1 TIME2
TC      SCALPREP    # STORE SCALER IN MPAC, MPAC+1
TC      POSTAND +3   # T1,T2,SCALER NOT COMPATIBLE
EXTEND   # T1,T2 AND SCALER OK
DCS     SCALSAVE
DAS     MPAC      # FORM DP DIFFERENCE OF POST-STANDBY SCALER

# Page 1448
CAF     BIT10      # MINUS PRE-STANDBY SCALER AND SHIFT RIGHT
TC      SHORTMP    # 5 TO ALIGN BITS WITH TIME1 TIME2.
CAF     ZERO
TS      MPAC +2    # NEEDED FOR TP AGREE
TC      TPAGREE    # MAKE DP DIFF AGREE
CCS     MPAC
TC      POSTCOM    # IF DP DIFF NET +, NO SCALER OVERFLOW
TC      POSTCOM    # BETWEEN PRE AND POST STANDBY.
TC      +1         # IF DP DIFF NET -, SCALER OVERFLOWED.  ADD
CAF     BIT10      # BIT 10 TO HIGH DIFF TO CORRECT.
ADS     MPAC

```

July 12, 2016

701

```
POSTCOM      EXTEND      # C(MPAC,+1) IS MAGNITUDE OF DELTA SCALER.
              DCA        TIME2SAV      # PRE-STANDBY TIME1 TIME2
              DAS        MPAC
              TC         TPAGREE        # FORCE SIGN AGREEMENT
              DXCH       MPAC           # UPDATED VALUE FOR T1,T2.
              DAS        TIME2         # LOAD UPDATED VALUE INTO T1,T2, WITH
              TC         DOWNFLAG       # CLEAR NODOFLAG
              ADRES      NODOFLAG

              TC         GOTOP00H
```

This code is written to file `src/IMU-MODE-SWITCHING-ROUTINES.s`.

B.40 IMU PERFORMANCE TEST 2

```

702  <src/IMU-PERFORMANCE-TEST-2.s 702>≡
      # Copyright:    Public domain.
      # Filename:     IMU_PERFORMANCE_TEST_2.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         373-381
      # Mod history:   2009-05-17 RSB   Adapted from the corresponding
      #               Luminary131 file, using page
      #               images from Luminary 1A.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969
      #
      # Page 373
      # NAME --        IMU PERFORMANCE TESTS 2
      #
      # DATE --        MARCH 20, 1967
      #
      # BY --          SYSTEM TEST GROUP 864-6900 EXT. 1274
      #
      # MODNO. --      ZERO
      #
      # FUNCTIONAL DESCRIPTION
      #
      # POSITIONING ROUTINES FOR THE IMU PERFORMANCE TESTS AS WELL AS SOME OF
      # THE TESTS THEMSELVES. FOR A DESCRIPTION OF THESE SUBROUTINES AND THE
      # OPERATING PROCEDURES (TYPICALLY) SEE STG MEMO 685. THEORETICAL REF. E-1973

```

	BANK	33
	SETLOC	IMU2
	BANK	
	EBANK=	POSITON
	COUNT*	\$\$/P07
REDO	TC	NEWMODEX
	MM	07
GEOIMUTT	TC	IMUZERR
IMUBACK	CA	ZERO
	TS	NDXCTR
	TS	TORQNDX
	TS	TORQNDX +1
	TS	OVFLOWCK
NBPOSPL	CA	DEC17
	TS	ZERONDX
	CA	XNBADR
	TC	ZEROING
	CA	HALF
	TS	XNB
GUESS	TC	INTPRET
LATAZCHK	DLOAD	SL2
		LATITUDE
	STODL	DSPTM1 +1
		AZIMUTH
	RTB	EXIT
		1ST02S
	XCH	MPAC
	TS	DSPTM1
	CAF	VN0641
	TC	BANKCALL
	CADR	GOFLASH
	TC	ENDTEST1
	TC	+2
	TC	-5
# Page 374	TC	INTPRET
	SLOAD	RTB
		DSPTM1
		CDULOGIC
	STORE	AZIMUTH
	SLOAD	SR2
		DSPTM1 +1
	STORE	LATITUDE
	COS	DCOMP

	SL1		
	STODL	WANGI	
		LATITUDE	
	SIN	SL1	
	STODL	WANGO	
		AZIMUTH	
	PUSH	SIN	
	STORE	YNB	+2
	STODL	ZNB	+4
	COS		
	STORE	YNB	+4
	DCOMP		
POSGMBL	STCALL	ZNB	+2
		CALCGA	
	EXIT		
	TC	BANKCALL	
	CADR	IMUCOARS	
	CAF	BIT14	# IF BIT14 SET, GIMBAL LOCK
	MASK	FLAGWRD3	
	EXTEND		
	BZF	+2	
	INCR	NDXCTR	# +1 IF IN GIMBAL LOCK, OTHERWISE 0
	TC	DOWNFLAG	
	ADRES	GLOKFAIL	# RESET GIMBAL LOCK FLAG
	TC	IMUSLLLG	
	CCS	NDXCTR	# IF ONE GO AND DO A PIPA TEST ONLY
	TC	PIPACHK	# ALIGN AND MEASURE VERTICAL PIPA RATE
	TC	FINIMUDD	
	EXTEND		
	DCA	PERFDLAY	
	TC	LONGCALL	# DELAY WHILE SUSPENSION STABILIZES
	EBANK=	POSITON	
	2CADR	GOESTIMS	
	CA	ESTICADR	
	TC	JOBSLEEP	
GOESTIMS	CA	ESTICADR	
	TC	JOBWAKE	
	TC	TASKOVER	
ESTICADR	CADR	ESTIMS	
TORQUE	CA	ZERO	
# Page 375			
	TS	DSPTM2	
	CA	DRIFTI	
	TS	DSPTM2	+1
	INDEX	POSITON	

	TS	SOUTHDR -1	
	TC	SHOW	
PIPACHK	INDEX	NDXCTR	# PIPA TEST
	TC	+1	
	TC	EARTH*	
	CA	DEC17	# ALLOW PIP COUNTER TO OVERFLOW 17 TIMES
	TS	DATAPL +4	# IN THE ALLOTTED TIME INTERVAL
	CA	DEC58	
	TS	LENGTHOT	
	CA	ONE	
	TS	RESULTCT	
	CA	ZERO	
	INDEX	PIPINDEX	
	TS	PIPA	
	TS	DATAPL	
	TC	CHECKG	
	INHINT		
	CAF	TWO	
	TC	TWIDDLE	
	EBANK=	XSM	
	ADRES	PIPATASK	
	TC	ENDOFJOB	
PIPATASK	EXTEND		
	DIM	LENGTHOT	
	CA	LENGTHOT	
	EXTEND		
	BZMF	STARTPIP	
	CAF	BIT10	
	TC	TWIDDLE	
	EBANK=	XSM	
	ADRES	PIPATASK	
STARTPIP	CAF	PRI020	
	TC	FINDVAC	
	EBANK=	XSM	
	2CADR	PIPJOBB	
	TC	TASKOVER	
PIPJOBB	INDEX	NDXCTR	
	TC	+1	
	TC	EARTH*	
	CA	LENGTHOT	
# Page 376	EXTEND		

	BZMF	+2	
	TC	ENDOFJOB	
	CA	FIVE	
	TS	RESULTCT	
	TC	CHECKG	
	CCS	DATAPL +1	
	TC	+4	
	TC	CCSHOLE	
	CS	DATAPL +4	
	TS	DATAPL +4	
	EXTEND		
	DCS	DATAPL	
	DAS	DATAPL +4	
	TC	INTPRET	
	DLOAD	DSU	
		DATAPL +6	
		DATAPL +2	
	BPL	CALL	
		AINGOTN	
		OVERFFIX	
AINGOTN	PDDL	DDV	
		DATAPL +4	
	DMPR	RTB	
		DEC585	# DEC585 HAS BEEN REDEFINED FOR LEM
		SGNAGREE	
	STORE	DSPTM2	
	EXIT		
	CCS	NDXCTR	
	TC	COAALIGN	# TAKE PLATFORM OUT OF GIMBAL LOCK
	TC	SHOW	
VERTDRFT	CA	3990DEC	# ABOUT 1 HOUR VERTICAL DRIFT TEST
	TS	LENGTHOT	
	INDEX	POSITON	
	CS	SOUTHDR -2	
	TS	DRIFTT	
	CCS	PIPINDEX	# OFFSET PLATFORM TO MISS PIP DEAD-ZONES
	TCF	PON4	# Z-UP IN POS 4
PON2	CS	BIT5	# X-UP
	ADS	ERCOMP +2	
	CA	BIT5	
	ADS	ERCOMP +4	
	TCF	PON	
PON4	CS	BIT5	
	ADS	ERCOMP +2	

July 12, 2016

707

```

      CA      BIT5
      ADS     ERCOMP
      PON     TC      EARTH*
# Page 377
      CA      ZERO
      TS      ERVECTOR
      TS      ERVECTOR +1
      GUESS1  CAF     POSMAX
      TS      TORQNDX
      TS      TORQNDX +1
      CA      CDUX
      TS      LOSVEC
      TC      ESTIMS
      VALMIS  CA      DRIFT0
      TS      DSPTM2 +1
      CA      ZERO
      TS      DSPTM2
      TC      SHOW

      ENDTEST1 TC      DOWNFLAG
      ADRES    IMUSE
      CS       ZERO
      TC       NEWMODEA
      TC       ENDEXT

# Page 378
      OVERFFIX DAD     DAD
      DAD     DPPOSMAX
      DAD     ONEDPP
      RVQ

      COAALIGN EXTEND
      QXCH     ZERONDX
      CA       ZERO
      TS       THETAD
      TS       THETAD +1
      TS       THETAD +2
      TC       BANKCALL
      CADR     IMUCOARS
      ALIGNCOA TC      BANKCALL
      CADR     IMUSTALL
      TC       SOMERR2
      TC       ZERONDX

      IMUSLLLG EXTEND
      QXCH     ZERONDX

# ALLOW ONLY SOUTH GYRO EARTH RATE COMPENS
# COARSE ALIGN SUBROUTINE
```

	TC	ALIGNCOA	
FINIMUDD	EXTEND		
	QXCH	ZERONDX	
	TC	BANKCALL	
	CADR	IMUFINE	
	TC	ALIGNCOA	
IMUZERR	EXTEND		
	QXCH	ZERONDX	
	TC	BANKCALL	
	CADR	IMUZERO	
	TC	ALIGNCOA	
CHECKG	EXTEND		# PIP PULSE CATCHING ROUTINE
	QXCH	QPLACE	
	TC	+6	
CHECKG1	RELINT		
	CA	NEWJOB	
	EXTEND		
	BZMF	+6	
	TC	CHANG1	
	INHINT		
	INDEX	PIPINDEX	
	CS	PIPAX	
	TS	ZERONDX	
	INHINT		
# Page 379	INDEX	PIPINDEX	
	CA	PIPAX	
	AD	ZERONDX	
	EXTEND		
	BZF	CHECKG1	
	INDEX	PIPINDEX	
	CA	PIPAX	
	INDEX	RESULTCT	
	TS	DATAPL	
	TC	FINETIME	
	INDEX	RESULTCT	
	TS	DATAPL +1	
	INDEX	RESULTCT	
	LXCH	DATAPL +2	
	RELINT		
ENDCHKG	TC	QPLACE	
ZEROING	TS	L	

July 12, 2016

709

ZEROING1	TCF	+2
	TS	ZERONDX
	CAF	ZERO
	INDEX	L
	TS	0
	INCR	L
	CCS	ZERONDX
	TCF	ZEROING1
	TC	Q

Page 380

ERTHRVSE	DLOAD	PDDL		
		SCHZEROS	# PD24 = (SIN	-COS
		LATITUDE		0) (OMEG/MS)
	COS	DCOMP		
	PDDL	SIN		
		LATITUDE		
	VDEF	VXSC		
		OMEG/MS		
	STORE	ERVECTOR		
	RTB			
		LOADTIME		
	STOVL	TMARK		
		SCHZEROS		
	STORE	ERCOMP		
	RVQ			

EARTH	ITA	RTB
		S2
		LOADTIME
	STORE	TEMPTIME
	DSU	BPL
		TMARK
		ERTHR
	CALL	

ERTHR		OVERFFIX
	SL	VXSC
		9D
		ERVECTOR
	MXV	VAD
		XSM
		ERCOMP
	STODL	ERCOMP
		TEMPTIME
	STORE	TMARK
	AXT,1	RTB

	ECADR	ERCOMP	
		PULSEIMU	
	GOTO		
		S2	
EARTH*	EXTEND		
	QXCH	QPLACES	
	TC	INTPRET	
	CALL		
		EARTH*	
	EXIT		
	TC	IMUSLLLG	
	TC	QPLACES	
SHOW	EXTEND		
# Page 381			
	QXCH	QPLACE	
SHOW1	CA	POSITON	
	TS	DSPTM2 +2	
	CA	VB06N98	
	TC	BANKCALL	
	CADR	GOFLASH	
	TC	ENDTEST1	# V34
	TC	QPLACE	# V33
	TCF	SHOW1	
3990DEC	DEC	3990	
VB06N98	VN	0698	
VN0641	VN	0641	
DEC17	=	ND1	
DEC58	DEC	58	
OGCPL	ECADR	OGC	
1SECX	=	1SEC	
XNBADR	GENADR	XNB	
XSMADR	GENADR	XSM	
	BLOCK	2	
	COUNT*	\$\$/P07	
FINETIME	INHINT		# RETURNS WITH INTERRUPT INHIBITED
	EXTEND		
	READ	LOSCALAR	
	TS	L	
	EXTEND		
	RXOR	LOSCALAR	
	EXTEND		
	BZF	+4	
	EXTEND		

July 12, 2016

711

```

      READ  LOSCALAR
      TS    L
+4      CS    POSMAX
      AD    L
      EXTEND
      BZF    FINETIME +1
      EXTEND
      READ  HISCALAR
      TC Q
```

This code is written to file `src/IMU-PERFORMANCE-TEST-2.s`.

B.41 IMU PERFORMANCE TESTS 4

```

712  <src/IMU-PERFORMANCE-TESTS-4.s 712>≡
      # Copyright:    Public domain.
      # Filename:     IMU_PERFORMANCE_TESTS_4.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         382-389
      # Mod history:   2009-05-17 RSB   Adapted from the corresponding
      #               Luminary131 file, using page
      #               images from Luminary 1A.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969
      #
      # Page 382
      # PROGRAM --      IMU PERFORMANCE TESTS 4
      # DATE --         NOV 15, 1966
      # BY --           GEORGE SCHMIDT IL7-146 EXT 1126
      # MOD NO-ZERO
      #
      # FUNCITONAL DESCRIPTION
      #
      # THIS SECTION CONSISTS OF THE FILTER FOR THE GYRO DRIFT TESTS. NO COMPASS
      # IS DONE IN LEM. FOR A DESCRIPTION OF THE FILTER SEE E-1973. THIS
      # SECTION IS ENTERED FROM IMU 2. IT RETURNS THERE AT END OF TEST.
      #
      # EARTH,OGC ZERO,ERTHRVSE
      #
      # NORMAL EXIT

```



```

#
# LENGTHOT GOES TO ZERO -- RETURN TO IMU PERF TESTS 2 CONTROL
#
# ALARMS
#
# 1600  OVERFLOW IN DRIFT TEST
# 1601  BAD IMU MODING IN ANY ROUTINE THAT USES IMUSTALL
#       OUTPUT
#
# FLASHING DISPLAY OF RESULTS -- CONTROLLED IN IMU PERF TESTS 2
#
# DEBRIS
#
# ALL CENTRALS -- ALL OF EBANK XSM

```

```

# Page 383

```

```

BANK      33
SETLOC    IMU4
BANK
COUNT*   $$/P07

```

```

EBANK=    XSM

```

```

ESTIMS      INHINT
CAE         1SECXT
TC          TWIDDLE
EBANK=      XSM
ADRES       ALLOOP
CAF         ZERO          # ZERO THE PIPAS
TS          PIPAX
TS          PIPAY
TS          PIPAZ
RELINT
CA          77DECML
TS          ZERONDX
CA          ALXXXZ
TC          ZEROING
TC          INTPRET
SLOAD
           SCHZEROS
STOVL       GCOMPSW -1
           INTVAL  +2
STOVL       ALX1S
           SCHZEROS
STORE       DELVX
STORE       GCOMP

```

	SLOAD		
		TORQNDX	
	DCOMP	BMN	
		VERTSKIP	
	CALL		
		ERTHRVSE	
VERTSKIP	EXIT		
	TC	SLEEPIE +1	
# Page 384			
ALLOOP	CA	OVFLOWCK	
	EXTEND		
	BZF	+2	
	TC	TASKOVER	
	CCS	ALTIM	
	CA	A	# SHOULD NEVER HIT THIS LOCATION
	TS	ALTIMS	
	CS	A	
	TS	ALTIM	
	CS	ONE	
	AD	GEOCOMPS	
	EXTEND		
	BZF	+4	
	CA	LENGTHOT	
	EXTEND		
	BZMF	+5	
	CAE	1SECXT	
	TC	TWIDDLE	
	EBANK=	XSM	
	ADRES	ALLOOP	
	CAF	ZERO	
	XCH	PIPAX	
	TS	DEL VX	
	CAF	ZERO	
	XCH	PIPAY	
	TS	DEL VY	
	CAF	ZERO	
	XCH	PIPAZ	
	TS	DEL VZ	
SPECSTS	CAF	PRI020	
	TC	FINDVAC	
	EBANK=	XSM	
	2CADR	ALFLT	# START THE JOB
	TC	TASKOVER	

July 12, 2016

715

Page 385

ALFLT	CCS	GEOCOMPS	
	TC	+2	
	TC	NORMLOP	
	TC	BANKCALL	
	CADR	1/PIPA	
NORMLOP	TC	INTPRET	
	DLOAD		
		INTVAL	
	STOVL	S1	
		DELVX	
	VXM	VSL1	
		XSM	
	DLOAD	DCOMP	
		MPAC +3	
	STODL	DPIPAY	
		MPAC +5	
	STORE	DPIPAZ	
	SETPD	AXT,1	
		0	
		8D	
	SLOAD	DCOMP	
		GEOCOMPS	
	BMN		
		PERFERAS	
ALCGKK	SLOAD	BMN	
		ALTIMS	
		ALFLT3	
ALKCG	AXT,2	LXA,1	# LOADS SLOPES AND TIME CONSTANTS AT RQST
		12D	
		ALX1S	
ALKCG2	DLOAD*	INCR,1	
		ALFDK	+144D,1
	DEC	-2	
	STORE	ALDK	+10D,2
	TIX,2	SXA,1	
		ALKCG2	
		ALX1S	
ALFLT3	AXT,1		
		8D	
DEMLP	DLOAD*	DMP	
		DPIPAY	+8D,1
		PIPASC	
	SLR	BDSU*	

Page 386

	9D	
	INTY	+8D,1
STORE	INTY	+8D,1
PDDL	DMP*	
	VELSC	
	VLAUN	+8D,1
SL2R		
DSU	STADR	
STORE	DELM	+8D,1
STORE	DELM	+10D,1
TIX,1	AXT,2	
	DEMLP	
	4	
ALILP	DLOAD*	DMPR*
	ALK	+4,2
	ALDK	+4,2
STORE	ALK	+4,2
TIX,2	AXT,2	
	ALILP	
	8D	
ALKLP	LXC,1	SXA,1
		CMPX1
		CMPX1
	DLOAD*	DMPR*
	ALK	+1,1
	DELM	+8D,2
DAD*		
	INTY	+8D,2
STORE	INTY	+8D,2
DLOAD*	DAD*	
	ALK	+12D,2
	ALDK	+12D,2
STORE	ALK	+12D,2
DMPR*	DAD*	
	DELM	+8D,2
	INTY	+16D,2
STORE	INTY	+16D,2
DLOAD*	DMP*	
	ALSK	+1,1
	DELM	+8D,2
SL1R	DAD*	
	VLAUN	+8D,2
STORE	VLAUN	+8D,2
TIX,2	AXT,1	
	ALKLP	

```

                                8D

LOOSE      DLOAD*  PDDL*
                                ACCWD  +8D,1
                                VLAUN  +8D,1
                                PDDL*  VDEF
                                POSNV  +8D,1
                                MXV    VSL1
                                TRANS1

# Page 387

                                DLOAD
                                MPAC
STORE      POSNV  +8D,1
DLOAD
                                MPAC  +3
STORE      VLAUN  +8D,1
DLOAD
                                MPAC  +5
STORE      ACCWD  +8D,1
TIX,1
                                LOOSE

                                AXT,2  AXT,1      # EVALUATE SINES AND COSINES
                                6
                                2
BOOP      DLOAD*  DMPR
                                ANGX   +2,1
                                GEORGEJ

                                SR2R
                                PUSH   SIN
                                SL3R   XAD,1
                                X1
                                STORE  16D,2
                                DLOAD
                                COS
                                STORE  22D,2      # COSINES
                                TIX,2
                                BOOP

PERFERAS   EXIT
                                CA      EBANK7
                                TS      EBANK
                                EBANK=  ATINGINC
                                TC      ATINGINC      # GOTO ERASABLE TO CALCULATE ONLY TO RETN

#                                CAUTION

```

```

#
# THE ERASABLE PROGRAM THAT DOES THE CALCULATIONS MUST BE LOADED
# BEFORE ANY ATTEMPT IS MAKE TO RUN THE IMU PERFORMANCE TEST

EBANK=  AZIMUTH
CCS     LENGTHOT
TC      SLEEPIE
CCS     TORQNDX
TCF     +2
TC      SETUPER1
CA      CDUX
TS      LOSVEC  +1      # FOR TROUBLESHOOTING VD POSNS 2$4

# Page 388
SETUPER1 TC      INTPRET
          DLOAD   PDDL      # ANGLES FROM DRIFT TEST ONLY
          ANGZ
          ANGY
          PDDL    VDEF
          ANGX
          VCOMP   VXSC
          GEORGEJ
          MXV     VSR1
          XSM
          STORE   OGC
          EXIT

          CA      OGCPL
          TC      BANKCALL
          CADR    IMUPULSE
          TC      IMUSLLLG
          CCS     TORQNDX      # ONLY POSITIVE IF IN VERTICAL DRIFT TEST
          TC      VALMIS
          TC      INTPRET
          CALL
          ERTHRVSE
          EXIT
          TC      TORQUE

SLEEPIE  TS      LENGTHOT      # TEST NOT OVER-DECREMENT LENGTHOT
          CCS     TORQNDX      # ARE WE DOING VERTDRIFT
          TC      EARTH*
          TC      ENDOFJOB

SOMEERRR CA      EBANK5
          TS      EBANK

```

July 12, 2016

719

	CA	ONE	
	TS	OVFLOWCK	# STOP ALLOOP FROM CALLING ITSELF
	TC	ALARM	
	OCT	1600	
	TC	ENDTEST1	
SOMERR2	CAF	OCT1601	
	TC	VARALARM	
	TC	DOWNFLAG	
	ADRES	IMUSE	
	TC	ENDOFJOB	
OCT1601	OCT	01601	
DEC585	OCT	06200	# 3200 B+14 ORDER IS IMPORTANT
SCHZEROS	2DEC	.00000000	
# Page 389	2DEC	.00000000	
	OCT	00000	
ONEDPP	OCT	00000	# ORDER IS IMPORTANT
	OCT	00001	
INTVAL	OCT	4	
	OCT	2	
	DEC	144	
	DEC	-1	
SOUPPLY	2DEC	.93505870	# INITIAL GAINS FOR PIP OUTPUTS
	2DEC	.26266423	# INITIAL GAINS/4 FOR ERECTION ANGLES
77DECML	DEC	77	
ALXXXZ	GENADR	ALX1S -1	
PIPASC	2DEC	.13055869	
VELSC	2DEC	-.52223476	# 512/980.402
ALSK	2DEC	.17329931	# SSWAY VEL GAIN X 980.402/4096
	2DEC	-.00835370	# SSWAY ACCEL GAIN X 980.402/4096
GEORGEJ	2DEC	.63661977	
GEORGEK	2DEC	.59737013	

This code is written to file src/IMU-PERFORMANCE-TESTS-4.s.

B.42 INFLIGHT ALIGNMENT ROUTINES

720 *<src/INFLIGHT-ALIGNMENT-ROUTINES.s 720>≡*

```
# Copyright:      Public domain.
# Filename:       INFLIGHT_ALIGNMENT_ROUTINES.agc
# Purpose:       Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Ron Burkey <info@sandroid.org>.
# Website:       www.ibiblio.org/apollo.
# Pages:        1355-1364
# Mod history:   2009-05-14 RSB   Adapted from the Colossus249/ file of the
#                               same name, using Comanche055 page images.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum. The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
# thanks to both. The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo. If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#       Colossus 2A

# Page 1355

      BANK      22
      SETLOC    INFLIGHT
      BANK

      EBANK=    XSM

# CALCGTA COMPUTES THE GYRO TORQUE ANGLES REQUIRED TO BRING THE STABLE MEMBER INTO THE
#
# THE INPUT IS THE DESIRED STABLE MEMBER COORDINATES REFERRED TO PRESENT STABLE MEMBER
# HALF-UNIT VECTORS ARE STORED AT XDC, YDC, AND ZDC.
#
# THE OUTPUTS ARE THE THREE GYRO TORQUE ANGLES TO BE APPLIED TO THE Y, Z, AND X GYROSCOPES
# MGC, AND OGC RESPECTIVELY.
```


	COUNT	23/INFLT	
CALCGTA	ITA	DLOAD	# PUSHDOWN 00-03, 16D-27D, 34D-37D
		S2	# XDC = (XD1 XD2 XD3)
		XDC	# YDC = (YD1 YD2 YD3)
	PDDL	PDDL	# ZDC = (ZD1 ZD2 ZD3)
		HI6ZEROS	
		XDC +4	
	DCOMP	VDEF	
	UNIT		
	STODL	ZPRIME	# ZP = UNIT(-XD3 0 XD1) = (ZP1 ZP2 ZP3)
		ZPRIME	
	SR1		
	STODL	SINTH	# SIN(IGC) = ZP1
		ZPRIME +4	
	SR1		
	STCALL	COSTH	# COS(IGC) = ZP3
		ARCTRIG	
	STODL	IGC	# Y GYRO TORQUING ANGLE FRACTION OF REV.
		XDC +2	
	SR1		
	STODL	SINTH	# SIN(MGC) = XD2
		ZPRIME	
	DMP	PDDL	
		XDC +4	# PD00 = (ZP1)(XD3)
		ZPRIME +4	
	DMP	DSU	
		XDC	# MPAC = (ZP3)(XD1)
	STADR		
	STCALL	COSTH	# COS(MGC) = MPAC - PD00
		ARCTRIG	
# Page 1356	STOVL	MGC	# Z GYRO TORQUING ANGLE FRACTION OF REV.
		ZPRIME	
	DOT		
		ZDC	
	STOVL	COSTH	# COS(OGC) = ZP . ZDC
		ZPRIME	
	DOT		
		YDC	
	STCALL	SINTH	# SIN(OGC) = ZP . YDC

ARCTRIG

STCALL OGC # X GYRO TORQUING ANGLE FRACTION OF REV.
S2

Page 1357

ARCTRIG COMPUTES AN ANGLE GIVEN THE SINE AND COSINE OF THIS ANGLE.

#

THE INPUTS ARE SIN/4 AND COS/4 STORED DP AT SINTH AND COSTH.

#

THE OUTPUT IS THE CALCULATED ANGLE BETWEEN +.5 AND -.5 REVOLUTIONS AND STORED AT TH

AVAILABLE AT MPAC.

ARCTRIG	DLOAD	ABS SINTH	# PUSHDOWN 16D-21D
	DSU	BMN QTSN45 TRIG1	# ABS(SIN/4) - SIN(45)/4 # IF (-45,45) OR (135,-135)
	DLOAD	SL1 COSTH	# (45,135) OR (-135,-45)
	ACOS	SIGN SINTH	
	STORE RVQ	THETA	# X = ARCCOS(COS) WITH SIGN(SIN)
TRIG1	DLOAD	SL1 SINTH	# (-45,45) OR (135,-135)
	ASIN STODL	THETA COSTH	# X = ARCSIN(SIN) WITH SIGN(SIN)
	BMN	TRIG2	# IF (135,-135)
	DLOAD	RVQ THETA	# X = ARCSIN(SIN) (-45,45)
TRIG2	DLOAD	SIGN HIDPHALF SINTH	# (135,-135)
	DSU	THETA	
	STORE RVQ	THETA	# X = .5 WITH SIGN(SIN) - ARCSIN(SIN) # (+) - (+) OR (-) - (-)

Page 1358

SMNB, NBSM, AND AXISROT, WHICH USED TO APPEAR HERE, HAVE BEEN
 # COMBINED IN A ROUTINE CALLED AX*SR*T, WHICH APPEARS AMONG THE POWERED
 # FLIGHT SUBROUTINES.

Page 1359

CALCGA COMPUTES THE CDU DRIVING ANGLES REQUIRED TO BRING THE STABLE MEMBER INTO THE DESIRED C
 #

THE INPUTS ARE 1) THE NAVIGATION BASE COORDINATES REFERRED TO ANY COORDINATE SYSTEM. THE TH
 # VECTORS ARE STORED AT XNB, YNB, AND ZNB. 2) THE DESIRED STABLE MEMBER COORDINATES REFERRED T
 # COORDINATE SYSTEM ARE STORED AT XSM, YSM, AND ZSM.

#

THE OUTPUTS ARE THE THREE CDU DRIVING ANGLES AND ARE STORED SP AT THETAD, THETAD +1, AND THET

CALCGA	SETPD		# PUSHDOWN 00-05, 16D-21D, 34D-37D
		0	
	VLOAD	VXV	
		XNB	# XNB = OGA (OUTER GIMBAL AXIS)
		YSM	# YSM = IGA (INNER GIMBAL AXIS)
	UNIT	PUSH	# PDO = UNIT(OGA X IGA) = MGA
	DOT	ITA	
		ZNB	
		S2	
	STOVL	COSTH	# COS(OG) = MGA . ZNB
		0	
	DOT		
		YNB	
	STCALL	SINTH	# SIN(OG) = MGA . YNB
		ARCTRIG	
	STOVL	OGC	
		0	
	VXV	DOT	# PROVISION FOR MG ANGLE OF 90 DEGREES
		XNB	
		YSM	
	SL1		
	STOVL	COSTH	# COS(MG) = IGA . (MGA X OGA)
		YSM	
	DOT		
		XNB	
	STCALL	SINTH	# SIN(MG) = IGA . OGA
		ARCTRIG	
	STORE	MGC	
	ABS	DSU	
		.166...	

```

                                BPL
                                GIMLOCK1      # IF ANGLE GREATER THAN 60 DEGREES

CALCGA1      VLOAD  DOT
                                ZSM
                                0
                                STOVL  COSTH      # COS(IG) = ZSM . MGA
                                XSM

# Page 1360
                                DOT    STADR
                                STCALL  SINTH      # SIN(IG) = XSM . MGA
                                ARCTRIG

                                STOVL  IGC
                                OGC
                                RTB    BONCLR
                                V1STO2S
                                CPHIFLAG
                                S2
                                STCALL  THETAD
                                S2

GIMLOCK1     EXIT
                                TC      ALARM
                                OCT     00401
                                TC      UPFLAG      # GIMBAL LOCK HAS OCCURRED
                                ADRES    GLOKFAIL

                                TC      INTPRET
                                GOTO     CALCGA1

```

Page 1361

AXISGEN COMPUTES THE COORDINATES OF ONE COORDINATE SYSTEM REFERRED TO ANOTHER COORDINATE SYSTEM

THE INPUTS ARE 1) THE STAR1 VECTOR REFERRED TO COORDINATE SYSTEM A STORED AT STARAD +6.
REFERRED TO COORDINATE SYSTEM A STORED AT STARAD +6. 3) THE STAR1 VECTOR REFERRED TO COORDINATE SYSTEM B
AT LOCATION 6 OF THE VAC AREA. 4) THE STAR2 VECTOR REFERRED TO COORDINATE SYSTEM B STORED AT
THE VAC AREA.

THE OUTPUT DEFINES COORDINATE SYSTEM A REFERRED TO COORDINATE SYSTEM B. THE THREE COORDINATES
AT LOCATIONS XDC, XDC +6, XDC +12D, AND STARAD, STARAD +6, STARAD +12D.

```

AXISGEN      AXT,1  SSP      # PUSHDOWN 00-30D, 34D-37D
                                STARAD +6
                                S1

```

```

STARAD  -6

SETPD
0
AXISGEN1  VLOAD*  VXV*      # 06D  UA = S1
STARAD  +12D,1  #      STARAD +00D      UB = S1
STARAD  +18D,1
UNIT      # 12D  VA = UNIT(S1 X S2)
STORE    STARAD  +18D,1  #      STARAD +06D      VB = UNIT(S1 X S2)
VLOAD*
STARAD  +12D,1

VXV*  VSL1
STARAD  +18D,1  # 18D  WA = UA X VA
STORE  STARAD  +24D,1  #      STARAD +12D      WB = UB X VB

TIX,1
AXISGEN1

AXC,1  SXA,1
6
30D

AXT,1  SSP
18D
S1
6

AXT,2  SSP
6
S2
2

AXISGEN2  XCHX,1  VLOAD*
30D      # X1=-6 X2=+6  X1=-6 X2=+4  X1=-6 X2=+2
0,1

# Page 1362
VXSC*  PDVL*      # J=(UA) (UB1)  J=(UA) (UB2)  J=(UA) (UB3)
STARAD  +6,2
6,1
VXSC*
STARAD  +12D,2
STOVL*  24D      # K=(VA) (VB1)  J=(VA) (VB2)  J=(VA) (VB3)
12D,1

```

```

VXSC*  VAD
        STARAD  +18D,2  # L=(WA)(WB1)    J=(WA)(WB2)    J=(WA)(WB3)
VAD     VSL1
        24D
XCHX,1  UNIT
        30D
STORE   XDC      +18D,1  # XDC = L+J+K    YDC = L+J+K    ZDC = L+J+K

TIX,1
        AXISGEN3

AXISGEN3  TIX,2
          AXISGEN2

VLOAD
        XDC
STOVL    STARAD
        YDC
STOVL    STARAD  +6
        ZDC
STORE    STARAD  +12D

RVQ

# Page 1363
QTSN45   2DEC    .1768
.166...  2DEC    .1666666667

# Page 1364 (empty page)

```

This code is written to file `src/INFLIGHT-ALIGNMENT-ROUTINES.s`.

B.43 INPUT OUTPUT CHANNEL BIT DESCRIPTIONS

727 *<src/INPUT-OUTPUT-CHANNEL-BIT-DESCRIPTIONS.s 727>≡*

```
# Copyright:      Public domain.
# Filename:       INPUT_OUTPUT_CHANNEL_BIT_DESCRIPTIONS.agc
# Purpose:       Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:       Onno Hommes <ohommes@cmu.edu>.
# Website:       www.ibiblio.org/apollo.
# Pages:        0054-0060
# Mod history:   2009-05-14 OH   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum. The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum. Many thanks to both. The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo. If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969
#
# Page 54
#
# *** CHANNEL DESCRIPTION WORDS ARE ALLOCATED IN ERASABLE ASSIGNMENTS ***
#
# CHANNEL 1      IDENTICAL TO COMPUTER REGISTER L (0001)
#
# CHANNEL 2      IDENTICAL TO COMPUTER REGISTER Q (0002)
#
# CHANNEL 3      HISCALAR: INPUT CHANNEL; MOST SIGNIFICANT 14 BITS FROM 33 STAGE BINARY COUNTER.
#               FACTOR IS B23 IN CSEC, SO MAX VALUE ABOUT 23.3 HOURS AND LEAST SIGNIFICANT BIT
#
# CHANNEL 4      LOSCALAR: INPUT CHANNEL; NEXT MOST SIGNIFICANT 14 BITS FROM THE 33 STAGE BINARY
#               ASSOCIATED WITH CHANNEL 3. SCALE FACTOR IS B9 IN CSEC. SO MAX VAL IS 5.12 SEC A
#               SIGNIFICANT BIT IS 1/3200 SEC. SCALE FACTOR OF D.P. WORD WITH CHANNEL 3 IS B23
```

```

# CHANNEL 5    PYJETS:  OUTPUT CHANNEL; PITCH RCS JET CONTROL.  (REACTION CONTROL SYSTEM)
# CHANNEL 6    ROLLJETS: OUTPUT CHANNEL; ROLL RCS JET CONTROL.  (REACTION CONTROL SYSTEM)
# CHANNEL 7    SUPERBNK: OUTPUT CHANNEL; NOT RESET BY RESTART; FIXED EXTENSION BITS
#              APPROPRIATE FIXED MEMORY BANK IF FBANK IS 30 OCTAL OR MORE.  USES BITS 1-14
# CHANNEL 10   OUTO:  OUTPUT CHANNEL; REGISTER USED TO TRANSMIT LATCHING-RELAY DRIVING DATA TO
#              THE DISPLAY SYSTEM.  BITS 15-12 ARE SET TO THE ROW NUMBER (1-14 OCTAL) OF THE
#              CHANGED AND BITS 11-1 CONTAIN THE REQUIRED SETTINGS FOR THE RELAYS IN THE
# CHANNEL 11   DSALMOUT: OUTPUT CHANNEL; REGISTER WHOSE BITS ARE USED FOR ENGINE ON/OFF AND
#              DRIVE INDIVIDUAL INDICATORS OF THE DISPLAY SYSTEM.  BITS 1-7 ARE A REGISTER OF
#              INDICATORS
#              BIT 1      ISS WARNING
#              BIT 2      LIGHT COMPUTER ACTIVITY LAMP
#              BIT 3      LIGHT UPLINK ACTIVITY LAMP
#              BIT 4      LIGHT TEMP CAUTION LAMP
#              BIT 5      LIGHT KEYBOARD RELEASE LAMP
#              BIT 6      FLASH VERB AND NOUN LAMPS
#              BIT 7      LIGHT OPERATOR ERROR LAMP
# Page 55
#              BIT 8      SPARE
#              BIT 9      TEST CONNECTOR OUTBIT
#              BIT 10     CAUTION RESET
#              BIT 11     SPARE
#              BIT 12     SPARE
#              BIT 13     ENGINE ON
#              BIT 14     ENGINE OFF
#              BIT 15     SPARE
# CHANNEL 12   CHAN12: OUTPUT CHANNEL; BITS USED TO DRIVE NAVIGATION AND SPACECRAFT
#              BIT 1      ZERO RR CDU; CDU'S GIVE RRADAR INFORMATION FOR LM
#              BIT 2      ENABLE CDU RADAR ERROR COUNTERS
#              BIT 3      NOT USED
#              BIT 4      COARSE ALIGN ENABLE OF IMU
#              BIT 5      ZERO IMU CDU'S
#              BIT 6      ENABLE IMU ERROR COUNTER, CDU ERROR COUNTER.
#              BIT 7      SPARE
#              BIT 8      DISPLAY INERTIAL DATA
#              BIT 9      -PITCH GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#              BIT 10     +PITCH GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#              BIT 11     -ROLL GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#              BIT 12     +ROLL GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#              BIT 13     LR POSITION 2 COMMAND

```



```
#          BIT 14          ENABLE RENDEZVOUS RADAR LOCK-ON; AUTO ANGLE TRACK'G
#          BIT 15          ISS TURN ON DELAY COMPLETE
```

```
# Page 56
```

```
# CHANNEL 13  CHAN13: OUTPUT CHANNEL.
```

```
#
#          BIT 1          RADAR C          PROPER SETTING OF THE A,B,C MATRIX
#          BIT 2          RADAR B          SELECTS CERTAIN RADAR
#          BIT 3          RADAR A          PARAMETERS TO BE READ.
#          BIT 4          RADAR ACTIVITY
#          BIT 5          NOT USED (CONNECTS AN ALTERNATE INPUT TO UPLINK)
#          BIT 6          BLOCK INPUTS TO UPLINK CELL
#          BIT 7          DOWNLINK TELEMETRY WORD ORDER CODE BIT
#          BIT 8          RHC COUNTER ENABLE (READ HAND CONTROLLER ANGLES)
#          BIT 9          START RHC READ INTO COUNTERS IS BIT 8 SET
#          BIT 10         TEST ALARMS, TEST DSKY LIGHTS
#          BIT 11         ENABLE STANDBY
#          BIT 12         RESET TRAP 31-A          ALWAYS APPEAR TO BE SET TO 0
#          BIT 13         RESET TRAP 31-B          ALWAYS APPEAR TO BE SET TO 0
#          BIT 14         RESET TRAP 32           ALWAYS APPEAR TO BE SET TO 0
#          BIT 15         ENABLE T6 RUPT
```

```
# CHANNEL 14  CHAN14: OUTPUT CHANNEL; USED TO CONTROL COMPUTER COUNTER CELLS (CDU, GYRO, SPAC
```

```
#
#          BIT 1          OUTLINK ACTIVITY (NOT USED)
#          BIT 2          ALTITUDE RATE OR ALTITUDE SELECTOR
#          BIT 3          ALTITUDE METER ACTIVITY
#          BIT 4          THRUST DRIVE ACTIVITY FOR DESCENT ENGINE
#          BIT 5          SPARE
#          BIT 6          GYRO ENABLE POWER FOR PULSES
#          BIT 7          GYRO SELECT B          PAIR OF BITS IDENTIFIES AXIS OF
#          BIT 8          GYRO SELECT A          GYRO SYSTEM TO BE TORQUED.
#          BIT 9          GYRO TORQUING COMMAND IN NEGATIVE DIRECTION.
```

```
# Page 57
```

```
#          BIT 10         GYRO ACTIVITY
#          BIT 11         DRIVE CDU S
#          BIT 12         DRIVE CDU T
#          BIT 13         DRIVE CDU Z
#          BIT 14         DRIVE CDU Y
#          BIT 15         DRIVE CDU X
```

```
# CHANNEL 15  MNKEYIN: INPUT CHANNEL; KEY CODE INPUT FROM KEYBOARD OF DSKY, SENSED BY PROGRAM
#              PROGRAM INTERRUPT #5 IS RECEIVED.  USED BITS 5-1
```

```
# CHANNEL 16  NAVKEYIN: INPUT CHANNEL; OPTICS MARK INFORMATION AND NAVIGATION PANEL DSKY (CM)
#              CONTROL (LM) SENSED BY PROGRAM THEN PROGRAM INTERRUPT #6 IS RECEIVED.  USES BIT
```

```

#
#          BIT 1          NOT ASSIGNED
#          BIT 2          NOT ASSIGNED
#          BIT 3          OPTICS X-AXIS MARK SIGNAL FOR ALIGN OPTICAL TSCOPE
#          BIT 4          OPTICS Y-AXIS MARK SIGNAL FOR AOT
#          BIT 5          OPTICS MARK REJECT SIGNAL
#          BIT 6          DESCENT+ ; CREW DESIRED SLOWING RATE OF DESCENT
#          BIT 7          DESCENT- ; CREW DESIRED SPEEDING UP RATE OF D'CENT

```

```

# NOTE: ALL BITS IN CHANNELS 30-33 ARE INVERTED AS SENSED BY THE PROGRAM, SO THAT A V
# THAT THE INDICATED SIGNAL IS PRESENT.

```

```

# CHANNEL 30    INPUT CHANNEL
#
#          BIT 1          ABORT WITH DESCENT STAGE
#          BIT 2          UNUSED
#          BIT 3          ENGINE ARMED SIGNAL
#          BIT 4          ABORT WITH ASCENT ENGINE STAGE
#          BIT 5          AUTO THROTTLE; COMPUTER CONTROL OF DESCENT ENGINE

```

```

# Page 58

```

```

#          BIT 6          DISPLAY INERTIAL DATA
#          BIT 7          RR CDU FAIL
#          BIT 8          SPARE
#          BIT 9          IMU OPERATE WITH NO MALFUNCTION
#          BIT 10         LM COMPUTER (NOT AGS) HAS CONTROL OF LM.
#          BIT 11         IMU CAGE COMMAND TO DRIVE IMU GIMBAL ANGLES TO 0.
#          BIT 12         IMU CDU FAIL (MALFUNCTION OF IMU CDU,S)
#          BIT 13         IMU FAIL (MALFUCTION OF IMU STABILIZATION LOOPS)
#          BIT 14         ISS TURN ON REQUESTED
#          BIT 15         TEMPERATURE OF STABLE MEMBER WITHIN DESIGN LIMITS

```

```

# CHANNEL 31    INPUT CHANNEL; BITS ASSOCIATED WITH THE ATTITUDE CONTROLLER, TRANSLAT
# AND SPACECRAFT ATTITUDE CONTROL; USED BY RCS DAP.

```

```

#
#          BIT 1          ROTATION (BY RHC) COMMANDED IN POSITIVE PITCH DIRECT
#                          ALSO POSITIVE ELEVATION CHANGE FOR LANDING POINT DES
#          BIT 2          AS BIT 1 EXCEPT NEGATIVE PITCH AND ELEVATION.
#          BIT 3          ROTATION (BY RHC) COMMANDED IN POSITIVE YAW DIRECTION
#          BIT 4          AS BIT 3 EXCEPT NEGATIVE YAW
#          BIT 5          ROTATION (BY RHC) COMMANDED IN POSITIVE ROLL DIRECTIO
#                          ALSO POSITIVE AZIMUTH CHANGE FOR LANDING POINT DESIGN
#          BIT 6          AS BIT 5 EXCEPT NEGATIVE ROLL AND AZIMUTH
#          BIT 7          TRANSLATION IN +X DIRECTION COMMANDED BY THC
#          BIT 8          TRANSLATION IN -X DIRECTION COMMANDED BY THC
#          BIT 9          TRANSLATION IN +Y DIRECTION COMMANDED BY THC
#          BIT 10         TRANSLATION IN -Y DIRECTION COMMANDED BY THC

```

```

#          BIT 11          TRANSLATION IN +Z DIRECTION COMMANDED BY THC
#          BIT 12          TRANSLATION IN -Z DIRECTION COMMANDED BY THC
# Page 59
#          BIT 13          ATTITUDE HOLD MODE ON SCS MODE CONTROL SWITCH
#          BIT 14          AUTO STABILIZATION OF ATTITUDE ON SCS MODE SWITCH
#          BIT 15          ATTITUDE CONTROL OUT OF DETENT (RHC NOT IN NEUTRAL)

# CHANNEL 32  INPUT CHANNEL.
#
#          BIT 1           THRUSTERS 2 & 4 DISABLED BY CREW
#          BIT 2           THRUSTERS 5 & 8 DISABLED BY CREW
#          BIT 3           THRUSTERS 1 & 3 DISABLED BY CREW
#          BIT 4           THRUSTERS 6 & 7 DISABLED BY CREW
#          BIT 5           THRUSTERS 14 & 16 DISABLED BY CREW
#          BIT 6           THRUSTERS 13 & 15 DISABLED BY CREW
#          BIT 7           THRUSTERS 9 & 12 DISABLED BY CREW
#          BIT 8           THRUSTERS 10 & 11 DISABLED BY CREW
#          BIT 9           DESCENT ENGINE DISABLED BY CREW
#          BIT 10          APPARENT DESCENT ENGINE GIMBAL FAILURE
#          BIT 14          INDICATES PROCEED KEY IS DEPRESSED

# CHANNEL 33  CHAN33: INPUT CHANNEL; FOR HARDWARE STATUS AND COMMAND INFORMATION.  BITS 15-11
#              FLOP BITS RESET BY A CHANNEL "WRITE" COMMAND THAT ARE RESET BY A RESTART & BY T
#
#          BIT 1           SPARE
#          BIT 2           RR AUTO-POWER ON
#          BIT 3           RR RANGE LOW SCALE
#          BIT 4           RR DATA GOOD
#          BIT 5           LR RANGE DATA GOOD
#          BIT 6           LR POS1
#          BIT 7           LR POS2
# Page 60
#          BIT 8           LR VEL DATA GOOD
#          BIT 9           LR RANGE LOW SCALE
#          BIT 10          BLOCK UPLINK INPUT
#          BIT 11          UPLINK TOO FAST
#          BIT 12          DOWNLINK TOO FAST
#          BIT 13          PIPA FAIL
#          BIT 14          WARNING OF REPEATED ALARMS: RESTART, COUNTER FAIL, VOLTAGE FAIL
#          BIT 15          LGC OSCILLATOR STOPPED

# CHANNEL 34  DNT M1: OUTPUT CHANNEL; DOWNLINK 1: FIRST OF TWO WORDS SERIALIZATION.
# CHANNEL 35  DNT M2: OUTPUT CHANNEL; DOWNLINK 2: SECOND OF TWO WORDS SERIALIZATION.

```

B.44 INTEGRATION INITIALIZATION

```

732  <src/INTEGRATION-INITIALIZATION.s 732>≡
# Copyright:      Public domain.
# Filename:       INTEGRATION_INITIALIZATION.agc
# Purpose:        Part of the source code for Colossus 2A, AKA Comanche 055.
#                 It is part of the source code for the Command Module's (CM)
#                 Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:     yaYUL
# Contact:        Ron Burkey <info@sandroid.org>.
# Website:        www.ibiblio.org/apollo.
# Pages:          1309-1333
# Mod history:    2009-05-15 RSB   Adapted from the Colossus249/ file of the
#                 same name, using Comanche055 page images.
#                 2009-05-20 RSB   Corrections:  fixed an interpreter instruction,
#                 fixed a SETLOC.
#                 2009-05-23 RSB   In SETCOAST, corrected MOONTHIS to AMOONFLG.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum.  The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum.  Many
# thanks to both.  The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo.  If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#       Colossus 2A
#
# Page 1309
# 1.0 INTRODUCTION
# -----
#
# FROM A USER'S POINT OF VIEW, ORBITAL INTEGRATION IS ESSENTIALLY THE SAME AS THE 278
# PROGRAM.  THE SAME ENTRANCES TO THE PROGRAM WILL BE MAINTAINED, THE SAME STALLING P
# OUTPUT WILL STILL BE VIA THE PUSHLIST.  THE PRIMARY DIFFERENCES TO A USER INVOLVE T
# TERMINATING INTEGRATION AT A SPECIFIC FINAL RADIUS AND THE DIFFERENCE IN STATE VEC
# THE LUNAR SPHERE OF INFLUENCE.
#
# IN ORDER TO MAKE THE CSM(LEM)PREC AND CSM(LEM)CONIC ENTRANCES SIMILAR TO FLIGHT 278

```

```

# WILL ITSELF SET THE FINAL RADIUS (RFINAL) TO 0 SO THAT REACHING THE DESIRED TIME ONLY WILL TE
# INTEGRATION. THE DP REGISTER RFINAL MUST BE SET BY USERS OF INTEGRVS AND INTEGRV, AND MUST E
# CALL TC INTSTALL.
#
# WHEN THE LM IS ON THE LUNAR SURFACE (INDICATED BY LUNAR SURFACE FLAG SET) CALLS TO LEMCONIC,
# INTEGRV WITH VINFLAG = 0 WILL RESULT IN THE USE OF THE PLANETARY INERTIAL ORIENTATION SUBROUT
# BOTH THE LM'S POSITION AND VELOCITY IN THE REFERENCE COORDINATE SYSTEM.
# THE PROGRAM WILL PROVIDE OUTPUT AS IF INTEGRATION WAS USED. THAT IS, THE PUSHLIST WILL BE SE
# THE PERMANENT STATE VECTOR UPDATED WHEN SPECIFIED BY AN INTEGRV CALL.
#
# USERS OF INTEGRVS DESIRING INTEGRATION (INTYPFLG = 0) SHOULD NOTE THAT THE OBLATENESS PERTURE
# IN LUNAR ORBIT IS TIME DEPENDENT. THEREFORE, THE USER SHOULD SUPPLY AN INITIAL STATE VECTOR
# TIME AND THE DESIRED TIME (TDEC1) ALSO AT SOME REAL TIME. FOR CONIC "INTEGRATION" THE USER M
# AS THE INITIAL TIME AND DELTA TIME AS THE DESIRED TIME.
#
# 2.0 CENTRAL DESCRIPTION
# -----
#
# THE INTEGRATION PROGRAM OPERATES AS A CLOSED INTERPRETIVE SUBROUTINE AND PERFORMS THESE FUNCT
#     1) INTEGRATES (PRECISION OR CONIC) EITHER CSM OR LM STATE VECTOR
#     2) INTEGRATES THE W-MATRIX
#     3) PERMANENT OR TEMPORARY UPDATE OF THE STATE VECTOR
#
# THERE ARE SIX ENTRANCES TO THE INTEGRATION PROGRAM. FOUR OF THESE (CSMPREC, LEMPREC, CSMCONI
# ALL THE FLAGS REQUIRED IN THE INTEGRATION PROGRAM ITSELF TO CAUSE THE PRECISION OR CONIC INTE
# THE LM OR CSM STATE VECTOR, AS THE NAMES SUGGEST. ONE ENTRANCE (INTEGRVS) PERMITS THE CALLIN
# PROVIDE A STATE VECTOR TO BE INTEGRATED. THE CALLING PROGRAM MUST SET THE FLAGS INDICATING (
# CONIC INTEGRATION, (2) IN OR OUT OF LUNAR SPHERE, (3) MIDCOURSE OR NOT, AND THE INTEGRATION P
# THE FLAG SETTING TO BYPASS W-MATRIX INTEGRATION. THE LAST ENTRANCE (INTEGRV, USED IN GENERAL
# NAVIGATION PROGRAMS) PERMITS THE CALLER TO SET FIVE FLAGS (NOT MOONFLAG OR MIDFLAG) BUT NOT T
# VECTOR. ANY PROGRAM WHICH CALLS INTEGRVS OR INTEGRV MUST CALL INTSTALL BEFORE IT SETS THE IN
# AND/OR STATE VECTOR.
#
# THREE SETS OF 42 REGISTERS AND 2 FLAGS ARE USED FOR THE STATE VECTORS. TWO SETS, WHICH MAY N
# USED FOR THE PERMANENT STATE VECTORS FOR THE CSM AND LM. THE THIRD SET, WHICH MAY BE OVERLAY
# IS NOT BEING DONE, IS USED IN THE COMPUTATIONS.
#
# THE PERMANENT STATE VECTORS WILL BE PERIODICALLY UPDATED SO THAT THE VECTORS WILL NOT BE OLDE
# THE PERMANENT STATE VECTORS WILL ALSO BE UPDATED WHENEVER THE W-MATRIX IS INTEGRATED OR WHEN
# SETS STATEFLG (THE NAVIGATION PROGRAMS P20, P22.)
#
# Page 1310
# APPENDIX B OF THE USERS' GUIDE LISTS THE STATE VECTOR QUANTITIES.
#
# 2.1 RESTARTS
#

```


July 12, 2016

735

```

#      RCVSM(LEM)          CONIC POSITION          METERS          2          2
#      EQUALS RRECTCSM(LEM) IF
#      TCCSM(LEM) = 0
#
#      VCVCSM(LEM)          CONIC VELOCITY          M/CSEC          7          5
#      EQUALS VRECTCSM(LEM) IF
#      TCCSM(LEM) = 0
#
#      TCCSM(LEM)          TIME SINCE RECTIFICATION  CSECS          28          28
#      CUSTOMARILY 0
#
#      XKEPCSM(LEM)          RDOT OF KEPLER'S EQUATION  1/2          17          16
#      0 IF TCCSM(LEM) = 0  M          2          2
#
#      CMOONFLG          PERMANENT FLAGS CORRESPONDING  0          0
#      CMIDFLAG          TO MOONFLAG AND MIDFLAG      0,1        0,1
#      LMOONFLG          C = CSM, L = LM              0          0
#      LMIDFLAG          0,1        0,1
#
#      SURFFLAG          LUNAR SURFACE FLAG            0,1        0,1
#
# IN ADDITION, IF (L)CMIDFLAG IS SET, THE INITIAL INPUT VALUES FOR LUNAR
# SOLAR EPHEMERIDES SUBROUTINE AND PLANETARY INERTIAL ORIENTATION SUB-
# ROUTINE MUST BE PRESET.
#
# OUTPUT
# AFTER EVERY CALL TO INTEGRATION
#
#      EARTH  MOON
#      29      29
#      OD      RATT  POSITION          METERS          2          2
#
#      7      7
#      6D      VATT  VELOCITY          M/CSEC          2          2
#
#      28      28
#      12D     TAT   TIME              2          2
#
#      29      27
#      14D     RATT1 POSITION          METERS          2          2
#
#      7      5
#      20D     VATT1 VELOCITY          M/CSEC          2          2
#

```

```

#                                     3   2           36   30
#          26D      MU(P)   MU           M /CS           2   2
#
#          X1                MUTABLE ENTRY           -2   -10D
#
#          X2                COORDINT
#          X2                COORDINATE SYSTEM ORIGIN           0   2
#                          (THIS, NOT MOONFLAG, SHOULD BE
# Page 1312
#                          USED TO DETERMINE ORIGIN.)
#
# IN ADDITION TO THE ABOVE, THE PERMANENT STATE VECTOR IS UPDATED WHENEVER
# STATEFLG WAS SET AND WHENEVER A W-MATRIX IS TO BE INTEGRATED.  THE PUSH
# COUNTER IS SET TO 0 AND OVERFLOW IS CLEARED BEFORE RETURNING TO THE
# CALLING PROGRAM.
#
# 4.0 CALLING SEQUENCES AND SAMPLE CODE
# -----
#
#      A) PRECISION ORBITAL INTEGRATION.  CSMPREC, LEMPREC ENTRANCES
#          L-X      STORE TIME TO 96T5791T5 T 95 PUS L9ST (T4531)
#          L        CALL
#          L+1      CSMPREC (OR LEMPREC)
#          L+2      RETURN
#      INPUT                                           28
#          TDEC1 (PD 32D) TIME TO INTEGRATE TO...CENTISECONDS SCALED 2
#      OUTPUT
#          THE DATA LISTED IN SECTION 3.2 PLUS
#          RQVV     POSITION VECTOR OF VEHICLE WITH RESPECT TO SECONDARY
#          BODY... METERS B-29 ONLY IF MIDFLAG = DIMOFLAG = 1
#      B) CONIC INTEGRATION.  CSMCONIC, LEMCONIC ENTRANCES
#          L-X      STORE TIME IN PUSH LIST (TDEC1)
#          L        CALL
#          L+1      CSMCONIC (OR LEMCONIC)
#      INPUT/OUTPUT
#          SAME AS PRECISION INTEGRATION, EXCEPT RQVV NOT SET
#      C) INTEGRATE GIVEN STATE VECTOR.  INTEGRVS ENTRANCE
#          CALL
#
#          INTSTALL
#
#          VLOAD
#
#          POSITION VECTOR
#          RCV
#          VELOCITY VECTOR
#          VCV
#          TIME STATE VECTOR VALID
#          TET

```



```

#                               FINAL RADIUS
#                               STORE          RFINAL
#                               SET(CLEAR)    SET(CLEAR)
#                               INTYPFLAG
#                               MOONFLAG
#                               SET(CLEAR)    DLOAD
#                               DESIRED TIME
#                               STCALL        TDEC1
#                               INTEGRVS
#
# INPUT
# RCV      POSITION VECTOR          METERS
# VCV      VELOCITY VECTOR        M/CSEC
# TET      TIME OF STATE VECTOR (MAY = 0) CSEC B-28
# Page 1313
# TDEC1    TIME TO INTEGRATE TO      CSEC B-28 (PD 32D)
#          (MAY BE INCREMENT IF TET=0)
#
# OUTPUT
# SAME AS FOR PRECISION OR CONIC INTEGRATION,
# DEPENDING ON INTYPFLG.
# D) INTEGRATE STATE VECTOR.  INTGRV ENTRANCE
# L-X      STORE TIME IN PUSH LIST (TDEC1) (MAY BE DONE AFTER CALL TO INTSTALL)
# L-8      CALL
# L-7
# L-6      SET(CLEAR)              SET(CLEAR)
# L-5      VINTFLAG                1=CSM, 0=LM
# L-4      INTYPFLAG                1=CONIC, 0=PRECISION
# L-3      SET(CLEAR)              SET(CLEAR)
# L-2      DIMOFLAG                1=W-MATRIX, 0=NO W-MATRIX
# L-1      D6OR9FLG                1=9X9, 0=6X6
# L        SET                    DLOAD
# L+1      STATEFLG                DESIRE PERMANENT UPDATE
# L+2      FINAL RAD.              OF STATE VECTOR
# L+3      STCALL                  RFINAL
# L+4      INTEGRV
# L        CALL
# L+1      INTEGRV                  NORMAL USE -- WILL UPDATE STATE
# L+2      RETURN                  VECTOR IF DIMOFLAG=1. (STATEFLG IS
#                                ALWAYS RESET IN INTEGRATION AFTER
#                                IT USED.)
#
# INPUT
# TDEC1 (PD 32D) TIME TO INTEGRATE TO      CSEC B-28
#
# OUTPUT
# SAME AS FOR PRECISION OR CONIC INTEGRATION
# THE PROGRAM WILL SET MOONFLAG, MIDFLAG DEPENDING ON
# THE PERMANENT STATE VECTOR REPRESENTATION.

```

	SETLOC	INTINIT	
	BANK		
	EBANK=	RRECTCSM	
	COUNT	13/INTIN	
STATEINT	TC	PHASCHNG	
	OCT	00052	
	CAF	PRI05	
	TC	FINDVAC	
	EBANK=	RRECTCSM	
	2CADR	STATINT1	
STATINT1	TC	TASKOVER	
	TC	INTPRET	
	BON	RTB	
		QUITFLAG	
		NOINT	# NO STATEINT IF V96
		LOADTIME	
# Page 1314			
	STORE	TDEC1	
	CLEAR	CALL	
		V96ONFLG	
		INTSTALL	
	SET	CALL	
		NODOFLAG	
		SETIFLGS	
	GOTO		
		STATEUP	
600SECS	2DEC	60000	
ENDINT	CLEAR	EXIT	
		STATEFLG	
	TC	PHASCHNG	
	OCT	20032	
	EXTEND		
	DCA	600SECS	
	TC	LONGCALL	
	EBANK=	RRECTHIS	
	2CADR	STATEINT	
SETIFLGS	TC	ENDOFJOB	
	SET	CLEAR	
		STATEFLG	
		INTYPFLG	
	CLEAR	CLEAR	
		DIMOFLEG	

```

                                D6OR9FLG
NOINT      RVQ
           EXIT
           TC      PHASCHNG
           OCT      2

           TC      DOWNFLAG
           ADRES    QUITFLAG

           TC      ENDOFJOB

# ATOPCSM TRANSFERS RRECT TO RRECT +41 TO RRECTCSM TO RRECTCSM +41
#
# CALLING SEQUENCE
#      L      CALL
#      L+1
#      ATOPCSM
#
# NORMAL EXIT AT L+2

ATOPCSM    STQ      RTB
           S2
           MOVEACSM
           SET      CALL
# Page 1315
           CMOONFLG
           SVDWN1
           BON      CLRG0
           MOONFLAG
           S2
           CMOONFLG
           S2
MOVEACSM    TC      SETBANK
           TS      DIFEQCNT      # INITIALIZE INDEX
           INDEX    DIFEQCNT
           CA      RRECT
           INDEX    DIFEQCNT
           TS      RRECTCSM
           CCS      DIFEQCNT      # IS TRANSFER COMPLETE
           TCF      MOVEACSM +1    # NO-LOOP
           TC      DANZIG          # COMPLETE -- RETURN

# PTOACSM TRANSFERS RRECTCSM TO RRECTCSM +41 TO RRECT TO RRECT +41
#
# CALLING SEQUENCE
#      L      CALL
#      PTOACSM

```

```

#
# NORMAL EXIT AT L+2

PTOACSM      RTB      BON
                MOVEPCSM
                CMOONFLG
                SETMOON
CLRMOON      CLEAR    SSP
                MOONFLAG
                PBODY
                0
                RVQ
SETMOON      SET      SSP
                MOONFLAG
                PBODY
                2
                RVQ
MOVEPCSM     TC        SETBANK
                TS      DIFEQCNT
                INDEX   DIFEQCNT
                CA      RRECTCSM
                INDEX   DIFEQCNT
                TS      RRECT
                CCS     DIFEQCNT
                TCF     MOVEPCSM +1
                TC      DANZIG

```

Page 1316

ATOPLEM TRANSFERS RRECT TO RRECT +41 TO RRECTLEM TO RRECTLEM +41

```

ATOPLEM      STQ      RTB
                S2
                MOVEALEM
                SET     CALL
                LMOONFLG
                SVDWN2
                BON     CLRG0
                MOONFLAG
                S2
                LMOONFLG
                S2
MOVEALEM     TC        SETBANK
                TS      DIFEQCNT
                INDEX   DIFEQCNT
                CA      RRECT
                INDEX   DIFEQCNT
                TS      RRECTLEM

```

July 12, 2016

741

CCS DIFEQCNT
TCF MOVEALEM +1
TC DANZIG

PTOALEM TRANSFERS RRECTLEM TO RRECTLEM +41 TO RRECT TO RRECT +41

PTOALEM BON RTB
SURFFLAG
USEPIOS
MOVEPLEM
BON GOTO
LMOONFLG
SETMOON
CLRMOON
MOVEPLEM TC SETBANK
TS DIFEQCNT
INDEX DIFEQCNT
CA RRECTLEM
INDEX DIFEQCNT
TS RRECT
CCS DIFEQCNT
TCF MOVEPLEM +1
TC DANZIG

USEPIOS SETPD VLOAD
0
RLS
PDDL PUSH
TDEC1

Page 1317

STODL TET
5/8
CALL
RP-TO-R
STOVL RCV
ZUNIT
STODL OD
TET
STODL 6D
5/8
SET CALL
MOONFLAG
RP-TO-R
VXV VXSC
RCV
OMEGMOON

NEEDED FOR SETTING X1 ON EXIT

```

                                STOVL   VCV
                                ZEROVEC
                                STORE   TDELTA V
                                AXT,2   SXA,2
                                           2
                                           PBODY
                                STCALL  TNUV
                                           A-PCHK
OMEGMOON      2DEC*   2.66169947 E-8 B+23*

SETBANK       CAF     INTBANK
              TS      BBANK
              CAF     FORTYONE
              TC      Q
              EBANK=  RRECTCSM
INTBANK       BBCON   INTEGRV

```

```

# SPECIAL PURPOSE ENTRIES TO ORBITAL INTEGRATION.  THESE ROUTINES PROVIDE ENTRANCES TO
# APPROPRIATE SWITCHES SET OR CLEARED FOR THE DESIRED INTEGRATION.
#
# CSMPREC AND LEMPREC PERFORM ORBIT INTEGRATION BY THE ENCKE METHOD TO THE TIME INDICATED.
# ACCELERATIONS DUE TO OBLATENESS ARE INCLUDED.  NO W-MATRIX INT. IS DONE.
# THE PERMANENT STATE VECTOR IS NOT UPDATED.
# CSMCONIC AND LEMCONIC PERFORM ORBIT INTEG. BY KEPLER'S METHOD TO THE TIME INDICATED.
# NO DISTURBING ACCELERATIONS ARE INCLUDED.  IN THE PROGRAM FLOW THE GIVEN
# STATE VECTOR IS RECTIFIED BEFORE SOLUTION OF KEPLER'S EQUATION.
#
# THE ROUTINES ASSUME THAT THE CSM (LEM) STATE VECTOR IN P-MEM IS VALID.
# SWITCHES SET PRIOR TO ENTRY TO THE MAIN INTEG. PROG ARE AS FOLLOWS:
#
#           CSMPREC           CSMCONIC           LEMPREC           LEMCONIC
# VINTFLAG   SET              SET              CLEAR           CLEAR
# INTYPFLG   CLEAR           SET              CLEAR           SET
# DIMOFLAG   CLEAR           CLEAR           CLEAR           CLEAR
# Page 1318
#
# CALLING SEQUENCE
#   L-X      STORE   TDEC1
#   L        CALL              (STCALL TDEC1)
#   L+1      CSMPREC      (CSMCONIC, LEMPREC, LEMCONIC)
#
# NORMAL EXIT TO L+2
#
# SUBROUTINES CALLED
#   INTEGRV1
#   PRECOUT FOR CSMPREC AND LEMPREC
#   CONICOUT FOR CSMCONIC AND LEMCONIC

```

```

#
# OUTPUT -- SEE PAGE 2 OF THIS LOG SECTION
#
# INPUT
#      TDEC1          TIME TO INTEGRATE TO.  CSECS B-28

CSMPREC      STQ      CALL
                  X1
                  INTSTALL
                  SXA,1 SET
                  IRETURN
                  VINTFLAG

IFLAGP      SET      CLEAR
                  PRECIFLG
                  DIMOFLAG
                  CLRG0
                  INTYPFLG
                  INTEGRV1

LEMPREC      STQ      CALL
                  X1
                  INTSTALL
                  SXA,1 CLRG0
                  IRETURN
                  VINTFLAG
                  IFLAGP

CSMCONIC     STQ      CALL
                  X1
                  INTSTALL
                  SXA,1 SET
                  IRETURN
                  VINTFLAG

IFLAGC      CLEAR    SETGO
                  DIMOFLAG
                  INTYPFLG
                  INTEGRV1

LEMCONIC     STQ      CALL
                  X1

# Page 1319
                  INTSTALL
                  SXA,1 CLRG0
                  IRETURN
                  VINTFLAG
                  IFLAGC

```

```

INTEGRVS      SET      SSP
                PRECIFLG
                PBODY
                0
                BOF     SSP
                MOONFLAG
                +3
                PBODY
                2
                STQ     VLOAD
                IRETURN
                ZEROVEC
                STORE   TDELTA
                STCALL  TNUV
                RECTIFY
                CLEAR   SET
                DIMOFLAG
                NEWIFLG
                SETGO
                RPQFLAG
                ALOADED

```

```

# INTEGRV IS AN ENTRY TO ORBIT INTEGRATION WHICH PERMITS THE CALLER,
# NORMALLY THE NAVIGATION PROGRAM, TO SET THE INTEG. FLAGS.  THE ROUTINE
# IS ENTERED AT INTEGRV1 BY CSMPREC ET. AL. AND AT ALOADED BY INTEGRVS.
# THE ROUTINE SETS UP A-MEMORY IF ENTERED AT INTEGRV,1 AND SETS THE INTEG.
# PROGRAM FOR PRECISION OR CONIC.
#
# THE CALLER MUST FIRST CALL INTSTALL TO CHECK IF INTEG. IS IN USE BEFORE
# SETTING ANY FLAGS.
#
# THE FLAGS WHICH SHOULD BE SET OR CLEARED ARE
#      VINTFLAG      (IGNORED WHEN ENTERED FROM INTEGRVS)
#      INTYPFLG
#      DIMOFLAG
#      D6OR9FLG
#
# CALLING SEQUENCE
#      L-X      CALL
#      L-Y              INTSTALL
#      L-1      SET OR CLEAR ALL FOUR FLAGS.  ALSO CAN SET STATEFLG IF DESIRED
#              AND DIMOFLAG IS CLEAR.
#      L          CALL
#      L+1              INTEGRV
#
# INITIALIZATION

```



```

#      FLAGS AS ABOVE
#      STORE TIME TO INTEGRATE TO IN TDEC1
#
# Page 1320
# OUTPUT
#      RATT      AS
#      VATT      DEFINED
#      TAT       BEFORE

INTEGRV      STQ
INTEGRV1     SET      IRETURN
              SET      RPQFLAG
              NEWIFLG
INTEGRV2     SSP
              QPRET
              ALOADED
              BON      GOTO
              VINTFLAG
              PTOACSM
              PTOALEM
              SETLOC   INTINIT1
              BANK
ALOADED      DLOAD
              TDEC1
              STORE    TDEC
              BOFF     GOTO
              INTYPFLG
              TESTLOOP
              RVCON
              SETLOC   INTINIT
              BANK
A-PCHK      BOF      CALL
              MIDFLAG
              ANDOUT   # DON'T MAKE ORIGIN CHANGE CHECK
              CHKSWTCH
              BPL      CALL
              ANDOUT   # NO ORIGIN CHANGE
              ORIGCHNG # MAKE THE SWITCH
ANDOUT      BOFCLR   EXIT
              STATEFLG
              RECTOUT
              TC       PHASCHNG
              OCT      04022
              TC       UPFLAG      # PHASE CHANGE HAS OCCURRED BETWEEN
              ADRES    REINTFLG    # INTSTALL AND INTWAKE

```

```

TC      INTPRET
SSP

      QPRET
      PHEXIT
BON     GOTO
      VINTFLAG
      ATOPCSM

# Page 1321
      ATOPLEM
PHEXIT  CALL
RECTOUT SETPD  GRP2PC
      CALL
      0
      RECTIFY
VLOAD   VSL*
      RRECT
      0,2
PDVL    VSL*      # RATT TO PDO
      VRECT
      0,2
PDDL    PDVL      # VATT TO PD6   TAT TO PD12
      TET
      RRECT
PDVL    PDDL*
      VRECT
      MUEARTH,2
PUSH    AXT,1
DEC     -10
BON     AXT,1
      MOONFLAG
      +2
DEC     -2
INTEXIT SETPD  BOV
      0
      +1
      CLEAR
      MIDAVFLG
      CLEAR
      AVEMIDSW      # ALLOW UPDATE OF DOWNLINK STATE VECTOR
      PRECIFLG
SLOAD   EXIT
      IRETURN
CA      MPAC
INDEX   FIXLOC
TS      QPRET
TC      INTWAKE

```

RVCON SETS UP ORBIT INTEGRATION TO DO A CONIC SOLUTION FOR POSITION AND
VELOCITY FOR THE INTERVAL (TET-TDEC)

```
RVCON      DLOAD  DSU
           TDEC
           TET
           STCALL TAU.
           RECTIFY
           CALL
           KEPPREP
           DLOAD  DAD
           TC
```

Page 1322

```
TET
STCALL    TET
RECTOUT
```

Page 1323

```
TESTLOOP  BOF      CLRG0
           QUITFLAG
           +3
           STATEFLG
           INEXIT      # STOP INTEGRATION
           +3          SETPD  LXA,2
           10D
           PBODY
           VLOAD  ABVAL
           RCV
           PUSH   CLEAR      # RC TO 10D
           MIDFLAG
           DSU*   BMN        # MIDFLAG=0 IF R G.T. RMP
           RME,2
           +3
           SET
           MIDFLAG
NORFINAL  DLOAD  DMP
           10D
           34D
           SR1R  DDV*
           MUEARTH,2
           SQRT  DMP
           .3D
           SR3   SR4        # DT IS TRUNCATED TO A MULTIPLE
           DLOAD SL
           MPAC
```

```

                                15D                #      OF 128 CSECS.
                                PUSH      BOV
                                MAXDT
                                BDSU      BMN
                                DT/2MAX
                                MAXDT
DT/2COMP      DLOAD      DSU
                                TDEC
                                TET
                                RTB       SL
                                SGNAGREE
                                8D
                                STORE     DT/2      # B-19
                                BOV       ABS
                                GETMAXDT
                                DSU       BMN        # IS TIME TO INTEG. TO GR THAN MAXTIME
                                12D
                                POOHCHK
USEMAXDT      DLOAD      SIGN
                                12D
                                DT/2
# Page 1324
                                STCALL    DT/2
                                POOHCHK
MAXDT         DLOAD      PDDL              # EXCHANGE DT/2MAX WITH COMPUTED MAX.
                                DT/2MAX
                                GOTO
                                DT/2COMP
GETMAXDT      RTB
                                SIGNMPAC
                                STCALL    DT/2
                                USEMAXDT
POOHCHK       DLOAD      ABS
                                DT/2
                                DSU       BMN
                                DT/2MIN
                                A-PCHK
                                SLOAD      BHIZ
                                MODREG
                                +3
                                GOTO
                                TIMESTEP
                                BON        # WAS THIS CALL VIA CSM(LEM)PREC
                                PRECIFLG
                                TIMESTEP    # YES
                                DLOAD      DSU

```

		DT/2	
		12D	
	BMN	BOFCLR	
		A-PCHK	
		NEWIFLG	
		TIMESTEP	
	DLOAD	DSU	
		TDEC	
		TET	
	BMN		# NO BACKWARD INTEGRATION
		INTEXIT	
	PDDL	SR4	
		DT/2	# IS 4(DT) LS (TDEC - TET)
	SR2R	BDSU	
	BMN	GOTO	
		INTEXIT	
		TIMESTEP	
DT/2MIN	2DEC	3 B-20	
DT/2MAX	2DEC	4000 E2 B-20	
INTSTALL	EXIT		
	CAF	ZERO	
ALLSTALL	TS	L	
	CA	RASFLAG	
	INDEX	L	
# Page 1325			
	MASK	INTBITAB	# IS THIS STALL AREA FREE
	EXTEND		
	BZF	OKTOGRAB	# YES
	INDEX	L	
	CAF	WAKESTAL	
	TC	JOBSLEEP	
INTWAKEO	EXIT		
	TCF	INTWAKE1	
INTWAKE	CS	RASFLAG	# IS THIS INSTALLED ROUTINE TO BE
	MASK	REINTBIT	# RESTARTED
	CCS	A	
	TC	INTWAKE1	# NO
	INDEX	FIXLOC	
	CA	QPRET	
	TS	TBASE2	# YES, DON'T RESTART WITH SOMEONE ELSE'S Q
	TC	PHASCHNG	

	OCT	04022	
	CA	TBASE2	
	INDEX	FIXLOC	
	TS	QPRET	
	CAF	REINTBIT	
	MASK	RASFLAG	
	EXTEND		
	BZF	GOBAC	# DON'T INTWAKE IF WE CAME HERE VIA RESTART
INTWAKE1	CAF	ZERO	
WAKE	TS	STALTEM	# INDEX OF ANY STALL USER
WAKE1	INDEX	STALTEM	
	CAF	WAKESTAL	
	INHINT		
	TC	JOBWAKE	
	CCS	LOCCTR	
	TCF	WAKE1	# MAY BE MORE TO WAKE UP
FORTYONE	DEC	41	
	INDEX	STALTEM	
	CS	INTBITAB	
	MASK	RASFLAG	
	TS	RASFLAG	# RELEASE STALL AREA
	RELINT		
	TCF	GOBAC	
OKTOGRAB	INDEX	L	# NO, WAIT UNTIL AVAILABLE
	CAF	INTFLBIT	
	ADS	RASFLAG	
GOBAC	TC	INTPRET	
	RVQ		
# Page 1326			
ERASTAL1	EXIT		
	CAF	ONE	
	TCF	ALLSTALL	
ERASTAL2	EXIT		
	CAF	TWO	
	TCF	ALLSTALL	
ERASWAK1	CAF	ONE	
	TCF	WAKE	
ERASWAK2	CAF	TWO	
	TCF	WAKE	
WAKESTAL	CADR	INTSTALL +1	
	CADR	ERASTAL1 +1	
	CADR	ERASTAL2 +1	
STALTEM	EQUALS	MPAC	

```

INTBITAB      OCT      20100
               OCT      10040
               OCT      04020

```

```

# Page 1327

```

```

# AVETOMID

```

```

#

```

```

# THIS ROUTINE PERFORMS THE TRANSITION FROM A THRUSTING PHASE TO THE COAST
# PHASE BY INITIALIZING THIS VEHICLE'S PERMANENT STATE VECTOR WITH THE
# VALUES LEFT BY THE AVERAGEG ROUTINE IN RN,VN,PIPTIME.

```

```

#

```

```

# BEFORE THIS IS DONE THE W-MATRIX, IF IT'S VALID (OR WFLAG OR RENDWFLT IS
# SET) IS INTEGRATED FORWARD TO PIPTIME WITH THE PRE-THRUST STATE VECTOR.

```

```

#

```

```

# IN ADDITION, THE OTHER VEHICLE IS INTEGRATED (PERMANENT) TO PIPTIME.

```

```

#

```

```

# FINALLY TRKMKCNT IS ZEROED.

```

```

      SETLOC  INTINIT2
      BANK

```

```

      COUNT*  $$/INTIN
AVETOMID     STQ   BON

```

```

      EGRESS
      RENDWFLG
      INT/W

```

```

      # W-MATRIX VALID, GO INTEGRATE IT

```

```

      BON

```

```

      ORBWFLAG
      INT/W

```

```

      # W-MATRIX VALID, GO INTEGRATE IT.

```

```

SETCOAST

```

```

      AXT,2  CALL
            2

```

```

      # NOW MOVE PROPERLY SCALED RN,UN AS WELL AS
      # PIPTIME TO INTEGRATION ERASABLES.

```

```

      INTSTALL
      BON   AXT,2
            AMOONFLG
            +2
            0

```

```

      VLOAD  VSR*
            RN
            0,2

```

```

      STORE  RRECT

```

```

      STODL  RCV
            PIPTIME

```

```

      STOVL  TET
            VN

```

```

      VSR*   CALL

```

```

                                0,2
                                MINIRECT      # FINISH SETTING UP STATE VECTOR
RTB      SSP
                                MOVATHIS      # PUT TEMP STATE VECTOR INTO PERMANENT
                                TRKMKCNT
                                0
SET      BON
                                CMOONFLG
                                AMOONFLG
                                +3
CLEAR    CMOONFLG

BON      DLOAD      # NOW DO LM
                                SURFFLAG
                                FAZAB5      # NO COASTING LM
                                PIPTIME
STCALL   TDEC1
                                SETIFLGS
CLEAR    CALL
                                VINTFLAG
                                INTEGRV
GOTO

INT/W    DLOAD      EGRESS
                                CALL
                                PIPTIME      # INTEGRATE W THRU BURN
                                INTSTALL
SET      SET
                                DIMOFLAG      # DO W-MATRIX
                                AVEMIDSW      # SO WON'T CLOBBER RN,VN,PIPTIME
CLEAR    SET
                                D6OR9FLG
                                VINTFLAG
STCALL   TDEC1
                                INTEGRV
GOTO
                                SETCOAST

```

Page 1329

MIDTOAV1

#

THIS ROUTINE INTEGRATES (PRECISION) TO THE TIME SPECIFIED IN TDEC1.

IF, AT THE END OF AN INTEGRATION TIME STEP, CURRENT TIME PLUS A DELTA

TIME (SEE TIMEDELT....BASED ON THE COMPUTATION TIME FOR ONE TIME STEP)

IS GREATER THAN THE DESIRED TIME, ALARM 1703 IS SET AND THE INTEGRATION


```

# IS DONE TO THE CURRENT TIME.
# RETURN IS IN BASIC TO THE RETURN ADDRESS PLUS ONE.
#
# IF THE INTEGRATION IS FINISHED TO THE DESIRED TIME, RETURN IS IN BASIC
# TO THE RETURN ADDRESS.
#
# IN EITHER CASE, BEFORE RETURNING, THE EXTRAPOLATED STATE VECTOR IS TRANSFERRED
# FROM R,VATT TO R,VN1 -- PIPTIME1 IS SET TO THE FINISHING INTEGRATION
# TIME AND MPAC IS SET TO THE DELTA TIME --
#           TAT MINUS CURRENT TIME

# MIDTOAV2
#
# THIS ROUTINE INTEGRATES THIS VEHICLE'S STATE VECTOR TO THE CURRENT TIME PLUS
# INCREMENTS OF TIMEDELT SUCH THAT THE DIFFERENCE BETWEEN CURRENT TIME
# AND THE STATE VECTOR TIME AT THE END OF THE LAST STEP IS AT LEAST 5.6
# SECS.
# NO INPUTS ARE REQUIRED OF THE CALLER.  RETURN IS IN BASIC TO THE RETURN
# ADDRESS WITH THE ABOVE TRANSFERS TO R,VN1 -- PIPTIME1 -- AND MPAC DONE

                SETLOC  INTINIT
                BANK
                EBANK=  IRETURN1
MIDTOAV2        STQ      CALL
                  IRETURN1
                  INTSTALL
                DLOAD    CLEAR
                  TIMEDELT
                  MID1FLAG
                STCALL   T-TO-ADD
                  ENTMID2

MIDTOAV1        STQ      CALL
                  IRETURN1
                  INTSTALL
                SET      RTB
                  MID1FLAG
                  LOADTIME
                DAD      BDSU          # INITIAL CHECK, IS TDEC1 IN THE FUTURE
                  TIMEDELT
                  TDEC1
                BPL      CALL
                  ENTMID1

# Page 1330
                NOTIME          # NO, SET ALARM, SWITCH TO MIDTOAV2

```

ENTMID2	RTB	DAD	
		LOADTIME	
		T-TO-ADD	
	STORE	TDEC1	
ENTMID1	CLEAR	CALL	
		DIMOFLLG	# NO W-MATRIX
		THISVINT	
	CLEAR	SET	
		INTYPFLG	
		MIDAVFLG	# LET INTEG. KNOW THE CALL IS FOR MIDTOAV.
	CALL		
		INTEGRV	# GO INTEGRATE
	SXA,2	SXA,1	
		RTX2	
		RTX1	
	CLEAR	SLOAD	
		AMOUNFLG	
		RTX2	
	BZE	SET	
		+2	
		AMOUNFLG	
	VLOAD		
		RATT	
	STOVL	RN1	
		VATT	
	STODL	VN1	
		TAT	
	STORE	PIPTIME1	
	EXIT		
		INHINT	
		EXTEND	
	DCS	TIME2	
	DAS	MPAC	
	TC	TPAGREE	
	CA	IRETURN1	
	TC	BANKJUMP	
CKMID2	BOF	RTB	
		MID1FLAG	
		MID2	
		LOADTIME	
	DAD	BDSU	
		TIMEDELT	
		TDEC	

July 12, 2016

755

```
# Page 1331

BPL      CALL
          TESTLOOP      # YES

          NOTIME

TIMEINC   RTB      DAD
          LOADTIME
          T-TO-ADD
          STCALL    TDEC
          TESTLOOP

MID2      DLOAD    DSU
          TDEC
          TET
          ABS      DSU
          3CSECS
          BPL
          TIMEINC

          RTB      BDSU      # SEE IF 5.6 SECS. AVAILABLE TO CALLER
          LOADTIME
          TET
          DSU      BPL
          5.6SECS
          A-PCHK      # YES. GET OUT.

          DLOAD    DAD
          T-TO-ADD
          TIMEDELT
          STCALL    T-TO-ADD
          TIMEINC
          NOTIME    CLEAR    EXIT      # TOO LATE
          MID1FLAG
          INCR      IRETURN1
          TC        ALARM      # SET ERROR EXIT (CALLOC +2)
          OCT      1703      # INSUFFICIENT TIME FOR INTEGRATION --
          TC        INTPRET    #          TIG WILL BE SLIPPED...
          DLOAD
          TIMEDELT
          STORE     T-TO-ADD
          RVQ

3CSECS    2DEC      3

TIMEDELT  2DEC      1250
```

```

5.6SECS      2DEC      560

                BANK      27
                SETLOC    UPDATE2
                BANK
                EBANK=    INTWAKUQ

# Page 1332
COUNT*      $$/INTIN

INTWAKUQ      =          INTWAK1Q      # TEMPORARY UNTIL NAME OF INTWAK1Q IS CHNG

INTWAKEU      RELINT
                EXTEND
                QXCH      INTWAKUQ      # SAVE Q FOR RETURN

                TC          INTPRET

                SLOAD      BZE
                                UPSVFLAG
                                INTWAKUP
                                # IS THIS A CSM/LEM STATE VECTOR UPDATE
                                # REQUEST.  IF NOT GO TO INTWAKUP.

                VLOAD
                                # MOVE PRECT(6) AND VRECT(6) INTO
                                # RCV(6) AND VCV(6) RESPECTIVELY.
                STOVL      RCV
                                VRECT
                CALL
                                # NOW GO TO 'RECTIFY +13D' TO
                                # STORE VRECT INTO VCV AND ZERO OUT
                                RECTIFY +13D
                                # TDELTA(6),TNUV(6),TC(2), AND XKEP(2)
                SLOAD      ABS
                                # COMPARE ABSOLUTE VALUE OF 'UPSVFLAG'
                                UPSVFLAG
                                # TO 'UPDATE MOON STATE VECTOR CODE'
                DSU          BZE
                                # TO DETERMINE WHETHER THE STATE VECTOR TO
                                UPMNSVCD
                                # BE UPDATED IS IN THE EARTH OR LUNAR
                                INTWAKEM
                                # SPHERE OF INFLUENCE.....
                AXT,2      CLRG0
                                # EARTH SPHERE OF INFLUENCE.
                DEC          0
                                MOONFLAG
                                INTWAKEC

INTWAKEM      AXT,2      SET
                DEC          2
                                # LUNAR SPHERE OF INFLUENCE.
                                MOONFLAG

INTWAKEC      SLOAD      BMN
                                # COMMON CODING AFTER X2 INITIALIZED AND
                                # MOONFLAG SET (OR CLEARED).
                                UPSVFLAG
                                # IS THIS A REQUEST FOR A LEM OR CSM
                                INTWAKLM
                                # STATE VECTOR UPDATE.....
                CALL
                                # UPDATE CSM STATE VECTOR
                                ATOPCSM

```

```

                                CLEAR  GOTO
                                ORBWFLAG
                                INTWAKEX

INTWAKLM      CALL              # UPDATE LM STATE VECTOR
                                ATOPLEM

INTWAKEX      CLEAR
                                RENDWFLG

INTWAKUP      SSP      CALL              # REMOVE 'UPDATE STATE VECTOR INDICATOR'
# Page 1333    0
                                UPSVFLAG
                                0
                                INTWAKEO      # RELEASE 'GRAB' OF ORBIT INTEG.
                                EXIT

                                TC      PHASCHNG
                                OCT      04026
                                TC      INTWAKUQ

UPMNSVCD      OCT      2
                                OCT      0

GRP2PC        STQ      EXIT
                                GRP2SVQ
                                TC      PHASCHNG
                                OCT      04022
                                TC      INTPRET
                                GOTO
                                GRP2SVQ

```

This code is written to file `src/INTEGRATION-INITIALIZATION.s`.

B.45 INTER-BANK COMMUNICATION

```

758  <src/INTER-BANK-COMMUNICATION.s 758>≡
      # Copyright:    Public domain.
      # Filename:     INTER-BANK_COMMUNICATION.agc
      # Purpose:      Part of the source code for Comanche, build 055. It
      #               is part of the source code for the Command Module's
      #               (CM) Apollo Guidance Computer (AGC), Apollo 11.
      # Assembler:    yaYUL
      # Reference:     pp. 1103-1106
      # Contact:       Ron Burkey <info@sandroid.org>
      # Website:       http://www.ibiblio.org/apollo.
      # Mod history:   2009-05-08 RSB  Adapted from Colossus249/ file of same name
      #               and page images. Corrected various typos
      #               in the transcription of program comments,
      #               and these should be back-ported to
      #               Colossus249.
      #
      # The contents of the "Comanche055" files, in general, are transcribed
      # from scanned documents.
      #
      # Assemble revision 055 of AGC program Comanche by NASA
      # 2021113-051. April 1, 1969.
      #
      # This AGC program shall also be referred to as Colossus 2A
      #
      # Prepared by
      #           Massachusetts Institute of Technology
      #           75 Cambridge Parkway
      #           Cambridge, Massachusetts
      #
      # under NASA contract NAS 9-4065.
      #
      # Refer directly to the online document mentioned above for further
      # information. Please report any errors to info@sandroid.org.
      #
      # Page 1103
      # THE FOLLOWING ROUTINE CAN BE USED TO CALL A SUBROUTINE IN ANOTHER BANK. IN THE BANK
      # CADR OF THE SUBROUTINE IMMEDIATELY FOLLOWS THE TC BANKCALL INSTRUCTION, WITH C(A) A

      BLOCK      02
      COUNT      02/BANK

      BANKCALL    DXCH    BUF2          # SAVE INCOMING A,L.
                  INDEX   Q            # PICK UP CADR.
                  CA      0

```

July 12, 2016

759

```

                INCR    Q                # SO WE RETURN TO THE LOC. AFTER THE CADR.

# SWCALL IS IDENTICAL TO BANKCALL, EXCEPT THAT THE CADR ARRIVES IN A.

SWCALL          TS      L
                LXCH    FBANK           # SWITCH BANKS, SAVING RETURN.
                MASK    LOW10           # GET SUB-ADDRESS OF CADR.
                XCH     Q               # A,L NOW CONTAINS DP RETURN.
                DXCH    BUF2           # RESTORING INPUTS IF THIS IS A BANKCALL.
                INDEX   Q
                TC      10000           # SETTING Q TO SWRETURN

SWRETURN        XCH     BUF2 +1         # COMES HERE TO RETURN TO CALLER. C(A,L)
                XCH     FBANK           # ARE PRESERVED FOR RETURN.
                XCH     BUF2 +1
                TC      BUF2

# THE FOLLOWING ROUTINE CAN BE USED AS A UNILATERAL JUMP WITH C(A,L) PRESERVED AND THE CADR IMMEDIATE
# FOLLOWING THE TC POSTJUMP INSTRUCTION.

POSTJUMP        XCH     Q               # SAVE INCOMING C(A).
                INDEX   A               # GET CADR.
                CA      0

# BANKJUMP IS THE SAME AS POSTJUMP, EXCEPT THAT THE CADR ARRIVES IN A.

BANKJUMP        TS      FBANK
                MASK    LOW10
                XCH     Q               # RESTORING INPUT C(A) IF THIS WAS A
Q+10000         INDEX   Q               # POSTJUMP.
PRI012          TCF     10000           # PRI012 = TCF 10000 = 12000

# Page 1104
# THE FOLLOWING ROUTINE GETS THE RETURN CADR SAVED BY SWCALL OR BANKCALL AND LEAVES IT IN A.

MAKECADR        CAF     LOW10
                MASK    BUF2
                AD      BUF2 +1
                TC      Q

SUPDACAL        TS      MPTMP
                XCH     FBANK           # SET FBANK FOR DATA.
                EXTEND
                ROR     SUPERBNK        # SAVE FBANK IN BITS 15-11, AND
                XCH     MPTMP           # SUPERBANK IN BITS 7-5.
                MASK    LOW10
```

```

XCH      L                # SAVE REL. ADR. IN BANK, FETCH SUPERBITS
INHINT
EXTEND
WRITE    SUPERBNK        # SET SUPERBANK FOR DATA.
INDEX    L
CA       10000           # PINBALL (FIX MEM DISP) PREVENTS DCA HERE
XCH      MPTMP           # SAVE 1ST WD, FETCH OLD FBANK AND SBANK.
EXTEND
WRITE    SUPERBNK        # RESTORE SUPERBANK.
RELINT
TS       FBANK           # RESTORE FBANK.
CA       MPTMP           # RECOVER FIRST WORD OF DATA.
RETURN   # 24 WDS. DATACALL 516 MU, SUPDACAL 432 MU

```

Page 1105

THE FOLLOWING ROUTINES ARE IDENTICAL TO BANKCALL AND SWCALL EXCEPT THAT THEY ARE US

```

IBNKCALL  DXCH    RUPTREG3    # USES RUPTREG3,4 FOR DP RETURN ADDRESS.
           INDEX   Q
           # Was CAF --- RSB 2009
           CA      0
           INCR    Q

```

```

ISWCALL   TS       L
           LXCH    FBANK
           MASK    LOW10
           XCH     Q
           DXCH    RUPTREG3
           INDEX   Q
           TC      10000

```

```

ISWRETRN  XCH      RUPTREG4
           XCH      FBANK
           XCH      RUPTREG4
           TC       RUPTREG3

```

2. USPRCADR ACCESSES INTERPRETIVE CODING IN OTHER THAN THE USER'S FBANK. THE CALL

```

#      L      TC      USPRCADR
#      L+1    CADR    INTPRETX    # INTPRETX IS THE INTERPRETIVE CODING
#                                     # RETURN IS TO L+2

```

```

USPRCADR  TS       LOC        # SAVE A
           CA       BIT8
           TS       EDOP       # EXIT INSTRUCTION TO EDOP
           CA       BBANK
           TS       BANKSET    # USER'S BBANK TO BANKSET

```


INDEX	Q	
CA	0	
TS	FBANK	# INTERPRETIVE BANK TO FBANK
MASK	LOW10	# YIELDS INTERPRETIVE RELATIVE ADDRESS
XCH	Q	# INTERPRETIVE ADDRESS TO Q, FETCHING L+1
XCH	LOC	# L+1 TO LOC, RETRIEVING ORIGINAL A
TCF	Q+10000	

Page 1106

THERE ARE FOUR POSSIBLE SETTINGS FOR CHANNEL 07. (CHANNEL 07 CONTAINS SUPERBANK SETTING.)

#

# SUPERBANK	SETTING	S-REG. VALUE	PSEUDO-FIXED BANK NUMBERS	OCTAL PSEUDO ADDRESSES	
# -----	-----	-----	-----	-----	
# SUPERBANK 3	OXX	2000 - 3777	30 - 37	70000 - 107777	(WHERE XX CAN BE
#					WILL USUALLY BE
# SUPERBANK 4	100	2000 - 3777	40 - 47	110000 - 127777	(AS FAR AS IT C
#					ONLY BANKS 40-4
#					AND ARE PRESENT
# SUPERBANK 5	101	2000 - 3777	50 - 57	130000 - 147777	(PRESENTLY NOT
#					THE USER)
# SUPERBANK 6	110	2000 - 3777	60 - 67	150000 - 167777	(PRESENTLY NOT
#					THE USER)

*** THIS ROUTINE MAY BE CALLED BY ANY PROGRAM LOCATED IN BANKS 00 - 27. I.E., NO PROGRAM LIV

SUPERBANK SHOULD USE SUPERSW. ***

#

SUPERSW MAY BE CALLED IN THIS FASHION:

#	CAF	ABBCON	WHERE -- ABBCON BCON SOMETHING --
#	TCR	SUPERSW	(THE SUPERBNK BITS ARE IN THE BCON)
#	
#	.	.	
#	.	.	

OR IN THIS FASHION:

#	CAF	SUPERSET	WHERE SUPERSET IS ONE OF THE FOUR AVAILABLE
#	TCR	SUPERSW	SUPERBANK BIT CONSTANTS:
#	SUPER011 OCTAL 60
#	.	.	SUPER100 OCTAL 100
#	.	.	SUPER101 OCTAL 120
#	.	.	SUPER110 OCTAL 140

SUPERSW

EXTEND

WRITE SUPERBNK

WRITE BITS 7-6-5 OF THE ACCUMULATOR INTO
CHANNEL 07

TC Q

TC TO INSTRUCTION FOLLOWING

TC SUPERSW

This code is written to file `src/INTER-BANK-COMMUNICATION.s`.

B.46 INTERPRETER

763

<src/INTERPRETER.s 763>≡

```
# Copyright:    Public domain.
# Filename:     INTERPRETER.agc
# Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
#               It is part of the source code for the Command Module's (CM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:   yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1107-1199
# Mod history:  2009-05-08 RSB   Adapted from the Luminary131/ file of the
#               same name, using Comanche055 page images.
#               2009-05-20 RSB   Corrections:  P00D00 -> P00D00, fixed a
#               "Page N" reference.
#               2009-05-21 RSB   Corrected definition of 5B10, which overflowed
#               integer arithmetic.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum.  The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum.  Many
# thanks to both.  The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo.  If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 055 of AGC program Comanche by NASA
#       2021113-051.  10:28 APR. 1, 1969
#
#       This AGC program shall also be referred to as
#               Colossus 2A
#
# Page 1107
# SECTION 1:  DISPATCHER
#
# ENTRY TO THE INTERPRETER.  INTPRET SETS LOC TO THE FIRST INSTRUCTION, BANKSET TO THE BBANK OF
# OBJECT INTERPRETIVE PROGRAM, AND INTBIT15 TO THE BIT15 CONTENTS OF FBANK.  INTERPRETIVE PROGRAMS
# VIRTUALLY ALL BANKS PRESENT UNDER ANY SUPER-BANK SETTING, WITH THE RESTRICTION THAT PROGRAMS
# (BIT15 OF FBANK = 1) DO NOT REFER TO LOWBANKS, AND VICE-VERSA.  THE INTERPRETER DOES NOT SWITCH
# E-BANK SWITCHING OCCURS WHENEVER GENERAL ERASABLE (100-3777) IS ADDRESSED.
```

```

COUNT*  $$/INTER
INTPRET  RELINT
EXTEND
QXCH     LOC
+2       CA      BBANK
        TS      BANKSET
        MASK    BIT15
        TS      INTBIT15

        TS      EDOP
        TCF     NEWOPS

INTRSM   LXCH    BBANK
        TCF     INTPRET +3

# DLOAD LOADS MPAC, MPAC +1, LEAVING ZERO IN MPAC +2.

DLOAD    EXTEND
        INDEX   ADDRWD
        DCA     0
SLOAD2   DXCH    MPAC
        CAF     ZERO

# Page 1108
# AT THE END OF MOST INSTRUCTIONS, CONTROL IS GIVEN TO DANZIG TO DISPATCH THE NEXT OF

        TS      MPAC +2
NEWMODE   TS      MODE
DANZIG    CA      BANKSET
        TS      BBANK
NOIBNKSW  CCS     EDOP
        TCF     OPJUMP

        CCS     NEWJOB
        TCF     CHANG2

        INCR    LOC

# ITRACE (1) REFERS TO "NEWOPS"
NEWOPS   INDEX   LOC

```

SET LOC TO THE WORD FOLLOWING THE T

INTERPRETIVE BRANCHES FINISH HERE.

GET 15TH BIT FOR INDEXABLE ADDRESS

MAKE SURE NO INSTRUCTIONS LEFT OVER

PICK UP OP CODE PAIR AND BEGIN.

RESUME SUSPENDED INTERPRETIVE JOB

LOAD DP C(C(ADDRWD)) INT MPAC,MPAC

ZERO MPAC +2

AND DECLARE DP MODE

PROLOGUE FOR MODE-CHANGING INSTRUCT

SET BBANK BEFORE TESTING NEWJOB SO

IT MAY BE SAVED DIRECTLY BY CHANJO

SEE IF AN ORDER CODE IS LEFT OVER

THE LAST PAIR RETRIEVED. IF SO, E

EDOP IS SET TO ZERO ON ITS RE-EDIT

SEE IF A JOB OF HIGHER PRIORITY IS

PRESENT, AND IF SO, CHANGE JOBS.

ADVANCE THE LOCATION COUNTER.

ENTRY TO BEGIN BY PICKING OP CODE P

July 12, 2016

765

	CA	0	# MAY BE AN OPCODE PAIR OR A STORE CODE.
	CCS	A	# TEST SIGN AND GET DABS(A).
	TCF	DOSTORE	# PROCESS STORE CODE.
LOW7	OCT	177	
	TS	EDOP	# OP CODE PAIR. LEAVE THE OTHER IN EDOP
	MASK	LOW7	# WHERE CCS EDOP WILL HONOR IT NEXT.
OPJUMP	TS	CYR	# LOWWD ENTERS HERE IF A RIGHT-HAND OP
	CCS	CYR	# CODE IS TO BE PROCESSED. TEST PREFIXES.
	TCF	OPJUMP2	# TEST SECOND PREFIX BIT.
	TCF	EXIT	# +0 OP CODE IS EXIT

Page 1109

PROCESS ADDRESSES WHICH MAY BE DIRECT, INDEXED, OR REFERENCE THE PUSHDOWN LIST.

ADDRESS	MASK	BIT1	# SEE IF ADDRESS IS INDEXED. CYR CONTAINED
	CCS	A	# 400XX, SO BIT 1 IS NOW AS IT WAS IN CYR.
	TCF	INDEX	# FORM INDEXED ADDRESS.
DIRADRES	INDEX	LOC	# LOOK AHEAD TO NEXT WORD TO SEE IF
OCT40001	CS	1	# ADDRESS IS GIVEN.
	CCS	A	
	TCF	PUSHUP	# IF NOT.
NEG4	DEC	-4	
	INCR	LOC	# IF SO, TO SHOW WE PICKED UP A WORD.
	TS	ADDRWD	

Page 1110

FINAL DIGESTION OF DIRECT ADDRESSES OF OP CODES WITH 01 PREFIX IS DONE HERE. IN EACH CASE,
REQUIRED 12-BIT SUB-ADDRESS IS LEFT IN ADDRWD, WITH ANY REQUIRED E OR F BANK SWITCHING DONE.
THAN 45D ARE TAKEN TO BE RELATIVE TO THE WORK AREA. THE OP CODE IS NOW IN BITS 1-5 OF CYR WITH

	AD	-ENDVAC	# SEE IF ADDRESS RELATIVE TO WORK AREA.
	CCS	A	
	AD	-ENDERAS	# IF NOT, SEE IF IN GENERAL ERASABLE.
	TCF	IERASTST	
NETZERO	CA	FIXLOC	# IF SO, LEAVE THE MODIFIED ADDRESS IN
	ADS	ADDRWD	# ADDRWD AND DISPATCH.
ITR15	INDEX	CYR	# THIS INDEX MAKES THE NEXT INSTRUCTION
	7	INDJUMP -1	# TCF INDJUMP + OP, EDITING CYR.

IERASTST	EXTEND		
	BZMF	GEADDR	# GO PROCESS GENERAL-ERASABLE ADDRESS
	MASK	LOW10	# FIXED BANK ADDRESS. RESTORE AND ADDRESS
	AD	LOW10	# SWITCH BANKS AND LEAVE SUBADDRESS 1
	XCH	ADDRWD	# ADDRWD FOR OPERAND RETRIEVAL. (THIS
	AD	INTBIT15	# METHOD PRECLUDES USE OF THE LAST
	TS	FBANK	# LOCATION IN EACH FBANK.)
ITR12	INDEX	CYR	
	7	INDJUMP -1	
GEADDR	MASK	LOW8	
	AD	OCT1400	
	XCH	ADDRWD	
	TS	EBANK	
ITR10	INDEX	CYR	
	7	INDJUMP -1	

Page 1111

THE FOLLOWING ROUTINE PROCESSES INTERPRETIVE INDEXED ADDRESSES. AN INTERPRETER IN

CONTAIN THE ADDRESS OF ANY ERASABLE REGISTER (0-42 BEING RELATIVE TO THE VAC AREA)

BANK, OR ANY INTEGER IN THAT RANGE.

DODLOAD*	CAF	DLOAD*	# STODL* COMES HERE TO PROCESS LOAD A
	TS	CYR	# (STOVL* ENTERS HERE).
INDEX	CA	FIXLOC	# SET UP INDEX LOCATION.
	TS	INDEXLOC	
	INCR	LOC	# (ADDRESS ALWAYS GIVEN).
	INDEX	LOC	
	CS	0	
	CCS	A	# INDEX 2 IF ADDRESS STORED COMPLEMEN
	INCR	INDEXLOC	
	NOOP		
	TS	ADDRWD	# 14 BIT ADDRESS TO ADDRWD.
	MASK	HIGH4	# IF ADDRESS GREATER THAN 2K, ADD INT
	EXTEND		
	BZF	INDEX2	
	CA	INTBIT15	
	ADS	ADDRWD	
INDEX2	INDEX	INDEXLOC	
	CS	X1	
	ADS	ADDRWD	# DO AUGMENT, IGNORING AND CORRECTING

July 12, 2016

767

	MASK	HIGH9	# SEE IF ADDRESS IS IN WORK AREA.
	EXTEND		
	BZF	INDWORK	
	MASK	HIGH4	# SEE IF IN FIXED BANK.
	EXTEND		
	BZF	INDERASE	
	CA	ADDRWD	# IN FIXED -- SWITCH BANKS AND CREATE
	TS	FBANK	# SUB-ADDRESS
	MASK	LOW10	
	AD	2K	
	TS	ADDRWD	
ITR11	INDEX	CYR	
	3	INDJUMP -1	
INDWORK	CA	FIXLOC	# MAKE ADDRWD RELATIVE TO WORK AREA.
	TCF	ITR13 -1	
INDERASE	CA	OCT1400	
	XCH	ADDRWD	
	TS	EBANK	
	MASK	LOW8	
	ADS	ADDRWD	
-1			
# Page 1112			
ITR13	INDEX	CYR	
	3	INDJUMP -1	

Page 1113

PUSH-UP ROUTINES. WHEN NO OPERAND ADDRESS IS GIVEN, THE APPROPRIATE OPERAND IS TAKEN FROM THE
LIST. IN MOST CASES THE MODE OF THE RESULT (VECTOR OR SCALAR) OF THE LAST ARITHMETIC OPERATION
IS THE SAME AS THE TYPE OF OPERAND DESIRED (ALL ADD/SUBTRACT ETC.). EXCEPTIONS TO THIS GENERAL
RULE ARE LISTED BELOW (NOTE THAT IN EVERY CASE THE MODE REGISTER IS LEFT INTACT):

- #
- # 1. VXSC AND V/SC WANT THE OPPOSITE TYPE OF OPERAND, E.G., IF THE LAST OPERATION YIELDED A VECTOR
RESULT, VXSC WANTS A SCALAR.
 - # 2. THE LOAD CODES SHOULD LOAD THE ACCUMULATOR INDEPENDENT OF THE RESULT OF THE LAST OPERATION.
INCLUDES VLOAD, DLOAD, TLOAD, PDDL, AND PDVL (NO PUSHUP WITH SLOAD).
 - # 3. SOME ARITHMETIC OPERATIONS REQUIRE A STANDARD TYPE OF OPERAND REGARDLESS OF THE TYPE OF OPERAND.
THIS INCLUDES SIGN WANTING DP AND TAD REQUIRING TP.

PUSHUP	CAF	OCT23	# IF THE LOW 5 BITS OF CYR ARE LESS THAN
	MASK	CYR	# 20, THIS OP REQUIRES SPECIAL ATTENTION.
	AD	-OCT10	# (NO -0).

	CCS	A	
	TCF	REGUP	# FOR ALL CODES GREATER THAN OCT 7.
-OCT10	OCT	-10	
	AD	NEG4	# WE NOW HAVE 7 -- OP CODE (MOD4). SEE IF
	CCS	A	# THE OP CODE (MOD4) IS THREE (REVERSE).
	INDEX	A	# NO -- THE MODE IS DEFINITE. PICK UP THE
	CS	NO.WDS	
	TCF	REGUP +2	
	INDEX	MODE	# FOR VXSC AND V/SC WE WANT THE REQUIRED
	CS	REVCNT	# PUSHLOC DECREMENT WITHOUT CHANGING THE
	TCF	REGUP +2	# MODE AT THE IS TIME.
REGUP	INDEX	MODE	# MOST ALL OP CODES PUSHUP HERE.
	CS	NO.WDS	
+2	ADS	PUSHLOC	
	TS	ADDRWD	
ITR14	INDEX	CYR	
	7	INDJUMP -1	# (THE INDEX MAKES THIS A TCF.)
	OCT	2	# REVERSE PUSHUP DECREMENT. VECTOR TAKES 2
REVCNT	OCT	6	# WORDS, SCALAR TAKES 6.
	OCT	6	
NO.WDS	OCT	2	# CONVENTIONAL DECREMENT IS 6 WORDS VECTOR
OCTAL3	OCT	3	# 2 IN DP, AND 3 IN TP.
	OCT	6	
# Page 1114			
# TEST THE SECOND PREFIX BIT TO SEE IF THIS IS A MISCELLANEOUS OR A UNARY/SHORT SHIFT			
OPJUMP2	CCS	CYR	# TEST SECOND PREFIX BIT.
	TCF	OPJUMP3	# TEST THIRD BIT TO SEE IF UNARY OR SHIFT
-ENDVAC	DEC	-45	
# THE FOLLOWING ROUTINE PROCESSES ADDRESSES OF SUFFIX CLASS 10. THEY ARE BASICALLY V			
# IN THE RANGE 0-52, ERASABLE ECADR CONSTANTS FROM 100-3777, AND FCADRS ABOVE THAT.			
# IN CONTRAST TO SUFFIX 1, IN WHICH ONLY THE LOW ORDER 14 ARE AVAILABLE.			
15BITADR	INCR	LOC	# (ENTRY HERE FROM STCALL).
	INDEX	LOC	# PICK UP ADDRESS WORD.
	CA	0	
	TS	POLISH	# WE MAY NEED A SUBADDRESS LATER.

	CAF	LOW7+2K	# THESE INSTRUCTIONS ARE IN BANK 1.
	TS	FBANK	
	MASK	CYR	
ITR7	INDEX	A	
	TCF	MISCJUMP	

Page 1115

COMPLETE THE DISPATCHING OF UNARY AND SHORT SHIFT OPERATIONS.

OPJUMP3	TS	FBANK	# CALL IN BANK 0 (BIT5S 11-15 OF A ARE 0.)
	CCS	CYR	# TEST THIRD PREFIX BIT.
	INDEX	A	# THE DECREMENTED UNARY CODE IS IN BITS
	TCF	UNAJUMP	# 1-4 OF A (ZERO, EXIT, HAS BEEN DETECTED)
	CCS	MODE	# IT'S A SHORT SHIFT CODE. SEE IF PRESENT
	TCF	SHORTT	# SCALAR OR VECTOR.
	TCF	SHORTT	
	TCF	SHORTV	# CALLS THE APPROPRIATE ROUTINE.
FBANKMSK	EQUALS	BANKMASK	
LVBUF	ADRES	VBUF	

Page 1116

THE FOLLOWING IS THE JUMP TABLE FOR OP CODES WHICH MAY HAVE INDEXABLE ADDRESSES OR MAY PUSH U

INDJUMP	TCF	VLOAD	# 00 -- LOAD MPAC WITH A VECTOR.
	TCF	TAD	# 01 -- TRIPLE PRECISION ADD TO MPAC.
	TCF	SIGN	# 02 -- COMPLEMENT MPAC (V OR SC) IF X NEG.
	TCF	VXSC	# 03 -- VECTOR TIMES SCALAR.
	TCF	CGOTO	# 04 -- COMPUTED GO TO.
	TCF	TLOAD	# 05 -- LOAD MPAC WITH TRIPLE PRECISION.
	TCF	DLOAD	# 06 -- LOAD MPAC WITH A DP SCALAR.
	TCF	V/SC	# 07 -- VECTOR DIVIDED BY A SCALAR.
	TCF	SLOAD	# 10 -- LOAD MPAC IN SINGLE PRECISION.
	TCF	SSP	# 11 -- SET SINGLE PRECISION INTO X.
	TCF	PDDL	# 12 -- PUSH DOWN MPAC AND RE-LOAD IN DP.
	TCF	MXV	# 13 -- MATRIX POST-MULTIPLIED BY VECTOR.
	TCF	PDVL	# 14 -- PUSH DOWN AND VECTORLOAD.
	TCF	CCALL	# 15 -- COMPUTED CALL.
	TCF	VXM	# 16 -- MATRIX PRE-MULTIPLIED BY VECTOR.
	TCF	TSLC	# 17 -- NORMALIZE MPAC (SCALAR ONLY).
	TCF	DMPR	# 20 -- DP MULTIPLY AND ROUND.
	TCF	DDV	# 21 -- DP DIVIDE BY.

TCF	BDDV	# 22 -- DP DIVIDE INTO.
TCF	GSHIFT	# 23 -- GENERAL SHIFT INSTRUCTION
TCF	VAD	# 24 -- VECTOR ADD.
TCF	VSU	# 25 -- VECTOR SUBTRACT.
TCF	BVSU	# 26 -- VECTOR SUBTRACT FROM.
TCF	DOT	# 27 -- VECTOR DOT PRODUCT.
TCF	VXV	# 30 -- VECTOR CROSS PRODUCT.
TCF	VPROJ	# 31 -- VECTOR PROJECTION.
TCF	DSU	# 32 -- DP SUBTRACT.
TCF	BDSU	# 33 -- DP SUBTRACT FROM.
TCF	DAD	# 34 -- DP ADD.
TCF	+0	# 35 -- AVAILABLE
TCF	DMP1	# 36 -- DP MULTIPLY.
TCF	SETPD	# 37 -- SET PUSH DOWN POINTER (DIRECT ONLY)

CODES 10 AND 14 MUST NOT PUSH UP. CODE 04 MAY BE USED FOR VECTOR DECLARE BEFORE PUSH

Page 1117

THE FOLLOWING JUMP TABLE APPLIES TO INDEX, BRANCH, AND MISCELLANEOUS INSTRUCTIONS.

MISCJUMP	TCF	AXT	# 00 -- ADDRESS TO INDEX TRUE.
	TCF	AXC	# 01 -- ADDRESS TO INDEX COMPLEMENTED.
	TCF	LXA	# 02 -- LOAD INDEX FROM ERASABLE.
	TCF	LXC	# 03 -- LOAD INDEX FROM COMPLEMENT OF ERAS.
	TCF	SXA	# 04 -- STORE INDEX IN ERASABLE.
	TCF	XCHX	# 05 -- EXCHANGE INDEX WITH ERASABLE.
	TCF	INCR	# 06 -- INCREMENT INDEX REGISTER.
	TCF	TIX	# 07 -- TRANSFER ON INDEX.
	TCF	XAD	# 10 -- INDEX REGISTER ADD FROM ERASABLE.
	TCF	XSU	# 11 -- INDEX SUBTRACT FROM ERASABLE.
	TCF	BZE/GOTO	# 12 -- BRANCH ZERO AND GOTO
	TCF	BPL/BMN	# 13 -- BRANCH PLUS AND BRANCH MINUS.
	TCF	RTB/BHIZ	# 14 -- RETURN TO BASIC AND BRANCH HI ZERO.
	TCF	CALL/ITA	# 15 -- CALL AND STORE QPRET.
	TCF	SW/	# 16 -- SWITCH INSTRUCTIONS AND AVAILABLE.
	TCF	BOV(B)	# 17 -- BRANCH ON OVERFLOW TO BASIC OR INT.

Page 1118

THE FOLLOWING JUMP TABLE APPLIES TO UNARY INSTRUCTIONS

	COUNT*	\$\$/INTER	
	BANK	0	# 00 -- EXIT -- DETECTED EARLIER.
UNAJUMP	TCF	SQRT	# 01 -- SQUARE ROOT.
	TCF	SINE	# 02 -- SIN.
	TCF	COSINE	# 03 -- COS.

TCF	ARCSIN	# 04 -- ARC SIN.
TCF	ARCCOS	# 05 -- ARC COS.
TCF	DSQ	# 06 -- DP SQUARE.
TCF	ROUND	# 07 -- ROUND TO DP.
TCF	COMP	# 10 -- COMPLEMENT VECTOR OR SCALAR
TCF	VDEF	# 11 -- VECTOR DEFINE.
TCF	UNIT	# 12 -- UNIT VECTOR.
TCF	ABVALABS	# 13 -- LENGTH OF VECTOR OR MAG OF SCALAR.
TCF	VSQ	# 14 -- SQUARE OF LENGTH OF VECTOR.
TCF	STADR	# 15 -- PUSH UP ON STORE CODE.
TCF	RVQ	# 16 -- RETURN VIA QPRET.
TCF	PUSH	# 17 -- PUSH MPAC DOWN.

Page 1119

SECTION 2 LOAD AND STORE PACKAGE.

#

A SET OF EIGHT STORE CODES IS PROVIDED AS THE PRIMARY METHOD OF STORING THE MULTI-PURPOSE
 # ACCUMULATOR (MPAC). IF IN THE DANZIG SECTION LOC REFERS TO AN ALGEBRAICALLY POSITIVE WORD, I
 # STORE CODE WITH A CORRESPONDING ERASABLE ADDRESS. MOST OF THESE CODES ARE TWO ADDRESS, SPECI
 # FOLLOWING THE STORE CODE IS TO BE USED AS AN ADDRESS FROM WHICH TO RE-LOAD MPAC. FOUR OPTION
 #

#	1. STORE	STORE MPAC. THE E ADDRESS MAY BE INDEXED.
#	2. STODL	STORE MPAC AND RE-LOAD IT IN DP WITH THE NEXT ADDRESS (THE LOAD MAY BE
#	3. STOVL	STORE MPAC AND RE-LOAD A VECTOR (AS ABOVE).
#	4. STCALL	STORE AND DO A CALL (BOTH ADDRESSES MUST BE DIRECT HERE).

#

STODL AND STOVL WILL TAKE FROM THE PUSH-DOWN LIST IF NO LOAD ADDRESS IS GIVEN.

BLOCK 3

COUNT 03/INTER

STADR	CA	BANKSET	# THE STADR CODE (PUSHUP UP ON STORE
	TS	FBANK	# ADDRESS) ENTERS HERE.
	INCR	LOC	
ITR1	INDEX	LOC	# THE STORECODE WAS STORED COMPLEMENTED TO
	CS	0	# MAKE IT LOOK LIKE AN OPCODE PAIR.
	AD	NEGONE	# (YUL CAN'T REMOVE 1 BECAUSE OF EARLY CCS)
DOSTORE	TS	ADDRWD	
	MASK	LOW11	# ENTRY FROM DISPATCHER. SAVE THE ARASABLE
	XCH	ADDRWD	# ADDRESS AND JUMP ON THE STORE CODE NO.
	MASK	B12T14	
	EXTEND		
	MP	BIT5	# EACH TRANSFER VECTOR ENTRY IS TWO WORDS.

```
ITR0          INDEX  A
              TCF    STORJUMP
```

```
# Page 1120
```

```
# STORE CODE JUMP TABLE.  CALLS THE APPROPRIATE STORING ROUTINE AND EXITS TO DANZIG
```

```
# A SUPPLIED OPERATION CODE.
```

```
#
```

```
# STORE STORE,1 AND STORE,2 RETURN TO DANZIG, THUS RESETTING THE EBANK TO ITS STATE
```

```
STORJUMP      TC      STORE      # STORE.
              TCF     DANZIG      # PICK UP NEW OP CODE(S).
              TC      STORE,1
              TCF     DANZIG
              TC      STORE,2
              TCF     DANZIG

              TC      STORE      # STODL.
              TCF     DODLOAD

              TC      STORE      # STODL WITH INDEXED LOAD ADDRESS.
              TCF     DODLOAD*

              TC      STORE      # STOVL.
              TCF     DOVLOAD

              TC      STORE      # STOVL WITH INDEXED LOAD ADDRESS.
              TCF     DOVLOAD*
              TC      STORE      # STOTC.
              CAF     CALLCODE
              TS      CYR
              TCF     15BITADR    # GET A 15 BIT ADDRESS.
```

```
# Page 1121
```

```
# STORE CODE ADDRESS PROCESSOR.
```

```
STORE,1       INDEX   FIXLOC
              CS      X1
              TCF     PRESTORE

STORE,2       INDEX   FIXLOC
              CS      X2
PRESTORE      ADS     ADDRWD      # RESULTANT ADDRESS IS IN ERASABLE.

STORE        CS      ADDRWD
              AD      DEC45
              CCS     A           # DOES THE ADDRESS POINT TO THE WORK AREA?
```

July 12, 2016

773

	CA	FIXLOC	# YES.
	TCF	AHEAD5	
	CA	OCT1400	# NO. SET EBANK & MAKE UP SUBADDRESS.
	XCH	ADDRWD	
	TS	EBANK	
	MASK	LOW8	
AHEAD5	ADS	ADDRWD	

Page 1122

STORING ROUTINES. STORE DP, TP, OR VECTOR AS INDICATED BY MODE.

STARTSTO EXTEND # MPAC,+1 MUST BE STORED IN ANY EVENT.
ITRACE (5) REFERS TO "STARTSTO".

DCA	MPAC
INDEX	ADDRWD
DXCH	0

CCS	MODE
TCF	TSTORE
TC	Q

VSTORE	EXTEND
DCA	MPAC +3
INDEX	ADDRWD
DXCH	2

EXTEND	
DCA	MPAC +5
INDEX	ADDRWD
DXCH	4
TC	Q

TSTORE	CA	MPAC +2
	INDEX	ADDRWD
	TS	2
	TC	Q

Page 1123

ROUTINES TO BEGIN PROCESSING OF THE SECOND ADDRESS ASSOCIATED WITH ALL STORE-TYPE CODES EXCEPT
ITSELF.

DODLOAD	CAF	DLOADCOD	
	TS	CYR	
	TCF	DIRADRES	# GO GET A DIRECT ADDRESS.

DOVLOAD	CAF	VLOADCOD
---------	-----	----------

	TS	CYR	
	TCF	DIRADRES	
DOVLOAD*	CAF	VLOAD*	
	TCF	DODLOAD* +1	# PROLOGUE TO INDEX ROUTINE.

Page 1124

THE FOLLOWING LOAD INSTRUCTIONS ARE PROVIDED FOR LOADING THE MULTI-PURPOSE ACCUMUL

TLOAD	INDEX	ADDRWD	
	CA	2	# LOAD A TRIPLE PRECISION ARGUMENT INTO
	TS	MPAC +2	# THE FIRST THREE MPAC REGISTERS, WITH THE
	EXTEND		# CONTENTS OF THE OTHER FOUR IRRELEVANT.
	INDEX	ADDRWD	
	DCA	0	
	DXCH	MPAC	
TMODE	CAF	ONE	
	TCF	NEWMODE	# DECLEAR TRIPLE PRECISION MODE.
SLOAD	ZL		# LOAD A SINGLE PRECISION NUMBER INTO
	INDEX	ADDRWD	# MPAC, SETTING MPAC+1,2 TO ZERO. THE
	CA	0	# CONTENTS OF THE REMAINING MPAC REGISTERS
	TCF	SLOAD2	# ARE IRRELEVANT.
VLOAD	EXTEND		# LOAD A DOUBLE PRECISION VECTOR INTO
	INDEX	ADDRWD	# MPAC,+1, MPAC+3,4, AND MPAC+5,6. THE
	DCA	0	# CONTENTS OF MPAC +2 ARE IRRELEVANT.
	DXCH	MPAC	
ENDVLOAD	EXTEND		# PDVL COMES HERE TO FINISH UP FOR DP, TP.
	INDEX	ADDRWD	
	DCA	2	
	DXCH	MPAC +3	
	+4	EXTEND	# TPDVL FINISHES HERE.
		INDEX	ADDRWD
		DCA	4
		DXCH	MPAC +5
VMODE	CS	ONE	# DECLARE VECTOR MODE.
	TCF	NEWMODE	

Page 1125

THE FOLLOWING INSTRUCTIONS ARE PROVIDED FOR STORING OPERANDS IN THE PUSHDOWN LIST:

#	1.	PUSH	PUSHDOWN AND NO LOAD.
#	2.	PDDL	PUSHDOWN AND DOUBLE PRECISION LOAD.

July 12, 2016

775

```
#      3.      PDVL      PUSHDOWN AND VECTOR LOAD.

PDDL      EXTEND
INDEX     ADDRWD      # LOAD MPAC,+1, PUSHING THE FORMER
DCA       0           # CONTENTS DOWN.
DXCH      MPAC
INDEX     PUSHLOC
DXCH      0

INDEX     MODE      # ADVANCE THE PUSHDOWN POINTER APPRO-
CAF       NO.WDS    # PRIATELY.
ADS       PUSHLOC

CCS       MODE
TCF       ENDPUSH
TCF       ENDDPUSH

TS        MODE      # NOW DP.
TS        MPAC +2
DXCH      MPAC +3    # PUSH DOWN THE REST OF THE VECTOR HERE.
INDEX     PUSHLOC
DXCH      0 -4

DXCH      MPAC +5
INDEX     PUSHLOC
DXCH      0 -2

TCF       DANZIG

ENDDPUSH  TS        MPAC +2      # SET MPAC +2 TO ZERO AND EXIT ON DP.
TCF       DANZIG

ENDTPUSH  TS        MODE
          XCH       MPAC +2      # ON TRIPLE, SET MPAC +2 TO ZERO, PUSHING
          INDEX     PUSHLOC      # DOWN THE OLD CONTENTS
          TS        0 -1
          TCF       DANZIG
          +2

# Page 1126
# PDVL -- PUSHDOWN AND VECTOR LOAD

PDVL      EXTEND      # RELOAD MPAC AND PUSH DOWN ITS CONTENTS.
INDEX     ADDRWD
DCA       0
DXCH      MPAC
INDEX     PUSHLOC
```

```

                                DXCH      0

                                INDEX     MODE          # ADVANCE THE PUSHDOWN POINTER.
                                CAF        NO.WDS
                                ADS        PUSHLOC

                                CCS        MODE          # TEST PAST MODE.
                                TCF        TPDVL
                                TCF        ENDVLOAD      # JUST LOAD LAST FOUR REGISTERS ON DP.

VPDVL                          EXTEND
                                INDEX     ADDRWD        # PUSHDOWN AND RE-LOAD LAST TWO COMPONENTS
                                DCA        2
                                DXCH      MPAC +3
                                INDEX     PUSHLOC
                                DXCH      0 -4

                                EXTEND
                                INDEX     ADDRWD
                                DCA        4
                                DXCH      MPAC +5
                                INDEX     PUSHLOC
                                DXCH      0 -2

                                TCF        DANZIG

TPDVL                          EXTEND
                                INDEX     ADDRWD        # ON TP, WE MUST LOAD THE Y COMPONENT
                                DCA        2              # BEFORE STORING MPAC +2 IN CASE THIS IS A
                                DXCH      MPAC +3        # PUSHUP.

                                CA         MPAC +2
                                INDEX     PUSHLOC        # IN DP.
                                TS         0 -1
                                TCF        ENDVLOAD +4

# SSP (STORE SINGLE PRECISION) IS EXECUTED HERE.

SSP                            INCR      LOC          # PICK UP THE WORD FOLLOWING THE GIVEN
                                INDEX     LOC          # ADDRESS AND STORE IT AT X.
                                CA         0
STORE1                         INDEX     ADDRWD        # SOME INDEX AND MISCELLANEOUS OPS END
                                TS         0            # HERE.

# Page 1127
                                TCF        DANZIG

```


Page 1128

SEQUENCE CHANGING AND SUBROUTINE CALLING OPTIONS.

#

THE FOLLOWING OPERATIONS ARE AVAILABLE FOR SEQUENCING CHANGING, BRANCHING, AND CALLING SUBROUTINE

#	1.	GOTO	GO TO.
#	2.	CALL	CALL SUBROUTINE SETTING QPRET.
#	3.	CGOTO	COMPUTED GO TO.
#	4.	CCALL	COMPUTED CALL.
#	7.	BPL	BRANCH IF MPAC POSITIVE OR ZERO.
#	8.	BZE	BRANCH IF MPAC ZERO.
#	9.	BMN	BRANCH IF MPAC NEGATIVE NON-ZERO.

CCALL	INCR	LOC	# MAINTAIN LOC FOR QPRET COMPUTATION
-------	------	-----	--------------------------------------

INDEX	LOC
-------	-----

Was CAF --- RSB 2009.

CA	0	# GET BASE ADDRESS OF CADR LIST.
----	---	----------------------------------

INDEX	ADDRWD
-------	--------

AD	0	# ADD INCREMENT.
----	---	------------------

TS	FBANK	# SELECT DESIRED CADR.
----	-------	------------------------

MASK	LOW10
------	-------

INDEX	A
-------	---

CAF	10000
-----	-------

TS	POLISH
----	--------

CALL	CA	BANKSET	# FOR ANY OF THE CALL OPTIONS, MAKE UP THE
	MASK	BANKMASK	# ADDRESS OF THE NEXT OP-CODE PAIR/STORE
	AD	BANKMASK	# CODE AND LEAVE IT IN QPRET. NOTE THAT
	AD	LOC	# BANKMASK = -(2000 - 1).
	INDEX	FIXLOC	
	TS	QPRET	

GOTO	CA	POLISH	# BASIC BRANCHING SEQUENCE.	
	+1	MASK	HIGH4	
		EXTEND		
		BZF	GOTOERS	# SEE IF ADDRESS POINTS TO FIXED OR ERAS.
	+4	CA	BANKSET	# SET EBANK PART OF BBANK. NEXT, SET UP
		TS	BBANK	# FBANK. THE COMBINATION IS PICKED UP &
		CA	POLISH	# PUT INTO BANKSET AT INTPRET +2.
		TS	FBANK	
		MASK	LOW10	
		AD	2K	
		TS	LOC	
		TCF	INTPRET +3	

EBANK=	1400	# SO YUL DOESN'T CUSS THE "CA 1400" BELOW.
--------	------	--

GOTOERS	CA	POLISH	# THE GIVEN ADDRESS IS IN ERASABLE -- SEE
	AD	-ENDVAC	# IF RELATIVE TO THE WORK AREA.
	CCS	A	
	CA	POLISH	# GENERAL ERASABLE.
	TCF	GOTOG	

Page 1129

	CA	FIXLOC	# WORK AREA.
	AD	POLISH	
	INDEX	A	# USE THE GIVEN ADDRESS AS THE ADDRESS OF
	CA	0	# THE BRANCH ADDRESS.
	TS	POLISH	
	TCF	GOTO +1	# ALLOWS ARBITRARY INDIRECTNESS LEVELS.

GOTOG	TS	EBANK	
	MASK	LOW8	
	INDEX	A	# USE THE GIVEN ADDRESS AS THE ADDRESS OF
	CA	1400	# THE BRANCH ADDRESS.
	TS	POLISH	
	TCF	GOTO +1	

CGOTO	INDEX	LOC	# COMPUTED GO TO. PICK UP ADDRESS OF CADR
	CA	1	# LIST
	INDEX	ADDRWD	# ADD MODIFIER.
	AD	0	
	TS	FBANK	# SELECT GOTO ADDRESS
	MASK	LOW10	
	INDEX	A	
	CA	10000	
	TS	POLISH	
	TCF	GOTO +1	# WITH ADDRESS IN A.

SWBRANCH	CA	BANKSET	# SWITCH INSTRUCTIONS WHICH ELECT TO
	TS	FBANK	# BRANCH COME HERE TO DO SO.
	INDEX	LOC	
	CA	1	
	TS	POLISH	
	TCF	GOTO +1	

Page 1130

TRIPLE PRECISION BRANCHING ROUTINE. IF CALLING TC IS AT L, RETURN IS AS FOLLOWS:

#	L+1	IF MPAC IS GREATER THAN ZERO.
#	L+2	IF MPAC IS EQUAL TO +0 OR -0.
#	L+3	IF MPAC IS LESS THAN ZERO.

BRANCH	CCS	MPAC
--------	-----	------

```

TC      Q
TCF     +2      # ON ZERO.
TCF     NEG

CCS     MPAC +1
TC      Q
TCF     +2
TCF     NEG

CCS     MPAC +2
TC      Q
TCF     +2
TCF     NEG

Q+1      INDEX  Q
TC      1

NEG      INDEX  Q      # IF FIRST NON-ZERO REGISTER WAS NEGATIVE.
TC      2

Q+2      =      NEG

# ITRACE (3) REFERS TO "EXIT".

EXIT      CA      BANKSET      # RESTORE USER'S BANK SETTING, AND LEAVE
          TS      BBANK      # INTERPRETIVE MODE.
          INDEX   LOC
          TC      1

# Page 1131
# SECTION 3 -- ADD/SUBTRACT PACKAGE.
#
# THE FOLLOWING OPERATIONS ARE PROVIDED FOR ADDING TO AND SUBTRACTING FROM THE MULTI-PURPOSE AC
# MPAC:
#   1.      DAD      DOUBLE PRECISION ADD.
#   2.      DSU      DOUBLE PRECISION SUBTRACT.
#   3.      BDSU     DOUBLE PRECISION SUBTRACT FROM.
#   4.      TAD      TRIPLE PRECISION ADD.
#   5.      VAD      VECTOR ADD.
#   6.      VSU      VECTOR SUBTRACT.
#   7.      BVSU     VECTOR SUBTRACT FROM.
# THE INTERPRETIVE OVERFLOW INDICATOR OVFind IS SET NON-ZERO IF OVERFLOW OCCURS IN ANY OF THE A

VSU      CAF      BIT15      # CHANGES 0 TO DCS.
          TCF     +2

```

VAD	CAF	PRI030	# CHANGES 0 TO DCA.
	ADS	ADDRWD	
	EXTEND		
	INDEX	ADDRWD	
	READ	HISCALAR	# DCA 2 OR DCS 2
	DAS	MPAC +3	
	EXTEND		# CHECK OVERFLOW.
	BZF	+2	
	TC	OVERFLWY	
	EXTEND		
	INDEX	ADDRWD	
	READ	CHAN5	# DCA 4 OR DCS 4
	DAS	MPAC +5	
	EXTEND		
	BZF	+2	
	TC	OVERFLWZ	
	EXTEND		
	INDEX	ADDRWD	
	READ	LCHAN	# DCA 0 OR DCS 0
	TCF	ENDVXV	
DAD	EXTEND		
	INDEX	ADDRWD	
	DCA	0	
ENDVXV	DAS	MPAC	# VXV FINISHES HERE.
	EXTEND		
	BZF	DANZIG	
# Page 1132			
SETOVF	TC	OVERFLOW	
	TCF	DANZIG	
# Page 1133			
DSU	EXTEND		
	INDEX	ADDRWD	
	DCS	0	
	TCF	ENDVXV	
OVERFLWZ	TS	L	# ENTRY FOR THIRD COMPONENT.
	CAF	FIVE	
	TCF	+3	
OVERFLWY	TS	L	# ENTRY FOR SECOND COMPONENT.
	CAF	THREE	

```

XCH      L

OVERFLOW  INDEX  A          # ENTRY FOR 1ST COMP OR DP (L=0).
          CS      LIMITS    # PICK UP POSMAX OR NEGMAX.
          TS      BUF
          EXTEND
          AUG      A
          INDEX    L
          ADS      MPAC +1
          TS      7
          CAF      ZERO
          AD        BUF
          INDEX    L
          ADS      MPAC
          TS      7
          TC        Q          # NO OVERFLOW EXIT.
          TCF      SETOVF2     # SET OVFLND AND EXIT.

BVSU      EXTEND
          INDEX    ADDRWD
          DCA      2
          DXCH     MPAC +3
          EXTEND
          DCOM
          DAS      MPAC +3
          EXTEND
          BZF      +2
          TC        OVERFLWY

          EXTEND
          INDEX    ADDRWD
          DCA      4
          DXCH     MPAC +5
          EXTEND
          DCOM
          DAS      MPAC +5
          EXTEND
          BZF      +2
          TC        OVERFLWZ

# Page 1134
BDSU      EXTEND
          INDEX    ADDRWD
          DCA      0
          DXCH     MPAC
          EXTEND

```

DCOM
TCF ENDVXV

Page 1135
TRIPLE PRECISION ADD ROUTINE.

TAD EXTEND
 INDEX ADDRWD
 DCA 1 # ADD MINOR PARTS FIRST.
 DAS MPAC +1
 INDEX ADDRWD
 AD 0
 AD MPAC
 TS MPAC
 TCF DANZIG

 TCF SETOVF # SET OVFind IF SUCH OCCURS.

Page 1136
ARITHMETIC SUBROUTINES REQUIRED IN FIXED-FIXED.

1. DMPSUB DOUBLE PRECISION MULTIPLY, MULTIPLY THE CONTENTS OF MPAC,+1
IS IN ADDRWD AND LEAVE A TRIPLE-PRECISION RESULT IN MPAC.
2. ROUND SUB ROUND THE TRIPLE PRECISION CONTENTS OF MPAC TO DOUBLE PRECISION.
3. DOT SUB TAKE THE DOT PRODUCT OF THE VECTOR IN MPAC AND THE VECTOR WHO
AND LEAVE THE TRIPLE PRECISION RESULT IN MPAC.
4. POLY USING THE CONTENTS OF MPAC AS A DP ARGUMENT, EVALUATE THE POLY
COEFFICIENTS IMMEDIATELY FOLLOW THE TC POLY INSTRUCTION (SEE

DMP INDEX Q # BASIC SUBROUTINE FOR USE BY PINBALL, ETC
 # Was CAF --- RSB 2009.
 CA 0
 INCR Q
 -1 TS ADDRWD # (PROLOGUE FOR SETTING ADDRWD.)

DMP SUB INDEX ADDRWD # GET MINOR PART OF OPERAND AT C(ADDRWD).
 CA 1
 TS MPAC +2 # THIS WORKS FOR SQUARING MPAC AS WELL.
 CAF ZERO # SET MPAC +1 TO ZERO SO WE CAN ACCUMULATE
 XCH MPAC +1 # THE PARTIAL PRODUCTS WITH DAS
 TS MPTMP # INSTRUCTIONS.
 EXTEND
 MP MPAC +2 # MINOR OF MPAC X MINOR OF C(ADDRWD).

 XCH MPAC +2 # DISCARD MINOR PART OF ABOVE RESULT AND
 EXTEND # FORM MAJOR OF MPAC X MINOR OF C(ADDRWD).
 MP MPAC

```

                                DAS      MPAC +1      # GUARANTEED NO OVERFLOW.

                                INDEX     ADDRWD      # GET MAJOR PART OF ARGUMENT AT C(ADDRWD).
                                CA         0
                                XCH       MPTEMP      # SAVE AND BRING OUT MINOR OF MPAC.
DMPSUB2      EXTEND
                                MP        MPTEMP      # MAJOR OF C(ADDRWD) X MINOR OF MPAC.
                                DAS       MPAC +1      # ACCUMULATE, SETTING A TO NET OVERFLOW.

                                XCH       MPAC        # SETTING MPAC TO 0 OR +-1.
                                EXTEND
                                MP        MPTEMP      # MAJOR OF MPAC X MAJOR OF C(ADDRWD).
                                DAS       MPAC        # GUARANTEED NO OVERFLOW.
                                TC        Q          # 49 MCT = .573 MS. INCLUDING RETURN.

```

Page 1137

ROUND MPAC TO DOUBLE PRECISION, SETTING OVFINF ON THE RARE EVENT OF OVERFLOW.

```

ROUND SUB      CAF      ZERO      # SET MPAC +2 = 0 FOR SCALARS AND CHANGE
+1            TS       MODE      # MODE TO DP.

VROUND        XCH      MPAC +2    # BUT WE NEEDN'T TAKE THE TIME FOR VECTORS.
              DOUBLE
              TS       L
              TC       Q

              AD       MPAC +1    # ADD ROUNDING BIT IF MPAC +2 WAS GREATER
              TS       MPAC +1    # THAN .5 IN MAGNITUDE.
              TC       Q

              AD       MPAC      # PROPAGATE INTERFLOW.
              TS       MPAC
              TC       Q

SETOVF2       TS       OVFINF     # (RARE).
              TC       Q

```

Page 1138

THE DOT PRODUCT SUBROUTINE USUALLY FORMS THE DOT PRODUCT OF THE VECTOR IN MPAC WITH A STANDARD
REGISTER VECTOR WHOSE ADDRESS IS IN ADDRWD. IN THIS CASE C(DOTINC) ARE SET TO 2. VXM, HOWEVER
6 SO THAT DOTSUB DOTS MPAC WITH A COLUMN VECTOR OF THE MATRIX IN QUESTION IN THIS CASE.

```

PREDOT        CAF      TWO        # PROLOGUE TO SET DOTINC TO 2.
              TS       DOTINC

DOTSUB        EXTEND

```

```

QXCH  DOTRET      # SAVE RETURN
TC     DMPSUB      # DOT X COMPONENTS.
DXCH   MPAC +3     # POSITION Y COMPONENT OF MPAC FOR
DXCH   MPAC        # MULTIPLICATION WHILE SAVING RESULT IN
DXCH   BUF         # THREE WORD BUFFER, BUF.
CA     MPAC +2
TS     BUF +2

CA     DOTINC      # ADVANCE ADDRWD TO Y COMPONENT OF
ADS    ADDRWD      # OTHER ARGUMENT.
TC     DMPSUB
DXCH   MPAC +1     # ACCUMULATE PARTIAL PRODUCTS.
DAS    BUF +1
AD     MPAC
AD     BUF
TS     BUF
TCF    +2
TS     OVFIN      # IF OVERFLOW OCCURS.

DXCH   MPAC +5     # MULTIPLY Z COMPONENTS.
DXCH   MPAC
CA     DOTINC
ADS    ADDRWD
TC     DMPSUB
ENDDOT DXCH   BUF +1     # LEAVE FINAL ACCUMULATION IN MPAC.
DAS    MPAC +1
AD     MPAC
AD     BUF
TS     MPAC
TC     DOTRET

TC     OVERFLOW    # ON OVERFLOW HERE.
TC     DOTRET

# Page 1139
# DOUBLE PRECISION POLYNOMIAL EVALUATOR
#
#           N           N-1
#   THIS ROUTINE EVALUATES A X  + A  X  + ... + A  X  + A  LEAVING THE DP RESU
#                       N      N-1              1      0
#
# THE ROUTINE HAS TWO ENTRIES
#
#   1      ENTRY THRU POWRSERS.  THE COEFFICIENTS MAY BE EITHER IN FIXED OR ERA
#   TC POWRSERS, AND THE RETURN IS TO LOC(TC POWRSERS)+1.  THE ENTERING I
#   A      SP      LOC-3      ADDRESS FOR REFERENCING COEF
#   L      SP      N-1      N IS THE DEGREE OF THE POWER

```


July 12, 2016

785

```
#           MPAC   DP      X           ARGUMENT
#           LOC-2N DP      A(0)
#           ...
#           LOC    DP      A(N)
#
# 2. ENTRY THRU POLY. THE CALL TO POLY AND THE ENTERING DATA MUST BE AS FOLLOWS
#           MPAC           DP      X           ARGUMENT
#           LOC           TC      POLY
#           LOC+1         DP      A(0)
#           ...
#           LOC+2N+2      DP      A(N)           RETURN IS TO LOC+2N+4

POWRSERS    EXTEND
QXCH  POLYRET      # RETURN ADDRESS
TS    POLISH       # POWER SERIES ADDRESS
LXCH  POLYCNT      # N-1 TO COUNTER
TCF   POLYCOM      # SKIP SET UP BY POLY

POLY        INDEX   Q
# Was CAF --- RSB 2009.
CA       0
TS       POLYCNT    # N-1 TO COUNTER
DOUBLE
AD       Q
TS       POLISH     # L(A(N))-3 TO POLISH
AD       FIVE
TS       POLYRET    # STORE RETURN ADDRESS

POLYCOM     CAF     LVBUF      # INCOMING X WILL BE MOVED TO VBUF, SO
TS          ADDRWD      # SET ADDRWD SO DMPSUB WILL MPY BY VBUF.

EXTEND
INDEX  POLISH
DCA    3

# Page 1140
DXCH   MPAC        # LOAD A(N) INTO MPAC
DXCH   VBUF        # SAVING X IN VBUF
TCF    POLY2

POLYLOOP   TS       POLYCNT    # SAVE DECREMENTD LOOP COUNTER
CS        TWO
ADS       POLISH     # REGRESS COEFFICIENT POINTER

POLY2      TC       DMPSUB     # MULTIPLY BY X
EXTEND
```

	INDEX	POLISH	
	DCA	1	# ADD IN NEXT COEFFICIENT
	DAS	MPAC	# USER'S RESPONSIBILITY TO ASSURE NO OVFLOW
	CCS	POLYCNT	
	TCF	POLYLOOP	
	TC	POLYRET	# RETURN CALLER
# Page 1141			
# MISCELLANEOUS MULTI-PRECISION ROUTINES REQUIRED IN FIXED-FIXED BUT NOT USED BY THE			
DPAGREE	CAF	ZERO	# DOUBLE PRECISION ENTRY --
	TS	MPAC +2	# ZERO LOW-ORDER WORD
TPAGREE	LXCH	Q	# FORCE SIGN AGREEMENT AMONG THE TRIPLE
	TC	BRANCH	# PRECISION CONTENTS OF MPAC. RETURNING
	TCF	ARG+	# WITH SIGNUM OF THE INPUT IN A.
	TCF	ARGZERO	
	CS	POSMAX	# IF NEGATIVE.
	TCF	+2	
ARG+	CAF	POSMAX	
	TS	Q	
	EXTEND		
	AUG	A	# FORMS +-1.0.
	AD	MPAC +2	
	TS	MPAC +2	
	CAF	ZERO	
	AD	Q	
	AD	MPAC +1	
	TS	MPAC +1	
	CAF	ZERO	
	AD	Q	# Q STILL HAS POSMAX OR NEGMAX IN IT.
	AD	MPAC	
ARGZERO2	TS	MPAC	# ALWAYS SKIPPING UNLESS ARGZERO.
	TS	MPAC +1	
	TC	L	# RETURN VIA L.
ARGZERO	TS	MPAC +2	# SET ALL THREE MPAC REGISTERS TO ZERO.
	TCF	ARGZERO2	
# SHORTMP MULTIPLIES THE TP CONTENTS OF MPAC BY THE SINGLE PRECISION NUMBER ARRIVING			
SHORTMP	TS	MPTMP	
	EXTEND		

July 12, 2016

787

	MP	MPAC +2	
	TS	MPAC +2	
SHORTMP2	CAF	ZERO	# SO SUBSEQUENT DAS WILL WORK.
	XCH	MPAC +1	
	TCF	DMPSUB2	

Page 1142

DMPNSUB MULTIPLIES THE DP FRACTION ARRIVING IN MPAC BY THE SP
INTEGER ARRIVING IN A. THE DP PRODUCT DEPARTS BOTH IN MPAC AND IN
A AND L. NOTE THAT DMPNSUB NORMALLY INCREASES THE MAGNITUDE OF THE
CONTENTS OF MPAC. THE CUSTOMER MUST INSURE THAT $B(A) \times B(MPAC, MPAC+1)$
AND $B(A) \times B(MPAC)$ ARE LESS THAN 1 IN MAGNITUDE, WHERE B, AS IS OBVIOUS,
INDICATES THE ARRIVING CONTENTS.

DMPNSUB	TS	DMPNTEMP	
	EXTEND		
	MP	MPAC +1	
	DXCH	MPAC	# LOW PRODUCT TO MPAC, HIGH FACTOR TO A
	EXTEND		
	MP	DMPNTEMP	
	CA	L	
	ADS	MPAC	# COMPLETING THE PRODUCT IN MPAC
	EXTEND		
	DCA	MPAC	# BRINGING THE PRODUCT INTO A AND L
	TC	Q	

Page 1143

MISCELLANEOUS VECTOR OPERATIONS. INCLUDED HERE ARE THE FOLLOWING.

#	1.	DOT	DP VECTOR DOT PRODUCT.
#	2.	VXV	DP VECTOR CROSS PRODUCT.
#	3.	VXSC	DP VECTOR TIMES SCALAR.
#	4.	V/SC	DP VECTOR DIVIDED BY SCALAR.
#	5.	VPROJ	DP VECTOR PROJECTION. ($(MPAC.X)MPAC$).
#	6.	VXM	DP VECTOR POST-MULTIPLIED BY MATRIX.
#	7.	MXV	DP VECTOR PRE-MULTIPLIED BY MATRIX.

DOT	TC	PREDOT	# DO THE DOT PRODUCT AND EXIT, CHANGING
DMODE	CAF	ZERO	# THE MODE TO DP SCALAR.
	TCF	NEWMODE	

MXV	CAF	TWO	# SET UP MATINC AND DOTINC FOR ROW
	TS	MATINC	# VECTORS.
	TCF	VXM/MXV	# GO TO COMMON PORTION.

VXM	CS	TEN	# SET MATINC AND DOTINC TO REFER TO MATRIX
	TS	MATINC	# AS THREE COLUMN VECTORS.

CAF SIX

Page 1144

COMMON PORTION OF MXV AND VXM.

VXM/MXV TS DOTINC

ITRACE (2) REFERS TO "VXM/MXV".

TC MPACVBUF # SAVE VECTOR IN MPAC FOR FURTHER USE.

TC DOTSUB # GO DOT TO GET X COMPONENT OF ANSWER.

EXTEND

DCA VBUF # MOVE MPAC VECTOR BACK INTO MPAC, SAVING
DXCH MPAC # NEW X COMPONENT IN BUF2.

DXCH BUF2

EXTEND

DCA VBUF +2

DXCH MPAC +3

EXTEND

DCA VBUF +4

DXCH MPAC +5

CA MATINC # INITIALIZE ADDRWD FOR NEXT DOT PRODUCT.

ADS ADDRWD # FORMS HAS ADDRESS OF NEXT COLUMN(ROW).

TC DOTSUB

DXCH VBUF # MORE GIVEN VECTOR BACK TO MPAC, SAVING Y

DXCH MPAC # COMPONENT OF ANSWER IN VBUF +2.

DXCH VBUF +2

DXCH MPAC +3

DXCH VBUF +4

DXCH MPAC +5

CA MATINC # FORM ADDRESS OF LAST COLUMN OR ROW.

ADS ADDRWD

TC DOTSUB

DXCH BUF2 # ANSWER NOW COMPLETE. PUT COMPONENTS INTO

DXCH MPAC # PROPER MPAC REGISTERS.

DXCH MPAC +5

DXCH VBUF +2

DXCH MPAC +3

TCF DANZIG # EXIT.

Page 1145

VXSC -- VECTOR TIMES SCALAR.

July 12, 2016

789

```
VXSC      CCS      MODE      # TEST PRESENT MODE.
          TCF      DVXSC     # SEPARATE ROUTINE WHEN SCALAR IS IN MPAC.
          TCF      DVXSC

VVXSC     TC       DMPSUB     # COMPUTE X COMPONENT
          TC       VROUND     # AND ROUND IT.
          DXCH     MPAC +3     # PUT Y COMPONENT INTO MPAC SAVING MPAC IN
          DXCH     MPAC        # MPAC +3.
          DXCH     MPAC +3

          TC       DMPSUB     # DO SAME FOR Y AND Z COMPONENTS.
          TC       VROUND
          DXCH     MPAC +5
          DXCH     MPAC
          DXCH     MPAC +5

          TC       DMPSUB
          TC       VROUND
VROTATEX  DXCH     MPAC        # EXIT USED TO RESTORE MPAC AFTER THIS
          DXCH     MPAC +5     # TYPE OF ROTATION. CALLED BY VECTOR SHIFT
          DXCH     MPAC +3     # RIGHT, V/SC, ETC.
          DXCH     MPAC
          TCF      DANZIG

# Page 1146
# DP VECTOR PROJECTION ROUTINE.

VPROJ     TC       PREDOT     # (MPAC.X)MPAC IS COMPUTED AND LEFT IN
          CS       FOUR       # MPAC. DO DOT AND FALL INTO DVXSC.
          ADS      ADDRWD

# VXSC WHEN SCALAR ARRIVES IN MPAC AND VECTOR IS AT X.

DVXSC     EXTEND                     # SAVE SCALAR IN MPAC +3 AND GET X
          DCA      MPAC             # COMPONENT OF ANSWER.
          DXCH     MPAC +3
          TC       DMPSUB
          TC       VROUND

          CAF      TWO              # ADVANCE ADDRWD TO Y COMPONENT OF X.
          ADS      ADDRWD
          EXTEND
          DCA      MPAC +3          # PUT SCALAR BACK INTO MPAC AND SAVE
          DXCH     MPAC             # X RESULT IN MPAC +5.
          DXCH     MPAC +5
          TC       DMPSUB
```

```

TC      VROUND

CAF      TWO
ADS      ADDRWD      # TO Z COMPONENT.
DXCH     MPAC +3      # BRING SCALAR BACK, PUTTING Y RESULT IN
DXCH     MPAC          # THE PROPER PLACE.
DXCH     MPAC +3
TC       DMPSUB
TC       VROUND

DXCH     MPAC          # PUT Z COMPONENT IN PROPER PLACE, ALSO
DXCH     MPAC +5      # POSITIONING X.
DXCH     MPAC

TCF      VMODE      # MODE HAS CHANGED TO VECTOR.

```

Page 1147

VECTOR CROSS PRODUCT ROUTINE CALCULATES $(X_M - X_M, X_M - X_M, X_M - X_M)$ WHERE M IS
$\begin{matrix} & & 3 & 2 & 2 & 3 & 1 & 3 & 3 & 1 & 2 & 1 & 1 & 2 \end{matrix}$
MPAC AND X THE VECTOR AT THE GIVEN ADDRESS.

```

VXV      EXTEND
DCA      MPAC +5      # FORM UP M3X1, LEAVING M1 IN VBUF.
DXCH     MPAC
DXCH     VBUF
TC       DMPSUB      # BY X1.

EXTEND
DCS      MPAC +3      # CALCULATE -X1M2, SAVING X1M3 IN VBUF +2.
DXCH     MPAC
DXCH     VBUF +2
TC       DMPSUB

CAF      TWO          # ADVANCE ADDRWD TO X2.
ADS      ADDRWD
EXTEND
DCS      MPAC +5      # PREPARE TO GET -X2M3, SAVING -X1M2 IN
DXCH     MPAC          # MPAC +5.
DXCH     MPAC +5
TC       DMPSUB

EXTEND
DCA      VBUF          # GET X2M1, SAVING -X2M3 IN VBUF +4.
DXCH     MPAC
DXCH     VBUF +4
TC       DMPSUB

```

```

CAF      TWO      # ADVANCE ADDRWD TO X3.
ADS      ADDRWD
EXTEND
DCS      VBUF      # GET -X3M1, ADDING X2M1 TO MPAC +5 TO
DXCH     MPAC      # COMPLETE THE Z COMPONENT OF THE ANSWER.
DAS      MPAC +5

EXTEND
BZF      +2
TC       OVERFLWZ

TC       DMPSUB
DXCH     VBUF +2   # MOVE X1M3 TO MPAC +3 SETTING UP FOR X3M2
DXCH     MPAC +3   # AND ADD -X3M1 TO MPAC +3 TO COMPLETE THE
DXCH     MPAC      # Y COMPONENT OF THE RESULT.
DAS      MPAC +3

EXTEND
BZF      +2

# Page 1148
TC OVERFLWY

TC       DMPSUB
DXCH     VBUF +4   # GO ADD -X2M3 TO X3M2 TO COMPLETE THE X
TCF      ENDVXV    # COMPONENT (TAIL END OF DAD).

# THE MPACVBUF SUBROUTINE SAVES THE VECTOR IN MPAC IN VBUF WITHOUT CLOBBERING MPAC.

MPACVBUF EXTEND      # CALLED BY MXV, VXM, AND UNIT.
DCA      MPAC
DXCH     VBUF
EXTEND
DCA      MPAC +3
DXCH     VBUF +2
EXTEND
DCA      MPAC +5
DXCH     VBUF +4
TC       Q         # RETURN TO CALLER.

# DOUBLE PRECISION SIGN AGREE ROUTINE.  ARRIVE WITH INPUT IN A+L.  OUTPUT IS IN A + L.

ALSIGNAG CCS      A      # TEST UPPER PART.
TCF      UPPOS     # IT IS POSITIVE
TC       Q         # ZERO
TCF      UPNEG     # NEGATIVE

```

	TC	Q	# ZERO
UPPOS	XCH	L	# SAVE DECREMENTED UPPER PART.
	AD	HALF	
	AD	HALF	
	TS	A	# SKIPS ON OVERFLOW
	TCF	+2	
	INCR	L	# RESTORE UPPER TO ORIGINAL VALUE
	XCH	L	# SWAP A + L BANCK.
	TC	Q	
UPNEG	XCH	L	# SAVE COMPLEMENTED + DECREMENTED UPPER PT
	AD	NEGMAX	
	AD	NEGONE	
	TS	A	
	TCF	+2	# DON'T INCREMENT IF NO OVERFLOW.
	INCR	L	
	XCH	L	
	COM		# MAKE NEGATIVE AGAIN.
	TC	Q	

Page 1149

INTERPRETIVE INSTRUCTIONS WHOSE EXECUTION CONSISTS OF PRINCIPALLY CALLING SUBROUTIN

DMP1	TC	DMPSUB	# DMP INSTRUCTIONS
	TCF	DANZIG	
DMPR	TC	DMPSUB	
	TC	ROUND SUB +1	# (C(A) = +0).
	TCF	DANZIG	
DDV	EXTEND		
	INDEX	ADDRWD	# MOVE DIVIDENT INTO BUF.
	DCA	0	
	TCF	BDDV +4	
BDDV	EXTEND		# MOVE DIVISOR INTO MPAC SAVING MPAC, THE
	INDEX	ADDRWD	# DIVIDEND, IN BUF.
	DCA	0	
	DXCH	MPAC	
+4	DXCH	BUF	
	CAF	ZERO	# DIVIDE ROUTINES IN BANK 0.
	TS	FBANK	
	TCF	DDV/BDDV	
SETPD	CA	ADDRWD	# MUST SET TO WORK AREA, OR EBANK TROUBLE.

	TS	PUSHLOC	
	TCF	NOIBNKS	# NO FBANK SWITCH REQUIRED.
TSLC	CAF	ZERO	# SHIFTING ROUTINES LOCATED IN BANK 00.
	TS	FBANK	
	TCF	TSLC2	
GSHIFT	CAF	LOW7	# USED AS MASK AT GENSHIFT. THIS PROCESSES
	TS	FBANK	# ANY SHIFT INSTRUCTION (EXCEPT TSLC) WITH
	TCF	GENSHIFT	# AN ADDRESS (ROUTINES IN BANK 0).

Page 1150

THE FOLLOWING IS THE PROLOGUE TO V/SC. IF THE PRESENT MODE IS VECTOR, IT SAVES THE SCALAR AT
 # AND CALLS THE V/SC ROUTINE IN BANK 0. IF THE PRESENT MODE IS SCALAR, IT MOVES THE VECTOR AT
 # THE SCALAR IN MPAC IN BUF BEFORE CALLING THE V/SC ROUTINE IN BANK 0.

V/SC	CCS	MODE	
	TCF	DV/SC	# MOVE VECTOR INTO MPAC.
	TCF	DV/SC	
VV/SC	EXTEND		
	INDEX	ADDRWD	
	DCA	0	
V/SC1	DXCH	BUF	# IN BOTH CASES, VECTOR IS NOW IN MPAC AND
	CAF	ZERO	# SCALAR IN BUF.
	TS	FBANK	
	TCF	V/SC2	
DV/SC	EXTEND		
	INDEX	ADDRWD	
	DCA	2	
	DXCH	MPAC +3	
	EXTEND		
	INDEX	ADDRWD	
	DCA	4	
	DXCH	MPAC +5	
	CS	ONE	# CHANGE MODE TO VECTOR.
	TS	MODE	
	EXTEND		
	INDEX	ADDRWD	
	DCA	0	
	DXCH	MPAC	
	TCF	V/SC1	

Page 1151

SIGN AND COMPLEMENT INSTRUCTIONS.

SIGN	INDEX	ADDRWD	# CALL COMP INSTRUCTION IF WORD AT X IS
	CCS	0	# NEGATIVE NON-ZERO.
	TCF	DANZIG	
	TCF	+2	
	TCF	COMP	# DO THE COMPLEMENT.
CCSL	INDEX	ADDRWD	
	CCS	1	
	TCF	DANZIG	
	TCF	DANZIG	
	TCF	COMP	
	TCF	DANZIG	
COMP	EXTEND		# COMPLEMENT DP MPAC IN EVERY CASE.
	DCS	MPAC	
	DXCH	MPAC	
	CCS	MODE	# EITHER COMPLEMENT MPAC +3 OR THE REST OF
	TCF	DCOMP	# THE VECTOR ACCUMULATOR.
	TCF	DCOMP	
	EXTEND		# VECTOR COMPLEMENT.
	DCS	MPAC +3	
	DXCH	MPAC +3	
	EXTEND		
	DCS	MPAC +5	
	DXCH	MPAC +5	
	TCF	DANZIG	
DCOMP	CS	MPAC +2	
	TS	MPAC +2	
	TCF	DANZIG	

Page 1152

THE FOLLOWING SHORT SHIFT CODES REQUIRE NO ADDRESS WORD:

#	1.	SR1 TO SR4	SCALAR SHIFT RIGHT.
#	2.	SR1R TO SR4R	SCALAR SHIFT RIGHT AND ROUND.
#	3.	SL1 TO SL4	SCALAR SHIFT LEFT.
#	4.	SL1R TO SL4R	SCALAR SHIFT LEFT AND ROUND.
#	5.	VSR1 TO VSR8	VECTOR SHIFT RIGHT (ALWAYS ROUNDS).
#	6.	VSL1 TO VSL8	VECTOR SHIFT LEFT (NEVER ROUNDS).

THE FOLLOWING CODES REQUIRE AN ADDRESS WHICH MAY BE INDEXED:*

#	1.	SR	SCALAR SHIFT RIGHT.
#	2.	SRR	SCALAR SHIFT RIGHT AND ROUND.

```

#      3.      SL      SCALAR SHIFT LEFT.
#      4.      SLR     SCALAR SHIFT LEFT AND ROUND.
#      5.      VSR     VECTOR SHIFT RIGHT.
#      6.      VSL     VECTOR SHIFT LEFT.
# * IF THE ADDRESS IS INDEXED, AND THE INDEX MODIFICATION RESULTS IN A NEGATIVE SHIFT COUNT, A
# ABSOLUTE VALUE OF THE COUNT IS DONE IN THE OPPOSITE DIRECTION.

```

```

      BANK      00

```

```

      COUNT     00/INTER

```

```

SHORTT      CAF      SIX      # SCALAR SHORT SHIFTS COME HERE.  THE SHIFT
            MASK     CYR      # COUNT-1 IS NOW IN BITS 2-3 OF CYR.  THE
            TS       SR       # ROUNDING BIT IS IN BIT1 AT THIS POINT.

```

```

            CCS     CYR      # SEE IF RIGHT OR LEFT SHIFT DESIRED.
            TCF     TSSL     # SHIFT LEFT.

```

```

SRDDV      DEC      20      # MPTMP SETTING FOR SR BEFORE DDV.

```

```

TSSR      INDEX    SR      # GET SHIFTING BIT.
            CAF     BIT14
            TS      MPTMP

```

```

            CCS     CYR      # SEE IF A ROUND IS DESIRED.
RIGHTR     TC      MPACSRND  # YES -- SHIFT RIGHT AND ROUND.
            TCF     NEWMODE  # SET MODE TO DP (C(A) = 0).
MPACSHR    CA      MPTMP     # DO A TRIPLE PRECISION SHIFT RIGHT.

```

```

            EXTEND
            MP      MPAC +2
            TS      MPAC +2  # (EXIT FROM SQRT AND ABVAL).
            CA      MPTMP
            EXTEND

```

```

# Page 1153

```

```

            MP      MPAC      # SHIFT MAJOR PART INTO A,L AND PLACE IN
            DXCH    MPAC      # MPAC,+1.
            CA      MPTMP
            EXTEND
            MP      L        # ORIGINAL C(MPAC +1).
            DAS     MPAC +1  # GUARANTEED NO OVERFLOW.
            TCF     DANZIG

```

```

# MPAC SHIFT RIGHT AND ROUND SUBROUTINES

```

```

MPACSRND    CA      MPAC +2  # WE HAVE TO DO ALL THREE MULTIPLIES SINCE
            EXTEND          # MPAC +1 AND MPAC +2 MIGHT HAVE SIGN

```

```

MP      MPTMP      # DISAGREEMENT WITH A SHIFT RIGHT OF L.
XCH     MPAC +1
EXTEND
MP      MPTMP
XCH     MPAC +1      # TRIAL MINOR PART.
AD      L

VSHR2   DOUBLE      # (FINISH VECTOR COMPONENT SHIFT RIGHT
TS      MPAC +2      # AND ROUND.)
TCF     +2
ADS     MPAC +1      # GUARANTEED NO OVERFLOW.

CAF     ZERO
TS      MPAC +2
XCH     MPAC          # SETTING TO ZERO SO FOLLOWING DAS WORKS.
EXTEND
MP      MPTMP
DAS     MPAC          # AGAIN NO OVERFLOW.
TC      Q

VSHRRND CA      MPTMP      # ENTRY TO SHIFT RIGHT AND ROUND MPAC WHEN
EXTEND      # MPAC CONTAINS A VECTOR COMPONENT.
MP      MPAC +1
TS      MPAC +1
XCH     L
TCF     VSHR2          # GO ADD ONE IF NECESSARY AND FINISH.

# Page 1154
# ROUTINE FOR SHORT SCALAR SHIFT LEFT (AND MAYBE ROUND).

TSSL    CA      SR      # GET SHIFT COUNT FOR SR.
+1      TS      MPTMP

+2      EXTEND      # ENTRY HERE FROM SL FOR SCALARS.
DCA     MPAC +1      # SHIFTING LEFT ONE PLACE AT A TIME IS
DAS     MPAC +1      # FASTER THAN DOING THE WHOLE SHIFT WITH
AD      MPAC          # MULTIPLIES ASSUMING THAT FREQUENCY OF
AD      MPAC          # SHIFT COUNTS GOES DOWN RAPIDLY AS A
TS      MPAC          # FUNCTION OF THEIR MAGNITUDE.
TCF     +2
TS      OVFINF        # OVERFLOW. (LEAVES OVERFLOW-CORRECTED
                        # RESULT ANYWAY).
CCS     MPTMP        # LOOP ON DECREMENTED SHIFT COUNT.
TCF     TSSL +1

CCS     CYR          # SEE IF ROUND WANTED.

```

July 12, 2016

797

ROUND	TC	ROUND SUB	# YES -- ROUND AND EXIT.
	TCF	DANZIG	# SL LEAVES A ZERO IN CYR FOR NO ROUND.
	TCF	DANZIG	# NO -- EXIT IMMEDIATELY

Page 1155

VECTOR SHIFTING ROUTINES.

SHORTV	CAF	LOW3	# SAVE 3 BIT SHIFT COUNT -- 1 WITHOUT
	MASK	CYR	# EDITING CYR.
	TS	MPTEMP	
	CCS	CYR	# SEE IF LEFT OR RIGHT SHIFT.
	TCF	VSSL	# VECTOR SHIFT LEFT.
OCT176	OCT	176	# USED IN PROCESSED SHIFTS WITH - COUNT.
VSSR	INDEX	MPTEMP	# (ENTRY FROM SR). PICK UP SHIFTING BIT.
	CAF	BIT14	# MPTEMP CONTAINS THE SHIFT COUNT - 1.
	TS	MPTEMP	
	TC	VSHRRND	# SHIFT X COMPONENT.
	DXCH	MPAC	# SWAP X AND Y COMPONENTS.
	DXCH	MPAC +3	
	DXCH	MPAC	
	TC	VSHRRND	# SHIFT Y COMPONENT.
	DXCH	MPAC	# SWAP Y AND Z COMPONENTS.
	DXCH	MPAC +5	
	DXCH	MPAC	
	TC	VSHRRND	# SHIFT Z COMPONENT.
	TCF	VROTATEX	# RESTORE COMPONENTS TO PROPER PLACES.

Page 1156

VECTOR SHIFT LEFT -- DONE ONE PLACE AT A TIME.

-1	TS	MPTEMP	# SHIFTING LOOP.
VSSL	EXTEND		
	DCA	MPAC	
	DAS	MPAC	
	EXTEND		
	BZF	+2	
	TC	OVERFLOW	
	EXTEND		
	DCA	MPAC +3	
	DAS	MPAC +3	

```

EXTEND
BZF      +2
TC       OVERFLWY

```

```

EXTEND
DCA      MPAC +5
DAS      MPAC +5
EXTEND
BZF      +2
TC       OVERFLWZ

```

```

CCS      MPTMP          # LOOP ON DECREMENTED SHIFT COUNTER.
TCF      VSSL -1
TCF      DANZIG         # EXIT.

```

Page 1157

TSLC -- TRIPLE SHIFT LEFT AND COUNT. SHIFTS MPAC LEFT UNTIL GREATER THAN .5 IN MAGNITUDE.
 # THE COMPLEMENT OF THE NUMBER OF SHIFTS REQUIRED IN X.

```

TSLC2      TS      MPTMP          # START BY ZEROING SHIFT COUNT (IN A NOW).
            TC      BRANCH        # EXIT WITH NO SHIFTING IF ARGUMENT ZERO.
            TCF     +2
            TCF     ENDTSLC       # STORES ZERO SHIFT COUNT IN THIS CASE.

            TC      TPAGREE       # MAY CAUSE UPSHIFT OF ONE EXTRA PLACE.

            CA      MPAC
            TCF     TSLCTEST      # BEGIN NORMALIZATION LOOP.

TSLCLOOP    INCR     MPTMP          # INCREMENT SHIFT COUNTER.
            EXTEND
            DCA      MPAC +1
            DAS      MPAC +1
            AD       MPAC
            ADS      MPAC

TSLCTEST    DOUBLE          # SEE IF (ANOTHER) SHIFT IS REQUIRED
            OVSK
            TCF      TSLCLOOP     # YES -- INCREMENT COUNT AND SHIFT AGAIN.

ENDTSLC     CS      MPTMP
            TCF     STORE1        # STORE SHIFT COUNT AND RETURN TO DANZIG.

```

Page 1158

THE FOLLOWING ROUTINE PROCESSES THE GENERAL SHIFT INSTRUCTIONS SR, SRR, SL, AND SLL.
 # THE GIVEN ADDRESS IS DECODED AS FOLLOWS:
 # BITS 1-7 SHIFT COUNT (SUBADDRESS) LESS THAN 125 DECIMAL.

```

#      BIT 8      PSEUDO SIGN BIT (DETECTS CHANGE IN SIGN IN INDEXED SHIFTS).
#      BIT 9      0 FOR LEFT SHIFT, AND 1 FOR RIGHT SHIFT.
#      BIT 10     1 FOR TERMINAL ROUND ON SCALAR SHIFTS, 0 OTHERWISE
#      BITS 11-13 0.
#      BIT 14     1.
#      BIT 15     0.
# THE ABOVE ENCODING IS DONE BY THE YUL SYSTEM.

```

```

GENSHIFT      MASK  ADDRWD      # GET SHIFT COUNT, TESTING FOR ZERO.
              CCS   A           # (ARRIVES WITH C(A) = LOW7).
              TCF   GENSHFT2    # IF NON-ZERO, PROCEED WITH DECREMENTED CT

              CAF   BIT10       # ZERO SHIFT COUNT. NO SHIFTS NEEDED BUT
              MASK  ADDRWD      # WE MIGHT HAVE TO ROUND MPAC ON SLR AND
              CCS   A           # SRR (SCALAR ONLY).
              TC    ROUND SUB
              TCF   DANZIG

GENSHFT2      TS     MPTEMP      # DECREMENTED SHIFT COUNT TO MPTEMP.
              CAF   BIT8        # TEST MEANING OF LOW SEVEN BIT COUNT IN
              EXTEND                                     # MPTEMP NOW.
              MP     ADDRWD
              MASK  LOW2        # JUMPS ON SHIFT DIRECTION (BIT8) AND
              INDEX A           #
              TCF   +1          # ORIGINAL SHIFT DIRECTION (BIT 9)
              TCF   RIGHT-      # NEGATIVE SHIFT COUNT FOR SL OR SLR.
              TCF   LEFT        # SL OR SLR.
              TCF   LEFT-       # NEGATIVE SHIFT COUNT WITH SR OR SRR.

```

```

# Page 1159

```

```

# GENERAL SHIFT RIGHT

```

```

RIGHT         CCS   MODE        # SET IF VECTOR OR SCALAR.
              TCF   GENSCR
              TCF   GENSCR

              CA     MPTEMP      # SEE IF SHIFT COUNT LESS THAN 14D.
              AD     NEG12
              EXTEND
              BZMF   VSSR        # IF SO, BRANCH AND SHIFT IMMEDIATELY.

              AD     NEGONE      # IF NOT, REDUCE MPTEMP BY A TOTAL OF 14.
              TS     MPTEMP      # AND DO A SHIFT RIGHT AND ROUND BY 14.
              CAF    ZERO        # THE ROUND AT THIS STAGE MAY INTRODUCE A
              TS     L           # ONE BIT ERROR IN A SHIFT RIGHT 15D.
              XCH    MPAC

```

	XCH	MPAC +1	
	TC	SETROUND	# X COMPONENT NOW SHIFTED, SO MAKE UP THE
	DAS	MPAC	# ROUNDING QUANTITY (0 IN A AND 0 OR +-1
			# IN L).
	XCH	MPAC +3	# REPEAT THE ABOVE PROCESS FOR Y AND Z/
	XCH	MPAC +4	
	TC	SETROUND	
	DAS	MPAC +3	# NO OVERFLOW ON THESE ADDS.
	XCH	MPAC +5	
	XCH	MPAC +6	
	TC	SETROUND	
	DAS	MPAC +5	
	CCS	MPTMP	# SEE IF DONE, DOING FINAL DECREMENT.
	TS	MPTMP	
	TCF	VRIGHT2	
BIASLO	DEC	.2974 B-1	# SQRT CONSTANT
	TCF	DANZIG	
SETROUND	DOUBLE		# MAKES UP ROUNDING QUANTITY FROM ARRIVING
	TS	MPAC +2	# C(A). L IS ZERO INITIALLY.
	CAF	ZERO	
	XCH	L	
	TC	Q	# RETURN AND DO THE DAS, RESETTNG L TO 0.
# Page 1160			
# PROCESS SR AND SRR FOR SCALARS.			
GENSCR	CA	MPTMP	# SEE IF THE ORIGINAL SHIFT COUNT WAS LESS
+1	AD	NEG12	# THAN 14D.
	EXTEND		
	BZMF	DOSSHFT	# DO THE SHIFT IMMEDIATELY IF SO.
+4	AD	NEGONE	# IF NOT, DECREMENT SHIFT COUNT BY 14D AND
	TS	MPTMP	# SHIFT MPAC RIGHT 14 PLACES.
	CAF	ZERO	
	XCH	MPAC	
	XCH	MPAC +1	
	TS	MPAC +2	
	CCS	MPTMP	# SEE IF FINISHED, DO FINAL DECREMENT.
	TS	MPTMP	
	TC	GENSCR +1	
SLOPEHI	DEC	.5884	# SQRT CONSTANT.
	CAF	BIT10	# FINISHED WITH SHIFT. SEE IF ROUND

	MASK	ADDRWD	# WANTED.
	CCS	A	
	TC	ROUND SUB	
	TCF	DANZIG	# DO SO AND/OR EXIT.
DOSSHFT	INDEX	MPTEMP	# PICK UP SHIFTING BIT.
	CAF	BIT14	
	TS	MPTEMP	
	CAF	BIT10	# SEE IF TERMINAL ROUND DESIRED.
	MASK	ADDRWD	
	CCS	A	
	TCF	RIGHTR	# YES.
	TCF	MPACSHR	# JUST SHIFT RIGHT.

Page 1161

PROCESS THE RIGHT- (SL(R) WITH A NEGATIVE COUNT), LEFT-, AND LEFT OPTIONS.

RIGHT-	CS	MPTEMP	# GET ABSOLUTE VALUE - 1 OF SHIFT COUNT
	AD	OCT176	# UNDERSTANDING THAT BIT8 (PSEUDO-SIGN)
	TS	MPTEMP	# WAS 1 INITIALLY.
	TCF	RIGHT	# DO NORMAL SHIFT RIGHT.
LEFT-	CS	OCT176	# SAME PROLOGUE TO LEFT FOR INDEXED RIGHT
	AD	MPTEMP	# SHIFT WHOSE NET SHIFT COUNT IS NEGATIVE
	COM		
	TS	MPTEMP	
LEFT	CCS	MODE	# SINCE LEFT SHIFTING IS DONE ONE PLACE AT
	TCF	GENSCL	# A TIME, NO COMPARISON WITH 14 NEED BE
	TCF	GENSCL	# DONE. FOR SCALARS, SEE IF TERMINAL ROUND
	TCF	VSSL	# DESIRED. FOR VECTORS, SHIFT IMMEDIATELY.
GENSCL	CS	ADDRWD	# PUT ROUNDING BIT (BIT 10 OF ADDRWD) INTO
	EXTEND		# BIT 15 OF CYR WHERE THE ROUNDING BIT OF
	MP	BIT6	# A SHORT SHIFT LEFT WOULD BE
	TS	CYR	
	TCF	TSSL +2	# DO THE SHIFT.

Page 1162

SCALAR DIVISION INSTRUCTIONS, DDV AND BDDV, ARE EXECUTED HERE. AT THIS POINT, THE DIVIDEND IS IN BUF.
AND THE DIVISOR IS IN BUF.

DDV/BDDV	CS	ONE	# INITIALIZATION
	TS	DVSIGN	# +-1 FOR POSITIVE QUOTIENT -- -0 FOR NEG.
	TS	DVNORMCT	# DIVIDENT NORMALIZATION COUNT.
	TS	MAXDVSW	# NEAR-ONE DIVIDE FLAG.

	CCS	BUF	# FORCE BUF POSITIVE WITH THE MAJOR PART
	TCF	BUFPOS	# NON-ZERO.
	TCF	+2	
	TCF	BUFNEG	
BUFZERO	TS	MPAC +2	# ZERO THIS.
	TC	TPAGREE	# FORCE SIGN AGREEMENT BEFORE OVERFLOW
	CCS	MPAC	# TEST TO SEE IF MPAC NON-ZERO. (TOO BIG)
	TCF	OVF+	# MAJOR PART OF DIVIDEND IS POSITIVE NON-0
	TCF	+2	
	TCF	OVF+ -1	# MAJOR PART OF DIVIDEND IS NEG. NON-ZERO
	XCH	BUF +1	# SHIFT DIVIDENT AND DIVISOR LEFT 14
	XCH	BUF	
	XCH	MPAC +1	
	XCH	MPAC	
	CCS	BUF	# TRY AGAIN ON FORMER MINOR PART.
	TCF	BUF+	
	TCF	+2	# OVERFLOW ON ZERO DIVISOR.
	TCF	BUF-	
SGNDVOVF	CS	MPAC	# SIGN OF MPAC DETERMINES SIGN OF RESULT.
	EXTEND		
	BZMF	+2	
DVOVF	INCR	DVSIGN	# NEGMAX IN MPAC PERHAPS.
	CAF	POSMAX	# ON DIVISION OVERFLOW OF ANY SORT, SET
	TS	MPAC	# SET DP MPAC TO +-POSMAX.
	TC	FINALDV +3	
	CAF	ONE	# SET OVEFLOW INDICATOR AND EXIT.
	TS	OVFIND	
	TC	DANZIG	
OVF+ -1	INCR	DVSIGN	
	CS	BUF +1	# LOAD LOWER ORDER PART OF DIVISOR.
	TCF	SGNDVOVF	# GET SIGN OF RESULT.
BUF-	EXTEND		# IF BUF IS NEGATIVE, COMPLEMENT IT AND
	DCS	BUF	# MAINTAIN DVSIGN FOR FINAL QUOTIENT SIGN.
	DXCH	BUF	
	INCR	DVSIGN	# NOW -0.
# Page 1163			
BUF+	CCS	MPAC	# FORCE MPAC POSITIVE, CHECKING FOR ZERO
	TCF	MPAC+	# DIVIDEND IN THE PROCESS.

	TCF	+2	
	TCF	MPAC-	
	CCS	MPAC +1	
	TCF	MPAC+	
	TCF	DANZIG	# EXIT IMMEDIATELY ON ZERO DIVIDEND.
	TCF	MPAC-	
	TCF	DANZIG	
MPAC-	EXTEND		# FORCE MPAC POSITIVE AS BUF IN BUF-.
	DCS	MPAC	
	DXCH	MPAC	
	INCR	DVSIGN	# NOW +1 OR -0.
# Page 1164			
MPAC+	CS	MPAC	# CHECK FOR DIVISION OVERFLOW. IF THE
	AD	NEGONE	# MAJOR PART OF THE DIVIDEND IS LESS THAN
	AD	BUF	# THE MAJOR PART OF THE DIVISOR BY AT
	CCS	A	# LEAST TWO, WE CAN PROCEED IMMEDIATELY
	TCF	DVNORM	# WITHOUT NORMALIZATION PRODUCING A DVMAX.
-1/2+2	OCT	60001	# USED IN SQRTSUB.
	TCF	+1	# IF THE ABOVE DOES NOT HOLD, FORCE SIGN
	CAF	HALF	# AGREEMENT IN NUMERATOR AND DENOMINATOR
	DOUBLE		# TO FACILITATE OVERFLOW AND NEAR-ONE
	AD	MPAC +1	# CHECKING.
	TS	MPAC +1	
	CAF	ZERO	
	AD	POSMAX	
	ADS	MPAC	
	CAF	HALF	# SAME FOR BUF.
	DOUBLE		
	AD	BUF +1	
	TS	BUF +1	
	CAF	ZERO	
	AD	POSMAX	
	ADS	BUF	
	CS	MPAC	# CHECK MAGNITUDE OF SIGN-CORRECTED
	AD	BUF	# OPERANDS.
	CCS	A	
	TCF	DVNORM	# DIVIDE OK -- WILL NOT BECOME MAXOV CASE.
LBUF2	ADRES	BUF2	
	TCF	DVOVF	# DIVISOR NOT LESS THAN DIVIDEND -- OVF.
	TS	MAXDVSW	# IF THE MAJOR PARTS OF THE DIVIDEND AND

	CS	MPAC +1	# DIVISOR ARE EQUAL, A SPECIAL APPROXIMA-
	AD	BUF +1	# TION IS USED (PROVIDED THE DIVISION IS
	EXTEND		# POSSIBLE, OF COURSE).
	BZMF	DVOVF	
	TCF	DVNORM	# IF NO OVERFLOW.
# Page 1165			
BUFNORM	EXTEND		# ADD -1 TO AUGMENT SHIFT COUNT AND SHIFT
	AUG	DVNORMCT	# LEFT ONE PLACE.
	EXTEND		
	DCA	BUF	
	DAS	BUF	
DVNORM	CA	BUF	# SEE IF DIVISOR NORMALIZED YET.
	DOUBLE		
	OVSF		
	TCF	BUFNORM	# NO -- SHIFT LEFT ONE AND TRY AGAIN.
	DXCH	MPAC	# CALL DIVIDEND NORMALIZATION SEQUENCE
	INDEX	DVNORMCT	# PRIOR TO DOING THE DIVIDE.
	TC	MAXTEST	
	TS	MPAC +2	# RETURNS WITH DIVISION DONE AND C(A) = 0.
	TCF	DANZIG	
BUFPOS	CCS	A	
	TCF	BUF+	# TO BUF+ IF BUF IS GREATER THAN +1.
	CS	BUF +1	# IF BUF IS +1, FORCING SIGN AGREEMENT
	EXTEND		# MAY CAUSE BUF TO BECOME ZERO.
	BZMF	BUF+	# BRANCH IF SIGNS AGREE.
	CA	HALF	# SIGNS DISAGREE. FORCE AGREEMENT.
	DOUBLE		
+6	ADS	BUF +1	
	CA	ZERO	
	TS	BUF	
	TCF	BUFZERO	
BUFNEG	CCS	A	
	TCF	BUF-	# TO BUF- IF BUF IS LESS THAN -1.
	CA	BUF +1	# IF BUF IS -1, FORCING SIGN AGREEMENT
	EXTEND		# MAY CAUSE BUF TO BECOME ZERO.
	BZMF	BUF-	# BRANCH IF SIGNS AGREE.

July 12, 2016

805

```
CS      HALF      # SIGNS DISAGREE.  FORCE AGREEMENT.
TCF     BUFPOS +6
```

Page 1166

THE FOLLOWING ARE PROLOGUES TO SHIFT THE DIVIDEND ARRIVING IN A AND L BEFORE THE DIVIDE.

```
-21D    LXCH      SR      # SPECIAL PROLOGUE FOR UNIT WHEN THE
        EXTEND    # LENGTH OF THE ARGUMENT WAS NOT LESS THAN
        MP        HALF   # .5.  IN THIS CASE, EACH COMPONENT MUST BE
        XCH       L      # SHIFTED RIGHT ONE TO PRODUCE A HALF-UNIT
        AD        SR     # VECTOR.
        XCH       L
        TCF       GENDDV +1 # WITH DP DIVIDEND IN A,L.
```

```
DDOUBL      # PROLOGUE WHICH NORMALIZES THE DIVIDEND
DDOUBL      # WHEN IT IS KNOWN THAT NO DIVISION
DDOUBL      # OVEFLOW WILL OCCUR.
```

```
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
DDOUBL
```

```
DXCH      MPAC
```

```
MAXTEST    CCS      MAXDVSW      # 0 IF MAJORS MIGHT BE =, -1 OTHERWISE.
BIASHI     DEC      .4192 B-1    # SQRT CONSTANTS.
```

```
TCF       MAXDV      # CHECK TO SEE IF THAY ARE NOW EQUAL.
```

Page 1167

THE FOLLOWING IS A GENERAL PURPOSE DOUBLE PRECISION DIVISION ROUTINE. IT DIVIDES MPAC BY BUF
THE RESULT IN MPAC. THE FOLLOWING CONDITIONS MUST BE SATISFIED:

#

1. THE DIVISOR (BUF) MUST BE POSITIVE AND NOT LESS THAN .5.

#

2. THE DIVIDEND (MPAC) MUST BE POSITIVE WITH THE MAJOR PART OF MPAC STRICTLY LESS
(A SPECIAL APPROXIMATION, MAXDV, IS USED WHEN THE MAJOR PARTS ARE EQUAL).

#

UNDERSTANDING THAT $A/B = Q + S(R/B)$ WHERE $S = 2(-14)$ AND Q AND R ARE QUOTIENT AND REMAINDER,
TIVELY, THE FOLLOWING APPROXIMATION IS OBTAINED BY MULTIPLYING ABOVE AND BELOW BY C - SD AND
ORDER S-SQUARED (POSSIBLY INTRODUCING ERROR INTO THE LOW TWO BITS OF THE RESULT). SIGN AGREE

```

#
#      A + SB .      (R - CD)      A + SB
#      ----- = Q + S(-----) WHERE Q AND R ARE QUOTIENT AND REMAINDER OF ----- RES
#      C + SD      ( C   }      C
#
GENDDV      DXCH      MPAC      # WE NEED A AND B ONLY FOR FIRST DV.
+1          EXTEND      # (SPECIAL UNIT PROLOGUE ENTERS HERE).
            DV      BUF      # A NOW CONTAINS Q AND L, R.
            DXCH      MPAC
#
            CS      MPAC      # FORM DIVIDEND FOR MINOR PART OF RESULT.
            EXTEND
            MP      BUF +1
            AD      MPAC +1      # OVERFLOW AT THIS POINT IS POSITIVE SINCE
            OVSK      # R IS POSITIVE IN EVERY CASE.
            TCF      +5
#
            EXTEND      # OVERFLOW CAN BE REMOVED BY SUBTRACTING C
            SU      BUF      # (BUF) ONCE SINCE R IS ALWAYS LESS THAN C
            INCR      MPAC      # IN THIS CASE. INCR COMPENSATES SUBTRACT.
            TCF      +DOWN      # (SINCE C(A) IS STILL POSITIVE).
#
            +5          EXTEND      # C(A) CAN BE MADE LESS THAN C IN MAGNI-
            BZMF      -UP      # TUDE BY DIMINISHING IT BY C (SINCE C IS
# NOT LESS THAN .5) UNLESS C(A) = 0.
#
# Page 1168
+DOWN      EXTEND
            SU      BUF      # IF POSITIVE, REDUCE ONLY IF NECESSARY
            EXTEND      # SINCE THE COMPENSATING INCR MIGHT CAUSE
            BZF      +3      # OVERFLOW.
            EXTEND      # DON'T SUBTRACT UNLESS RESULT IS POSITIVE
            BZMF      ENDMAXDV      # OR ZERO.
#
            +3          INCR      MPAC      # KEEP SUBTRACT HERE AND COMPENSATE.
            TCF      FINALDV
#
            -UP          EXTEND      # IF ZERO, SET MINOR PART OF RESULT TO
            BZF      FINALDV +3      # ZERO.
#
            EXTEND      # IF NEGATIVE, ADD C TO A, SUBTRACTING ONE
            DIM      MPAC      # TO COMPENSATE. DIM IS OK HERE SINCE THE
            AD      BUF      # MAJOR PART NEVER GOES NEGATIVE.
#
# Page 1169
FINALDV      ZL      # DO DV TO OBTAIN MINOR PART OF RESULT.

```

```

                                EXTEND
                                DV      BUF
+3      TS      MPAC +1

                                CCS      DVSIGN      # LEAVE RESULT POSITIVE UNLESS C(DVSIGN).
                                TC      Q
                                TC      Q
                                TC      Q

                                EXTEND
                                DCS      MPAC
                                DXCH      MPAC
                                CAF      ZERO      # SO WE ALWAYS RETURN WITH C(A) = 0.
                                TC      Q

# Page 1170
# IF THE MAJOR PARTS OF THE DIVISOR AND DIVIDEND ARE EQUAL, BUT THE MINOR PARTS ARE SUCH THAT T
# DIVIDEND IS STRICTLY LESS THAN THE DIVISOR IN MAGNITUDE, THE FOLLOWING APPROXIMATION IS USED.
# ARE THE SAME AS THE GENERAL ROUTINE WITH THE ADDITION THAT SIGN AGREEMENT IS NECESSARY (B, C,
#
#      C + SB .      (C + B - D)
#      ----- = 37777 + S(-----)
#      C + SD      (      C      )
#
# THE DIVISION MAY BE PERFORMED IMMEDIATELY SINCE B IS STRICTLY LESS THAN D AND C IS NOT LESS T

MAXDV      CS      MPAC      # SEE IF MAXDV CASE STILL HOLDS AFTER
            AD      BUF      # NORMALIZATION.
            EXTEND
            BZF      +2
            TCF      GENDDV      # MPAC NOW LESS THAN BUFF -- DIVIDE AS USUAL.

+2      CAF      POSMAX      # SET MAJOR PART OF RESULT.
            TS      MPAC

            CS      BUF +1      # FORM DIVIDEND OF MINOR PART OF RESULT.
            AD      MPAC +1
            TCF      ENDMAXDV      # GO ADD C AND DO DIVIDE, ATTACHING SIGN
                                   # BEFORE EXITING.

# Page 1171
# VECTOR DIVIDED BY SCALAR, V/SC, IS EXECUTED HERE.  THE VECTOR IS NOW IN MPAC WITH SCALAR IN E

V/SC2      CS      ONE      # INITIALIZE DIVIDEND NORMALIZATION COUNT
            TS      DVNORMCT      # AND DIVISION SIGN REGISTER.
            TS      VBUF +5

```

	TC	VECAGREE	# FORCE SIGN AGREEMENT IN VECTOR
	DXCH	BUF	
	TC	ALSIGNAG	# SIGN AGREE BUF
	DXCH	BUF	
	CCS	BUF	# FORCE DIVISOR POSITIVE WITH MAJOR PART
	TCF	/BUF+	# NON-ZERO (IF POSSIBLE).
	TCF	+2	
	TCF	/BUF-	
	XCH	BUF +1	# SHIFT VECTOR AND SCALAR LEFT 14.
	XCH	BUF	
	XCH	MPAC +1	
	XCH	MPAC	
	EXTEND		# CHECK FOR OVERFLOW IN EACH CASE.
	BZF	+2	
	TCF	DVOVF	
	XCH	MPAC +4	
	XCH	MPAC +3	
	EXTEND		
	BZF	+2	
	TCF	DVOVF	
	XCH	MPAC +6	
	XCH	MPAC +5	
	EXTEND		
	BZF	+2	
	TCF	DVOVF	
	CCS	BUF	
	TCF	/BUF+	
	TCF	DVOVF	# ZERO DIVISOR - OVERFLOW.
	TCF	/BUF-	
	TCF	DVOVF	
/BUF-	EXTEND		# ON NEGATIVE, COMPLEMENT BUF AND MAINTAIN
	DCS	BUF	# DVSIGN IN VBUF +5.
	DXCH	BUF	
	INCR	VBUF +5	
# Page 1172			
/BUF+	EXTEND		
	DCA	BUF	# LEAVE ABS(ORIG DIVISOR) IN BUF2
	DXCH	BUF2	# FOR OVERFLOW TESTING

July 12, 2016

809

	TCF	/NORM	# NORMALIZE DIVISOR IN BUF.
/NORM2	EXTEND		# IF LESS THAN .5, AUGMENT DVNORMCT AND
	AUG	DVNORMCT	# DOUBLE DIVISOR.
	EXTEND		
	DCA	BUF	
	DAS	BUF	
/NORM	CA	BUF	# SEE IF DIVISOR NORMALIZED.
	DOUBLE		
	OVSF		
	TCF	/NORM2	# DOUBLE AND TRY AGAIN IF NOT.
	TC	V/SCDV	# DO X COMPONENT DIVIDE.
	DXCH	MPAC +3	# SUPPLY ARGUMENTS IN USUAL SEQUENCE.
	DXCH	MPAC	
	DXCH	MPAC +3	
	TC	V/SCDV	# Y COMPONENT.
	DXCH	MPAC +5	
	DXCH	MPAC	
	DXCH	MPAC +5	
	TC	V/SCDV	# Z COMPONENT.
	TCF	VROTATEX	# GO RE-ARRANGE COMPONENTS BEFORE EXIT.
# Page 1173			
# SUBROUTINE USED BY V/SC TO DIVIDE VECTOR COMPONENT IN MPAC,+1 BY THE SCALAR GIVEN IN BUF.			
V/SCDV	CA	VBUF +5	# REFLECTS SIGN OF SCALAR.
	TS	DVSIGN	
	CCS	MPAC	# FORCE MPAC POSITIVE, EXITING ON ZERO.
	TCF	/MPAC+	
	TCF	+2	
	TCF	/MPAC-	
	CCS	MPAC +1	
	TCF	/MPAC+	
	TC	Q	
	TCF	/MPAC-	
	TC	Q	
/MPAC-	EXTEND		# USUAL COMPLEMENTING AND SETTING OF SIGN.
	DCS	MPAC	
	DXCH	MPAC	

	INCR	DVSIGN	
/MPAC+	CS	ONE	# INITIALIZE NEAR-ONE SWITCH.
	TS	MAXDVSW	
	CS	MPAC	# CHECK POSSIBLE OVERFLOW.
	AD	BUF2	# UNNORMALIZED INPUT DIVISOR.
	CCS	A	
	TCF	DDVCALL	# NOT NEAR-ONE
	TCF	+2	# +0 IS JUST POSSIBLE
	TCF	DVOVF	# NO HOPE
	TS	MAXDVSW	# SIGNAL POSSIBLE NEAR-ONE CASE
	CS	MPAC +1	# SEE IF DIVISION CAN BE DONE
	AD	BUF2 +1	
	EXTEND		
	BZMF	DVOVF	
DDVCALL	DXCH	MPAC	# CALL PRE-DIVIDE NORMALIZATION.
	INDEX	DVNORMCT	
	TCF	MAXTEST	

Page 1174

SLOPELO DEC .8324

VECAGREE	XCH	Q	# SAVE Q IN A
	DXCH	MPAC	
	TC	ALSIGNAG	# SIGNAGREE MPAC
	DXCH	MPAC	
	DXCH	MPAC +3	
	TC	ALSIGNAG	# SIGN AGREE MPAC +3
	DXCH	MPAC +3	
	DXCH	MPAC +5	
	TC	ALSIGNAG	# SIGNAGREE MPAC +5
	DXCH	MPAC +5	
	TC	A	

Page 1175

THE FOLLOWING ROUTINE EXECUTES THE UNIT INSTRUCTION, WHICH TAKES THE UNIT OF THE V

UNIT	TC	VECAGREE	# FORCE SIGN AGREEMENT IN VECTOR
	TC	MPACVBUF	# SAVE ARGUMENT IN VBUF
	CAF	ZERO	# MUST SENSE OVERFLOW IN FOLLOWING DOT.
	XCH	OVFIND	
	TS	TEM1	
	TC	VSQSUB	# DOT MPAC WITH ITSELF.
	CA	TEM1	

```

XCH      OV FIND
EXTEND
BZF      +2
TCF      DVOVF
EXTEND
DCA      MPAC      # LEAVE THE SQUARE OF THE LENGTH OF THE
INDEX    FIXLOC    # ARGUMENT IN LVSQUARE.
DXCH     LVSQUARE

TC        SQRTSUB   # GO TAKE THE NORMALIZED SQUARE ROOT.

CCS      MPAC      # CHECK FOR UNIT OVERFLOW.
TCF      +5        # MPAC IS NOT LESS THAN .5 UNLESS
TS        L
INDEX    FIXLOC
DXCH     LV
TCF      DVOVF     # INPUT TO SQRTSUB WAS 0.

CS        FOURTEEN # SEE IF THE INPUT WAS SO SMALL THAT THE
AD        MPTEMP   # FIRST TWO REGISTERS OF THE SQUARE WERE 0
CCS      A
COM
TCF      SMALL    # IF SO, SAVE THE NEGATIVE OF THE SHIFT
                        # COUNT -15D.

TCF      LARGE    # (THIS IS USUALLY THE CASE.)

CS        THIRTEEN # IF THE SHIFT COUNT WAS EXACTLY 14, SET
TS        MPTEMP   # THE PRE-DIVIDE NORM COUNT TO -13D.

SMALL2   CA        MPAC      # SHIFT THE LENGTH RIGHT 14 BEFORE STORING
TS        L        # (SMALL EXITS TO THIS POINT).
CAF      ZERO
TCF      LARGE2    # GO TO STORE LENGTH AND PROCEED.

LARGE    CCS      MPTEMP   # MOST ALL CASES COME HERE.
TCF      LARGE3    # SEE IF NO NORMALIZATION WAS REQUIRED BY
CS        SRDDV    # Sqrt, AND IF SO, SET UP FOR A SHIFT
TS        MPTEMP   # RIGHT 1 BEFORE DIVIDING TO PRODUCE
EXTEND
DCA      MPAC      # THE DESIRED HALF UNIT VECTOR.

# Page 1176
TCF      LARGE2

# Page 1177
LARGE3   COM
TS        MPTEMP   # LEAVE NEGATIVE OF SHIFT COUNT-1 FOR
                        # PREDIVIDE LEFT SHIFT.

```

```

COM                                     # PICK UP REQUIRED SHIFTING BIT TO UNNORM-
INDEX  A                               # ALIZE THE SQRT RESULT.
CAF    BIT14
TS     BUF
EXTEND
MP     MPAC +1
XCH    BUF
EXTEND                                     # (UNNORMALIZE THE SQRT FOR LV).
MP     MPAC
XCH    L
AD     BUF
XCH    L

LARGE2  INDEX  FIXLOC
DXCH    LV                               # LENGTH NOW STORED IN WORK AREA.

CS      ONE
TS      MAXDVSW                          # NO MAXDV CASES IN UNIT.

DXCH    VBUF                             # PREPARE X COMPONENT FOR DIVIDE, SETTING
DXCH    MPAC                             # LENGTH OF VECTOR AS DIVISOR IN BUF.
DXCH    BUF
TC      UNITDV

DXCH    VBUF +2                          # DO Y AND Z IN USUAL FASHION SO WE CAN
DXCH    MPAC                             # EXIT THROUGH VROTATEX.
DXCH    MPAC +3
TC      UNITDV

DXCH    VBUF +4
DXCH    MPAC
DXCH    MPAC +5
TC      UNITDV
TCF     VROTATEX                          # AND EXIT.

# Page 1178
# IF THE LENGTH OF THE ARGUMENT VECTOR WAS LESS THAN 2(-28), EACH COMPONENT MUST BE S
# 14 PLACES BEFORE TEH DIVIDE, NOTE THAT IN THIS CASE, THE MAJOR PART OF EACH COMPON

SMALL   TS      MPTMP                     # NEGATIVE OF PRE-DIVIDE SHIFT COUNT.

CAF     ZERO                               # SHIFT EACH COMPONENT LEFT 14.
XCH     VBUF +1
XCH     VBUF
XCH     VBUF +3

```

```

XCH      VBUF +2
XCH      VBUF +5
XCH      VBUF +4

```

```

CS      MPTEMP
INDEX   A
CAF     BIT14
EXTEND
MP      MPAC
TCF     SMALL2

```

```

THIRTEEN = OCT15
FOURTEEN = OCT16
OCT16    = R1D1

```

Page 1179

THE FOLLOWING ROUTINE SETS UP THE CALL TO THE DIVIDE ROUTINES.

```

UNITDV      CCS      MPAC      # FORCE MPAC POSITIVE IF POSSIBLE, SETTING
            TCF      UMPAC+    # DVSIGN ACCORDING TO THE SIGN OF MPAC
            TCF      +2        # SINCE THE DIVISOR IS ALWAYS POSITIVE
            TCF      UMPAC-    # HERE.

```

```

            CCS      MPAC +1
            TCF      UMPAC+
            TC       Q          # EXIT IMMEDIATELY ON ZERO.
            TCF      UMPAC-
            TC       Q

```

```

UMPAC-      CS      ZERO      # IF NEGATIVE, SET -0 IN DVSIGN FOR FINAL
            TS      DVSIGN    # COMPLEMENT.
            EXTEND
            DCS      MPAC      # PICK UP ABSOLUTE VALUE OF ARG AND JUMP.
            INDEX   MPTEMP
            TCF      MAXTEST -1

```

```

UMPAC+      TS      DVSIGN    # SET DVSIGN FOR POSITIVE QUOTIENT.
            DXCH     MPAC
            INDEX   MPTEMP
            TCF      MAXTEST -1

```

Page 1180

MISCELLANEOUS UNARY OPERATIONS.

```

DSQ         TC      DSQSUB    # SQUARE THE DP CONTENTS OF MPAC.
            TCF     DANZIG

```

ABVALABS	CCS	MODE	# ABVAL OR ABS INSTRUCTION.
	TCF	ABS	# DO ABS ON SCALAR.
	TCF	ABS	
ABVAL	TC	VSQSUB	# DOT MPAC WITH ITSELF.
	LXCH	MODE	# MODE IS NOW DP (L ZERO AFTER DAS).
	EXTEND		# STORE SQUARE OF LENGTH IN WORK AREA.
	DCA	MPAC	
	INDEX	FIXLOC	
	DXCH	LVSQUARE	

Page 1181

PROGRAM DESCRIPTION -- SUBROUTINE SQRT

#

FUNCTIONAL DESCRIPTION -- DOUBLE PRECISION SQUARE ROOT ROUTINE

THIS PROGRAM TAKES THE SQUARE ROOT OF THE 27 OR 28 MOST SIGNIFICANT BITS IN TWO
NUMBERS -- MPAC, MPAC+1, AND MPAC+2. THE ROOT IS RETURNED DOUBLE PRECISION IN MPAC.

#

WARNING -- THIS SUBROUTINE USES A TRIPLE PRECISION INPUT. THE PROGRAMMER MUST ASSURE
ESPECIALLY IF THE CONTENTS OF MPAC IS SMALL OR ZERO. FOR DETAILS SEE STG MEN

#

CALLING SEQUENCE -- IN INTERPRETIVE MODE, I.E., FOLLOWING 'TC INTPRET', 'SQRT', NO
INPUT SCALING: THE BINARY POINT IS ASSUMED TO THE RIGHT OF BIT 15. THE ANSWER IS

#

SUBROUTINES -- GENSCR, MPACSHR, SQRTSUB, ABORT

#

ABORT EXIT MODE -- ABORTS ON NEGATIVE INPUT -1.2X10E-4 (77775 OCTAL) OR LESS.
DISPLAYS ERROR CODE 1302

TC ABORT

OCT 1302

#

DEBRIS -- LOCATIONS BUF, MPTEMP, ADDRWD ARE USED

SQRT	TC	SQRTSUB	# TAKE THE SQUARE ROOT OF MPAC.
	CCS	MPTEMP	# RETURNED NORMALIZED SQUARE ROOT. SEE IF
	TCF	+2	# ANY UN-NORMALIZATION REQUIRED AND EXIT
	TCF	DANZIG	# IF NOT.
	AD	NEG12	# A RIGHT SHIFT OF MORE THAN 13 COULD BE
	EXTEND		# REQUIRED IF INPUT WAS ZERO IN MPAC,+1.
	BZMF	SQRTSHFT	# GOES HERE IN MOST CASES.
	ZL		# IF A LONG SHIFT IS REQUIRED, GO TO
	LXCH	ADDRWD	# GENERAL RIGHT SHIFT ROUTINES.
	TCF	GENSCR +4	# ADDRWD WAS ZERO TO PREVENT ROUND.

SQRTSHFT	INDEX	MPTMP	# SELECT SHIFTING BIT AND EXIT THROUGH
	CAF	BIT15	# SHIFT ROUTINES.
	TS	MPTMP	
	CAF	ZERO	# TO ZERO MPAC +2 IN THE PROCESS.
	TCF	MPACSHR +3	
ABS	TC	BRANCH	# TEST SIGN OF MPAC AND COMPLEMENT IF
	TCF	DANZIG	
	TCF	DANZIG	
	TCF	COMP	
# Page 1182			
VDEF	CS	FOUR	# VECTOR DEFINE -- ESSENTIALLY TREATS
	ADS	PUSHLOC	# SCALAR IN MPAC AS X COMPONENT, PUSHES UP
	EXTEND		# FOR Y AND THEN AGAIN FOR Z.
	INDEX	A	
	DCA	2	
	DXCH	MPAC +3	
	EXTEND		
	INDEX	PUSHLOC	
	DCA	0	
	DXCH	MPAC +5	
	TCF	VMODE	# MODE IS NON VECTOR.
VSQ	TC	VSQSUB	# DOT MPAC WITH ITSELF.
	TCF	DMODE	# MODE IS NOW DP.
PUSH	EXTEND		# PUSH DOWN MPAC LEAVING IT LOADED.
	DCA	MPAC	
	INDEX	PUSHLOC	# PUSH DOWN FIRST TWO REGISTERS IN EACH
	DXCH	0	
	INDEX	MODE	# INCREMENT PUSHDOWN POINTER.
	CAF	NO.WDS	
	ADS	PUSHLOC	
	CCS	MODE	
	TCF	TPUSH	# PUSH DOWN MPAC +2.
	TCF	DANZIG	# DONE FOR DP.
	EXTEND		# ON VECTOR, PUSH DOWN Y AND Z COMPONENTS.
	DCA	MPAC +3	
	INDEX	PUSHLOC	
	DXCH	0 -4	
	EXTEND		

```

DCA      MPAC +5
INDEX    PUSHLOC
DXCH     0 -2
TCF      DANZIG

TPUSH    CA      MPAC +2
         TCF      ENDTPUSH +2

RVQ      INDEX    FIXLOC      # RVQ -- RETURN IVA QPRET.
         CA      QPRET
         TS      POLISH
         TCF      GOTO +4      # (ASSUME QPRET POINTS TO FIXED ONLY.)

# Page 1183
# THE FOLLOWING SUBROUTINES ARE USED IN SQUARING MPAC, IN BOTH THE SCALAR AND VECTOR
# SPECIAL CASES OF DMPSUB AND DOTSUB, PUT IN TO SAVE SOME TIME.

DSQSUB   CA      MPAC +1      # SQUARES THE SCALAR CONTENTS OF MPAC.
         EXTEND
         SQUARE
         TS      MPAC +2
         CAF     ZERO          # FORM 2(CROSS TERM).
         XCH     MPAC +1
         EXTEND
         MP      MPAC
         DDOUBL          # AND MAYBE OVEFLOW.
         DAS     MPAC +1      # AND SET A TO NET OVERFLOW.
         XCH     MPAC
         EXTEND
         SQUARE
         DAS     MPAC
         TC      Q

VSQSUB   EXTEND          # DOTS THE VECTOR IN MPAC WITH ITSELF.
         QXCH    DOTRET
         TC      DSQSUB      # SQUARE THE X COMPONENT.
         DXCH    MPAC +3
         DXCH    MPAC
         DXCH    BUF          # SO WE CAN END IN DOTSUB.
         CA      MPAC +2
         TS      BUF +2

         TC      DSQSUB      # SQUARE Y COMPONENT.
         DXCH    MPAC +1
         DAS     BUF +1
         AD      MPAC

```



```

AD      BUF
TS      BUF
TCF     +2
TS      OVFLND      # IF OVERFLOW.

DXCH    MPAC +5
DXCH    MPAC
TC      DSQSUB      # SQUARE Z COMPONENT.
TCF     ENDDOT      # END AS IN DOTSUB.

```

Page 1184

DOUBLE PRECISION SQUARE ROOT ROUTINE. TAKE THE SQUARE ROOT OF THE TRIPLE PRECISION (MPAC +2
IN NORMALIZATION) CONTENTS OF MPAC AND LEAVE THE NORMALIZED RESULT IN MPAC (C(MPAC) GREATER THAN
.5). THE RIGHT SHIFT COUNT (TC UNNORMALIZE) IS LEFT IN MPTMP.

```

SQRTSUB  CAF      ZERO      # START BY ZEROING RIGHT SHIFT COUNT.
          TS      MPTMP

          CCS      MPAC      # CHECK FOR POSITIVE ARGUMENT, SHIFTING
          TCF      SMPAC+    # FIRST SIGNIFICANT MPAC REGISTER INTO
          TCF      +2        # MPAC ITSELF.
          TCF      SQRTNEG   # SEE IF MAG OF ARGUMENT LESS THAN 10(-4).

          XCH      MPAC +2   # MPAC IS ZERO -- SHIFT LEFT 14.
          XCH      MPAC +1
          TS      MPAC
          CAF      SEVEN     # AUGMENT RIGHT SHIFT COUNTER.
          TS      MPTMP

          CCS      MPAC      # SEE IF MPAC NOW PNZ.
          TCF      SMPAC+
          TCF      +2
          TCF      ZEROANS   # NEGATIVE BUT LESS THAN 10(-4) IN MAG.

          XCH      MPAC +1   # ZERO -- SHIFT LEFT 14 AGAIN.
          TS      MPAC
          CAF      SEVEN     # AUGMENT RIGHT SHIFT COUNTER.
          ADS      MPTMP

          CCS      MPAC
          TCF      SMPAC+
          TC       Q         # Sqrt(0) = 0.
          TCF      ZEROANS
          TCF      FIXROOT   # DO NOT LEAVE SQRTSUB WITH -0 IN MPAC.

SQRTNEG  CCS      A         # ARGUMENT IS NEGATIVE, BUT SEE IF SIGN-

```

	TCF	SQRTABRT	# CORRECTED ARGUMENT IS LESS THAN $10(-4)$
	CCS	MPAC +1	# IN MAGNITUDE. IF SO, CALL ANSWER ZERO.
ZEROANS	CAF	ZERO	# FORCE ANSWER TO ZERO HERE.
	TCF	FIXROOT	
	TCF	SQRTABRT	
	TCF	FIXROOT	
SQRTABRT	TC	POOD00	
	OCT	1302	
# Page 1185			
SMPAC+	AD	-1/2+2	# SEE IF ARGUMENT GREATER THAN OR EQUAL TO
	EXTEND		# .5.
	BZMF	SRTEST	# IF SO, SEE IF LESS THAN .25.
	DXCH	MPAC	# WE WILL TAKE THE SQUARE ROOT OF MPAC/2.
	LXCH	SR	# SHIFT RIGHT 1 AND GO TO THE SQRT ROUTINE
	EXTEND		
	MP	HALF	
	DXCH	MPAC	
	XCH	SR	
	ADS	MPAC +1	# GUARANTEED NO OVERFLOW.
ARGHI	CAF	SLOPEHI	# ARGUMENT BETWEEN .25 AND .5, GET A
	EXTEND		# LINEAR APPROXIMATION FOR THIS RANGE.
	MP	MPAC	
	AD	BIASHI	# $X0/2 = (MPAC/2)(SLOPHI) + BIASHI/2$.
+4	TS	BUF	# $X0/2$ (ARGLO ENTERS HERE).
	CA	MPAC	# SINGLE-PRECISION THROUGHOUT.
	ZL		
	EXTEND		
	DV	BUF	# $(MPAC/2)/(X0/2)$
	EXTEND		
	MP	HALF	
	ADS	BUF	# $X1 = X0/2 + .5(MPAC/2)/(X0/2)$
	EXTEND		
	MP	HALF	# FORM UP $X1/2$.
	DXCH	MPAC	# SAVE AND BRING OUT ARGUMENT.
	EXTEND		# TAKE DP QUOTIENT WITH $X1$.
	DV	BUF	
	TS	BUF +1	# SAVE MAJOR PART OF QUOTIENT.
	CAF	ZERO	# FORM MINOR PART OF QUOTIENT USING
	XCH	L	# (REMAINDER,0).

```

EXTEND
DV      BUF
TS      L      # IN PREPARATION FOR DAS.
CA      BUF +1
DAS     MPAC    #  $X_2 = X_1/2 + (MPAC/2)X_1$ 

EXTEND      # OVERFLOWS IF ARG. NEAR POSMAX.
BZF      TCQBNK00
CAF      POSMAX
TS      MPAC
TS      MPAC +1
TCQBNK00 TC      Q      # RETURN TO CALLER TO UNNORMALIZE, ETC.

# Page 1186
SRTEST    AD      QUARTER      # ARGUMENT WAS LESS THAN .5, SEE IF LESS
EXTEND    # THAN .25.
BZMF      SQRTNORM      # IF SO, BEGIN NORMALIZATION.

DXCH      MPAC      # IF BETWEEN .5 AND .25, SHIFT RIGHT 1 AND
LXCH      SR        # START AT ARGLO.
EXTEND
MP        HALF
DXCH      MPAC
XCH       SR
ADS       MPAC +1    # NO OVERFLOW.

ARGLO     CAF      SLOPELO      # (NORMALIZED) ARGUMENT BETWEEN .125 AND
EXTEND    # .25
MP        MPAC
AD        BIASLO
TCF       ARGHI +4    # BEGIN SQUARE ROOT.

SQRTNM2   EXTEND      # SHIFT LEFT 2 AND INCREMENT RIGHT SHIFT
DCA       MPAC +1    # COUNT (FOR TERMINAL UNNORMALIZATION).
DAS       MPAC +1
AD        MPAC
ADS       MPAC      # (NO OVERFLOW).

SQRTNORM  INCR      MPTMP      # FIRST TIME THROUGH, JUST SHIFT LEFT 1
EXTEND    # (PUTS IN EFFECTIVE RIGHT SHIFT SINCE
DCA       MPAC +1    # WE WANT  $MPAC/2$ ).
DAS       MPAC +1
AD        MPAC
ADS       MPAC      # (AGAIN NO OVERFLOW).
DOUBLE
TS        CYL

```

```

NORMTEST      CCS      CYL          # SEE IF ARGUMENT NOW NORMALIZED AT
              CCS      CYL          # GREATER THAN .125.
              TCF      SQRTNM2      # NO -- SHIFT LEFT 2 MORE AND TRY AGAIN.
              TCF      ARGHI        # YES -- NOW BETWEEN .5 AND .25.
              TCF      ARGLO        # ARGUMENT NOW BETWEEN .25 AND .125.

# Page 1187
# TRIGONOMETRIC FUNCTION PACKAGE.
# THE FOLLOWING TRIGONOMETRIC FUNCTIONS ARE AVAILABLE AS INTERPRETIVE OPERATIONS.
# 1. SIN COMPUTES (1/2)SINE(2 PI MPAC).
# 2. COS COMPUTES (1/2)COSINE(2 PI MPAC).
# 3. ASIN COMPUTES (1/2PI)ARCSINE(2 MPAC).
# 4. ACOS COMPUTES (1/2PI)ARCCOSINE(2 MPAC).
#
# SIN-ASIN AND COS-ACOS ARE MUTUALLY INVERSE, I.E., SIN(ASIN(X)) = X.

COSINE        TC      BRANCH        # FINDS COSINE USING THE IDENTITY
              TCF      +3           # COS(X) = SIN(PI/2 - ABS(X)).
              TCF      PRESINE
              TCF      PRESINE

              +3      EXTEND
              DCS      MPAC
              DXCH      MPAC

PRESINE        CAF      QUARTER      # PI/2 SCALED.
              ADS      MPAC

SINE           DXCH      MPAC        # DOUBLE ARGUMENT.
              DDOUBL
              OVSK        # SEE IF OVERFLOW PRESENT.
              TCF      +3          # IF NOT, ARGUMENT OK AS IS.

              EXTEND          # IF SO, WE LOST (OR GAINED) PI, SO
              DCOM           # COMPLEMENT MPAC USING THE IDENTITY
                              # SIN(X-(+)PI) = SIN(-X).

              +3      DXCH      MPAC
              CA      MPAC        # SEE IF ARGUMENT GREATER THAN .5 IN
              DOUBLE          # MAGNITUDE. IF SO, REDUCE IT TO LESS THAN
              TS      L          # .5 (+-PI/2 SCALED) AS FOLLOWS:
              TCF      SN1

              INDEX      A          # IF POSITIVE, FORM PI - X, IF NEGATIVE
              CAF      NEG1/2 +1    # USE -PI -X.
              DOUBLE

```

```

EXTEND
SU      MPAC      # GUARANTEED NO OVERFLOW.
TS      MPAC
CS      MPAC +1
TS      MPAC +1

# Page 1188
SN1     EXTEND      # SET UP TO EVALUATE HASTINGS POLYNOMIAL
DCA     MPAC
DXCH    BUF2
TC      DSQSUB      # SQUARE MPAC.

TC      POLY      # EVALUATE FOURTH ORDER POLYNOMIAL.
DEC     3
2DEC    +.3926990796
2DEC    -.6459637111
2DEC    +.318758717
2DEC    -.074780249
2DEC    +.009694988

CAF     LBUF2      # MULTIPLY BY ARGUMENT AND SHIFT LEFT 2.
TC      DMPSUB -1

EXTEND
DCA     MPAC +1
DAS     MPAC +1
AD      MPAC
ADS     MPAC      # NEITHER SHIFT OVERFLOWS.
EXTEND
DCA     MPAC +1
DAS     MPAC +1
AD      MPAC
ADS     MPAC
TCF     DANZIG

# Page 1189
# ARCSIN/ARCCOS ROUTINE.

ARCSIN  CAF      LASINEX      # COMPUTE ARCSIN BY USING THE IDENTITY
TCF     +2              # ARCSIN(X) = PI/2 - ARCCOS(X).

ARCCOS  CAF      LDANZIG      # (EXITS IMMEDIATELY).
TS      ESCAPE
TC      BRANCH      # TEST SIGN OF INPUT.
TCF     ACOSST      # START IMMEDIATELY IF POSITIVE.
TCF     ACOSZERO     # ARCCOS(0) = PI/2 = .25.

```

	EXTEND		# IF NEGATIVE, USE THE IDENTITY
	DCS	MPAC	# ARCCOS(X) = PI - ARCCOS(-X), FORCING
	DXCH	MPAC	# ARGUMENT POSITIVE.
	CAF	TCSUBTR	# SET EXIT TO DO ABOVE BEFROE
	XCH	ESCAPE	# ARCSIN/ARCCOS CONSIDERATIONS.
	TS	ESCAPE2	
ACOSST	CS	HALF	# TEST MAGNITUDE OF INPUT.
	AD	MPAC	
	CCS	A	
	TCF	ACOSOVF	# THIS IS PROBABLY AN OVERFLOW CASE.
LASINEX	TCF	ASINEX	
	TCF	ACOSST2	# NO OVERFLOW -- PROCEED.
	CCS	MPAC +1	# IF MAJOR PART IS .5, CALL ANSWER 0
	CAF	ZERO	# UNLESS MINOR PART NEGATIVE.
	TCF	ACOS=0	
	TCF	ACOSST2	
ACOS=0	TS	MPAC +1	
	TS	MPAC	
	TC	ESCAPE	
ACOSST2	EXTEND		# NOW THAT ARGUMENT IS IN PROPER RANGE,
	DCS	MPAC	# BEGIN COMPUTATION. USE HASTINGS
	AD	HALF	# APPROXIMATION ARCCOS(X) = SQRT(1-X)P(X)
	DXCH	MPAC	# IN A SCALED VERSION WHERE P(X) IS A
	DXCH	BUF2	# SEVENTH ORDER POLYNOMIAL.
	TC	SQRTSUB	# RETURNS WITH NORMALIZED SQUARE ROOT.
	CCS	MPTMP	# SEE IF UN-NORMALIZATION REQUIRED.
	TCF	ACOSSHR	
# Page 1190			
ACOS3	DXCH	MPAC	# SET UP FOR POLYNOMIAL EVALUATION.
	DXCH	BUF2	
	DXCH	MPAC	
	TC	POLY	
	DEC	6	
	2DEC	+.353553385	# COEFFICIENTS ARE C 2(+I)/PISQRT(2) WHERE
	2DEC*	-.0483017006 B+1*	# I

```

2DEC*  +.0200273085 B+2*      # WEHRE C STANDS FOR ORIGINAL COEFFS.
2DEC*  -.0112931863 B+3*
2DEC*  +.00695311612 B+4*
2DEC*  -.00384617957 B+5*
2DEC*  +.001501297736 B+6*
2DEC*  -.000284160334 B+7*

CAF    LBUF2      # DO FINAL MULTIPLY AND GO TO ANY
TC     DMPSUB -1  # EPILOGUE SEQUENCES.
TC     ESCAPE

SUBTR   EXTEND      # EPILOGUE FOR NEGATIVE INPUTS TO ARCCOS.
DCS    MPAC
AD     HALF        # FORMS  $\pi - \arccos(-x) = \arccos(x)$ .
DXCH   MPAC
TC     ESCAPE2     # GO TO POSSIBLE ARCSIN EPILOGUE.

ASINEX  EXTEND
DCS    MPAC        # ARCSIN EPILOGUE -- GET ARCSIN(X)
AD     QUARTER     # =  $\pi/2 - \arccos(x)$ .
DXCH   MPAC
LDANZIG TCF        DANZIG

# Page 1191
ACOSSHR INDEX  A      # THE SHIFT RIGHT IS LESS THAN 14 SINCE
CAF     BIT14      # THE INPUT WAS NON-ZERO DP.
TS      MPTMP
TC      VSHRRND    # DP SHIFT RIGHT AND ROUND.
TCF     ACOS3      # PROCEED.

ACOSOVF EXTEND      # IF MAJOR PART WAS ONLY 1 MORE THAN .5,
BZF     ACOS=0      # CALL ANSWER ZERO.

ACOSABRT TC      ALARM # IF OVERFLOW, CALL ANSWER ZERO BUT
OCT     1301        # SOUND AN ALARM.

CAF     ZERO
TCF     ACOS=0

ACOSZERO CAF     QUARTER #  $\arccos(0) = \pi/2$ .
TCF     ACOS=0 +1      # SET MPAC AND EXIT VIA ESCAPE.

NEG12   DEC      -12
TCSUBTR TCF      SUBTR

```

```
# THE FOLLOWING INSTRUCTIONS ARE AVAILABLE FOR SETTING, MODIFYING, AND BRANCHING ON
#      1.      AXT      ADDRESS TO INDEX TRUE.
#      2.      AXC      ADDRESS TO INDEX COMPLEMENTED.
#      3.      LXA      LOAD INDEX FROM ERASABLE.
#      4.      LXC      LOAD INDEX COMPLEMENTED FROM ERASABLE.
#      5.      SXA      STORE INDEX IN ERASABLE.
#      6.      XCHX     EXCHANGE INDEX REGISTER WITH ERASABLE.
#      7.      INCR     INCREMENT INDEX REGISTER.
#      8.      XAD      ERASABLE ERASABLE ADD TO INDEX REGISTER.
#      9.      XSU      ERASABLE SUBTRACT FROM INDEX REGISTER.
#     10.      TIX      BRANCH ON INDEX REGISTER AND DECREMENT.
```

```
BANK      01
```

```
COUNT     01/INTER
```

```
AXT        TC      TAGSUB      # SELECT APPROPRIATE INDEX REGISTER.
```

```
CA         POLISH
```

```
XSTORE     INDEX   INDEXLOC    # CONTAINS C(FIXLOC) OR C(FIXLOC)+1
```

```
TS         X1
```

```
TCF        DANZIG
```

```
AXC        TC      TAGSUB
```

```
CS         POLISH
```

```
TC         XSTORE
```

```
LXA        TC      15ADRERS     # LOAD INDEX REGISTER FROM ERASABLE.
```

```
INDEX      POLISH
```

```
CA         0
```

```
TCF        XSTORE
```

```
LXC        TC      15ADRERS     # LOAD NDX REG FROM ERASABLE COMPLEMENTED.
```

```
INDEX      POLISH
```

```
CS         0
```

```
TCF        XSTORE
```

```
SXA        TC      15ADRERS     # STORE INDEX REGISTER IN ERASABLE.
```

```
INDEX      INDEXLOC
```

```
CA         X1
```

```
MSTORE1    INDEX   POLISH
```

```
TS         0
```

```
TCF        DANZIG
```

```
# Page 1193
```

```
XCHX       TC      15ADRERS     # EXCHANGE INDEX REGISTER WITH ERASABLE.
```

```
INDEX      POLISH
```


	CA	0	
	INDEX	INDEXLOC	
	XCH	X1	
	TCF	MSTORE1	
XAD	TC	15ADRERS	# ADD ERASABLE TO INDEX REGISTER.
	INDEX	POLISH	
	CA	0	
XAD2	INDEX	INDEXLOC	
	ADS	X1	# IGNORING OVERFLOWS.
	TCF	DANZIG	
INCR	TC	TAGSUB	# INCREMENT INDEX REGISTER.
	CA	POLISH	
	TCF	XAD2	
XSU	TC	15ADRERS	# SUBTRACT ERASABLE FROM INDEX REGISTER.
	INDEX	POLISH	
	CS	0	
	TCF	XAD2	
TIX	TC	TAGSUB	# BRANCH AND DECREMENT ON INDEX.
	INDEX	INDEXLOC	
	CS	S1	
	INDEX	INDEXLOC	
	AD	X1	
	EXTEND		# NO OPERATION IF DECREMENTED INDEX IS
	BZMF	DANZIG	# NEGATIVE OR ZERO.
DOTIXBR	INDEX	INDEXLOC	
	XCH	X1	# IGNORING OVERFLOWS.
	TCF	GOTO	# DO THE BRANCH USING THE CADR IN POLISH.
# Page 1194			
# SUBROUTINE TO CONVERT AN ERASABLE ADDRESS (11 BITS) TO AN EBANK SETTING AND SUBADDRESS.			
15ADRERS	CS	POLISH	
	AD	DEC45	
	CCS	A	# DOES THE ADDRESS POINT TO THE WORK AREA?
	CA	FIXLOC	# YES. ADD FIXLOC. EBANK OK AS IS.
	TCF	+5	
	CA	OCT1400	# NO. SET EBANK & MAKE UP SUBADDRESS.
	XCH	POLISH	
	TS	EBANK	

		MASK	LOW8	
+5		ADS	POLISH	# FALL INTO TAGSUB, AND RETURN VIA Q.

SUBROUTINE WHICH SETS THE ADDRESS OF THE SPECIFIED INDEX IN INDEXLOC. (ACTUALLY, T

TAGSUB	CA	FIXLOC	
	TS	INDEXLOC	
	CCS	CYR	# BIT 15 SPECIFIES INDEX.
	INCR	INDEXLOC	# 0 MEANS USE X2.
	TC	Q	
	TC	Q	# 1 FOR X1.

Page 1195

MISCELLANEOUS OPERATION CODES WITH DIRECT ADDRESSES. INCLUDED HERE ARE:

#	1.	ITA	STORE CPRET (RETURN ADDRESS) IN ERASABLE.
#	2.	CALL	CALL A SUBROUTINE, LEAVING RETURN IN QPRET.
#	3.	RTB	RETURN TO BASIC LANGUAGE AT THE GIVEN ADDRESS.
#	4.	BHIZ	BRANCH IF THE HIGHORDER OF MPAC IS ZERO (SINGLE PRECISION).
#	5.	BOV	BRANCH ON OVERFLOW.
#	6.	GOTO	SIMPLE SEQUENCE CHANGE.

RTB/BHIZ	CCS	CYR	
RTB	CA	POLISH	
	TC	SWCALL -1	# SO A "TC Q" FROM ROUTINE LEADS TO DANZIG

BHIZ	CCS	MPAC	
	TCF	DANZIG	
	TCF	GOTO	
	TCF	DANZIG	
	TCF	GOTO	

BOV(B)	CCS	OVFIND	# BRANCH ON OVERFLOW TO BASIC OR INTERP.
	TCF	+2	
	TCF	DANZIG	
	TS	OVFIND	
	CCS	CYR	
	TCF	RTB	# IF BASIC.

B5TOBB	OCT	360	
	TCF	GOTO	

Page 1196

BZE/GOTO	CCS	CYR	# SEE WHICH OP-CODE IS DESIRED.
	TC	BRANCH	# DO BZE.
	TCF	DANZIG	
	TCF	GOTO	# DO GOTO.

	TCF	DANZIG		
BPL/BMN	CCS	CYR		
	TCF	BPL		
5B10	#DEC	5	B+10	# SHIFTS OP CODE IN SWITCH INSTRUCTION ADR
	DEC	5	B-4	# RSB 2009
	TC	BRANCH		# DO BMN
	TCF	DANZIG		
	TCF	DANZIG		
	TCF	GOTO		# ONLY IF NNZ.
BPL	TC	BRANCH		
	TCF	GOTO		# IF POSITIVE OR ZERO.
	TCF	GOTO		
	TCF	DANZIG		
CALL/ITA	CCS	CYR		
	TCF	CALL		
	TC	CCSHOLE		
	TC	15ADRERS		# STORE QPRET. (TAGSUB AFTER 15ADRERS IS
	INDEX	FIXLOC		# SLOW IN THIS CASE, BUT SAVES STORAGE.)
	CA	QPRET		
	TCF	MSTORE1		

Page 1197

THE FOLLOWING OPERATIONS ARE AVAILABLE FOR ALTERING AND TESTING INTERPRETATIVE SWITCHES:

#	00	BONSET	SET A SWITCH AND DO A GOTO IF IT WAS ON.
#	01	SETGO	SET A SWITCH AND DO A GOTO.
#	02	BOFSET	SET A SWITCH AND DO A GOTO IF IT WAS OFF
#	03	SET	SET A SWITCH.
#	04	BONINV	INVERT A SWITCH AND BRANCH IF IT WAS ON.
#	05	INVGO	INVERT A SWITCH AND DO A GOTO.
#	06	BOFINV	INVERT A SWITCH AND BRANCH IF IT WAS OFF
#	07	INVERT	INVERT A SWITCH.
#	10	BONCLR	CLEAR A SWITCH AND BRANCH IF IT WAS ON.
#	11	CLRGO	CLEAR A SWITCH AND DO A GOTO.
#	12	BOFCLR	CLEAR A SWITCH AND BRANCH IF IT WAS OFF.
#	13	CLEAR	CLEAR A SWITCH.
#	14	BON	BRANCH IF A SWITCH WAS ON.
#	16	BOFF	BRANCH IF A SWITCH WAS OFF.

THE ADDRESS SUPPLIED WITH THE SWITCH INSTRUCTION IS INTERPRETED AS FOLLOWS:

#	BITS 1-4	SWITCH BIT NUMBER (1-15).
#	BITS 5-8	SWITCH OPERATION NUMBER
#	BITS 9-	SWITCH WORD NUMBER (UP TO 64 SWITCH WORDS).

THE ADDRESS ITSELF IS MADE UP BY THE YUL SYSTEM ASSEMBLER. THE BRANCH INSTRUCTIONS
 # ADDRESSES, THE SECOND TAKEN AS THE DIRECT (OR INDIRECT IF IN ERASABLE) ADDRESS OF

SWITCHES	CAF	LOW4	# LEAVE THE SWITCH BIT IN SWBIT.	
	MASK	POLISH		
	INDEX	A		
	CAF	BIT15	# (NUMBER FROM LEFT TO RIGHT.)	
	TS	SWBIT		
	CAF	BIT7	# LEAVE THE SWITCH NUMBER IN SWWORD.	
	EXTEND			
	MP	POLISH		
	TS	SWWORD		
	INHINT		# DURING SWITCH CHANGE SO RUPT CAN USE TOO	
	INDEX	A	# LEAVE THE SWITCH WORD ITSELF IN L.	
	CA	STATE		
	TS	Q	# Q WILL BE USED AS A CHANNEL.	
# Page 1198	CAF	BIT11		
	EXTEND		# DISPATCH SWITCH BIT OPERATION AS IN BITS	
	MP	POLISH	# 7-8 OF POLISH.	
	MASK	B3TOB4	# GETS 4X2-BIT CODE.	
	INDEX	A		
	TCF	+1		
	+1	CA	SWBIT	# 00 -- SET SWITCH IN QUESTION.
		EXTEND		
		ROR	QCHAN	
		TCF	SWSTORE	
	+5	CA	SWBIT	# 01 -- INVERT SWITCH.
		EXTEND		
		RXOR	QCHAN	
		TCF	SWSTORE	
	+9D	CS	SWBIT	# 10 -- CLEAR.
		MASK	Q	
SWSTORE		INDEX	SWWORD	
		TS	STATE	# NEW SWITCH WORD.
# Page 1199				
	+13D	RELINT		# 11 -- NOOP.
		CAF	BIT13	
		EXTEND		# DISPATCH SEQUENCE CHANGING OR BRANCING
		MP	POLISH	# CODE.

		MASK	B3TOB4	
		INDEX	A	
		TCF	+1	# ORIGINALLY STORED IN BITS 5-6
TEST	+1	CS	Q	# 00 -- BRANCH IF ON.
		MASK	SWBIT	
		CCS	A	
		TCF	SWSKIP	
	+5	TCF	SWBRANCH	# 01 -- GO TO.
		TCF	SWSKIP	# HERE ONLY ON BIT 15.
		TC	CCSHOLE	
		TC	CCSHOLE	
	+9D	CA	Q	# 10 -- BRANCH IF OFF.
		TCF	TEST	
B3TOB4		OCT	0014	
SWSKIP		INCR	LOC	
SW/		EQUALS	SWITCHES	
	+13D	TCF	DANZIG	# 11 -- NOOP.

This code is written to file `src/INTERPRETER.s`.

B.47 INTERPRETIVE CONSTANT

```

830  <src/INTERPRETIVE-CONSTANT.s 830>≡
      # Copyright:    Public domain.
      # Filename:     INTERPRETIVE_CONSTANT.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         1100-1101
      # Mod history:   2009-05-25 RSB   Adapted from the corresponding
      #               Luminary131 file, using page
      #               images from Luminary 1A.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969

      # Page 1100

      SETLOC  INTPRET1
      BANK

      COUNT*  $$/ICONS
      DP1/4TH 2DEC   .25

      UNITZ    2DEC   0
      UNITY    2DEC   0
      UNITX    2DEC   .5
      ZEROVECS 2DEC   0

```

```

                2DEC      0

                2DEC      0

DPHALF          =        UNITX
DPPOSMAX        OCT      37777
                OCT      37777

# Page 1101
# INTERPRETIVE CONSTANTS IN THE OTHER HALF-MEMORY

                SETLOC  INTPRET2
                BANK

                COUNT*  $$/ICONS
ZUNIT           2DEC      0

YUNIT           2DEC      0

XUNIT           2DEC      .5

ZEROVEC         2DEC      0

                2DEC      0

                2DEC      0

                OCT      77777          # -0, -6, -12 MUST REMAIN IN THIS ORDER
DFC-6           DEC      -6
DFC-12          DEC      -12
LODPMAX         2OCT     3777737777    # THESE TWO CONSTANTS MUST REMAIN
LODPMAX1        2OCT     3777737777    # ADJACENT AND THE SAME FOR INTEGRATION

ZERODP          =        ZEROVEC
HALFDP          =        XUNIT

```

This code is written to file `src/INTERPRETIVE-CONSTANT.s`.

B.48 INTERPRETIVE CONSTANTS

```

832  <src/INTERPRETIVE-CONSTANTS.s 832>≡
      # Copyright:    Public domain.
      # Filename:     INTERPRETIVE_CONSTANTS.agc
      # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
      #               It is part of the source code for the Command Module's (CM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         1205-1206
      # Mod history:   2009-05-13 RSB   Adapted from the Colossus249/ file of the
      #               same name, using Comanche055 page images.
      #
      # This source code has been transcribed or otherwise adapted from digitized
      # images of a hardcopy from the MIT Museum. The digitization was performed
      # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
      # thanks to both. The images (with suitable reduction in storage size and
      # consequent reduction in image quality as well) are available online at
      # www.ibiblio.org/apollo. If for some reason you find that the images are
      # illegible, contact me at info@sandroid.org about getting access to the
      # (much) higher-quality images which Paul actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 055 of AGC program Comanche by NASA
      #       2021113-051.  10:28 APR. 1, 1969
      #
      #       This AGC program shall also be referred to as
      #               Colossus 2A

      # Page 1205

      SETLOC  INTPRET1
      BANK

      COUNT   23/ICONS

      DP1/4TH  2DEC   .25

      UNITZ    2DEC   0

      UNITY    2DEC   0

      UNITX    2DEC   .5

```


ZEROVECS	2DEC	0
	2DEC	0
	2DEC	0

DPHALF	=	UNITX
DPPOSMAX	OCT	37777
	OCT	37777

Page 1206

INTERPRETIVE CONSTANTS IN THE OTHER HALF-MEMORY

	SETLOC	INTPRET2	
	BANK		
	COUNT	14/ICONS	
ZUNIT	2DEC	0	
YUNIT	2DEC	0	
XUNIT	2DEC	.5	
ZEROVEC	2DEC	0	
	2DEC	0	
	2DEC	0	
	OCT	77777	# -0, -6, -12 MUST REMAIN IN THIS ORDER
DEC-6	DEC	-6	
DEC-12	DEC	-12	
LODPMAX	2OCT	3777737777	# THESE TWO CONSTANTS MUST REMAIN
LODPMAX1	2OCT	3777737777	# ADJACENT AND THE SAME FOR INTEGRATION
ZERODP	=	ZEROVEC	
HALFDP	=	XUNIT	

This code is written to file `src/INTERPRETIVE-CONSTANTS.s`.

B.49 INTERRUPT LEAD INS

```

834  <src/INTERRUPT-LEAD-INS.s 834>≡
      # Copyright:    Public domain.
      # Filename:     INTERRUPT_LEAD_INS.agc
      # Purpose:      Part of the source code for Comanche, build 055.
      #               It is part of the source code for the Command Module's (CM)
      #               Apollo Guidance Computer (AGC), Apollo 11.
      # Assembler:    yaYUL
      # Reference:     pp. 131-132
      # Contact:       Ron Burkey <info@sandroid.org>,
      #               Fabrizio Bernardini <fabrizio@spacecraft.it>
      # Website:       http://www.ibiblio.org/apollo.
      # Mod history:   09/05/09 FB      Transcription of Batch FB-1 Assignment.
      #
      # The contents of the "Comanche055" files, in general, are transcribed
      # from scanned documents.
      #
      #       Assemble revision 055 of AGC program Comanche by NASA
      #       2021113-051.  April 1, 1969.
      #
      #       This AGC program shall also be referred to as Colossus 2A
      #
      #       Prepared by
      #
      #               Massachussets Institute of Technology
      #               75 Cambridge Parkway
      #               Cambridge, Massachusetts
      #
      #       under NASA contract NAS 9-4065.
      #
      # Refer directly to the online document mentioned above for further
      # information. Please report any errors to info@sandroid.org.

      # Page 131

      SETLOC  4000

      COUNT   02/RUPTS

      INHINT                                # GO
      CAF     GOBB
      XCH     BBANK
      TCF     GOPROG

      DXCH    ARUPT                        # T6RUPT
      EXTEND
      DCA     T6LOC

```

DTCB

DXCH	ARUPT	# T5RUPT
CS	TIME5	
AD	.5SEC	
TCF	T5RUPT	

DXCH	ARUPT	# T3RUPT
CAF	T3RPTBB	
XCH	BBANK	
TCF	T3RUPT	

DXCH	ARUPT	# T4RUPT
CAF	T4RPTBB	
XCH	BBANK	
TCF	T4RUPT	

DXCH	ARUPT	# KEYRUPT1
CAF	KEYRPTBB	
XCH	BBANK	
TCF	KEYRUPT1	

DXCH	ARUPT	# KEYRUPT2
CAF	MKRUPTBB	
XCH	BBANK	
TCF	MARKRUPT	

DXCH	ARUPT	# UPRUPT
CAF	UPRPTBB	
XCH	BBANK	
TCF	UPRUPT	

DXCH	ARUPT	# DOWNRUPT
CAF	DWNRPTBB	
XCH	BBANK	
TCF	DODOWNTM	

DXCH	ARUPT	# RADAR RUPT
CAF	RDRPTBB	
XCH	BBANK	
TCF	VHFREAD	

Page 132

DXCH	ARUPT	# HAND CONTROL RUPT
CAF	HCRUPTBB	
XCH	BBANK	

	TCF	RESUME +3	# NOT USED
GOBB	EBANK=	LST1	# RESTART USES E0,E3
	BBCON	GOPROG	
T3RPTBB	EBANK=	LST1	
	BBCON	T3RUPT	
KEYRPTBB	EBANK=	KEYTEMP1	
	BBCON	KEYRUPT1	
MKRPTBB	EBANK=	MRKBUF1	
	BBCON	MARKRUPT	
UPRPTBB	=	KEYRPTBB	
DWNRPTBB	EBANK=	DNTMBUFF	
	BBCON	DODOWNTM	
RDRPTBB	EBANK=	DATATEST	
	BBCON	VHFREAD	
HCRUPTBB	EBANK=	TIME1	# NOT USED
	BBCON	RESUME	
T4RPTBB	EBANK=	DSRUPTSW	
	BBCON	T4RUPT	
T5RPTBB	EBANK=	TIME1	
	BBCON	T5RUPT	
T5RUPT	EXTEND		
	BZMF	NOQBRSM	
	EXTEND		
	DCA	T5LOC	
	DTCB		

This code is written to file `src/INTERRUPT-LEAD-INS.s`.

B.50 JET SELECTION LOGIC

```

837  <src/JET-SELECTION-LOGIC.s 837>≡
    # Copyright:    Public domain.
    # Filename:     JET_SELECTION_LOGIC.agc
    # Purpose:      Part of the source code for Colossus 2A, AKA Comanche 055.
    #               It is part of the source code for the Command Module's (CM)
    #               Apollo Guidance Computer (AGC), for Apollo 11.
    # Assembler:    yaYUL
    # Contact:       Ron Burkey <info@sandroid.org>.
    # Website:       www.ibiblio.org/apollo.
    # Pages:         1039-1062
    # Mod history:   2009-05-13 RSB   Adapted from the Colossus249/ file of the
    #               same name, using Comanche055 page images.
    #
    # This source code has been transcribed or otherwise adapted from digitized
    # images of a hardcopy from the MIT Museum. The digitization was performed
    # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
    # thanks to both. The images (with suitable reduction in storage size and
    # consequent reduction in image quality as well) are available online at
    # www.ibiblio.org/apollo. If for some reason you find that the images are
    # illegible, contact me at info@sandroid.org about getting access to the
    # (much) higher-quality images which Paul actually created.
    #
    # Notations on the hardcopy document read, in part:
    #
    #       Assemble revision 055 of AGC program Comanche by NASA
    #       2021113-051.  10:28 APR. 1, 1969
    #
    #       This AGC program shall also be referred to as
    #               Colossus 2A
    #
    # Page 1039
    BANK      21
    SETLOC    DAPS4
    BANK
    COUNT     17/DAPJS
    EBANK=     KMPAC
    # EXAMINE CHANNEL 31 FOR TRANSLATION COMMANDS
    JETSLECT   LXCH   BANKRUPT
    CAF        DELTATT3      # = 60 MS  RESET TO EXECUTIVE PHASE1
    AD         T5TIME

```

```

TS      TIME5
TCF     +3
CAF     DELATT20      # = 20 MS  TO ASSURE A T5RUPT
TS      TIME5
CAF     =14MS         # RESET T6 TO INITIALIZE THE JET CHANNELS
TS      TIME6         # IN 14 MS
CAF     NEGMAX
EXTEND
WOR     CHAN13
EXTEND
QXCH    QRUPT
CAF     XLNMASK       # = 7700 OCT
EXTEND  # EXAMINE THE TRANSLATION
RXOR    CHAN31        # HAND CONTROLLER
MASK    XLNMASK
EXTEND
BZF     NOXLNCMD
TS      T5TEMP
EXTEND
MP      BIT9
MASK    THREE
TS      XNDX1         # AC QUAD  X-TRANSLATION INDEX
TS      XNDX2         # BD QUAD  X-TRANSLATION INDEX
CA      T5TEMP
EXTEND  # 1 = + XLN
MP      BIT7          # 2 = - XLN
MASK    THREE         # 3 = NO XLN
TS      YNDX          # Y-TRANSLATION INDEX

CA      T5TEMP
EXTEND
MP      BIT5
MASK    THREE
TS      ZNDX          # Z-TRANSLATION INDEX

CA      DAPDATR1      # SET ATTKALMN TO PICK UP FILTER GAINS FOR
MASK    BIT14         # TRANSLATIONS.
EXTEND  # CHECK DAPDATR1 BIT 14 FOR LEM ATTACHED.

BZF     NOLEM
CS      THREE         # IF LEM IS ON, SET ATTKALMN = -3
TCF     +2
CS      TWO           # IF LEM IS OFF, SET ATTKALMN = -2.
TS      ATTKALMN
CCS     XTRANS        # (+, -1, 0)
TS      XNDX1         # USING BD-X  ZERO XNDX1

```

Page 1040

NOLEM

	TCF	PWORD	
	TS	XNDX2	# USING AC-X ZERO XNDX2
	TCF	PWORD	
XLNMASK	OCT	7700	
DELTATT3	DEC	16378	# = 60 MS
DELATT20	DEC	16382	# = 20 MS
NOXLNCMD	TS	XNDX1	# ZERO ALL REQUESTS FOR TRANSLATION
	TS	XNDX2	
	TS	YNDX	
	TS	ZNDX	

PITCH COMMANDS TIMING(NO X-TRANS, NO QUAD FAILS) 32MCT

PWORD	CCS	TAU1	# CHECK FOR PITCH COMMANDS
	CAF	ONE	
	TCF	+2	# 0 = NO PITCH
	CAF	TWO	# +1 = + PITCH
	TS	PINDEX	# +2 = - PITCH
	CCS	RACFAIL	# FLAG FOR REAL AC QUAD FAILURES
	TCF	AFAILP	
	TCF	TABPCOM	# 0 = NO REAL AC FAILURES
	TCF	CFAILP	# + = A QUAD FAILED
	TCF	TABPCOM	# - = C QUAD FAILED
			# IF FAILURES ARE PRESENT IGNORE
			# X-TRANSLATIONS ON THIS AXIS
AFAILP	CAF	NINE	# IF FAILURE IS PRESENT 1JET OPERATION
	TCF	TABPCOM +2	# IS ASSUMED. IGNORE X-TRANSLATION
CFAILP	CAF	TWELVE	
	TCF	TABPCOM +2	
XLNNDX	DEC	0	# INDICES FOR TRANSLATION COMMANDS
	DEC	3	# FOR USE IN TABLE LOOK UP
	DEC	6	
	DEC	0	
TWELVE	=	OCT14	

TABLE LOOK UP FOR PITCH COMMANDS WITH AND WITHOUT X-TRANSLATION AND AC QUAD FAILURES PRESENT.
 # BITS 9, 10 CONTAIN THE NUMBER OF PITCH JETS USED TO PERFORM THE PITCH ROTATION
 # Page 1041

TABPCOM	INDEX	XNDX1
---------	-------	-------

CA	XLNNDX	
AD	PINDEX	
INDEX	A	
CA	PYTABLE	
MASK	PJETS	# =1417 OCT
TS	PWORD1	
EXTEND		
MP	BIT7	
TS	NPJETS	# = NO. OF PITCH JETS

YAW JET COMMANDS TIMING(N X-TRANS, NO QUAD FAILURES) 32MCT

YWORD	CCS	TAU2	# CHECK FOR YAW COMMANDS
	CAF	ONE	
	TCF	+2	
	CAF	TWO	
	TS	YINDEX	# YAW ROTATION INDEX
	CCS	RBDFAIL	# FLAG FOR B OR D QUAD FAILURES
	TCF	BFAILY	# 0 = NO BD FAILURE
	TCF	TABYCOM	# + - B QUAD FAILED
	TCF	DFAILY	# - = D QUAD FAILED
	TCF	TABYCOM	
BFAILY	CAF	NINE	
	TCF	TABYCOM +2	
DFAILY	CAF	TWELVE	
	TCF	TABYCOM +2	

Page 1042

TABLE FOR PITCH(YAW) COMMANDS

BITS 4,3,2,1 = PITCH, X-TRANSLATION JETS SELECTED

BITS 10,9 = NO. PITCH JETS USED TO PERFORM ROTATION

BITS 8,7,6,5 = YAW, X-TRANSLATION JETS SELECTED

BITS 12,11: NO. YAW JETS USED TO PERFORM ROTATION

			# ROT	TRANS	QUAD	BIAS
PYTABLE	OCT	0	# 0	0		0
	OCT	5125	# +	0		0
	OCT	5252	# -	0		0
	OCT	0231	# 0	+		3
	OCT	2421	# +	+		3
	OCT	2610	# -	+		3
	OCT	0146	# 0	-		6
	OCT	2504	# +	-		6
	OCT	2442	# -	-		6

July 12, 2016

841

OCT	0	# 0	A(B)	9
OCT	2421	# +	A(B)	9
OCT	2442	# -	A(B)	9
OCT	0	# 0	C(D)	12
OCT	2504	# +	C(D)	12
OCT	2610	# -	C(D)	12

MASKS FOR PITCH AND YAW COMMANDS

PJETS	OCT	1417
YJETS	OCT	6360

TABLE LOOK UP FOR YAW COMMANDS WITH AND WITHOUT X-TRANSLATION AND AC QUAD FAILURES PRESENT
BITS 11, 12 CONTAIN THE NUMBER OF YAW JETS USED TO PERFORM THE YAW ROTATION

TABYCOM	INDEX	XNDX2	
	CA	XLNNDX	
	AD	YINDEX	
	INDEX	A	
	CA	PYTABLE	
	MASK	YJETS	# = 6360 OCT
	TS	YWORD1	
	EXTEND		
	MP	BIT5	
	TS	NYJETS	# NO. OF YAW JETS USED TO PERFORM ROTATION

Page 1043

ROLL COMMANDS TIMING(NO Y,Z TRANS, NO QUAD FAILS) 45MCT

RWORD	CCS	TAU	# CHECK FOR ROLL COMMANDS
	CAF	ONE	
	TCF	+2	
	CAF	TWO	
	TS	RINDEX	
	CCS	ACORBD	# FLAG FOR AC OR BD QUAD SELECTION FOR
	TCF	BDROLL	# ROLL COMMANDS
	TCF	BDROLL	# +, +0 = BD ROLL
	TCF	+1	# -, -0 = AC ROLL
ACROLL	CCS	RACFAIL	# CHECK FOR REAL FAILURES
	TCF	RAFAIL	# ON AC QUADS
	TCF	RXLNS	
	TCF	RCFAIL	
	TCF	RXLNS	

RAFAIL	CAF	NINE	# QUAD FAILURE WILL GET
	TCF	TABRCOM	# 1-JET OPERATION
RCFAIL	CAF	TWELVE	
	TCF	TABRCOM	
XLN1NDX	DEC	0	
	DEC	1	# INDICES FOR TRANSLATION
	DEC	2	
	DEC	0	

TABLE LOOK UP FOR AC-ROLL COMMANDS WITH AND WITHOUT Y-TRANSLATION AND ACQUAD FAILURE
 # BITS 9,10,11 CONTAIN THE MAGNITUDE AND DIRECTION OF THE ROLL

RXLNS	INDEX	YNDX	# NO AC QUAD FAILURES
	CA	XLNNDX	# INCLUDE +,-,0, Y-TRANSLATION
TABRCOM	AD	RINDEX	
	INDEX	A	
	CA	RTABLE	
	MASK	ACRJETS	# = 3760 OCT
	TS	RWORD1	

CHECK FOR Z-TRANSLATIONS ON BD

BDZCHECK	CA	ZNDX	
	EXTEND		
	BZMF	NOBDZ	# NO Z-TRANSLATION

Page 1044

TABLE LOOK UP FOR BD Z-TRANSLATION WITH AND WITHOUT REAL BD QUAD FAILURES. Z-TRANSLATION
 # IS VALID AS LONG AS ROLL COMMANDS CAN BE SATISFIED WITH THE AC ROLL JETS. CRITERION:
 # COMMANDS = 0 (WITH Z-TRANSLATION) AND IF TAU = 0, THEN INCLUDE THE BD Z-TRANSLATION
 # ROLL COMMAND = 0, AND IF TAU NZ, THEN IGNORE THE BD Z-TRANSLATION

CCS	RBDFAIL	
CAF	THREE	
TCF	+2	
CAF	SIX	
INDEX	ZNDX	
AD	XLN1NDX	
INDEX	A	
CA	YZTABLE	
MASK	BDZJETS	# = 3417 OCT
AD	RWORD1	# ADD TO ROLL COMMANDS
TS	T5TEMP	# IF POSSIBLE. MUST CHECK TAU FIRST
EXTEND		

July 12, 2016

843

```

MP      BIT7      # DETERMINE THE NET ROLL COMMAND WITH
AD      =-4       # Z-TRANSLATION ADDED ON
TS      NRJETS    # NET NO. OF +,- ROLL JETS ON
EXTEND
BZF     TAUCHECK

ACRBDZ   CA      T5TEMP    # Z-TRANSLATION ACCEPTED EVEN THO WE MAY
TS      RWORD1   # HAVE INTRODUCED AN UNDESIRABLE ROLL
TCF     ROLLTIME  # BRANCH TO JET ON-TIME CALCULATIONS

TAUCHECK CCS     TAU
TCF     NOBDZ
TCF     ACRBDZ
TCF     NOBDZ
TCF     ACRBDZ

NOBDZ    CA      RWORD1    # Z-TRANSLATION NOT ACCEPTED
EXTEND
MP      BIT7
AD      =-2
TS      NRJETS
TCF     ROLLTIME    # BRANCH TO JET ON-TIME CALCULATION

# Page 1045
# BD QUAD SELECTION FOR ROLL COMMANDS

BDROLL   CCS     RBDFAIL
TCF      RBDFAIL
TCF      RZXLNS
TCF      RDBFAIL
TCF      RZXLNS
RBFAIL   CAF     NINE
TCF      TABRZCMD
RDBFAIL  CAF     TWELVE
TCF      TABRZCMD

RZXLNS   INDEX   ZNDX      # NO BD FAILURES
CA       XLNNDX   # +,-,0 Z-TRANSLATION PRESENT
TABRZCMD AD      RINDEX
INDEX    A
CA       RTABLE
MASK     BDRJETS    # = 34017 OCT
TS       RWORD1

ACYCHECK CA      YNDX      # ANY Y-TRANSLATION
EXTEND
```

	BZF	NOACY	# NO Y-TRANSLATION
	CCS	RACFAIL	
	CAF	THREE	
	TCF	+2	
	CAF	SIX	
	INDEX	YNDX	
	AD	XLN1NDX	
	INDEX	A	
	CA	YZTABLE	
	MASK	ACYJETS	# = 34360 OCT
	AD	RWORD1	
	TS	T5TEMP	
	EXTEND		# FOR EXPLANATION SEE CODING ON RTABLE
	MP	BIT4	
	AD	=-4	
	TS	NRJETS	# NO. OF NET ROLL JETS
	EXTEND		
	BZF	TAUCHCK	# IF NRJETS = 0
BDRACZ	CA	T5TEMP	# Y-TRANSLATION ACCEPTED
	TS	RWORD1	
	TCF	ROLLTIME	# BRANCH TO JET ON-TIME CALCULATIONS
TAUCHCK	CCS	TAU	
	TCF	NOACY	
	TCF	BDRACZ	
	TCF	NOACY	
	TCF	BDRACZ	
# Page 1046			
NOACY	CA	RWORD1	# Y-TRANSLATION NOT ACCEPTED
	EXTEND		
	MP	BIT4	
	AD	=-2	
	TS	NRJETS	
	TCF	ROLLTIME	
# Page 1047			
#			TABLE FOR ROLL, Y AND Z-TRANSLATION COMMANDS
#			
#			# EITHER AC OR BD ROLL MAY BE SELECTED. IF AC ROLL IS SELECTED, Y-TRANSLATIONS MAY BE
#			# PROVIDED THAT THERE ARE NO AC QUAD FAILURES. IF THERE ARE AC FAILURES, Y-TRANSLATIONS
#			# IN WHICH CASE THE ASTRONAUT SHOULD SWITCH TO BD ROLL.
#			
#			# IF BDROLL IS SELECTED, Z-TRANSLATIONS MAY BE SATISFIED SIMULTANEOUSLY PROVIDED THAT
#			# FAILURES. IF THERE ARE BD FAILURES, Z-TRANSLATION COMMANDS WILL BE IGNORED, IN WH

SWITCH TO AC ROLL.

#

NOTE THAT IF ONE QUAD FAILS (E.G. B FAILED), Z-TRANSLATION IS STILL POSSIBLE AND THAT THE UND
INTRODUCED BY THIS TRANSLATION WILL BE COMPENSATED BY THE TWO AC ROLL JETS ACTUATED BY THE AU

#

WORD MAKE UP...RTABLE

#

TWO WORDS, CORRESPONDING TO AC OR BD ROLL SELECTION, HAVE BEEN COMBINED INTO ONE TABLE. THE
TO AC ROLL HAS THE FOLLOWING INTERPRETATION:

#

BITS 9,10,11 ARE CODED TO GIVE THE NET ROLL TORQUE FOR THE WORD SELECTED. THE CODING I

#

BIT NO. 11 10 9 NO. OF ROLL JETS

#

0 0 0 -2

0 0 1 -1

0 1 0 0

0 1 1 +1

1 0 0 +2

#

THIS WORD MAY THEN BE ADDED TO THE WORD SELECTED FROM THE YZ-TRANSLATION TABLE, WHICH HAS THE
CODING AS ABOVE, AND THE NET ROLL DETERMINED BY SHIFTING THE RESULTANT WORD RIGHT 8 PLACES AN

#

THE WORD CORRESPONDING TO THE BD ROLL HAS A SIMILAR INTEPRETATION, EXCEPT THAT BITS 12, 13, 1
(AS ABOVE) TO GIVE THE NET ROLL TORQUE.

			# ROLL	TRANS	QUADFAIL	BIAS
RTABLE	OCT	11000	# 0			0
	OCT	22125	# +			0
	OCT	00252	# -			0
	OCT	11231	# 0	+Y(+Z)		3
	OCT	15421	# +	+Y(+Z)		3
	OCT	04610	# -	+Y(+Z)		3
	OCT	11146	# 0	-Y(-Z)		6
	OCT	15504	# +	-Y(-Z)		6
	OCT	04442	# -	-Y(-Z)		6
	OCT	11000	# 0		A(B)	9
	OCT	15504	# +		A(B)	9
	OCT	04610	# -		A(B)	9
	OCT	11000	# 0		C(D)	12
	OCT	15421	# +		C(D)	12
	OCT	04442	# -		C(D)	12

Page 1048

RTABLE MASKS:

```
# THE ROTATION COMMANDS (TAU'S), WHICH WERE DETERMINED FROM THE JET SWITCHING LOGIC (C
```

```

# OPERATION, MUST NOW BE UPDATED BY THE ACTUAL NUMBER OF JETS TO BE USED IN SATISFYING THESE CO
# ALSO BE DECREMENTED ACCORDING TO THE EXPECTED TORQUE GENERATED BY THE NEW COMMANDS ACTING OVE
# INTERVAL.
#
# IN ORDER TO MAINTAIN ACCURATE KNOWLEDGE OF VEHICLE ANGULAR RATES, WE MUST ALSO PROVIDE EXPECT
# (DFT'S, ALSO IN TERMS OF 1-JET OPERATION) FOR THE RATE FILTER.
#
# NOTE THAT TRANSLATIONS CAN PRODUCE ROTATIONS EVEN THOUGH NO ROTATIONS WERE CALLED FOR.  NEVER
# UPDATE DFT.
#
# WHEN THE ROTATIONS HAVE FINISHED, WE MUST PROVIDE CHANNEL INFORMATION TO THE T6 PROGRAM TO CO
# THE TRANSLATIONS.  THIS WILL BE DONE IN THE NEXT SECTION.  HOWEVER, TO INSURE THAT JETS ARE N
# THAN A MINIMUM IMPULSE (14MS), ALL JET CHANNEL COMMANDS WILL BE HELD FIXED FROM THE START OF
# AT LEAST 14MS UNTIL THE INITIALIZATION OF NEW COMMANDS.  MOREOVER, A 14MS ON-TIME WILL BE ADD
# COMMANDS GENERATED BY THE MANUAL CONTROLS OR THE JET SWITCHING LOGIC, AND ALL TRANSLATION COM
# ACTIVE FOR AT LEAST ONE CYCLE OF THE T5 PROGRAM (.1SEC)

# PITCH JET ON-TIME CALCULATION

PITCHTIM      CCS      TAU1
              TCF      PTAUPOS
              TCF      +2
              TCF      PTAUNEG
              TS       DFT1          # NO PITCH ROTATION
              TCF      PBYPASS      # COMMANDS

PTAUNEG       CS       NPJETS
              TS       NPJETS
PTAUPOS       CA       TAU1
              EXTEND
              INDEX    NPJETS
              MP       NJET
              TS       BLAST1
              AD       ==-.1SEC
              EXTEND
              BZMF     AD14MSP
              INDEX    NPJETS
              CA       DFTMAX      # THE PITCH ON-TIME IS GREATER THAN .1 SEC
              TS       DFT1
              COM
              ADS      TAU1          # UPDATE TAU1
              CAF      ==+.1SEC     # LIMIT THE LENGTH OF PITCH ROTATION
              TS       BLAST1      # COMMANDS TO 0.1 SEC SO THAT ONLY
              TCF      ASMBLWP     # X-TRANSLATIONS WILL CONTINUE ON SWITCH
              # OVER TO TVC
AD14MSP       CS       BLAST1      # SEE IF JET ON TIME IS LESS THAN

```

```

                                AD      =14MS      # MINIMUM IMPULSE TIME
                                EXTEND
                                BZMF     PBLASTOK    # IF SO LIMIT MINIMUM ON TIME TO 14 MS
                                CAF      =14MS

# Page 1051
PBLASTOK                      TS      BLAST1
                                CA      BLAST1
                                EXTEND
                                MP      NPJETS
                                LXCH    DFT1
                                TS      TAU1
                                TCF     ASMBLWP
                                # THE PITCH COMMANDS WILL BE COMPLETED
                                # WITHIN THE TS-CYCLE TIME
                                # FOR USE IN UPDATING RATE FILTER
                                # ZERO TAU1 (ACC CONTAINS ZERO)

# Page 1052
# YAW JET ON-TIME CALCULATION

YAWTIME                      CCS      TAU2
                                TCF     YTAUPOS
                                TCF     +2
                                TCF     YTAUNEG
                                TS      DFT2
                                TCF     YBYPASS    # NO YAW ROTATION COMMANDS

YTAUNEG                      CS      NYJETS
                                TS      NYJETS
YTAUPOS                      CA      TAU2
                                EXTEND
                                INDEX   NYJETS
                                MP      NJET
                                TS      BLAST2
                                AD      =-.1SEC
                                EXTEND
                                BZMF     AD14MSY
                                INDEX   NYJETS
                                CA      DFTMAX    # YAW COMMANDS WILL LAST LONGER THAN .1SEC
                                TS      DFT2
                                COM
                                ADS     TAU2      # DECREMENT TAU2
                                CAF     =+.1SEC   # LIMIT THE LENGTH OF YAW ROTATION COMMAND
                                TS      BLAST2    # TO 0.1 SEC SO THAT ONLY X-TRANSLATION
                                TCF     ASMBLWY   # WILL CONTINUE ON SWITCH OVER TO TVC

AD14MSY                      CS      BLAST2      # SEE IF JET ON-TIME LESS THAN
                                AD      =14MS     # MINIMUM IMPULSE TIME
                                EXTEND
                                BZMF     YBLASTOK  # IF SO, LIMIT MINIMUM ON-TIME TO 14 MS

```


July 12, 2016

849

```

CAF      =14MS
TS       BLAST2
YBLASTOK CA      BLAST2      # YAW COMMANDS WILL BE COMPLETED WITHIN
EXTEND   # THE T5CYCLE TIME
MP       NYJETS
LXCH     DFT2
TS       TAU2      # ZERO TAU2
TCF      ASMBLWY

# Page 1053
# ROLL ON-TIME CALCULATION:

ROLLTIME CCS      TAU
TCF      RBLAST
TCF      +2
TCF      RBLAST
INDEX    NRJETS
CA       DFTMAX    # UPDATE DFT EVEN THO NO ROLL COMMANDS ARE
TS       DFT       # PRESENT
TCF      RBYPASS

DEC      -480      # =-.3SEC
DEC      -320      # =-.2SEC
= -.1SEC DEC      -160      # =-.1SEC
DFTMAX   DEC      0       # 0
=+.1SEC  DEC      160      # =+.1SEC
DEC      320       # =+.2SEC
DEC      480       # =+.3SEC
=14MS    DEC      23       # =14MS

RBLAST   CA       TAU
EXTEND
INDEX    NRJETS
MP       NJET
TS       BLAST     # BLAST IS AN INTERMEDIATE VARIABLE
          # USED IN DETERMINING THE JET ON-TIMES

AD       =-.1SEC
EXTEND
BZMF     AD14MSR
INDEX    NRJETS    # THE ROLL ROTATION WILL LAST LONGER
CA       DFTMAX    # THAN THE T5 CYCLE TIME
TS       DFT
COM
ADS      TAU
CAF      =+.1SEC   # LIMIT THE LENGTH OF ROLL ROTATION
TS       BLAST     # COMMANDS TO 0.1 SEC SO THAT ONLY Y-Z
```

	TCF	ASMBLWR	# TRANSLATION COMMANDS CONTINUE
AD14MSR	CS	BLAST	# SEE IF THE JET ON-TIME LESS THAN
	AD	=14MS	# MINIMUM IMPULSE TIME
	EXTEND		
	BZMF	RBLASTOK	
	CAF	=14MS	# IF SO, LIMIT MINIMUM ON-TIME TO 14 MS
	TS	BLAST	
RBLASTOK	CA	BLAST	
	EXTEND		
	MP	NRJETS	
	LXCH	DFT	
	TS	TAU	# ZERO TAU
	TCF	ASMBLWR	

Page 1054

	DEC	-.333333	# = -1/3
	DEC	-.500000	# = -1.2
	DEC	-.999999	# = -1 (NEGMAX)
NJET	DEC	0	
	DEC	.999999	# = +1 (POS MAX)
	DEC	.500000	# = +1/2
	DEC	.333333	# = +1/3

Page 1055

WHEN THE ROTATION COMMANDS ARE COMPLETED, IT IS NECESSARY TO REPLACE THESE COMMANDS
CONTINUE ON WITH THE TRANSLATIONS IF ANY ARE PRESENT.

#

IN THIS SECTION THESE NEW COMMANDS ARE GENERATED AND STORED FOR REPLACEMENT OF THE
CORRESPONDING ROTATIONS ARE COMPLETED.

#

GENERATION OF THE SECOND PITCH(X-TRANS) WORD...PWORD2

ASMBLWP	CCS	RACFAIL	
	TCF	FPX2	# IF FAILURE ON AC IGNORE X-TRANSLATION
	TCF	+2	
	TCF	FPX2	
	INDEX	XNDX1	
	CA	XLNNDX	
	INDEX	A	
FPX2	CA	PYTABLE	
	MASK	PJETS	
	TS	PWORD2	
	TCF	YAWTIME	
PBYPASS	CA	PWORD1	# THE T6 PROGRAM WILL LOAD PWORD2

TS	PWORD2	# UPON ENTRY
CAF	ZERO	
TS	BLAST1	# THERE IS NO PWORD2
TCF	YAWTIME	

Page 1056

GENERATION OF THE SECOND ROLL (Y,Z) WORD (RWORD2)

ASMBLWR	CCS	YNDX	# CHECK FOR Y-TRANS
	TCF	ACBD2Y	
NO2Y	CAF	ZERO	
	TS	RWORD2	
	CCS	ZNDX	# CHECK FOR Z-TRANS
	TCF	ACBD2Z	
NO2Z	CAF	ZERO	
	ADS	RWORD2	
	TCF	PITCHTIM	# RWORD2 ASSEMBLED
ACBD2Y	CCS	ACORBD	
	TCF	AC2Y	# CAN DO Y-TRANS
	TCF	AC2Y	
	TCF	+1	# USING AC FOR ROLL
	CCS	RACFAIL	
	TCF	NO2Y	# USING AC AND AC HAS FAILED
	TCF	+2	
	TCF	NO2Y	# DITTO
	INDEX	YNDX	# NO FAILURES, CAN DO Y
	CA	XLNNDX	
	INDEX	A	
	CA	RTABLE	
	MASK	ACRJETS	
	TCF	NO2Y +1	
AC2Y	CCS	RACFAIL	
	CAF	THREE	
	TCF	+2	
	CAF	SIX	
	INDEX	YNDX	
	AD	XLN1NDX	
	INDEX	A	
	CA	YZTABLE	
	MASK	ACYJETS	
	TS	RWORD2	
	EXTEND		
	MP	BIT4	

```

# Page 1057
ACBD2Z      AD      =-2
            TS      NRJETS
            CS      BLAST
            AD      =+.1SEC
            EXTEND
            MP      NRJETS
            CA      L
            ADS     DFT
            TCF     NO2Y      +2

            CCS     ACORBD
            TCF     BDF2Z      # USING BD-ROLL
            TCF     BDF2Z      # MUST CHECK FOR BD FAILURES
            TCF     +1
            CCS     RBDFAIL    # USING AC FOR ROLL, CAN DO Z-TRANS
            CAF     THREE
            TCF     +2
            CAF     SIX
            INDEX   ZNDX
            AD      XLN1NDX
            INDEX   A
            CA      YZTABLE
            MASK    BDZJETS
            ADS     RWORD2
            EXTEND
            MP      BIT7
            AD      =-2
            TS      NRJETS
            CS      BLAST
            AD      =+.1SEC
            EXTEND
            MP      NRJETS
            CA      L
            ADS     DFT
            TCF     PITCHTIM

BDF2Z      CCS     RBDFAIL
            TCF     NO2Z      # USING BD-ROLL AND BD HAS FAILED
            TCF     +2
            TCF     NO2Z      # DITTO
            INDEX   ZNDX
            CA      XLNNDX
            INDEX   A
            CA      RTABLE
            MASK    BDRJETS
            TCF     NO2Z +1

```

```

RBYPASS      CA      RWORD1
              TS      RWORD2
              CAF      ZERO
              TS      BLAST
              TCF      PITCHTIM

```

Page 1058

GENERATION OF THE SECOND YAW (X-TRANS) WORD...YWORD2

```

ASMBLWY      CCS      RBDFAIL
              TCF      FYX2          # IF FAILURE ON BD IGNORE X-TRANSLATION
              TCF      +2
              TCF      FYX2
              INDEX    XNDX2
              CA      XLNNDX
              INDEX    A
FYX2          CA      PYTABLE
              MASK     YJETS
              TS      YWORD2
              TCF      T6SETUP

YBYPASS      CA      YWORD1
              TS      YWORD2
              CAF      ZERO
              TS      BLAST2

```

Page 1059

```

#
#                                     SORT THE JET ON-TIMES
#
# AT THIS POINT ALL THE CHANNEL COMMANDS AND JET ON-TIMES HAVE BEEN DETERMINED.  IN SUMMARY THE
#
#      RWORD1
#      RWORD2          BLAST
#
#      PWORD1
#      PWORD2          BLAST1
#
#      YWORD1
#      YWORD2          BLAST2
#
# IN THIS SECTION THE JET ON-TIMES ARE SORTED AND THE SEQUENCE OF T6 INTERRUPTS IS DETERMINED.
# THE SORTING PROCESS AND THE T6 PROGRAM, THE VARIABLES BLAST, BLAST1, BLAST2, ARE RESERVED AS
# WORDS.  THE LOWER PART OF THESE WORDS CONTAIN A BRANCH INDEX ASSOCIATED WITH THE ROTATION AXI
# ORDER WORD.

```

T6SETUP	CAF	ZERO	# BRANCH INDEX FOR ROLL
	TS	BLAST +1	
	CAF	FOUR	# BRANCH INDEX FOR PITCH
	TS	BLAST1 +1	
	CAF	ELEVEN	# BRANCH INDEX FOR YAW
	TS	BLAST2 +1	
	CS	BLAST	
	AD	BLAST1	
	EXTEND		
CHECKT23	BZMF	DXCHT12	# T1 OR T2
	CS	BLAST1	
	AD	BLAST2	
	EXTEND		
CALCDT6	BZMF	DXCHT23	
	CS	BLAST1	
	ADS	BLAST2	
	CS	BLAST	
	ADS	BLAST1	# END OF SORTING PROCEDURE
	EXTEND		# RESET T5LOC TO BEGIN PHASE1
	DCA	RCS2CADR	
	DXCH	T5LOC	
ENDJETS	CS	BIT1	# RESET BIT1 FOR INITIALIZATION OF
	MASK	RCSFLAGS	# T6 PROGRAM
	TS	RCSFLAGS	
	CS	ZERO	# RESET T5PHASE FOR PHASE1
	TS	T5PHASE	
	TCF	RESUME	# RESUME INTERRUPTED PROGRAM
	EBANK=	KMPAC	
RCS2CADR	2CADR	RCSATT	
# Page 1060			
DXCHT12	DXCH	BLAST	
	DXCH	BLAST1	
	DXCH	BLAST	
	TCF	CHECKT23	
DXCHT23	DXCH	BLAST1	
	DXCH	BLAST2	
	DXCH	BLAST1	
	CS	BLAST	
	AD	BLAST1	
	EXTEND		
	BZMF	+2	
	TCF	CALCDT6	

```

DXCH    BLAST
DXCH    BLAST1
DXCH    BLAST
TCF     CALCDT6

```

Page 1061

T6 PROGRAM AND CHANNEL SETUP

```

BANK     21
SETLOC   DAPS5
BANK

T6START  LXCH    BANKRUPT
          EXTEND
          QXCH    QRUPT
          CCS     TIME6          # CHECK TO SEE IF TIME6 WAS RESET
          TCF     RESUME         # AFTER T6RUPT OCCURRED (IN T5RUPT)
          TCF     +2             # IF SO WAIT FOR NEXT T6RUPT BEFORE
          TCF     RESUME         # TAKING ACTION

          CS      RCSFLAGS
          MASK    BIT1          # IF BIT1 IS 0 RESET TO 1
          EXTEND                                     # AND INITIALIZE CHANNEL
          BZF     T6RUPTOR
          ADS     RCSFLAGS
          CA      RWORD1
          EXTEND                                     # INITIALIZE CHANNELS 5,6 WITH WORD1
          WRITE   CHAN6
          CA      PWORD1
          AD      YWORD1
          EXTEND
          WRITE   CHAN5

T6RUPTOR CCS     BLAST
          TCF     ZBLAST        # ZERO BLAST1
          TCF     REPLACE       # REPLACE WORD1
          TCF     +2
          TCF     REPLACE

T6L1     CCS     BLAST1
          TCF     ZBLAST1
          TCF     REPLACE1
          TCF     +2
          TCF     REPLACE1

T6L2     CCS     BLAST2
          TCF     ZBLAST2
          TCF     REPLACE2

```

	TCF	RESUME	
	TCF	REPLACE2	
REPLACE	INDEX	BLAST +1	
	TC	REPLACER	
	CS	ONE	
	TS	BLAST	
	TCF	T6L1	
REPLACE1	INDEX	BLAST1 +1	
# Page 1062			
	TC	REPLACER	
	CS	ONE	
	TS	BLAST1	
	TCF	T6L2	
REPLACE2	INDEX	BLAST2 +1	
	TC	REPLACER	
	CS	ONE	
	TS	BLAST2	
	TCF	RESUME	
REPLACER	CA	RWORD2	# INITIALIZE CHANNELS 5,6 WITH WORD2
	EXTEND		
	WRITE	CHAN6	
	TC	Q	
REPLACEP	CA	YJETS	
	EXTEND		
	RAND	CHAN5	
	AD	PWORD2	
	EXTEND		
	WRITE	CHAN5	
	TC	Q	
REPLACEY	CA	PJETS	
	EXTEND		
	RAND	CHAN5	
	AD	YWORD2	
	EXTEND		
	WRITE	CHAN5	
	TC	Q	
ZBLAST	CAF	ZERO	
	XCH	BLAST	
	TCF	ENABT6	

July 12, 2016

857

```
ZBLAST1      CAF      ZERO
              XCH      BLAST1
              TCF      ENABT6
ZBLAST2      CAF      ZERO
              XCH      BLAST2
ENABT6        TS       TIME6
              CAF      NEGMAX
              EXTEND
              WOR      CHAN13      # ENABLE T6RUPT
              TCF      RESUME

# END OF T6 INTERRUPT

ENDSLECT      EQUALS
```

This code is written to file `src/JET-SELECTION-LOGIC.s`.

B.51 KALCMANU STEERING

```

858  <src/KALCMANU-STEERING.s 858>≡
      # Copyright:    Public domain.
      # Filename:     KALCMANU-STEERING.agc
      # Purpose:      Part of the source code for Comanche, build 055.
      #               It is part of the source code for the Command Module's (CM)
      #               Apollo Guidance Computer (AGC), Apollo 11.
      # Assembler:    yaYUL
      # Reference:     pp. 414-419
      # Contact:       Onno Hommes <ohommes@cmu.edu>.
      # Website:       www.ibiblio.org/apollo.
      # Mod history:   05/07/09 OH      Transcription Batch 1 Assignment
      #
      # The contents of the "Comanche055" files, in general, are transcribed
      # from scanned documents.
      #
      #       Assemble revision 055 of AGC program Comanche by NASA
      #       2021113-051.  April 1, 1969.
      #
      #       This AGC program shall also be referred to as Colossus 2A
      #
      #       Prepared by
      #
      #               Massachussets Institute of Technology
      #               75 Cambridge Parkway
      #               Cambridge, Massachusetts
      #
      #       under NASA contract NAS 9-4065.
      #
      # Refer directly to the online document mentioned above for further information.
      # Please report any errors to info@sandroid.org.
      #
      # Page 414
      # GENERATION OF STEERING COMMANDS FOR DIGITAL AUTOPILOT FREE FALL MANEUVERS
      #
      # NEW COMMANDS WILL BE GENERATED EVERY ONE SECOND DURING THE MANEUVER
      #
      BANK      15
      #
      SETLOC    KALCMON1
      BANK
      #
      EBANK=    BCDU
      #
      COUNT     22/KALC

```

July 12, 2016

859

```
NEWDELHI      CS      HOLDFLAG      # SEE IF MANEUVER HAS BEEN INTERRUPTED
EXTEND                                     # BY ASTRONAUT.
BZMF          NOGO      -2          # IF SO, TERMINATE KALCMANU
NEWANGL       TC      INTPRET
AXC,1         AXC,2
              MIS      # COMPUTE THE NEW MATRIX FROM S/C TO
              DEL      # STABLE MEMBER AXES
CALL
              MXM3
VLOAD        STADR
STOVL        MIS +12D      # CALCULATE NEW DESIRED CDU ANGLES
STADR
STOVL        MIS +6D
STADR
STORE        MIS
AXC,1        CALL
              MIS
              DCMTOCDU      # PICK UP THE NEW CDU ANGLES FROM MATRIX
RTB
              V1STO2S
STORE        NCDU      # NEW CDU ANGLES
BONCLR       EXIT
              CALCMAN2
              MANUSTAT      # TO START MANEUVER
CAF          TWO      # +0 OTHERWISE
INCRDCDU     TS      KSPNDX
DOUBLE
TS           KDPNDX
INDEX        KSPNDX
CA           NCDU      # NEW DESIRED CDU ANGLES
EXTEND
INDEX        KSPNDX
MSU          BCDU      # INITIAL S/C ANGLE OR PREVIOUS DESIRED
EXTEND      # CDU ANGLES
MP           QUADROT
INDEX        KDPNDX
DXCH         DELCDUX      # ANGEL INCREMENTS TO BE ADDED TO
# Page 415
INDEX        KSPNDX      # DCDU EVERY TENTH SEC
CA           NCDU      # BY LEM DAP
INDEX        KSPNDX
XCH          BCDU
INDEX        KDPNDX
TS           CDUXD
CCS          KSPNDX
TCF          INCRDCDU      # LOOP FOR THREE AXES
```

```

                                RELINT
# COMPARE PRESENT TIME WITH TIME TO TERMINATE MANEUVER

TMANUCHK      TC      TIMECHK
               TC      POSTJUMP
               CADR     CONTMANU

MANUSTAL      CAF      ONE
               TC      WAITLIST
               EBANK=   BCDU
               2CADR   MANUSTOP

                                RELINT
                                TCF      ENDOFJOB

TIMECHK       EXTEND
               DCS      TIME2
               DXCH     TTEMP
               EXTEND
               DCA      TM
               DAS      TTEMP
               CCS      TTEMP
               TC       Q
               TCF      +2
               TCF      2NDRETRN
               CCS      TTEMP +1
               TC       Q
               TCF      MANUOFF
               COM
MANUOFF       AD      1SEC
               EXTEND
               BZMF     2NDRETRN
2NDRETRN     INCR      Q
               INCR      Q
               INCR      Q
               TC       Q

               SETLOC   MANUSTUF
               BANK

# Page 416
MANUSTAT     EXIT
               EXTEND
               DCA      TIME2
               DAS      TM
# INITIALIZATION ROUTINE
# FOR AUTOMATIC MANEUVERS
# TM+TO      MANEUVER COMPLETION TIME

```

	CS	1SEC	
	TS	L	
	CS	ZERO	
	DAS	TM	# (TM+TO)-1
	INHINT		
	CS	ONE	# ENABLE AUTOPILOT TO PERFORM
	TS	HOLDFLAG	# AUTOMATIC MANEUVERS
	CS	RATEINDX	# SEE IF MANEUVERING AT HIGH RATE
	AD	SIX	
	EXTEND		
	BZMF	HIGHGAIN	
	TCF	+4	
HIGHGAIN	CS	RCSFLAGS	# IF SO, SET HIGH RATE FLAG (BIT 15 OF
	MASK	BIT15	# RCSFLAGS)
	ADS	RCSFLAGS	
	DXCH	BRATE	# X-AXIS MANEUVER RATE
	DXCH	WBODY	
	DXCH	BRATE +2	# Y-AXIS MANEUVER RATE
	DXCH	WBODY1	
	DXCH	BRATE +4	# Z-AXIS MANEUVER RATE
	DXCH	WBODY2	
	CA	BIASTEMP +1	# INSERT ATTITUDE ERROR BIASES
	TS	BIAS	# INTO AUTOPILOT
	CA	BIASTEMP +3	
	TS	BIAS1	
	CA	BIASTEMP +5	
	TS	BIAS2	
	CA	TIME1	
	AD	1SEC	
	XCH	NEXTIME	
	TC	POSTJUMP	
	CADR	INCRDCDU -1	
CONTMANU	INHINT		# CONTINUE WITH UPDATE PROCESS
	CS	TIME1	
	AD	NEXTIME	
	CCS	A	
	AD	ONE	
	TCF	MANUCALL	
	AD	NEGMAX	
	COM		
MANUCALL	TC	WAITLIST	
	EBANK=	BCDU	
	2CADR	UPDTCALL	
	RELINT		

Page 417

CAF	1SEC	# INCREMENT TIME FOR NEXT UPDATE
ADS	NEXTIME	
TCF	ENDOFJOB	

UPDTCALL	CAF	PRI026	# CALL FOR UPDATE
	TC	FINDVAC	# OF STEERING COMMANDS
	EBANK=	BCDU	
	2CADR	NEWDELHI	
	TC	TASKOVER	

Page 418

ROUTINE FOR TERMINATING AUTOMATIC MANEUVERS

	SETLOC	KALCMON3	
	BANK		
MANUSTOP	TC	STOPYZ	
	TC	IBNKCALL	
	CADR	LOADYZ	
ENDROLL	CA	CPHI	
	TS	CDUXD	# SET CDUXD TO THE COMMANDED OUTER GIMBAL
	TC	STOPRATE	
ENDMANU	CA	ATTPRIO	# RESTORE USERS PRIO
	TS	NEWPRIO	
	CA	ZERO	# ZERO ATTCADR
	DXCH	ATTCADR	
	TC	SPVAC	# RETURN TO USER OF GOMANUR
	TC	TASKOVER	
	SETLOC	STOPRAT	
	BANK		
STOPRATE	CAF	ZERO	
	TS	DELCDEX	
	TS	DELCDEX +1	# ZERO ROLL INCREMENTAL ANGLES
	TS	WBODY	# RATE
	TS	WBODY +1	
	TS	BIAS	# BIAS
	CS	BIT15	# MAKE SURE HIGH RATE FLAG (BIT 15 OF

```

                                MASK    RCSFLAGS    # RCSFLAGS) IS RESET.
                                TS      RCSFLAGS

STOPYZ    CAF    ZERO
                                TS      DELCDUY      # ZERO PITCH, YAW
                                TS      DELCDUY +1    # INCREMENTAL ANGLES
                                TS      DELCDUZ
                                TS      DELCDUZ +1
                                TS      WBODY1        # RATES
                                TS      WBODY1 +1
                                TS      WBODY2
                                TS      WBODY2 +1
                                TS      BIAS1        # BIASES
                                TS      BIAS2
                                TC      Q

                                SETLOC  MANUSTUF
                                BANK

# Page 419
ZEROERROR    CA    CDUX      # PICK UP CDU ANGLES AND STORE IN
                                TS      CDUXD        # CDU DESIRED
                                CA    CDUY
                                TS      CDUYD
                                CA    CDUZ
                                TS      CDUZD
                                TC      Q

                                SETLOC  KALCMON1
                                BANK

LOADCDUD    CA    CPHI      # STORE TERMINAL ANGLES INTO
                                TS      CDUXD        # COMMAND ANGLES
LOADYZ      CA    CTHETA
                                TS      CDUYD
                                CA    CPSI
                                TS      CDUZD
                                TC      Q

```

This code is written to file `src/KALCMANU-STEERING.s`.

B.52 KALMAN FILTER

```

864  <src/KALMAN-FILTER.s 864>≡
      # Copyright:    Public domain.
      # Filename:     KALMAN_FILTER.agc
      # Purpose:      Part of the source code for Luminary 1A build 099.
      #               It is part of the source code for the Lunar Module's (LM)
      #               Apollo Guidance Computer (AGC), for Apollo 11.
      # Assembler:    yaYUL
      # Contact:       Ron Burkey <info@sandroid.org>.
      # Website:       www.ibiblio.org/apollo.
      # Pages:         1470-1471
      # Mod history:   2009-05-27 RSB   Adapted from the corresponding
      #               Luminary131 file, using page
      #               images from Luminary 1A.
      #
      # This source code has been transcribed or otherwise adapted from
      # digitized images of a hardcopy from the MIT Museum. The digitization
      # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
      # the Museum. Many thanks to both. The images (with suitable reduction
      # in storage size and consequent reduction in image quality as well) are
      # available online at www.ibiblio.org/apollo. If for some reason you
      # find that the images are illegible, contact me at info@sandroid.org
      # about getting access to the (much) higher-quality images which Paul
      # actually created.
      #
      # Notations on the hardcopy document read, in part:
      #
      #       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
      #       16:27 JULY 14, 1969
      #
      # Page 1470
      EBANK=  NO.UJETS
      BANK    16
      SETLOC  DAPS1
      BANK
      COUNT*  $$/DAP
      RATELOOP
      CA      TWO
      TS      DAPTEMP6
      DOUBLE
      TS      Q
      INDEX   DAPTEMP6
      CCS     TJP
      TCF     +2

```



```

TCF      LOOPRATE
AD        -100MST6
EXTEND
BZMF      SMALLTJU
INDEX     DAPTEMP6
CCS       TJP
CA        -100MST6
TCF       +2
CS        -100MST6
INDEX     DAPTEMP6
ADS       TJP
INDEX     DAPTEMP6
CCS       TJP
CS        -100MS      # 0.1 AT 1
TCF       +2
CA        -100MS
LOOPRATE  EXTEND
INDEX     DAPTEMP6
MP        NO.PJETS
CA        L
INDEX     DAPTEMP6
TS        DAPTEMP1    # SIGNED TORQUE AT 1 JET-SEC FOR FILTER
EXTEND
MP        BIT10        # RESCALE TO 32; ONE BIT ABOUT 2 JET-MSEC
EXTEND
BZMF      NEGORK
STORTORK  INDEX     Q      # INCREMENT DOWNLIST REGISTER.
ADS       DOWNTORK      #      NOTE:  NOT INITIALIZED; OVERFLOWS.

CCS       DAPTEMP6
TCF       RATELOOP +1
TCF       ROTORQUE
CA        ZERO
SMALLTJU  INDEX     DAPTEMP6
XCH       TJP
EXTEND
# Page 1471
MP        ELEVEN      # 10.24 PLUS
CA        L
TCF       LOOPRATE
CA        DAPTEMP2
AD        DAPTEMP3
EXTEND
MP        1JACCR
TS        JETRATER
CS        DAPTEMP3

```

	AD	DAPTEMP2
	EXTEND	
	MP	1JACCQ
	TS	JETRATEQ
	TCF	BACKP
-100MST6	DEC	-160
NEGTORK	COM	
	INCR	Q
	TCF	STORTORK

This code is written to file `src/KALMAN-FILTER.s`.

B.53 KEYRUPT UPRUPT

```

867  <src/KEYRUPT-UPRUPT.s 867>≡
    # Copyright:    Public domain.
    # Filename:     KEYRUPT_UPRUPT.agc
    # Purpose:      Part of the source code for Comanche, build 055. It
    #               is part of the source code for the Command Module's
    #               (CM) Apollo Guidance Computer (AGC), Apollo 11.
    # Assembler:   yaYUL
    # Reference:    pp. 1449-1451
    # Contact:      Ron Burkey <info@sandroid.org>
    # Website:      http://www.ibiblio.org/apollo.
    # Mod history:  2009-05-07 RSB   Adapted from Colossus249 file of the same
    #               name, and page images. Corrected various
    #               typos in the transcription of program
    #               comments, and these should be back-ported
    #               to Colossus249.
    #
    # The contents of the "Comanche055" files, in general, are transcribed
    # from scanned documents.
    #
    # Assemble revision 055 of AGC program Comanche by NASA
    # 2021113-051. April 1, 1969.
    #
    # This AGC program shall also be referred to as Colossus 2A
    #
    # Prepared by
    #
    #               Massachussets Institute of Technology
    #               75 Cambridge Parkway
    #               Cambridge, Massachusetts
    #
    # under NASA contract NAS 9-4065.
    #
    # Refer directly to the online document mentioned above for further
    # information. Please report any errors to info@sandroid.org.

    # Page 1449

    BANK      14
    SETLOC    KEYRUPT
    BANK
    COUNT*    $$/KEYUP

KEYRUPT1     TS      BANKRUPT
             XCH     Q
             TS      QRUPT
             TC      LODSAMPT      # TIME IS SNATCHED IN RUPT FOR NOUN 65.

```

	CAF	LOW5	
	EXTEND		
KEYCOM	RAND	MNKEYIN	# CHECK IF KEYS 5M-1M ON
	TS	RUPTREG4	
	CS	FLAGWRD5	
	MASK	BIT15	
	ADS	FLAGWRD5	
ACCEPTUP	CAF	CHRPRI0	# (NOTE: RUPTREG4 = KEYTEMP1)
	TC	NOVAC	
	EBANK=	DSPCOUNT	
	2CADR	CHARIN	
	CA	RUPTREG4	
	INDEX	LOCCTR	
	TS	MPAC	# LEAVE 5 BIT KEY CODE IN MPAC FOR CHARIN
	TC	RESUME	
# Page 1450			
# UPRUPT PROGRAM			
UPRUPT	TS	BANKRUPT	
	XCH	Q	
	TS	QRUPT	
	TC	LODSAMPT	# TIME IS SNATCHED IN RUPT FOR NOUN 65.
	CAF	ZERO	
	XCH	INLINK	
	TS	KEYTEMP1	
	CAF	BIT3	# TURN ON UPACT LIGHT
	EXTEND		# (BIT 3 OF CHANNEL 11)
	WOR	DSALMOUT	
UPRPT1	CAF	LOW5	# TEST FOR TRIPLE CHAR REDUNDANCY
	MASK	KEYTEMP1	# LOW5 OF WORD
	XCH	KEYTEMP1	# LOW5 INTO KEYTEMP1
	EXTEND		
	MP	BIT10	# SHIFT RIGHT 5
	TS	KEYTEMP2	
	MASK	LOW5	# MID 5
	AD	HI10	
	TC	UPTTEST	
	CAF	BIT10	
	EXTEND		
	MP	KEYTEMP2	# SHIFT RIGHT 5
	MASK	LOW5	# HIGH 5
	COM		
	TC	UPTTEST	

```

UPOK      CS      ELRCODE      # CODE IS GOOD.  IF CODE = 'ERROR RESET',
          AD      KEYTEMP1      # CLEAR UPLOCKFL (SET BIT4 OF FLAGWRD7 = 0)
          EXTEND      # IF CODE DOES NOT = 'ERROR RESET', ACCEPT
          BZF      CLUPLOCK      # CODE ONLY IF UPLOCKFL IS CLEAR (=0).

          CAF      BIT4          # TEST UPLOCKFL FOR 0 OR 1
          MASK     FLAGWRD7
          CCS      A
          TC      RESUME          # UPLOCKFL = 1
          TC      ACCEPTUP        # UPLOCKFL = 0

CLUPLOCK   CS      BIT4          # CLEAR UPLOCKFL (I.E., SET BIT 4 OF
          MASK     FLAGWRD7      # FLAGWRD7 = 0)
          TS      FLAGWRD7
          TC      ACCEPTUP

TMFAIL2    CS      FLAGWRD7      # CODE IS BAD
          MASK     BIT4          # LOCK OUT FURTHER UPLINK ACTIVITY
          ADS      FLAGWRD7      # (BY SETTING UPLOCKFL = 1) UNTIL
          TC      RESUME          # 'ERROR RESET' IS SENT VIA UPLINK.

UPTEST     AD      KEYTEMP1

# Page 1451

          CCS      A
          TC      TMFAIL2
HI10       OCT     77740
          TC      TMFAIL2
          TC      Q

ELRCODE    OCT     22

```

'UPLINK ACTIVITY LIGHT' IS TURNED OFF BY

- ```

1. VBRELDSP
2. ERROR RESET
3. UPDATE PROGRAM (P27) ENTERED BY V70,V71,V72, AND V73.
#

```

# THE RECEPTION OF A BAD CODE (I.E., CCC FAILURE) LOCKS OUT FURTHER UPLINK ACTIVITY BY SETTING  
 # THIS INDICATION WILL BE TRANSFERRED TO THE GROUND BY THE DOWNLINK WHICH DOWNLINKS ALL FLAGWOR  
 # WHEN UPLINK ACTIVITY IS LOCKED OUT, IT CAN BE ALLOWED WHEN THE GROUND UPLINKS AND 'ERROR RESE  
 # (IT IS RECOMMENDED THAT THE 'ERROR LIGHT RESET' CODE IS PRECEDED BY 16 BITS THE FIRST OF WHI  
 # BY 15 ZEROS. THIS WILL ELIMINATE EXTRANEIOUS BITS FROM INLINK WHICH MAY HAVE BEEN LEFT OVER F  
 # FAILURE).

#

# UPLINK ACTIVITY IS ALSO ALLOWED (UNLOCKED) DURING FRESH START WHEN FRESH START SETS BIT4 OF F

This code is written to file `src/KEYRUPT-UPRUPT.s`.

## B.54 LAMBERT AIMPOINT GUIDANCE

```

871 <src/LAMBERT-AIMPOINT-GUIDANCE.s 871>≡
Copyright: Public domain.
Filename: LAMBERT_AIMPOINT_GUIDANCE.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 651-653
Mod history: 2009-05-18 RSB Transcribed from Luminary 099
page images.
2009-06-05 RSB Corrected 4 typos.
2009-06-07 RSB Fixed a typo.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 651
#
GENERAL LAMBERT AIMPOINT GUIDANCE **
WRITTEN BY RAMA M AIYAWAR
#
PROGRAM P-31 DESCRIPTION **
#
1. TO ACCEPT TARGETING PARAMETERS OBTAINED FROM A SOURCE EXTERNAL
TO THE LEM AND COMPUTE THERE FROM THE REQUIRED-VELOCITY AND
OTHER INITIAL CONDITIONS REQUIRED BY LM FOR DESIRED MANEUVER.
THE TARGETING PARAMETERS ARE TIG (TIME OF IGNITION), TARGET
VECTOR (RTARG), AND THE TIME FROM TIG UNTIL THE TARGET IS
REACHED (DELLT4), DESIRED TIME OF FLIGHT FROM RINIT TO RTARG.

```

```

ASSUMPTIONS **
#
1. THE TARGET PARAMETERS MAY HAVE BEEN LOADED PRIOR TO THE
EXECUTION OF THIS PROGRAM.
2. THIS PROGRAM IS APPLICABLE IN EITHER EARTH OR LUNAR ORBIT.
3. THIS PROGRAM IS DESIGNED FOR ONE-MAN OPERATION, AND SHOULD
BE SELECTED BY THE ASTRONAUT BY DSKY ENTRY V37 E31.

SUBROUTINES USED **
#
MANUPARM, TTG/N35, R02BOTH, MIDGIM, DISPMGA, FLAGDOWN, BANKCALL,
GOTOPOOH, ENDOFJOB, PHASCHNG, GOFLASHR, GOFLASH.
#
MANUPARM CALCULATES APOGEE, PERIGEE ALTITUDES AND DELTAV DESIRED
FOR THE MANEUVER.
#
TTG/N35 CLOCKTASK - UPDATES CLOCK.
#
MIDGIM CALCULATES MIDDLE GIMBAL ANGLE FOR DISPLAY.
#
R02BOTH IMU - STATUS CHECK ROUTINE.

DISPLAYS USED IN P-31LM **
#
V06N33 DISPLAY SOTRED TIG (IN HRS. MINS. SECS.)
V06N42 DISPLAY APOGEE, PERIGEE, DELTAV.
V16N35 DISPLAY TIME FROM TIG.
V06N45 TIME FROM IGNITION AND MIDDLE GIMBAL ANGLE.

ERASABLE INITIALIZATION REQUIRED **
#
TIG TIME OF IGNITION DP (B+28) CS.
#
DELT4 DESIRED TIME OF FLIGHT DP (B+28) CS
FROM RINIT TO RTARG.
#
RTARG RADIUS VECTOR OF TARGET POSITION VECTOR
RADIUS VECTOR SCALED TO (B+29)METERS IF EARTH ORBIT
Page 652
RADIUS VECTOR SCALED TO (B+27)METERS IF MOON ORBIT

OUTPUT **
#
HAPO APOGEE ALTITUDE
HPER PERIGEE ALTITUDE
VGDISP MAG. OF DELTAV FOR DISPLAY, SCALING B+7 M/CS EARTH

```



July 12, 2016

873

```
MAG. OF DELTAV FOR DISPLAY, SCALING B+5 M/CS MOON
MIDGIM MIDDLE GIMBAL ANGLE
XDELVFLG RESETS XDELVFLG FOR LAMBERT VG COMPUTATIONS

ALARMS OR ABORTS NONE **

RESTARTS ARE VIA GROUP 4 **

 SETLOC GLM
 BANK

 EBANK= SUBEXIT

 COUNT* $$/P31
P31 TC P20FLGON
 CAF V06N33 # T16
 TC VNP00H
 TC INTPRET
 CLEAR DLOAD
 UPDATFLG
 TIG
 STCALL TDEC1 # INTEGRATE STATE VECTORS TO TIG
 LEMPREC
 VLOAD SETPD
 RATT
 OD
 STORE RTIG
 STOVL RINIT
 VATT
 STORE VTIG
 STODL VINIT
 P30ZERO
 PUSH PDDL # E4 AND NUMIT = 0
 DELLT4
 DAD SXA,1
 TIG
 RTX1
 STORE TPASS4
 SXA,2 CALL
 RTX2
 INITVEL
 VLOAD PUSH

 DELVEET3
 STORE DELVSIN
 ABVAL CLEAR
```

# Page 653

```

 XDELVFLG
STCALL VGDISP
 GET.LVC
VLOAD PDVL
 RTIG
 VIPRIME
CALL PERIAP01
CALL SHIFTR1
CALL # LIMIT DISPLAY TO 9999.9 N. MI.
 MAXCHK
STODL HPER
 4D
CALL SHIFTR1
CALL # LIMIT DISPLAY TO 9999.9 N. MI.
 MAXCHK
STORE HAPO
EXIT
CAF V06N81 # DELVLVC
TC VNP00H
CAF V06N42 # HAPO, HPER, VGDISP
TC VNP00H
TC INTPRET
REVN1645 SET CALL # TRKMKCNT, TTOGO, +MGA
 FINALFLG
 VN1645
GOTO REVN1645

```

```

*** END OF LEMP30S .103 ***

```

This code is written to file `src/LAMBERT-AIMPOINT-GUIDANCE.s`.

## B.55 LANDING ANALOG DISPLAYS

875 *<src/LANDING-ANALOG-DISPLAYS.s 875>≡*

```
Copyright: Public domain.
Filename: LANDING_ANALOG_DISPLAYS.agc
Purpose: Part of the source code for Luminary, build 099. It
is part of the source code for the Lunar Module's
(LM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 898-907
Contact: Ron Burkey <info@sandroid.org>,
Fabrizio Bernardini <fabrizio@spacecraft.it>
Website: http://www.ibiblio.org/apollo.
Mod history: 05/06/09 FB Transcription Batch 4 Assignment.
#
The contents of the "Luminary099" files, in general, are transcribed
from scanned documents.
#
Assemble revision 001 of AGC program Luminary099 by NASA
2021112-061. July 14, 1969.
#
Prepared by
#
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.
```

# Page 898

```

BANK 21
SETLOC R10
BANK

EBANK= UNIT/R/
COUNT* $$/R10

LANDISP LXCH PIPCTR1 # UPDATE TBASE2 AND PIPCTR SIMULTANEOUSLY.
 CS TIME1
 DXCH TBASE2

 CS FLAGWRD7 # IS LANDING ANALOG DISPLAYS FLAG SET?
 MASK SWANDBIT
 CCS A
```

|            |        |          |                                             |
|------------|--------|----------|---------------------------------------------|
|            | TCF    | DISPRSET | # NO.                                       |
|            | CA     | IMODES33 | # BIT 7 = 0 (DO ALTRATE), =1 (DO ALT.)      |
|            | MASK   | BIT7     |                                             |
|            | CCS    | A        |                                             |
| ALTROUT    | TCF    | ALTOUT   |                                             |
|            | TC     | DISINDAT | # CHECK MODE SELECT SWITCH AND DIDFLG.      |
|            | CS     | IMODES33 |                                             |
|            | MASK   | BIT7     |                                             |
|            | ADS    | IMODES33 | # ALTERNATE ALTITUDE RATE WITH ALTITUDE.    |
|            | CAF    | BIT2     | # RATE COMMAND IS EXECUTED BEFORE RANGE.    |
|            | EXTEND |          |                                             |
| ARCOMP     | WOR    | CHAN14   | # ALTRATE (BIT2 = 1), ALTITUDE (BIT2 = 0).  |
|            | CA     | RUNIT    | # COMPUTE ALTRATE = RUNIT.VVECT M/CS *(-6). |
|            | EXTEND |          |                                             |
|            | MP     | VVECT    | # MULTIPLY X-COMPONENTS.                    |
|            | XCH    | RUPTREG1 | # SAVE SINGLE PRECISION RESULT M/CS*2(-6)   |
|            | CA     | RUNIT +1 | # MULTIPLY Y-COMPONENTS.                    |
|            | EXTEND |          |                                             |
|            | MP     | VVECT +1 |                                             |
|            | ADS    | RUPTREG1 | # ACCUMULATE PARTIAL PRODUCTS.              |
|            | CA     | RUNIT +2 | # MULTIPLY Z-COMPONENTS.                    |
|            | EXTEND |          |                                             |
|            | MP     | VVECT +2 |                                             |
|            | ADS    | RUPTREG1 | # ALTITUDE RATE IN M/CS *2(-6).             |
|            | CA     | ARCONV   | # CONVERT ALTRATE TO BIT UNITS (.5FPS/BIT)  |
|            | EXTEND |          |                                             |
|            | MP     | RUPTREG1 |                                             |
|            | DDOUBL |          |                                             |
|            | DDOUBL |          |                                             |
|            | XCH    | RUPTREG1 | # ALTITUDE RATE IN BIT UNITS*2(-14).        |
|            | CA     | DALTRATE | # ALTITUDE RATE COMPENSATION FACTOR.        |
|            | EXTEND |          |                                             |
|            | MP     | DT       |                                             |
|            | AD     | RUPTREG1 |                                             |
|            | TS     | ALTRATE  | # ALTITUDE RATE IN BIT UNITS*2(-14).        |
|            | CS     | ALTRATE  |                                             |
| # Page 899 | EXTEND |          | # CHECK POLARITY OF ALTITUDE RATE.          |
|            | BZMF   | +2       |                                             |
|            | TCF    | DATAOUT  | # NEGATIVE -- SEND POS. PULSES TO ALTM REG. |
|            | CA     | ALTRATE  | # POSITIVE OR ZERO -- SET SIGN BIT = 1 AND  |
|            | AD     | BIT15    | # SEND TO ALTM REGISTER. *DO NOT SEND +0*   |
| DATAOUT    | TS     | ALTM     | # ACTIVATE THE LANDING ANALOG DISPLAYS      |
|            | CAF    | BIT3     |                                             |
|            | EXTEND |          |                                             |
|            | WOR    | CHAN14   | # BIT3 DRIVES THE ALT/ALTRATE METER.        |

|            |        |            |                                              |
|------------|--------|------------|----------------------------------------------|
|            | TCF    | TASKOVER   | # EXIT                                       |
| ALTOUT     | TC     | DISINDAT   | # CHECK MODE SELECT SWITCH AND DIDFLG.       |
|            | CS     | BIT7       |                                              |
|            | MASK   | IMODES33   |                                              |
|            | TS     | IMODES33   | # ALTERNATE ALTITUDE RATE WITH ALTITUDE.     |
|            | CS     | BIT2       |                                              |
|            | EXTEND |            |                                              |
|            | WAND   | CHAN14     |                                              |
|            | CCS    | ALTBITS    | # = -1 IF OLD ALT. DATA TO BE EXTRAPOLATED.  |
|            | TCF    | +4         |                                              |
|            | TCF    | +3         |                                              |
|            | TCF    | OLDDATA    |                                              |
|            | TS     | ALTBITS    | # SET ALTBITS FROM -0 TO +0.                 |
|            | CS     | ONE        |                                              |
|            | DXCH   | ALTBITS    | # SET ALTBITS = -1 FOR SWITCH USE NEXT PASS. |
|            | DXCH   | ALTSAVE    |                                              |
|            | CA     | BIT10      | # NEW ALTITUDE EXTRAPOLATION WITH ALTRATE.   |
|            | XCH    | Q          |                                              |
|            | LXCH   | 7          | # ZL                                         |
|            | CA     | DT         |                                              |
|            | EXTEND |            |                                              |
|            | DV     | Q          | # RESCALE DT*2(-14) TO *2(-9) TIME IN CS.    |
|            | EXTEND |            |                                              |
|            | MP     | ARTOA2     | # .0021322 *2(+8)                            |
|            | TCF    | OLDDATA +1 | # RATE APPLIES FOR DT CS.                    |
| ZDATA2     | DXCH   | ALTSAVE    |                                              |
|            | TCF    | NEWDATA    |                                              |
| OLDDATA    | CA     | ARTOA      | # RATE APPLIES FOR .5 SEC. (4X/SEC. CYCLE)   |
|            | EXTEND |            |                                              |
|            | MP     | ALTRATE    | # EXTRAPOLATE WITH ALTITUDE RATE.            |
|            | DDOUBL |            |                                              |
|            | AD     | ALTSAVE +1 |                                              |
|            | TS     | ALTSAVE +1 |                                              |
|            | CAF    | ZERO       |                                              |
|            | ADS    | ALTSAVE    |                                              |
|            | CAF    | POSMAX     | # FORCE SIGN AGREEMENT ASSUMING A            |
|            | AD     | ONE        | # NON-NEGATIVE ALTSAVE.                      |
|            | AD     | ALTSAVE +1 | # IF ALTSAVE IS NEGATIVE, ZERO ALTSAVE       |
|            | TS     | ALTSAVE +1 | # AND ALTSAVE +1 AT ZERODATA.                |
| # Page 900 | CAF    | ZERO       |                                              |
|            | AD     | POSMAX     |                                              |
|            | AD     | ALTSAVE    |                                              |
|            | TS     | ALTSAVE    | # POSSIBLY SKIP TO NEWDATA.                  |

|          |        |            |                                            |
|----------|--------|------------|--------------------------------------------|
|          | TCF    | ZERODATA   |                                            |
| NEWDATA  | CCS    | ALTSAVE +1 |                                            |
|          | TCF    | +4         |                                            |
|          | TCF    | +3         |                                            |
|          | CAF    | ZERO       | # SET NEGATIVE ALTSAVE +1 TO +0.           |
|          | TS     | ALTSAVE +1 |                                            |
|          | CCS    | ALTSAVE    | # PROVIDE A 15 BIT UNSIGNED OUTPUT.        |
|          | CAF    | BIT15      | # THE HI-ORDER PART IS +1 OR +0.           |
|          | AD     | ALTSAVE +1 |                                            |
|          | TCF    | DATAOUT    | # DISPATCH UNSIGNED BITS TO ALTM REG.      |
| DISINDAT | EXTEND |            |                                            |
|          | QXCH   | LADQSAVE   | # SAVE RETURN TO ALTROUT +1 OR ALTOUT +1   |
|          | CAF    | BIT6       |                                            |
|          | EXTEND |            | # WISHETH THE ASTRONAUT THE ANALOG         |
|          | RAND   | CHAN30     | # DISPLAYS? I.E.,                          |
|          | CCS    | A          | # IS THE MODE SELECT SWITCH IN PGNC'S?     |
|          | TCF    | DISPRSET   | # NO. ASTRONAUT REQUESTS NO INERTIAL DATA  |
|          | CS     | FLAGWRD1   | # YES. CHECK STATUS OF DIDFLAG.            |
|          | MASK   | DIDFLBIT   |                                            |
|          | EXTEND |            |                                            |
|          | BZF    | SPEEDRUN   | # SET. PERFORM DATA DISPLAY SEQUENCE.      |
|          | CS     | FLAGWRD1   | # RESET. PERFORM INITIALIZATION FUNCTIONS. |
|          | MASK   | DIDFLBIT   |                                            |
|          | ADS    | FLAGWRD1   | # SET DIDFLAG.                             |
|          | CS     | BIT7       |                                            |
|          | MASK   | IMODES33   | # TO DISPLAY ALTRATE FIRST AND ALT. SECOND |
|          | TS     | IMODES33   |                                            |
|          | CS     | FLAGWRD0   | # ARE WE IN DESCENT TRAJECTORY?            |
|          | MASK   | R10FLBIT   |                                            |
|          | EXTEND |            |                                            |
|          | BZF    | TASKOVER   | # NO                                       |
|          | CAF    | BIT8       | # YES.                                     |
|          | EXTEND |            |                                            |
|          | WOR    | CHAN12     | # SET DISPLAY INERTIAL DATA OUTBIT.        |
|          | CAF    | ZERO       |                                            |
|          | TS     | TRAKLATV   | # LATERAL VELOCITY MONITOR FLAG            |
|          | TS     | TRAKFWDV   | # FORWARD VELOCITY MONITOR FLAG            |
|          | TS     | LATVMETR   | # LATVEL MONITOR METER                     |
|          | TS     | FORVMETR   | # FORVEL MONITOR METER                     |
|          | CAF    | BIT4       |                                            |
|          | TC     | TWIDDLE    |                                            |
|          | ADRES  | INTLZE     |                                            |
|          | TCF    | TASKOVER   |                                            |
| INTLZE   | CAF    | BIT2       |                                            |
|          | EXTEND |            |                                            |
|          | WOR    | CHAN12     | # ENABLE RR ERROR COUNTER.                 |

# Page 901

```

 CS IMODES33
 MASK BIT8
 ADS IMODES33 # SET INERTIAL DATA FLAG.
 TCF TASKOVER

SPEEDRUN CS PIPTIME +1 # UPDATE THE VELOCITY VECTOR
 AD TIME1 # COMPUTE T - TN
 AD HALF # CORRECT FOR POSSIBLE OVERFLOW OF TIME1.
 AD HALF
 XCH DT # SAVE FOR LATER USE
 CA 1SEC
 TS ITEMP5 # INITIALIZE FOR DIVISION LATER
 EXTEND
 DCA GDT/2 # COMPUTE THE X-COMPONENT OF VELOCITY.
 DDOUBL
 DDOUBL
 EXTEND
 MP DT
 EXTEND
 DV ITEMP5
 XCH VVECT # VVECT = G(T-TN) M/CS *2(-5)
 EXTEND
 DCA V # M/CS *2(-7)
 DDOUBL # RESCALE TO 2(-5)
 DDOUBL
 ADS VVECT # VVECT = VN + G(T-TN) M/CS *2(-5)
 CA PIPAX # DELV CM/SEC *2(-14)
 AD PIPATMPX # IN CASE PIPAX HAS BEEN ZEROED
 EXTEND
 MP KPIP1(5) # DELV M/CS *2(-5)
 ADS VVECT # VVECT = VN + DELV + GN(T-TN) M/CS *2(-5)
 EXTEND
 DCA GDT/2 +2 # COMPUTE THE Y-COMPONENT OF VELOCITY.
 DDOUBL
 DDOUBL
 EXTEND
 MP DT
 EXTEND
 DV ITEMP5
 XCH VVECT +1
 EXTEND
 DCA V +2
 DDOUBL
 DDOUBL
 ADS VVECT +1

```

# Page 902

CA PIPAY  
 AD PIPATMPY  
 EXTEND  
 MP KPIP1(5)  
 ADS VVECT +1

EXTEND  
 DCA GDT/2 +4  
 DDOUBL  
 DDOUBL  
 EXTEND  
 MP DT  
 EXTEND  
 DV ITEMP5  
 XCH VVECT +2  
 EXTEND  
 DCA V +4  
 DDOUBL  
 DDOUBL  
 ADS VVECT +2  
 CA PIPAZ  
 AD PIPATMPZ  
 EXTEND  
 MP KPIP1(5)  
 ADS VVECT +2

# COMPUTE THE Z-COMPONENT OF VELOCITY.

CAF BIT3  
 TC VARDELAY

# PAUSE 40 MS TO LET OTHER RUPTS IN.

CS FLAGWRD0  
 MASK R10FLBIT  
 CCS A  
 TCF +2  
 TC LADQSAVE

# ARE WE IN DESCENT TRAJECTORY?

# YES.

# NO.

CA DELVS  
 AD VVECT  
 TS ITEMP1  
 CA DELVS +2  
 AD VVECT +1  
 TS ITEMP2  
 CA DELVS +4  
 AD VVECT +2  
 TS ITEMP3  
 CA ITEMP1  
 EXTEND

# HI X OF VELOCITY CORRECTION TERM.

# HI X OF UPDATED VELOCITY VECTOR.

# =  $VX - DVX \text{ M/CS } *2(-5)$ .

# Y

# Y

# =  $VY - DVY \text{ M/CS } *2(-5)$ 

# Z

# Z

# =  $VZ - DVZ \text{ M/CS } *2(-5)$ 

# COMPUTE VHY, VELOCITY DIRECTED ALONG THE

# Y-COORDINATE.



July 12, 2016

881

```
Page 903
MP UHYP # HI X OF CROSS-RANGE HALF-UNIT VECTOR
XCH RUPTREG1
CA ITEMP2
EXTEND
MP UHYP +2 # Y
ADS RUPTREG1 # ACCUMULATE PARTIAL PRODUCTS.
CA ITEMP3
EXTEND
MP UHYP +4 # Z
ADS RUPTREG1

CA RUPTREG1
DOUBLE
XCH VHY # VHY=VMP.UHYP M/CS*2(-5) .
CA ITEMP1 # NO COMPUTE VHZ, VELOCITY DIRECTED ALONG
EXTEND # THE Z-COORDINATE.
MP UHZP # HI X OF DOWN-RANGE HALF-UNIT VECTOR.
XCH RUPTREG1
CA ITEMP2
EXTEND
MP UHZP +2 # Y
ADS RUPTREG1 # ACCUMULATE PARTIAL PRODUCTS.
CA ITEMP3
EXTEND
MP UHZP +4 # Z
ADS RUPTREG1
CA RUPTREG1
DOUBLE
XCH VHZ # VHZ = VMP.UHZP M/CS*2(-5) .
GET22/32 CAF EBANK6 # GET SIN(AOG),COS(AOG) FROM GPMATRIX.
TS EBANK
EBANK= M22
CA M22
TS ITEMP3
CA M32
TS ITEMP4
CAF EBANK7
TS EBANK
EBANK= UNIT/R/
LATFWDV CA ITEMP4 # COMPUTE LATERAL AND FORWARD VELOCITIES.
EXTEND
MP VHY
XCH RUPTREG1
CA ITEMP3
EXTEND
MP VHZ
```

```

ADS RUPTREG1 # = VHY(COS)AOG+VHZ(SIN)AOG M/CS *2(-5)
CA VELCONV # CONVERT LATERAL VELOCITY TO BIT UNITS.
EXTEND
MP RUPTREG1
DDOUBL
XCH LATVEL # LATERAL VELOCITY IN BIT UNITS *2(-14).
CA ITEMP4 # COMPUTE FORWARD VELOCITY.
EXTEND
MP VHZ
XCH RUPTREG1
CA ITEMP3
EXTEND
MP VHY
CS A
ADS RUPTREG1 # =VHZ(COS)AOG-VHY(SIN)AOG M/CS *2(-5).
Page 904
CA VELCONV # CONVERT FORWARD VELOCITY TO BIT UNITS.
EXTEND
MP RUPTREG1
DDOUBL
XCH FORVEL # FORWARD VELOCITY IN BIT UNITS *2(-14).
CS MAXVBITS # ACC.=-199.9989 FT./SEC.
TS ITEMP6 # -547 BIT UNITS (OCTAL) AT 0.5571 FPS/BIT
VMONITOR
CAF ONE # LOOP TWICE.
TS ITEMP5 # FORWARD AND LATERAL VELOCITY LANDING
INDEX ITEMP5 # ANALOG DISPLAYS MONITOR.
CCS LATVEL
TCF +4
TCF LVLIMITS
TCF +8D
TCF LVLIMITS
INDEX ITEMP5
CS LATVEL
AD MAXVBITS # +199.9989 FT.SEC.
EXTEND
BZMF CHKLASTY
TCF LVLIMITS
INDEX ITEMP5
CA LATVEL
AD MAXVBITS
EXTEND
BZMF +2
TCF LVLIMITS
CHKLASTY INDEX ITEMP5

```

|            |        |             |
|------------|--------|-------------|
|            | CCS    | LATVMETR    |
|            | TCF    | +4          |
|            | TCF    | LASTOK      |
|            | TCF    | +7          |
|            | TCF    | LASTOK      |
|            | INDEX  | ITEMP5      |
|            | CA     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | LASTPOSY +5 |
|            | TCF    | +5          |
|            | INDEX  | ITEMP5      |
|            | CS     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | LASTNEGY +4 |
| LASTOK     | INDEX  | ITEMP5      |
|            | CCS    | TRAKLATV    |
|            | TCF    | LASTPOSY    |
|            | TCF    | +2          |
|            | TCF    | LASTNEGY    |
|            | INDEX  | ITEMP5      |
| # Page 905 |        |             |
|            | CA     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | NEGVMAXY    |
|            | TCF    | POSVMAXY    |
| LASTPOSY   | INDEX  | ITEMP5      |
|            | CA     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | +2          |
|            | TCF    | POSVMAXY    |
|            | CS     | MAXVBITS    |
|            | TCF    | ZEROLSTY    |
| POSVMAXY   | INDEX  | ITEMP5      |
|            | CS     | LATVMETR    |
|            | AD     | MAXVBITS    |
|            | INDEX  | ITEMP5      |
|            | XCH    | RUPTREG3    |
|            | CAF    | ONE         |
|            | TCF    | ZEROLSTY +3 |
| LASTNEGY   | INDEX  | ITEMP5      |
|            | CA     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | NEGVMAXY    |
|            | CA     | MAXVBITS    |
|            | TCF    | ZEROLSTY    |
| NEGVMAXY   | INDEX  | ITEMP5      |

|            |        |             |
|------------|--------|-------------|
|            | CA     | LATVMETR    |
|            | AD     | MAXVBITS    |
|            | COM    |             |
|            | INDEX  | ITEMP5      |
|            | XCH    | RUPTREG3    |
|            | CS     | ONE         |
|            | TCF    | ZEROLSTY +3 |
| LVLIMITS   | INDEX  | ITEMP5      |
|            | CCS    | TRAKLATV    |
|            | TCF    | LATVPOS     |
|            | TCF    | +2          |
|            | TCF    | LATVNEG     |
|            | INDEX  | ITEMP5      |
|            | CS     | LATVMETR    |
|            | EXTEND |             |
|            | BZMF   | +2          |
|            | TCF    | NEGLMLV     |
|            | INDEX  | ITEMP5      |
|            | CS     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | LVMINLM     |
|            | AD     | ITEMP6      |
|            | INDEX  | ITEMP5      |
|            | AD     | LATVMETR    |
|            | EXTEND |             |
| # Page 906 | BZMF   | LVMINLM     |
|            | INDEX  | ITEMP5      |
|            | AD     | LATVEL      |
|            | EXTEND |             |
|            | INDEX  | ITEMP5      |
|            | SU     | LATVMETR    |
|            | TCF    | ZEROLSTY    |
| LATVPOS    | INDEX  | ITEMP5      |
|            | CS     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | LVMINLM     |
|            | TCF    | +5          |
| LATVNEG    | INDEX  | ITEMP5      |
|            | CA     | LATVEL      |
|            | EXTEND |             |
|            | BZMF   | LVMINLM     |
|            | INDEX  | ITEMP5      |
|            | CS     | LATVMETR    |
|            | TCF    | ZEROLSTY    |

|            |        |          |                                           |
|------------|--------|----------|-------------------------------------------|
| NEGLMLV    | INDEX  | ITEMP5   |                                           |
|            | CA     | LATVEL   |                                           |
|            | EXTEND |          |                                           |
|            | BZMF   | LVMINLM  |                                           |
|            | CA     | MAXVBITS |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | AD     | LATVMETR |                                           |
|            | COM    |          |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | AD     | LATVEL   |                                           |
|            | EXTEND |          |                                           |
|            | BZMF   | LVMINLM  |                                           |
|            | EXTEND |          |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | SU     | LATVEL   |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | AD     | LATVMETR |                                           |
|            | COM    |          |                                           |
|            | TCF    | ZEROLSTY |                                           |
| LVMINLM    | INDEX  | ITEMP5   |                                           |
|            | CS     | LATVMETR |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | AD     | LATVEL   |                                           |
| ZEROLSTY   | INDEX  | ITEMP5   |                                           |
|            | XCH    | RUPTREG3 |                                           |
|            | CAF    | ZERO     |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | TS     | TRAKLATV |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | CA     | RUPTREG3 |                                           |
|            | AD     | NEGO     | # AVOIDS +0 DINC HARDWARE MALFUNCTION     |
| # Page 907 |        |          |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | TS     | CDUTCMD  |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | CA     | RUPTREG3 |                                           |
|            | INDEX  | ITEMP5   |                                           |
|            | ADS    | LATVMETR |                                           |
|            | CCS    | ITEMP5   | # FIRST MONITOR FORWARD THEN LATERAL VEL. |
|            | TCF    | VMONITOR |                                           |
|            |        |          |                                           |
|            | CAF    | BITSET   | # DRIVE THE X-POINTER DISPLAY.            |
|            | EXTEND |          |                                           |
|            | WOR    | CHAN14   |                                           |
|            | TC     | LADQSAVE | # GO TO ALTROUT +1 OR TO ALTOUT +1        |
| ZERODATA   | CAF    | ZERO     | # ZERO ALTSAVE AND ALTSAVE +1             |

```

 TS L # NO NEGATIVE ALTITUDES ALLOWED.
 TCF ZDATA2

DISPRSET CS FLAGWRD0 # ARE WE IN DESCENT TRAJECTORY?
 MASK R10FLBIT
 EXTEND
 BZF ABORTON # NO.
 CAF BIT8 # YES.
 MASK IMODES33 # CHECK IF INERTIAL DATA JUST DISPLAYED.
 CCS A
 CAF BIT2 # YES. DISABLE RR ERROR COUNTER
 AD BIT8 # NO. REMOVE DISPLAY INERTIAL DATA
 COM
 EXTEND
 WAND CHAN12
ABORTON CS BITS8/7 # RESET INERTIAL DATA, INTERLEAVE FLAGS.
 MASK IMODES33
 TS IMODES33
 CS DIDFLBIT
 MASK FLAGWRD1
 TS FLAGWRD1 # RESET DIDFLAG.
 TCF TASKOVER

BITS8/7 OCT 00300 # INERTIAL DATA AND INTERLEAVE FLAGS.
BITSET = PRI06

```

This code is written to file `src/LANDING-ANALOG-DISPLAYS.s`.

## B.56 LATITUDE LONGITUDE SUBROUTINES

```

887 <src/LATITUDE-LONGITUDE-SUBROUTINES.s 887>≡
Copyright: Public domain.
Filename: LATITUDE_LONGITUDE_SUBROUTINES.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1236-1242
Mod history: 2009-05-14 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 1236
SUBROUTINE TO CONVERT RAD VECTOR AT GIVEN TIME TO LAT, LONG AND ALT
#
CALLING SEQUENCE
L-1 CALL
L LAT-LONG
#
SUBROUTINES USED
R-TO-RP, ARCTAN, SETGAMMA, SETRE
#
ERASABLE INIT. REQ.
AXO, -AYO, AZO, TEPHEM (SET AT LAUNCH TIME)
ALPHAV = POSITION VECTOR METERS B-29
MPAC -- TIME (CSECS B-28)

```

```
ERADFLAG =1, TO COMPUTE EARTH RADIUS, =0 FOR FIXED EARTH RADIUS
LUNAFLAG=0 FOR EARTH, 1 FOR MOON
```

```
#
OUTPUT
```

```
LATITUDE IN LAT (REVS. B-0)
LONGITUDE IN LONG (REVS. B-0)
ALTITUDE IN ALT METERS B-29
```

```
BANK 30
SETLOC LATLONG
BANK
```

```
COUNT 13/LT-LG
```

```
LAT-LONG EBANK= ALPHAV
 STQ SETPD
 INCORPEX
 OD
```

```
STOVL 6D # SAVE TIME IN 6-7D FOR R-TO-RP
 ALPHAV
```

```
PUSH ABVAL # 0-5D= R FOR R-TO-RP
STODL ALPHAM # ABS. VALUE OF R FOR ALT FORMULA BELOW
 ZEROVEC # SET MPAC=0 FOR EARTH, NON-ZERO FOR MOON
BOFF COS # USE COS(0) TO GET NON-ZERO IN MPAC
 LUNAFLAG # 0=EARTH, 1=MOON
 CALLRTRP
```

```
CALLRTRP CALL
 R-TO-RP # RP VECTOR CONVERTED FROM R B-29
UNIT # UNIT RP B-1
STCALL ALPHAV # U2= 1/2 SINL FOR SETRE SUBR BELOW
 SETGAMMA # SET GAMMA=B2/A2 FOR EARTH, =1 FOR MOON
CALL # SCALED B-1.
 SETRE # CALC RE METERS B-29
DLOAD DSQ
 ALPHAV
PDDL DSQ
 ALPHAV +2
```

```
Page 1237
```

```
DAD SQRT
DMP SL1R
 GAMRP
STODL COSTH # COS(LAT) B-1
 ALPHAV +4
STCALL SINTH # SIN(LAT) B-1
 ARCTAN
STODL LAT # LAT B0
```



```

 ALPHAV
STODL COSTH # COS(LONG) B-1
 ALPHAV +2
STCALL SINTH # SIN(LONG) B-1
 ARCTAN
STODL LONG # LONG. REVS B-0 IN RANGE -1/2 TO 1/2
 ALPHAM
DSU ALT= R-RE METERS B-29
 ERADM
STCALL ALT # EXIT WITH ALT METERS B-29
 INCORPEX

Page 1238
SUBROUTINE TO CONVERT LAT, LONG, ALT AT GIVEN TIME TO RADIUS VECTOR
#
CALLING SEQUENCE
L-1 CALL
L LALOTORV
#
SUBROUTINES USED
SETGAMMA, SETRE, RP-TO-R
#
ERASABLE INIT. REQ.
AXO, AYO, AZO, TEPHEM SET AT LAUNCH TIME
LAT -- LATITUDE (REVS B0)
LONG -- LONGITUDE (REVS B0)
ALT -- ALTITUDE (METERS) B-29
MPAC -- TIME (CSECS B-28)
ERADFLAG =1 TO COMPUTE EARTH RADIUS, =0 FOR FIXED EARTH RADIUS
LUNAFLAG=0 FOR EARTH, 1 FOR MOON
#
OUTPUT
R-VECTOR IN ALPHAV (METERS B-29)

LALOTORV STQ SETPD # LAT, LONG, ALT TO R VECTOR
 INCORPEX
 OD
 STCALL 6D # 6-7D = TIME FOR RP-TO-R
 SETGAMMA # GAMMA = B2/A2 FOR EARTH, 1 FOR MOON B-1
 DLOAD SIN # COS(LONG)COS(LAT) IN MPAC
 LAT # UNIT RP = SIN(LONG)COS(LAT) 2-3D
 DMPR PDDL # PD 2 GAMMA*SIN(LAT) 0-1D
 GAMRP
 LAT # 0-1D = GAMMA*SIN(LAT) B-2
 COS PDDL # PD4 2-3D= COS(LAT) B-1 TEMPORARILY
 LONG
 SIN DMPR # PD 2

```

```

 PDDL COS # PD 4 2-3D= SIN(LONG)COS(LAT) B-2
 LAT
 PDDL COS # PD 6 4-5D= COS(LAT) B-1 TEMPORARILY
 LONG
 DMPR VDEF # PD4 MPAC = COS(LONG)COS(LAT) B-2
 UNIT PUSH # 0-5D = UNIT RP FOR RP-TO-R SUBR.
 STCALL ALPHAV # ALPHAV +4= SINL FOR SETRE SUBR.
 SETRE # RE METERS B-29
 DLOAD BOFF # SET MPAC = 0 FOR EARTH, NON-ZERO FOR MOON
 ZEROVEC
 LUNAFLAG
 CALLRPRT
 CALLRPRT COS # USE COS(0) TO GET NON-ZERO IN MPAC
 CALL
 RP-TO-R # EXIT WITH UNIT R VECTOR IN MPAC
 STODL ALPHAV
 ERADM
Page 1239
 DAD VXSC # (RE + ALT)(UNIT R) METERS B-30
 ALT
 ALPHAV
 VSL1
 STCALL ALPHAV # R METERS B-29
 INCORPEX # EXIT WITH R IN METERS B-29

SUBROUTINE TO COMPUTE EARTH RADIUS
#
INPUT
1/2 SIN LAT IN ALPHAV +4
#
OUTPUT
EARTH RADIUS IN ERADM AND MPAC (METERS B-29)

GETERAD DLOAD DSQ
 ALPHAV +4 # SIN**2(L)
 SL1 BDSU
 DP1/2 # COS**2(L)
 DMPR BDSU
 EE
 DP1/2
 BDDV SQRT
 B2XSC
 SR4R
 STORE ERADM
 RVQ

```

891

|       |      |                 |                        |
|-------|------|-----------------|------------------------|
| B2XSC | 2DEC | .0179450689     | # B**2 SCALED B-51     |
| DP1/2 | =    | XUNIT           |                        |
| B2/A2 | 2DEC | .9933064884 B-1 | # GAMMA= B**2/A**2 B-1 |
| EE    | 2DEC | 6.6935116 E-3   | # (1-B**2/A**2) B-0    |
| ERAD  | 2DEC | 6373338 B-29    | # PAD RADIUS           |

|          |       |          |                         |
|----------|-------|----------|-------------------------|
| ARCTAN   | BOV   | CLROVFLW |                         |
| CLROVFLW | DLOAD | DSQ      |                         |
|          |       | SINTH    |                         |
|          | PDDL  | DSQ      |                         |
|          |       | COSTH    |                         |
|          | DAD   |          |                         |
|          | BZE   | SQRT     |                         |
|          |       | ARCTANXX | # ATAN=0/0. SET THETA=0 |
|          | BDDV  | BOV      |                         |
|          |       | SINTH    |                         |
|          |       | ATAN=90  |                         |
|          | SR1   | ASIN     |                         |
|          | STORE | THETA    |                         |
|          | PDDL  | BMN      |                         |
|          |       | COSTH    |                         |
|          |       | NEGCOS   |                         |
| NEGCOS   | DLOAD | RVQ      |                         |
|          | DLOAD | DCOMP    |                         |
|          | BPL   | DAD      |                         |
|          |       | NEGOUT   |                         |

```

 ARCTANXX STORE DP1/2
 RVQ THETA

 NEGOUT DSU GOTO
 DP1/2
 ARCTANXX
 ATAN=90 DLOAD SIGN
 LODP1/4
 SINTH
 STORE THETA
 RVQ

 2DZERO = DPZERO

```

```

Page 1241
***** SETGAMMA SUBROUTINE *****
SUBROUTINE TO SET GAMMA FOR THE LAT-LONG AND LALOTORV SUBROUTINES
#
GAMMA = B**2/A**2 FOR EARTH (B-1)
GAMMA = 1 FOR MOON (B-1)
#
CALLING SEQUENCE
L CALL
L+1 SETGAMMA
#
INPUT
LUNAFLAG=0 FOR EARTH, =1 FOR MOON
#
OUTPUT
GAMMA IN GAMRP (B-1)

```

```

 SETGAMMA DLOAD BOFF # BRANCH FOR EARTH
 B2/A2 # EARTH GAMMA
 LUNAFLAG
 SETGMEX
 SLOAD
 1B1 # MOON GAMMA
 SETGMEX STORE GAMRP
 RVQ
 GAMRP = 8D

```

```

Page 1242
***** SETRE SUBROUTINE *****
SUBROUTINE TO SET RE (EARTH OR MOON RADIUS)
#

```

```

RE = RM FOR MOON
RE = RREF FOR FIXED EARTH RADIUS OR COMPUTED RF FOR FISCHER ELLIPSOID
#
CALLING SEQUENCE
L CALL
L+1 SETRE
#
SUBROUTINES USED
CETERAD
#
INPUT
ERADFLAG = 0 FOR FIXED RE, 1 FOR COMPUTED RE
ALPHAV +4 = 1/2 SINL IF GETERAD IS CALLED
LUNAFLAG = 0 FOR EARTH, =1 FOR MOON
#
OUTPUT
ERADM = 504RM FOR MOON (METERS B-29)
ERADM = ERAD OR COMPUTED RE FOR EARTH (METERS B-29)

SETRE STQ DLOAD
 SETREX
 504RM
 BON DLOAD # BRANCH FOR MOON
 LUNAFLAG
 TSTRLSRM
 ERAD
 BOFF CALL # ERADFLAG=0 FOR FIXED RE, 1 FOR COMPUTED
 ERADFLAG
 SETRXX
 GETERAD
SETRXX STCALL ERADM # EXIT WITH RE OR RM METERS B-29
 SETREX
TSTRLSRM BON VLOAD # ERADFLAG=0, SET RO=RLS
 ERADFLAG # =1 RO=RM
 SETRXX
 RLS
 ABVAL SR2R # SCALE FROM B-27 TO B-29
 GOTO
 SETRXX
SETRX = S2
504RM 2DEC 1738090 B-29 # METERS B-29 (MOON RADIUS)

```

## B.57 LEM GEOMETRY

```

894 <src/LEM-GEOMETRY.s 894>≡
 # Copyright: Public domain.
 # Filename: LEM_GEOMETRY.agc
 # Purpose: Part of the source code for Luminary 1A build 099.
 # It is part of the source code for the Lunar Module's (LM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 320-325
 # Mod history: 2009-05-16 RSB Adapted from the corresponding
 # Luminary131 file, using page
 # images from Luminary 1A.
 #
 # This source code has been transcribed or otherwise adapted from
 # digitized images of a hardcopy from the MIT Museum. The digitization
 # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
 # the Museum. Many thanks to both. The images (with suitable reduction
 # in storage size and consequent reduction in image quality as well) are
 # available online at www.ibiblio.org/apollo. If for some reason you
 # find that the images are illegible, contact me at info@sandroid.org
 # about getting access to the (much) higher-quality images which Paul
 # actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
 # 16:27 JULY 14, 1969
 #
 # Page 320
 BANK 23
 SETLOC LEMGEOM
 BANK
 SBANK= LOWSUPER
 EBANK= XSM
 # THESE TWO ROUTINES COMPUTE THE ACTUAL STATE VECTOR FOR LM,CSM BY ADDING
 # THE CONIC R,V AND THE DEVIATIONS R,V. THE STATE VECTORS ARE CONVERTED TO
 # METERS B-29 AND METERS/CSEC B-7 AND STORED APPROPRIATELY IN RN,VN OR
 # R-OTHER,V-OTHER FOR DOWNLINK. THE ROUTINES' NAMES ARE SWITCHED IN THE
 # OTHER VEHICLE'S COMPUTER.
 #
 # INPUT

```

```

STATE VECTOR IN TEMPORARY STORAGE AREA
IF STATE VECTOR IS SCALED POS B27 AND VEL B5
SET X2 TO +2
IF STATE VECTOR IS SCALED POS B29 AND VEL B7
SET X2 TO 0
#
OUTPUT
R(T) IN RN, V(T) IN VN, T IN PIPTIME
OR
R(T) IN R-OTHER, V(T) IN V-OTHER (T IS DEFINED BY T-OTHER)

SVDWN2 COUNT* $$/GEOM
BOF RVQ # SW=1=AVETOMID DOING W-MATRIX INTEG.
 AVEMIDSW
 +1
VLOAD VSL*
 TDELTAV
 0 -7,2
VAD VSL*
 RCV
 0,2
STOVL RN
 TNUV
VSL* VAD
 0 -4,2
 VCV
VSL*
 0,2
STODL VN
 TET
STORE PIPTIME
RVQ

Page 321
SVDWN1 VLOAD VSL*
 TDELTAV
 0 -7,2
VAD VSL*
 RCV
 0,2
STOVL R-OTHER
 TNUV
VSL* VAD
 0 -4,2
 VCV
VSL*
 0,2

```

STORE V-OTHER  
RVQ

# Page 322

# THE FOLLOWING ROUTINE TAKES A HALF UNIT TARGET VECTOR REFERRED TO NAV BASE COORDINATE  
# GIMBAL ORIENTATIONS AT WHICH THE RR MIGHT SIGHT THE TARGET. THE GIMBAL ANGLES CORRESPONDING  
# ARE LEFT IN MODEA AND THOSE WHICH WOULD BE USED AFTER A REMODE IN MODEB. THIS ROUTINE  
# ANGLE LESS THAN 90 DEGS IN ABS VALUE WITH ARBITRARY SHAFT, WITH A CORRESPONDING DEGREE  
# SELECTION AND LIMIT CHECKING ARE DONE ELSEWHERE.

#

# THE MODE 1 CONFIGURATION IS CALCULATED FROM THE VECTOR AND THEN MODE 2 IS FOUND USING

#

# S(2) = 180 + S(1)

# T(2) = 180 - T(1)

#

# THE VECTOR ARRIVES IN MPAC WHERE TRG\*SMNG OR \*SMNB\* WILL HAVE LEFT IT.

RRANGLES

STORE 32D

DLOAD DCOMP

34D

# SINCE WE WILL FIND THE MODE 1 SHAFT

# ANGLE LATER, WE CAN FIND THE MODE 1

SETPD ASIN

# TRUNNION BY SIMPLY TAKING THE ARCSIN OF

0

# THE Y COMPONENT, THE ASIN GIVIN AN

PUSH BDSU

# ANSWER WHOSE ABS VAL IS LESS THAN 90 DEG.

LODPHALF

STODL 4

# MODE 2 TRUNNION TO 4.

L06ZEROS

STOVL 34D

# UNIT THE PROJECTION OF THE VECTOR

32D

# IN THE X-Z PLANE

UNIT BOVB

# IF OVERFLOW, TARGET VECTOR IS ALONG Y

LUNDESCH

# CALL FOR MANEUVER UNLESS ON LUNAR SURF

STODL 32D

# PROJECTION VECTOR.

32D

SR1 STQ

S2

STODL SINTH

# USE ARCTRIG SINCE SHAFT COULD BE ARB.

36D

SR1

STCALL COSTH

ARCTRIG

# Page 323

PUSH DAD

# MODE 1 SHAFT TO 2.

LODPHALF

STOVL 6

4

RTB

# FIND MODE 2 CDU ANGLES.



```

 2V1STO2S
STOVL MODEB
 0
RTB # MODE 1 ANGLES TO MODE A.
 2V1STO2S
STORE MODEA
EXIT

CS RADMODES # SWAP MODEA AND MODEB IF RR IN MODE 2.
MASK ANTENBIT
CCS A
TCF +4

DXCH MODEA
DXCH MODEB
DXCH MODEA

TC INTERPRET
GOTO

 S2

```

# Page 324

```

GIVEN RR TRUNNION AND SHAFT (T,S) IN TANGNB,+1, FIND THE ASSOCIATED
LINE OF SIGHT IN NAV BASE AXES. THE HALF UNIT VECTOR, .5(SIN(S)COS(T),
-SIN(T),COS(S)COS(T)) IS LEFT IN MPAC AND 32D.

```

```

 SETLOC INFLIGHT
 BANK

COUNT* $$/GEOM

RRNB SLOAD RTB
 TANGNB
 CDULOGIC
 SETPD PUSH # TRUNNION ANGLE TO 0
 0
 SIN DCOMP
 STODL 34D # Y COMPONENT

 COS PUSH # .5 COS(T) TO 0
 SLOAD RTB
 TANGNB +1
 CDULOGIC
RRNB1 PUSH COS # SHAFT ANGLE TO 2
 DMP SL1
 0
 STODL 36D # Z COMPONENT

```

```

 SIN DMP
 SL1
 STOVL 32D
 32D
 RVQ

THIS ENTRY TO RRNB REQUIRES THE TRUNNION AND SHAFT ANGLES IN MPAC AND MPAC +1 RESP

RRNBMPAC STODL 20D # SAVE SHAFT CDU IN 21.
 MPAC # SET MODE TO DP. (THE PRECEEDING STORE
 # MAY BE DP, TP OR VECTOR.)

 RTB SETPD
 CDULOGIC
 0
 PUSH SIN # TRUNNION ANGLE TO 0
 DCOMP
 STODL 34D # Y COMPONENT
 COS PUSH # .5COS(T) TO 0
 SLOAD RTB # PICK UP CDU'S.
 21D
 CDULOGIC
 GOTO
 RRNB1

Page 325
(This page has nothing on it.)

```

This code is written to file `src/LEM-GEOMETRY.s`.

## B.58 LUNAR AND SOLAR EPHEMERIDES SUBROUTINES

```

899 <src/LUNAR-AND-SOLAR-EPHEMERIDES-SUBROUTINES.s 899>≡
 # Copyright: Public domain.
 # Filename: LUNAR_AND_SOLAR_EPHEMERIDES_SUBROUTINES.agc
 # Purpose: Part of the source code for Comanche, build 055. It
 # is part of the source code for the Command Module's
 # (CM) Apollo Guidance Computer (AGC), Apollo 11.
 # Assembler: yaYUL
 # Reference: pp. 785-788
 # Contact: Ron Burkey <info@sandroid.org>
 # Website: http://www.ibiblio.org/apollo.
 # Mod history: 2009-05-12 RSB Adapted from Colossus249 file of the same
 # name and Comanche 055 page images.
 # 2009-07-26 RSB Added annotations related to computation
 # of the ephemeral(?) polynomials.
 #
 # The contents of the "Comanche055" files, in general, are transcribed
 # from scanned documents.
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. April 1, 1969.
 #
 # This AGC program shall also be referred to as Colossus 2A
 #
 # Prepared by
 #
 # Massachussets Institute of Technology
 # 75 Cambridge Parkway
 # Cambridge, Massachusetts
 #
 # under NASA contract NAS 9-4065.
 #
 # Refer directly to the online document mentioned above for further
 # information. Please report any errors to info@sandroid.org.
 #
 # Page 785
 # LUNAR AND SOLAR EPHEMERIDES SUBROUTINES
 #
 # FUNCTIONAL DESCRIPTION
 #
 # THESE SUBROUTINES ARE USED TO DETERMINE THE POSITION AND VELOCITY
 # VECTORS OF THE SUN AND THE MOON RELATIVE TO THE EARTH AT THE
 # SPECIFIED GROUND ELAPSED TIME INPUT BY THE USER.
 #

```

```

THE POSITION OF THE MOON IS STORED IN THE COMPUTER IN THE FORM OF
A NINTH DEGREE POLYNOMIAL APPROXIMATION WHICH IS VALID OVER A 15
DAY INTERVAL BEGINNING SHORTLY BEFORE LAUNCH. THEREFORE THE TIME
INPUT BY THE USER SHOULD FALL WITHIN THIS 15 DAY INTERVAL.
The 9th-degree polynomial spoken of here is a pad load, meaning
that it is not actually hardcoded into the software. Additional
information about calculating the polynomial can be found on the

Orbiter NASSP wiki, as well as information about calculation
of the
solar ephemerides.
#
LSPOS COMPUTES THE POSITION VECTORS OF THE SUN AND THE MOON.
#
LUNPOS COMPUTES THE POSITION VECTOR OF THE MOON.
#
LUNVEL COMPUTES THE VELOCITY VECTOR OF THE MOON.
#
SOLPOS COMPUTES THE POSITION VECTOR OF THE SUN.
#
CALLING SEQUENCE
#
DLOAD CALL
TIME GROUND ELAPSED TIME
SUBROUTINE LSPOS OR LUNPOS OR LUNVEL OR SOLPOS
#
INPUT
#
1) SPECIFIED GROUND ELAPSED TIME IN CS x B-28 LOADED IN MPAC.
#
2) TIMEMO -- TIME AT THE CENTER OF THE RANGE OVER WHICH THE LUNAR
POSITION POLYNOMIAL IS VALID IN CS x B-42.
#
3) VECOEM -- VECTOR COEFFICIENTS OF THE LUNAR POSITION POLYNOMIAL
LOADED IN DESCENDING SEQUENCE IN METERS/CS**N x B-2
#
4) RESO -- POSITION VECTOR OF THE SUN RELATIVE TO THE EARTH AT
TIMEMO IN METERS x B-38
#
5) VESO -- VELOCITY VECTOR OF THE SUN RELATIVE TO THE EARTH AT
TIMEMO IN METERS/CS x B-9
#
6) OMEGAES -- ANGULAR VELOCITY OF THE VECTOR RESO AT TIMEMO IN
REV/CS x B+26
#
ALL EXCEPT THE FIRST INPUT ARE INCLUDED IN THE PRE-LAUNCH

```

```
ERASABLE DATA LOAD.
#
OUTPUT -- LSPOS
Page 786
#
1) 2D OF VAC AREA CONTAINS THE POSITION VECTOR OF THE SUN RELATIVE
TO THE EARTH AT TIME INPUT BY THE USER IN METERS x B-38.
#
2) MPAC CONTAINS THE POSITION VECTOR OF THE MOON RELATIVE TO THE
EARTH AT TIME INPUT BY THE USER IN METERS x B-29
#
OUTPUT -- LUNPOS
#
MPAC CONTAINS THE POSITION VECTOR OF THE MOON RELATIVE TO THE
EARTH AT THE TIME INPUT BY USER IN METERS x B-29
#
OUTPUT -- LUNVEL
#
MPAC CONTAINS THE VELOCITY VECTOR OF THE MOON RELATIVE TO THE
EARTH AT THE TIME INPUT BY THE USER IN METERS/CS x B-7
#
OUTPUT -- SOLPOS
#
MPAC CONTAINS THE POSITION VECTOR OF THE SUN RELATIVE TO THE EARTH
AT TIME INPUT BY THE USER IN METERS x B-38.
#
SUBROUTINES USED
#
NONE
#
REMARKS
#
THE VAC AREA IS USED FOR STORAGE OF INTERMEDIATE AND FINAL RESULTS
OF COMPUTATIONS.
#
S1, X1, AND X2 ARE USED BY THESE SUBROUTINES.
#
PRELAUNCH ERASABLE DATA LOAD ARE ONLY ERASABLE STORAGE USED BY
THESE SUBROUTINES.
#
RESTARTS DURING OPERATION OF THESE SUBROUTINES MUST BE HANDLED BY
THE USER.

 BANK 36
 SETLOC EPHEM
 BANK
```

|            |        |            |   |                                          |
|------------|--------|------------|---|------------------------------------------|
|            | COUNT* | \$\$/EPHEM |   |                                          |
|            | EBANK= | END-E7     |   |                                          |
| LSPOS      | AXT,2  |            |   |                                          |
|            |        | RESA       | # | COMPUTES POSITION VECTORS OF BOTH THE    |
|            | AXT,1  | GOTO       | # | SUN AND THE MOON. THE POSITION VECTOR    |
|            |        | RES        | # | OF THE SUN IS STORED IN 2D OF THE VAC    |
|            |        | LSTIME     | # | AREA. THE POSITION VECTOR OF THE MOON    |
| LUNPOS     | AXT,1  | GOTO       | # | IS STORED IN MPAC.                       |
|            |        | REM        | # | COMPUTES THE POSITION VECTOR OF THE MOON |
|            |        | LSTIME     | # | AND STORES IT IN MPAC.                   |
| # Page 787 |        |            |   |                                          |
| LUNVEL     | AXT,1  | GOTO       | # | COMPUTES THE VELOCITY VECTOR OF THE MOON |
|            |        | VEM        | # | AND STORES IT IN MPAC.                   |
|            |        | LSTIME     |   |                                          |
| SOLPOS     | STQ    | AXT,1      | # | COMPUTES THE POSITION VECTOR OF THE SUN  |
|            |        | X2         | # | AND STORES IT IN MPAC.                   |
|            |        | RES        |   |                                          |
| LSTIME     | SETPD  | SR         |   |                                          |
|            |        | OD         |   |                                          |
|            |        | 14D        |   |                                          |
|            | TAD    | DCOMP      |   |                                          |
|            |        | TEPHEM     |   |                                          |
|            | TAD    | DCOMP      |   |                                          |
|            |        | TIMEMO     |   |                                          |
|            | SL     | SSP        |   |                                          |
|            |        | 16D        |   |                                          |
|            |        | S1         |   |                                          |
|            |        | 6D         |   |                                          |
|            | GOTO   |            |   |                                          |
|            |        | X1         |   |                                          |
| RES        | PUSH   | DMP        | # | PD- 2                                    |
|            |        | OMEGAES    |   |                                          |
|            | PUSH   | COS        | # | PD- 4                                    |
|            | VXSC   | PDDL       | # | PD- 8                                    |
|            |        | RESO       |   |                                          |
|            | SIN    | PDVL       | # | PD-10                                    |
|            |        | RESO       |   |                                          |
|            | PUSH   | UNIT       | # | PD-10                                    |
|            | VXV    | UNIT       |   |                                          |
|            |        | VESO       |   |                                          |
|            | VXV    | VSL1       | # | PD-10                                    |
|            | VXSC   | VAD        | # | PD-02                                    |
|            | VSL1   | GOTO       | # | RES IN METERS x B-38 IN MPAC.            |
|            |        | X2         |   |                                          |
| RESA       | STODL  | 2D         | # | RES IN METERS x B-38 IN 2D OF VAC. PD- 0 |

July 12, 2016

903

|            |       |               |                                   |       |
|------------|-------|---------------|-----------------------------------|-------|
| REM        | AXT,1 | PDVL          | #                                 | PD- 2 |
|            |       | 54D           |                                   |       |
|            |       | VECOEM        |                                   |       |
| REMA       | VXSC  | VAD*          |                                   |       |
|            |       | OD            |                                   |       |
|            |       | VECOEM +60D,1 |                                   |       |
|            | TIX,1 | VSL2          | # REM IN METERS x B-29 IN MPAC.   |       |
|            |       | REMA          |                                   |       |
|            | RVQ   |               |                                   |       |
| VEM        | AXT,1 | PDDL          | #                                 | PD- 2 |
|            |       | 48D           |                                   |       |
|            |       | NINEB4        |                                   |       |
|            | PUSH  | VXSC          | #                                 | PD- 4 |
|            |       | VECOEM        |                                   |       |
| VEMA       | VXSC  |               |                                   |       |
|            |       | OD            |                                   |       |
| # Page 788 |       |               |                                   |       |
|            | STODL | 4D            | #                                 | PD- 2 |
|            | DSU   | PUSH          | #                                 | PD- 4 |
|            |       | ONEB4         |                                   |       |
|            | VXSC* | VAD           |                                   |       |
|            |       | VECOEM +54D,1 |                                   |       |
|            |       | 4D            |                                   |       |
|            | TIX,1 | VSL2          | # VEM IN METERS/CS x B-7 IN MPAC. |       |
|            |       | VEMA          |                                   |       |
|            | RVQ   |               |                                   |       |
| NINEB4     | 2DEC  | 9.0 B-4       |                                   |       |
| ONEB4      | 2DEC  | 1.0 B-4       |                                   |       |

This code is written to file src/LUNAR-AND-SOLAR-EPHemerIDES-SUBROUTINES.s.

## B.59 LUNAR LANDING GUIDANCE EQUATIONS

```

904 <src/LUNAR-LANDING-GUIDANCE-EQUATIONS.s 904>≡
 # Copyright: Public domain.
 # Filename: LUNAR_LANDING_GUIDANCE_EQUATIONS.agc
 # Purpose: Part of the source code for Luminary 1A build 099.
 # It is part of the source code for the Lunar Module's (LM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: HARTMUTH GUTSCHE <hgutsche@explornet.com>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 798-828
 # Mod history: 2009-05-23 HG Transcribed from page images.
 # 2009-06-05 RSB Fixed a goofy thing that was apparently
 # legal in GAP but not in yaYUL. Eliminated
 # a couple of lines of code that shouldn't
 # have survived from Luminary 131 to here.
 # 2009-06-07 RSB Fixed a typo.
 #
 # This source code has been transcribed or otherwise adapted from
 # digitized images of a hardcopy from the MIT Museum. The digitization
 # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
 # the Museum. Many thanks to both. The images (with suitable reduction
 # in storage size and consequent reduction in image quality as well) are
 # available online at www.ibiblio.org/apollo. If for some reason you
 # find that the images are illegible, contact me at info@sandroid.org
 # about getting access to the (much) higher-quality images which Paul
 # actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
 # 16:27 JULY 14, 1969
 #
 # Page 798
 #
 # EBANK= E2DPS
 # COUNT* $$/F2DPS
 #
 # *****
 # LUNAR LANDING FLIGHT SEQUENCE TABLES
 # *****
 #
 # FLIGHT SEQUENCE TABLES ARE ARRANGED BY FUNCTION. THEY ARE REFERENCED USING AS AN I
 # WCHPHASE = -1 ---> IGNALG

```



```

WCHPHASE = 0 ----> BRAKQUAD
WCHPHASE = 1 ----> APPRQUAD
WCHPHASE = 2 ----> VERTICAL

ROUTINES FOR STARTING NEW GUIDANCE PHASES:

NEWPHASE TCF TTFINCR # IGNALG
 TCF TTFINCR # BRAKQUAD
 TCF STARTP64 # APPRQUAD
 TCF P65START # VERTICAL

PRE-GUIDANCE COMPUTATIONS:

PREGUIDE TCF CALCRGVG # IGNALG
 TCF RGVGCALC # BRAKQUAD
 TCF REDESIG # APPRQUAD
 TCF RGVGCALC # VERTICAL

GUIDANCE EQUATIONS:

WHATGUID TCF TTF/8CL # IGNALG
 TCF TTF/8CL # BRAKQUAD
 TCF TTF/8CL # APPRQUAD
 TCF VERTGUID # VERTICAL

POST GUIDANCE EQUATION COMPUTATIONS:

AFTRGUID TCF CGCALC # IGNALG
 TCF CGCALC # BRAKQUAD
 TCF CGCALC # APPRQUAD
 TCF STEER? # VERTICAL

Page 799
WINDOW VECTOR COMPUTATIONS:

WHATEXIT TCF EXGSUB # IGNALG
 TCF EXBRAK # BRAKQUAD
 TCF EXNORM # APPRQUAD

DISPLAY ROUTINES:

WHATDISP TCF P63DISPS # BRAKQUAD
 TCF P64DISPS # APPRQUAD
 TCF VERTDISP # VERTICAL

```

# ALARM ROUTINE FOR TTF COMPUTATION:

|         |     |         |            |
|---------|-----|---------|------------|
|         | TCF | 1406P00 | # IGNALG   |
| WHATALM | TCF | 1406ALM | # BRAKQUAD |
|         | TCF | 1406ALM | # APPRQUAD |

# INDICES FOR REFERENCING TARGET PARAMETERS

|           |     |    |            |
|-----------|-----|----|------------|
|           | OCT | 0  | # IGNALG   |
| TARGETDEX | OCT | 0  | # BRAKQUAD |
|           | OCT | 34 | # APPRQUAD |

\*\*\*\*\*  
 # ENTRY POINTS: ?GUIDSUB FOR THE IGNITION ALGORITHM, LUNLAND FOR SERVOUT  
 \*\*\*\*\*

# IGNITION ALGORITHM ENTRY: DELIVERS N PASSES OF QUADRATIC GUIDANCE

|          |      |             |                                     |
|----------|------|-------------|-------------------------------------|
| ?GUIDSUB | EXIT |             |                                     |
|          | CAF  | TWO         | # N = 3                             |
|          | TS   | NGUIDSUB    |                                     |
|          | TCF  | GUILDRET +2 |                                     |
| GUIDSUB  | TS   | NGUIDSUB    | # ON SUCCEEDING PASSES SKIP TTFINCR |
|          | TCF  | CALCRGVG    |                                     |

# NORMAL ENTRY: CONTROL COMES HERE FROM SERVOUT

|         |     |          |                                          |
|---------|-----|----------|------------------------------------------|
| LUNLAND | TC  | PHASCHNG |                                          |
|         | OCT | 00035    | # GROUP 5: RETAIN ONLY PIPA TASK         |
|         | TC  | PHASCHNG |                                          |
|         | OCT | 05023    | # GROUP 3: PROTECT GUIDANCE WITH PRIO 21 |
|         | OCT | 21000    | # JUST HIGHER THAN SERVICER'S PRIORITY   |

# Page 800

\*\*\*\*\*  
 # GUILDENSTERN: AUTO-MODES MONITOR (R13)  
 \*\*\*\*\*

COUNT\* \$\$/R13

# HERE IS THE PHILOSOPHY OF GUILDENSTERN: ON EVERY APPEARANCE OR DISAPPEARANCE  
 # DISCRETE TO SELECT P67 OR P66 RESPECTIVELY: ON EVERY APPEARANCE OF THE ATTITUDE-I  
 # UNLESS THE CURRENT PROGRAM IS P67 IN WHICH CASE THERE IS NO CHANGE

July 12, 2016

907

```
GUILDEN EXTEND # IS UN-AUTO-THROTTLE DISCRETE PRESENT?
STERN # RSB 2009: Not originally a comment.

 READ CHAN30
 MASK BIT5
 CCS A
 TCF STARTP67 # YES
P67NOW? TC CHECKMM # NO: ARE WE IN P67 NOW?
 DEC 67
 TCF STABL? # NO
STARTP66 TC FASTCHNG # YES
 TC NEWMODEX
DEC66 DEC 66
 EXTEND
 DCA HDOTDISP # SET DESIRED ALTITUDE RATE = CURRENT
 DXCH VDGVERT # ALTITUDE RATE.
STRTP66A TC INTPRET
 SLOAD PUSH
 PBIASZ
 SLOAD PUSH
 PBIASY
 SLOAD VDEF
 PBIASX
 VXSC SET
 BIASFACT
 RODFLAG
 STOVL VBIAS
 TEMX
 VCOMP
 STOVL OLDPIPAX
 ZEROVECS
 STODL DELVROD
 RODSCALE
 STODL RODSCAL1
 PIPTIME
 STORE LASTTPIP
 EXIT
 CAF ZERO
 TS FCOLD
 TS FWEIGHT
 TS FWEIGHT +1
VRTSTART TS WCHVERT
Page 801

 CAF TWO # WCHPHASE = 2 ----> VERTICAL: P65,P66,P67
 TS WCHPHOLD
 TS WCHPHASE
 TC BANKCALL # TEMPORARY, I HOPE HOPE HOPE
```

|                                |        |            |                                           |
|--------------------------------|--------|------------|-------------------------------------------|
|                                | CADR   | STOPRATE   | # TEMPORARY, I HOPE HOPE HOPE             |
|                                | TC     | DOWNFLAG   | # PERMIT X-AXIS OVERRIDE                  |
|                                | ADRES  | XOVINFLG   |                                           |
|                                | TC     | DOWNFLAG   |                                           |
|                                | ADRES  | REDFLAG    |                                           |
|                                | TCF    | VERTGUID   |                                           |
| STARTP67                       | TC     | NEWMODEX   | # NO HARM IN "STARTING" P67 OVER AND OVER |
|                                | DEC    | 67         | # SO NO NEED FOR A FASTCHNG AND NO NEED   |
|                                | CAF    | ZERO       | # TO SEE IF ALREADY IN P67.               |
|                                | TS     | RODCOUNT   |                                           |
|                                | CAF    | TEN        |                                           |
|                                | TCF    | VRTSTART   |                                           |
| STABL?                         | CAF    | BIT13      | # IS UN-ATTITUDE-HOLD DISCRETE PRESENT?   |
|                                | EXTEND |            |                                           |
|                                | RAND   | CHAN31     |                                           |
|                                | CCS    | A          |                                           |
|                                | TCF    | GUILDRET   | # YES ALL'S WELL                          |
| P66NOW?                        | CS     | MODREG     |                                           |
|                                | AD     | DEC66      |                                           |
|                                | EXTEND |            |                                           |
|                                | BZF    | RESTART?   |                                           |
|                                | CA     | RODCOUNT   | # NO. HAS THE ROD SWITCH BEEN "CLICKED"?  |
|                                | EXTEND |            |                                           |
|                                | BZF    | GUILDRET   | # NO. CONTINUE WITH AUTOMATIC LANDING     |
|                                | TCF    | STARTP66   | # YES. SWITCH INTO THE ROD MODE.          |
| RESTART?                       | CA     | FLAGWRD1   | # HAS THERE BEEN A RESTART?               |
|                                | MASK   | RODFLBIT   |                                           |
|                                | EXTEND |            |                                           |
|                                | BZF    | STRTP66A   | # YES. REINITIALIZE BUT LEAVE VDGVERT AS  |
|                                |        |            | # IS.                                     |
|                                | TCF    | VERTGUID   | # NO: CONTINUE WITH R.O.D.                |
| # *****                        |        |            |                                           |
| # INITIALIZATION FOR THIS PASS |        |            |                                           |
| # *****                        |        |            |                                           |
|                                | COUNT* | \$\$/F2DPS |                                           |
| GUILDRET                       | CAF    | ZERO       |                                           |
|                                | TS     | RODCOUNT   |                                           |

# Page 802

+2

```

EXTEND
DCA TPIP
DXCH TPIPOLD

```

```

TC FASTCHNG

```

```

EXTEND
DCA PIPTIME1
DXCH TPIP

```

```

EXTEND
DCA TTF/8
DXCH TTF/8TMP

```

```

CCS FLPASSO
TCF TTFINCR

```

```

BRSPOT1 INDEX WCHPHASE
 TCF NEWPHASE

```

```

ROUTINES TO START NEW PHASES

```

```

P65START TC NEWMODEX
 DEC 65
 CS TWO
 TS WCHVERT
 TC DOWNFLAG # PERMIT X-AXIS OVERRIDE
 ADRES XOVINFLG
 TCF TTFINCR

```

```

STARTP64 TC NEWMODEX
 DEC 64
 CA DELTTFAP # AUGMENT TTF/8
 ADS TTF/8TMP
 CA BIT12 # ENABLE RUPT10
 EXTEND
 WOR CHAN13
 TC DOWNFLAG # INITIALIZE REDESIGNATION FLAG
 ADRES REDFLAG

```

```

(CONTINUE TO TTFINCR)

```

```

INCREMENT TTF/8, UPDATE LAND FOR LUNAR ROTATION, DO OTHER USEFUL THINGS

#
TTFINCR COMPUTATIONS ARE AS FOLLOWS --
Page 803
TTF/8 UPDATED FOR TIME SINCE LAST PASS:
TTF/8 = TTF/8 + (TPIP - TPIPOLD)/8
LANDING SITE VECTOR UPDATED FOR LUNAR ROTATION:
#

LAND = /LAND/ UNIT(LAND - LAND(TPIP - TPIPOLD) * WM)
SLANT RANGE TO LANDING SITE, FOR DISPLAY:
#
RANGEDSP = ABVAL(LAND - R)

TTFINCR TC INTPRET
 DLOAD DSU
 TPIP
 TPIPOLD
 SLR PUSH # SHIFT SCALES DELTA TIME TO 2(17) CSECS
 11D
 VXSC VXV
 LAND
 WM
 BVSU RTB
 LAND
 NORMUNIT
 VXSC VSL1
 /LAND/
 STODL LANDTEMP
 EXIT

 DXCH MPAC
 DAS TTF/8TMP # NOW HAVE INCREMENTED TTF/8 IN TTF/8TMP

 TC FASTCHNG

 EXTEND
 DCA TTF/8TMP
 DXCH TTF/8

 EXTEND
 DCA LANDTEMP
 DXCH LAND
 EXTEND

```

```

DCA LANDTEMP +2
DXCH LAND +2
EXTEND
DCA LANDTEMP +4
DXCH LAND +4

```

# Page 804

```

TC TDISPSET
TC FASTCHNG # SINCE REDESIG MAY CHANGE LANDTEMP

```

```

BRSPOT2 INDEX WCHPHASE
 TCF PREGUIDE

```

```

LANDING SITE PERTURBATION EQUATIONS

```

```

REDESIG CA FLAGWRD6 # IS REDFLAG SET?
 MASK REDFLBIT
 EXTEND
 BZF RGVGCALC # NO: SKIP REDESIGNATION LOGIC

 CA TREDES # YES: HAS TREDES REACHED ZERO?
 EXTEND
 BZF RGVGCALC # YES: SKIP REDESIGNATION LOGIC

 INHINT
 CA ELINCR1
 TS ELINCR
 CA AZINCR1
 TS AZINCR
 TC FASTCHNG

 CA ZERO
 TS ELINCR1
 TS AZINCR1
 TS ELINCR +1
 TS AZINCR +1

 CA FIXLOC # SET PD TO 0
 TS PUSHLOC

 TC INTERPRET
 VLOAD VSU
 LAND
 R # ---- -

```

```

RTB PUSH # PUSH DOWN UNIT (LAND - R)
 NORMUNIT
VXV VSL1
 YNBPIP #
VXSC PDDL # PUSH DOWN - ELINCR(--- * UNIT(---- - R))
 ELINCR
 AZINCR
VXSC VSU
 YNBPIP
VAD PUSH # RESULTING VECTOR IS 1/2 REAL SIZE

Page 805

DLOAD DSU # MAKE SURE REDESIGNATION IS NOT
 0 # TOO CLOSE TO THE HORIZON.
 DEPRCRIT
BMN DLOAD
 REDES1
 DEPRCRIT
STORE 0
REDES1 DLOAD #
 DSU
 LAND
 R
DDV VXSC
 0
VAD UNIT
 R
VXSC VSL1
 /LAND/
STORE LANDTEMP
EXIT # LOOKANGL WILL BE COMPUTED AT RGVGCALC

TC FASTCHNG

EXTEND
DCA LANDTEMP
DXCH LAND
EXTEND
DCA LANDTEMP +2
DXCH LAND +2
EXTEND
DCA LANDTEMP +4
DXCH LAND +4

TCF RGVGCALC

```



```

COMPUTE STATE IN GUIDANCE COORDINATES

#
RGVGCALC COMPUTATIONS ARE AS FOLLOWS:--
VELOCITY RELATIVE TO THE SURFACE
#
----- - - - -
ANGTERM = V + R * WM
STATE IN GUIDANCE COORDINATES:
#
* - - - -
RGU = CG (R - LAND)
#
* - - - -
VGU = CG (V - WM * R)
Page 806 actually starts one line earlier but that would separate the markers from their vari
#
HORIZONTAL VELOCITY FOR DISPLAY
#
VHORIZ = 8 ABVAL (0, VG , VG)
2 1
#
DEPRESSION ANGLE FOR DISPLAY:
#
LOOKANGL = ARCSIN(UNIT(R - LAND).XMBPIP)
#
CALCRGVG TC INTPRET # IN IGNALG, COMPUTE V FROM INTEGRATION
 VLOAD MXV # OUTPUT AND TRIM CORRECTION TERM
 VATT1 # COMPUTED LAST PASS AND LEFT IN UNFC/2
 REFSMMAT
 VSR1 VAD
 UNFC/2
 STORE V
 EXIT

RGVGCALC TC INTPRET # ENTER HERE TO RECOMPUTE RG AND VG
 VLOAD VXV
 R
 WM
 VAD VSR2 # RESCALE TO UNITS OF 2(9) M/CS
 V
 STORE ANGTERM
 MXV
 CG # NO SHIFT SINCE ANGTERM IS DOUBLE SIZED
 STORE VGU
 PDDL VDEF # FORM (0,VG ,VG) IN UNITS OF 2(10) M/CS
 ZEROVECS # 2 1
 ABVAL SL3
 STOVL VHORIZ # VHORIZ FOR DISPLAY DURING P65.

```

```

 R
VSU PUSH #
 LAND # PUSH DOWN R - LAND
 VSL1
MXV CG
STORE RGU
ABVAL
STOVL RANGEDSP
RTB DOT # NOW IN MPAC IS SINE(LOOKANGL)/4
 NORMUNIT
 XNBPIP
EXIT

CA FIXLOC # RESET PUSH DOWN POINTER
TS PUSHLOC

Page 807

CA MPAC # COMPUTE LOOKANGLE ITSELF
DOUBLE
TC BANKCALL
CADR SPARCSIN -1
AD 1/2DEG
EXTEND
MP 180DEGS
TS LOOKANGL # LOOKANGL FOR DISPLAY DURING P64

BRSPOT3 INDEX WCHPHASE
 TCF WHATGUID

TTF/8 COMPUTATION

TTF/8CL TC INTERPTX
 DLOAD*
 JDG2TTF,1
STODL* TABLTTF +6 # A(3) = 8 JDG TO TABLTTF
 ADG2TTF,1 #
 STODL TABLTTF +4 # A(2) = 6 ADG TO TABLTTF
 VGU +4 #
 DMP DAD*
 3/4DP
 VDG2TTF,1
STODL* TABLTTF +2 # A(1) = (6 VGU + 18 VDG)/8 TO TABLTTF
 RDG +4,1 #
 DSU DMP

```

```

 RGU +4
 3/8DP
STORE TABLTTF # A(0) = -24 (RGU - RDG)/64 TO TABLTTF
EXIT # 2 2

CA BIT8
TS TABLTTF +10 # FRACTIONAL PRECISION FOR TTF TO TABLE

EXTEND
DCA TTF/8
DXCH MPAC # LOADS TTF/8 (INITIAL GUESS) INTO MPAC
CAF TWO # DEGREE - ONE
TS L
CAF TABLTTF L
TC ROOTPSRS # YIELDS TTF/8 IN MPAC
INDEX WCHPHASE
TCF WHATALM

EXTEND # GOOD RETURN
DCA MPAC # FETCH TTF/8 KEEPING IT IN MPAC
DXCH TTF/8 # CORRECTED TTF/8

Page 808
TC TDISPSET

(CONTINUE TO QUADGUID)

MAIN GUIDANCE EQUATION

#
AS PUBLISHED --
#
--- --- --- ---
6(VDG + VG) 12(RDG - RG)
ACG = ADG + ----- + -----
TTF (TTF)(TTF)
#
AS HERE PROGRAMMED --
#
--- ---
3 (1/4(RDG - RG) --- --)
- (----- + VDG + VG)
4 (TTF/8)
ACG = ----- + ADG
TTF/8
#
QUADGUID CS TTF/8
 AD LEADTIME # LEADTIME IS A NEGATIVE NUMBER

```

|        |          |                                          |
|--------|----------|------------------------------------------|
| AD     | POSMAX   | # SAFEGUARD THE COMPUTATIONS THAT FOLLOW |
| TS     | L        | # BY FORCING $-TTF*LEADTIME > OR = ZERO$ |
| CS     | L        |                                          |
| AD     | L        |                                          |
| ZL     |          |                                          |
| EXTEND |          |                                          |
| DV     | TTF/8    |                                          |
| TS     | BUF      | # - RATIO OF LAG-DIMINISHED TTF TO TTF   |
| EXTEND |          |                                          |
| SQUARE |          |                                          |
| TS     | BUF +1   |                                          |
| AD     | BUF      |                                          |
| XCH    | BUF +1   | # RATIO SQUARED - RATIO                  |
| AD     | BUF +1   |                                          |
| TS     | MPAC     | # COEFFICIENT FOR VGU TERM               |
| AD     | BUF +1   |                                          |
| INDEX  | FIXLOC   |                                          |
| TS     | 26D      | # COEFFICIENT FOR RDG-RGU TERM           |
| AD     | BUF +1   |                                          |
| INDEX  | FIXLOC   |                                          |
| TS     | 28D      | # COEFFICIENT FOR VDG TERM               |
| AD     | BUF      |                                          |
| AD     | POSMAX   |                                          |
|        |          |                                          |
| AD     | BUF +1   |                                          |
| AD     | BUF +1   |                                          |
| INDEX  | FIXLOC   |                                          |
| TS     | 30D      | # COEFFICIENT FOR ADG TERM               |
|        |          |                                          |
| CAF    | ZERO     |                                          |
| TS     | MODE     |                                          |
|        |          |                                          |
| TC     | INTPRETX |                                          |
| VXSC   | PDDL     |                                          |
|        | VGU      |                                          |
|        | 28D      |                                          |
| VXSC*  | PDVL*    |                                          |
|        | VDG,1    |                                          |
|        | RDG,1    |                                          |
| VSU    | V/SC     |                                          |
|        | RGU      |                                          |
|        | TTF/8    |                                          |
| VSR2   | VXSC     |                                          |
|        | 26D      |                                          |
| VAD    | VAD      |                                          |
| V/SC   | VXSC     |                                          |

```

 TTF/8
 3/4DP
 PDDL VXSC*
 30D
 ADG, 1
 VAD
AFCCALC1 VXM VSL1 # VERGUID COMES HERE
 CG
 PDVL V/SC
 GDT/2
 GSCALE
 BVSU STADR
 STORE UNFC/2 # UNFC/2 NEED NOT BE UNITIZED
 ABVAL
AFCCALC2 STODL /AFC/ # MAGNITUDE OF AFC FOR THROTTLE
 UNFC/2 # VERTICAL COMPONENT
 DSQ PDDL
 UNFC/2 +2 # OUT-OF-PLANE
 DSQ PDDL
 HIGHESTF
 DDV DSQ
 MASS #
 DSU DSU # AMAXHORIZ = SQRT(ATOTAL - A2 - A2)
 BPL DLOAD #
 AFCCALC3 1 0
 ZEROVECS
AFCCALC3 SQRT DAD
 UNFC/2 +4
Page 810
 BPL BDSU
 AFCCLEND
 UNFC/2 +4
 STORE UNFC/2 +4
AFCCLEND EXIT
 TC FASTCHNG
 CA WCHPHASE # PREPARE FOR PHASE SWITCHING LOGIC
 TS WCHPHOLD
 INCR FLPASSO # INCREMENT PASS COUNTER

BRSPOT4 INDEX WCHPHASE
 TCF AFTRGUID

```

```

ERECT GUIDANCE-STABLE MEMBER TRANSFORMATION MATRIX

```

```

CGCALC CAF EBANK5
 TS EBANK
 EBANK= TCGIBRAK
 EXTEND
 INDEX WCHPHASE
 INDEX TARGTDEX
 DCA TCGFBRAK
 INCR BBANK
 INCR BBANK
 EBANK= TTF/8
 AD TTF/8
 XCH L
 AD TTF/8
 CCS A
 CCS L
 TCF EXTLOGIC
 TCF EXTLOGIC
 NOOP

 TC INTERPX
 VLOAD UNIT
 LAND
 STODL CG
 TTF/8
 DMP* VXSC
 GAINBRAK,1 # NUMERO MYSTERIOSO
 ANGTERM

 VAD
 LAND
 VSU RTB
 R
 NORMUNIT

Page 811

 VXV RTB
 LAND
 NORMUNIT
 STOVL CG +6 # SECOND ROW
 CG
 VXV VSL1
 CG +6
 STORE CG +14
 EXIT

(CONTINUE TO EXTLOGIC)
#

```

```

PREPARE TO EXIT

#
DECIDE (1) HOW TO EXIT, AND (2) WHETHER TO SWITCH PHASES
#
EXTLOGIC INDEX WCHPHASE # WCHPHASE = 1 APPRQUAD
 CA TENDBRAK # WCHPHASE = 0 BRAKQUAD
 AD TTF/8
EXSPOT1 EXTEND
 INDEX WCHPHASE
 BZMF WHATEXIT
 TC FASTCHNG
 CA WCHPHOLD
 AD ONE
 TS WCHPHASE
 CA ZERO
 TS FLPASSO # RESET FLPASSO
 INDEX WCHPHOLD
 TCF WHATEXIT

ROUTINES FOR EXITING FROM LANDING GUIDANCE

#
1. EXGSUB IS THE RETURN WHEN GUIDSUB IS CALLED BY THE IGNITION ALGORITHM.
2. EXBRAK IN THE EXIT USED DURING THE BRAKING PHASE. IN THIS CASE UNIT(R) IS THE WINDOW P
3. EXNORM IS THE EXIT USED AT OTHER TIMES DURING THE BURN.
(EXOVFLOW IS A SUBROUTINE OF EXBRAK AND EXNORM CALLED WHEN OVERFLOW OCCURRED ANYWHERE IN GUID

EXGSUB TC INTERPRET # COMPUTE TRIM VELOCITY CORRECTION TERM.
Page 812
 VLOAD RTB
 UNFC/2
 NORMUNIT
 VXSC VXSC
 ZOOMTIME
 TRIMACCL
 STORE UNFC/2
 EXIT
 CCS NGUIDSUB

```

|            |       |          |                                          |
|------------|-------|----------|------------------------------------------|
|            | TCF   | GUIDSUB  |                                          |
|            | CCS   | NIGNLOOP |                                          |
|            | TCF   | +3       |                                          |
|            | TC    | ALARM    |                                          |
|            | OCT   | 01412    |                                          |
| +3         | TC    | POSTJUMP |                                          |
|            | CADR  | DDUMCALC |                                          |
| EXBRAK     | TC    | INTPRET  |                                          |
|            | VLOAD |          |                                          |
|            |       | UNIT/R/  |                                          |
|            | STORE | UNWC/2   |                                          |
|            | EXIT  |          |                                          |
|            | TCF   | STEER?   |                                          |
| EXNORM     | TC    | INTPRET  |                                          |
|            | VLOAD | VSU      |                                          |
|            |       | LAND     |                                          |
|            |       | R        |                                          |
|            | RTB   |          |                                          |
|            |       | NORMUNIT |                                          |
|            | STORE | UNWC/2   | # UNIT(LAND - R) IS TENTATIVE CHOICE     |
|            | VXV   | DOT      |                                          |
|            |       | XNBPIP   |                                          |
|            |       | CG +6    |                                          |
|            | EXIT  |          | # WITH PROJ IN MPAC 1/8 REAL SIZE        |
|            | CS    | MPAC     | # GET COEFFICIENT FOR CG +14             |
|            | AD    | PROJMAX  |                                          |
|            | AD    | POSMAX   |                                          |
|            | TS    | BUF      |                                          |
|            | CS    | BUF      |                                          |
|            | ADS   | BUF      | # RESULT IS 0 IF PROJMAX - PROJ NEGATIVE |
|            | CS    | PROJMIN  | # GET COEFFICIENT FOR UNIT(LAND - R)     |
|            | AD    | MPAC     |                                          |
|            | AD    | POSMAX   |                                          |
|            | TS    | BUF +1   |                                          |
|            | CS    | BUF +1   |                                          |
| # Page 813 | ADS   | BUF +1   | # RESULT IS 0 IF PROJ - PROJMIN NEGATIVE |
| UNWCLOOP   | CAF   | FOUR     |                                          |
|            | MASK  | SIX      |                                          |
|            | TS    | Q        |                                          |



July 12, 2016

921

```

CA EBANK5
TS EBANK
EBANK= CG
CA BUF
EXTEND
INDEX Q
MP CG +14
INCR BBANK
EBANK= UNWC/2
INDEX Q
DXCH UNWC/2
EXTEND
MP BUF +1
INDEX Q
DAS UNWC/2
CCS Q
TCF UNWCLOOP

INCR BBANK
EBANK= PIF

STEER? CA FLAGWRD2 # IF STEERSW DOWN NO OUTPUTS
 MASK STEERBIT
 EXTEND
 BZF RATESTOP

EXVERT CA OVFFIND # IF OVERFLOW ANYWHERE IN GUIDANCE
 EXTEND # DON'T CALL THROTTLE OR FINDCDUW
 BZF +13

EXOVFLOW TC ALARM # SOUND THE ALARM NON-ABORTIVELY
 OCT 01410

RATESTOP CAF BIT13 # ARE WE IN ATTITUDE-HOLD?
 EXTEND
 RAND CHAN31
 EXTEND
 BZF DISPEXIT # YES

 TC BANKCALL # NO: DO A STOPRATE
 CADR STOPRATE

 TCF DISPEXIT

GDUMP1 TC THROTTLE
Page 814
```

```

TC INTPRET
CALL
 FINDCDUW -2
EXIT

(CONTINUE TO DISPEXIT)

GUIDANCE LOOP DISPLAYS

DISPEXIT EXTEND # KILL GROUP 3: DISPLAYS WILL BE
DCA NEG0 # RESTORED BY NEXT GUIDANCE CYCLE.
DXCH -PHASE3

+3 CS FLAGWRD8 # IF FLUNDISP IS SET, NO DISPLAY THIS PASS
MASK FLUNDBIT
EXTEND
BZF ENDLLJOB # TO PICK UP THE TAG

INDEX WCHPHOLD
TCF WHATDISP

-2 TC PHASCHNG # KILL GROUP 5
OCT 00035

P63DISPS CAF V06N63
DISPCOMN TC BANKCALL
CADR REGODSPR

ENDLLJOB TCF ENDOFJOB

P64DISPS CA TREDES # HAS TREDES REACHED ZERO?
EXTEND
BZF RED-OVER # YES: CLEAR REDESIGNATION FLAG

CS FLAGWRD6 # NO: IS REDFLAG SET?
MASK REDFLBIT
EXTEND
BZF REDES-OK # YES: DO STATIC DISPLAY

CAF V06N64 # OTHERWISE USE FLASHING DISPLAY
TC BANKCALL
CADR REFLASHR
TCF GOTOP00H # TERMINATE
TCF P64CEED # PROCEED PERMIT REDESIGNATIONS

```

```

Page 815 TCF P64DISPS # RECYCLE

 TCF ENDLLJOB

P64CEED CAF ZERO
 TS ELINCR1
 TS AZINCR1

 TC UPFLAG # ENABLE REDESIGNATION LOGIC
 ADRES REDFLAG

 TCF ENDOFJOB

RED-OVER TC DOWNFLAG
 ADRES REDFLAG

REDES-OK CAF V06N64
 TCF DISPCOMN

VERTDISP CAF V06N60
 TCF DISPCOMN

```

```

GUIDANCE FOR P65

```

```

VERTGUID CCS WCHVERT
 TCF P67VERT # POSITIVE NON-ZERO ---> P67
 TCF P66VERT # +0

```

```

#
THE P65 GUIDANCE EQUATION IS AS FOLLOWS --

```

```

---- ---
V2FG - VGU
ACG = -----
TAUVERT

```

```

P65VERT TC INTPRET
 VLOAD VSU
 V2FG
 VGU
 V/SC GOTO
 TAUVERT
 AFCCALC1

```

```

Page 816

```

```

```

# GUIDANCE FOR P66

# \*\*\*\*\*

|         |      |          |
|---------|------|----------|
| P66VERT | TC   | POSTJUMP |
|         | CADR | P66VERTA |

|         |     |          |                      |
|---------|-----|----------|----------------------|
| P67VERT | TC  | PHASCHNG | # TERMINATE GROUP 3. |
|         | OCT | 00003    |                      |

|       |          |
|-------|----------|
| TC    | INTPRET  |
| VLOAD | GOTO     |
|       | V        |
|       | VHORCOMP |

|        |            |
|--------|------------|
| SETLOC | P66LOC     |
| BANK   |            |
| COUNT* | \$\$/F2DPS |

|         |        |         |
|---------|--------|---------|
| RODTASK | CAF    | PRI022  |
|         | TC     | FINDVAC |
|         | EBANK= | DVCNTR  |
|         | 2CADR  | RODCOMP |

|     |          |
|-----|----------|
| TCF | TASKOVER |
|-----|----------|

|          |     |          |                      |
|----------|-----|----------|----------------------|
| P66VERTA | TC  | PHASCHNG | # TERMINATE GROUP 3. |
|          | OCT | 00003    |                      |

|       |         |
|-------|---------|
| CAF   | 1SEC    |
| TC    | TWIDDLE |
| ADRES | RODTASK |

|         |          |
|---------|----------|
| RODCOMP | INHINT   |
|         | CAF      |
|         | XCH      |
|         | EXTEND   |
|         | MP       |
|         | DAS      |
|         | ZERO     |
|         | RODCOUNT |
|         | RODSCAL1 |
|         | VDGVERT  |

# UPDATE DESIRED ALTITUDE RATE.

|        |  |                                |
|--------|--|--------------------------------|
| EXTEND |  | # SET OLDPIPAX,Y,Z = PIPAX,Y,Z |
|--------|--|--------------------------------|

|      |          |
|------|----------|
| DCA  | PIPAX    |
| DXCH | OLDPIPAX |
| DXCH | RUPTREG1 |
| CA   | PIPAZ    |
| XCH  | OLDPIPAZ |
| XCH  | RUPTREG3 |

# SET RUPTREG1,2,3 = OLDPIPAX,Y,Z

```

Page 817
EXTEND
DCA TIME2 # SHAPSHOT TIME OF PIPA READING.

DXCH THISTPIP

CA OLDPIPAX
AD PIPATMPX
TS MPAC # MPAC(X) = PIPAX + PIPATMPX
CA OLDPIPAY
AD PIPATMPY
TS MPAC +3 # MPAC(Y) = PIPAY + PIPATMPY
CA OLDPIPAZ
AD PIPATMPZ
TS MPAC +5 # MPAC(Z) = PIPAZ + PIPATMPZ

CS OLDPIPAX
AD TEMX
AD RUPTREG1
TS DELVROD
CS OLDPIPAY
AD TEMY
AD RUPTREG2
TS DELVROD +2
CS OLDPIPAZ
AD TEMZ
AD RUPTREG3
TS DELVROD +4

CAF ZERO
TS MPAC +1 # ZERO LO-ORDER MPAC COMPONENTS
TS MPAC +4
TS MPAC +6
TS TEMX # ZERO TEMX, TEMY, AND TEMZ SO WE WILL
TS TEMY # KNOW WHEN READACCS CHANGES THEM.
TS TEMZ
CS ONE
TS MODE
TC INTERPRET
ITRPNT1 VXSC PDDL # SCALE MPAC TO M/CS *2(-7) AND PUSH (6)
 KPIP1
 THISTPIP
DSU
 PIPTIME
STORE 30D # 30-31D CONTAINS TIME IN CS SINCE PIPTIME
DDV PDVL # (8)

```

# Page 818

|       |          |   |                                     |
|-------|----------|---|-------------------------------------|
|       | 4SEC(28) |   |                                     |
|       | GDT/2    |   |                                     |
| VSU   | VXSC     | # | (6)                                 |
|       | VBIAS    |   |                                     |
| VSL2  | VAD      |   |                                     |
|       | V        |   |                                     |
| VAD   | STADR    | # | (0)                                 |
| STOVL | 24D      | # | # STORE UPDATED VELOCITY IN 24-29D  |
|       | R        |   |                                     |
| UNIT  |          |   |                                     |
| STORE | 14D      |   |                                     |
| DOT   | SL1      |   |                                     |
|       | 24D      |   |                                     |
| STODL | HDOTDISP | # | # UPDATE HDOTDISP RATE FOR NOUN 63. |
|       | 30D      |   |                                     |
| SL    | DMP      |   |                                     |
|       | 11D      |   |                                     |
|       | HDOTDISP |   |                                     |
| DAD   | DSU      |   |                                     |
|       | 36D      |   |                                     |
|       | /LAND/   |   |                                     |
| STODL | HCALC1   | # | # UPDATE HCALC1 FOR NOUN 63.        |
|       | HDOTDISP |   |                                     |
| BDSU  | DDV      |   |                                     |
|       | VDGVERT  |   |                                     |
|       | TAUROD   |   |                                     |
| PDVL  | ABVAL    | # | (2)                                 |
|       | GDT/2    |   |                                     |
| DDV   | SR2      |   |                                     |
|       | GSCALE   |   |                                     |
| STORE | 20D      |   |                                     |
| DAD   |          | # | (0)                                 |
| PDVL  | CALL     | # | (2)                                 |
|       | UNITX    |   |                                     |
|       | CDU*NBSM |   |                                     |
| DOT   |          |   |                                     |
|       | 14D      |   |                                     |
| STORE | 22D      |   |                                     |
| BDDV  | STADR    | # | (0)                                 |
| STOVL | /AFC/    |   |                                     |
|       | DELVRD   |   |                                     |
| VXSC  | VAD      |   |                                     |
|       | KPIP1    |   |                                     |
|       | VBIAS    |   |                                     |
| ABVAL | PDDL     | # | (2)                                 |

|            |       |                      |                                 |                  |
|------------|-------|----------------------|---------------------------------|------------------|
|            | DSU   | THISTPIP<br>PDDL     | #                               | (4)              |
|            |       | LASTTPIP             |                                 |                  |
|            | STODL | THISTPIP<br>LASTTPIP | #                               | (2)              |
|            | DDV   | BDDV                 | #                               | (0)              |
|            |       | SHFTFACT             |                                 |                  |
|            | PDDL  | DMP                  | #                               | (2)              |
|            |       | FWEIGHT              |                                 |                  |
|            |       | BIT1H                |                                 |                  |
|            | DDV   | DDV                  |                                 |                  |
|            |       | MASS                 |                                 |                  |
|            |       | SCALEFAC             |                                 |                  |
| # Page 819 | DAD   | PDDL                 | #                               | (4)              |
|            |       | OD                   |                                 |                  |
|            |       | 20D                  |                                 |                  |
|            | DDV   | DSU                  | #                               | (2)              |
|            |       | 22D                  |                                 |                  |
|            | DMP   | DAD                  |                                 |                  |
|            |       | LAG/TAU              |                                 |                  |
|            |       | /AFC/                |                                 |                  |
|            | PDDL  | DDV                  | #                               | (4)              |
|            |       | MAXFORCE             |                                 |                  |
|            |       | MASS                 |                                 |                  |
|            | PDDL  | DDV                  | #                               | (6)              |
|            |       | MINFORCE             |                                 |                  |
|            |       | MASS                 |                                 |                  |
|            | PUSH  | BDSU                 | #                               | (8)              |
|            |       | 2D                   |                                 |                  |
|            | BMN   | DLOAD                | #                               | (6)              |
|            |       | AFCSPOT              |                                 |                  |
|            | DLOAD | PUSH                 | #                               | (6)              |
|            | BDSU  | BPL                  |                                 |                  |
|            |       | 2D                   |                                 |                  |
|            |       | AFCSPOT              |                                 |                  |
|            | DLOAD |                      | #                               | (4)              |
| AFCSPOT    | DLOAD |                      | #                               | (2), (4), OR (6) |
|            | SETPD |                      | #                               | (2)              |
|            |       | 2D                   |                                 |                  |
|            | STODL | /AFC/                | #                               | (0)              |
| ITRPNT2    | EXIT  |                      |                                 |                  |
|            | DXCH  | MPAC                 | # MPAC = MEASURED ACCELARATION. |                  |
|            | TC    | BANKCALL             |                                 |                  |
|            | CADR  | THROTTLE +3          |                                 |                  |
|            | TC    | INTPRET              |                                 |                  |

```

 VLOAD # PICK UP UPDATED VELOCITY VECTOR.
 24D
VHORCOMP VSL2 VAD
 DELVS
 VSR2 PDVL
 R
 UNIT VXSC
 HDOTDISP
 VSL1 BVSU
 ABVAL
 STORE VHORIZ
 EXIT
 TC BANKCALL # PUT UP V06N60 DISPLAY BUT AVOID PHASCHNG
 CADR DISPEXIT +3

BIT1H OCT 00001
SHFTFACT 2DEC 1 B-17
Page 820
BIASFACT 2DEC 655.36 B-28

REDESIGNATOR TRAP

 BANK 11
 SETLOC F2DPS*11
 BANK

 COUNT* $$/F2DPS

PITFALL XCH BANKRUPT
 EXTEND
 QXCH QRUPT

 TC CHECKMM # IF NOT IN P64, NO REASON TO CONTINUE
 DEC 64
 TCF RESUME

 EXTEND
 READ CHAN31
 COM
 MASK ALL4BITS
 TS ELVIRA
 CAF TWO
 TS ZERLINA
 CAF FIVE

```



|       |          |
|-------|----------|
| TC    | TWIDDLE  |
| ADRES | REDESMON |
| TCF   | RESUME   |

## # REDESIGNATOR MONITOR (INITIATED BY PITFALL)

|          |        |          |
|----------|--------|----------|
| PREMON1  | TS     | ZERLINA  |
| PREMON2  | CAF    | SEVEN    |
|          | TC     | VARDELAY |
| REDESMON | EXTEND |          |
|          | READ   | 31       |
|          | COM    |          |
|          | MASK   | ALL4BITS |
|          | XCH    | ELVIRA   |
|          | TS     | L        |
|          | CCS    | ELVIRA   |
|          | TCF    | PREMON2  |

# DO ANY BITS APPEAR THIS PASS?

# Y: CONTINUE MONITOR

|     |          |
|-----|----------|
| CCS | L        |
| TCF | COUNT'EM |

# N: ANY LAST PASS?

# Y: COUNT 'EM, RESET RUPT, TERMINATE

# Page 821

|          |        |          |
|----------|--------|----------|
|          | CCS    | ZERLINA  |
|          | TCF    | PREMON1  |
| RESETRPT | CAF    | BIT12    |
|          | EXTEND |          |
|          | WOR    | CHAN13   |
|          | TCF    | TASKOVER |

# N: HAS ZERLINA REACHED ZERO YET?

# N: DIMINISH ZERLINA, CONTINUE

# Y: RESET RUPT. TERMINATE

|          |        |          |
|----------|--------|----------|
| COUNT'EM | CAF    | BIT13    |
|          | EXTEND |          |
|          | RAND   | CHAN31   |
|          | EXTEND |          |
|          | BZF    | RESETRPT |

# ARE WE IN ATTITUDE-HOLD?

# YES: SKIP REDESIGNATION LOGIC.

|      |         |
|------|---------|
| CA   | L       |
| MASK | -AZBIT  |
| CCS  | A       |
| CS   | AZEACH  |
| ADS  | AZINCR1 |

# NO.

-AZ

|      |         |
|------|---------|
| CA   | L       |
| MASK | +AZBIT  |
| CCS  | A       |
| CA   | AZEACH  |
| ADS  | AZINCR1 |
| CA   | L       |
| MASK | -ELBIT  |

+AZ

|     |      |          |  |
|-----|------|----------|--|
|     | CCS  | A        |  |
| -EL | CS   | ELEACH   |  |
|     | ADS  | ELINCR1  |  |
|     | CA   | L        |  |
|     | MASK | +ELBIT   |  |
|     | CCS  | A        |  |
| +EL | CA   | ELEACH   |  |
|     | ADS  | ELINCR1  |  |
|     | TCF  | RESETRPT |  |

# THESE EQUIVALENCES ARE BASED ON GSOP CHAPTER 4, REVISION 16 OF P64LM

|        |   |      |          |
|--------|---|------|----------|
| +ELBIT | = | BIT2 | # -PITCH |
| -ELBIT | = | BIT1 | # +PITCH |
| +AZBIT | = | BIT5 |          |
| -AZBIT | = | BIT6 |          |

# Page 822

|          |     |        |              |
|----------|-----|--------|--------------|
| ALL4BITS | OCT | 00063  |              |
| AZEACH   | DEC | .03491 | # 2 DEGREES  |
| ELEACH   | DEC | .00873 | # 1/2 DEGREE |

# \*\*\*\*\*

# R.O.D. TRAP

# \*\*\*\*\*

|          |        |            |                                           |
|----------|--------|------------|-------------------------------------------|
|          | BANK   | 20         |                                           |
|          | SETLOC | RODTRAP    |                                           |
|          | BANK   |            |                                           |
|          | COUNT* | \$\$/F2DPS | # *****                                   |
| DESCBITS | MASK   | BIT7       | # COME HERE FROM MARKRUPT CODING WITH BIT |
|          | CCS    | A          | # 7 OR 6 OF CHANNEL 16 IN A; BIT 7 MEAN   |
|          | CS     | TWO        | # - RATE INCREMENT, BIT 6 + INCREMENT.    |
|          | AD     | ONE        |                                           |
|          | ADS    | RODCOUNT   |                                           |
|          | TCF    | RESUME     | # TRAP IS RESET WHEN SWITCH IS RELEASED   |
|          | BANK   | 31         |                                           |
|          | SETLOC | F2DPS*31   |                                           |
|          | BANK   |            |                                           |
|          | COUNT* | \$\$/F2DPS |                                           |

# \*\*\*\*\*

# DOUBLE PRECISION ROOT FINDER SUBROUTINE (BY ALLAN KLUMPP)

```

```

```
#
```

```
#
```

```
N N-1
```

```
ROOTPSRS FINDS ONE ROOT OF THE POWER SERIES A X + A X + ... + A X + A
```

```
N N-1 1 0
```

```
USING NEWTON'S METHOD STARTING WITH AN INITIAL GUESS FOR THE ROOT. THE ENTERING DATA MUST BE
```

```
A SP LOC-3 ADRES FOR REFERENCING PWR COF TABL
```

```
L SP N-1 N IS THE DEGREE OF THE POWER SERIES
```

```
MPAC DP X INITIAL GUESS FOR ROOT
```

```
#
```

```
LOC-2N DP A(0)
```

```
#
```

```
LOC DP A(N)
```

```
LOC+2 SP PRECROOT PREC RQD OF ROOT (AS FRACT OF 1ST GUESS)
```

```
#
```

```
Page 823
```

```
THE DP RESULT IS LEFT IN MPAC UPON EXIT, AND A SP COUNT OF THE ITERATIONS TO CONVERGENCE IS I
```

```
RETURN IS NORMALLY TO LOC(TC ROOTPSRS)+3. IF ROOTPSRS FAILS TO CONVERGE TO IN 8 PASSES, RETU
```

```
OUTPUTS ARE NOT TO BE TRUSTED.
```

```
#
```

```
PRECAUTION: ROOTPSRS MAKES NO CHECKS FOR OVERFLOW OR FOR IMPROPER USAGE. IMPROPER USAGE COU
```

```
PRECLUDE CONVERGENCE OR REQUIRE EXCESSIVE ITERATIONS. AS A SPECIFIC EXAMPLE, ROOTPSRS FORMS
```

```
COEFFICIENT TABLE BY MULTIPLYINE EACH A(I) BY I, WHERE I RANGES FROM 1 TO N. IF AN ELEMENT O
```

```
COEFFICIENT TABLE = 1 OR >1 IN MAGNITUDE, ONLY THE EXCESS IS RETAINED. ROOTPSRS MAY CONVERGE
```

```
ROOT NONETHELESS, BUT IT MAY TAKE AN EXCESSIVE NUMBER OF ITERATIONS. THEREFORE THE USER SHOU
```

```
1. USER'S RESPONSIBILITY TO ASSUR THAT I X A(I) < 1 IN MAGNITUDE FOR ALL I.
```

```
2. USER'S RESPONSIBILITY TO ASSURE OVERFLOW WILL NOT OCCUR IN EVALUTATING EITHER THE R
```

```
POWER SERIES. THIS OVERFLOW WOULD BE PRODUCED BY SUBROUTINE POWRSERS, CALLED BY RO
```

```
PRECLUDE EVENTUAL CONVERGENCE.
```

```
3. AT PRESENT, ERASABLE LOCATIONS ARE RESERVED ONLY FOR N UP TO 5. AN N IN EXCESS OF
```

```
ALL ERASABLES USED BY ROOTPSRS ARE UNSWITCHED LOCATED IN THE REGION FROM MPAC-33 OC
```

```
4. THE ITERATION COUNT RETURNED IN MPAC+2 MAY BE USED TO DETECT ABNORMAL PERFORMANCE.
```

```
#
```

```
STORE ENTERING DATA, INITIALIZE ERASABLES
```

```
ROOTPSRS EXTEND
```

```
QXCH RETROOT # RETURN ADRES
```

```
TS PWRPTR # PWR TABLE POINTER
```

```
DXCH MPAC +3 # PWR TABLE ADRES, N-1
```

```
CA DERTABLL
```

```
TS DERPTR # DER TABL POINTER
```

```
TS MPAC +5 # DER TABL ADRES
```

```
CCS MPAC +4 # NO POWER SERIES DEGREE 1 OR LESS
```

```
TS MPAC +6 # N-2
```

```
CA ZERO # MODE USED AS ITERATION COUNTER. MODE
```

```
TS MODE # MUST BE POS SO ABS WON'T COMP MPAC+3 ETC.
```

```

COMPUTE CRITERION TO STOP ITERATING
EXTEND
DCA MPAC # FETCH ROOT GUESS, KEEPING IT IN MPAC
DXCH ROOTPS # AND IN ROOTPS
INDEX MPAC +3 # PWR TABLE ADRES
CA 5 # PRECROOT TO A
TC SHORTMP # YIELDS DP PRODUCT IN MPAC
TC USPRCADR
CADR ABS # YIELDS ABVAL OF CRITERION ON DX IN MPAC
DXCH MPAC
DXCH DXCRIT # CRITERION

SET UP DER COF TABL

Page 824
EXTEND
INDEX PWRPTR
DCA 3
DXCH MPAC # A(N) TO MPAC

CA MPAC +4 # N-1 TO A

DERCLOOP TS PWRCNT # LOOP COUNTER
AD ONE
TC DMPNSUB # YIELDS DERCOF = I X A(I) IN MPAC
EXTEND
INDEX PWRPTR
DCA 1
DXCH MPAC # (I-1) TO MPAC, FETCHING DERCOF
INDEX DERPTR
DXCH 3 # DERCOF TO DER TABLE
CS TWO
ADS PWRPTR # DECREMENT PWR POINTER
CS TWO
ADS DERPTR # DECREMENT DER POINTER
CCS PWRCNT
TCF DERCLOOP

CONVERGE ON ROOT
ROOTLOOP EXTEND
DCA ROOTPS # FETCH CURRENT ROOT
DXCH MPAC # LEAVE IN MPAC
EXTEND
DCA MPAC +5 # LOAD A, L WITH DER TABL ADRES, N-2
TC POWRSERS # YIELDS DERIVATIVE IN MPAC
EXTEND

```

```

DCA ROOTPS
DXCH MPAC # CURRENT ROOT TO MPAC, FETCHING DERIVATIVE
DXCH BUF # LEAVE DERIVATIVE IN BUF AS DIVISOR
EXTEND
DCA MPAC +3 # LOAD A, L WITH PWR TABL ADRES, N-1
TC POWRSERS # YIELDS RESIDUAL IN MPAC

TC USPRCADR
CADR DDV/BDDV # YIELDS -DX IN MPAC

EXTEND
DCS MPAC # FETCH DX, LEAVING -DX IN MPAC
DAS ROOTPS # CORRECTED ROOT NOW IN ROOTPS

TC USPRCADR
CADR ABS # YIELDS ABS(DX) IN MPAC
EXTEND

Page 825
DCS DXCRIT
DAS MPAC # ABS(DX)-ABS(DXCRIT) IN MPAC

CA MODE
MASK BIT4 # KLUMPP SAYS GIVE UP AFTER EIGHT PASSES
CCS A
BADROOT TC RETROOT

INCR MODE # INCREMENT ITERATION COUNTER
CCS MPAC # TEST HI ORDER DX
TCF ROOTLOOP
TCF TESTLODX
TCF ROOTSTOR
TESTLODX CCS MPAC +1 # TEST LO ORDER DX
TCF ROOTLOOP
TCF ROOTSTOR
TCF ROOTSTOR
ROOTSTOR DXCH ROOTPS
DXCH MPAC
CA MODE
TS MPAC +2 # STORE SP ITERATION COUNT IN MPAC+2
INDEX RETROOT
TCF 2

DERTABLL ADRES DERCOFN -3

```

```

TRASHY LITTLE SUBROUTINES

```

# \*\*\*\*\*

|          |       |          |                                        |
|----------|-------|----------|----------------------------------------|
| INTPRETX | INDEX | WCHPHASE | # SET X1 ON THE WAY TO THE INTERPRETER |
|          | CS    | TARGTDEX |                                        |
|          | INDEX | FIXLOC   |                                        |
|          | TS    | X1       |                                        |
|          | TCF   | INTPRET  |                                        |

|          |        |          |
|----------|--------|----------|
| TDISPSET | CA     | TTF/8    |
|          | EXTEND |          |
|          | MP     | TSCALINV |
|          | DXCH   | TTFDISP  |

|        |          |                                  |
|--------|----------|----------------------------------|
| CA     | EBANK5   | # TREDES BECOMES ZERO TWO PASSES |
| TS     | EBANK    | # BEFORE TCGFAPPR IS REACHED     |
| EBANK= | TCGFAPPR |                                  |
| CA     | TCGFAPPR |                                  |
| INCR   | BBANK    |                                  |
| INCR   | BBANK    |                                  |
| EBANK= | TTF/8    |                                  |

# Page 826

|        |          |
|--------|----------|
| AD     | TTF/8    |
| EXTEND |          |
| MP     | TREDESCL |
| AD     | -DEC103  |
| AD     | NEGMAX   |
| TS     | L        |
| CS     | L        |
| AD     | L        |
| AD     | +DEC99   |
| AD     | POS MAX  |
| TS     | TREDES   |
| CS     | TREDES   |
| ADS    | TREDES   |
| TC     | Q        |

|         |     |          |
|---------|-----|----------|
| 1406P00 | TC  | P00D00   |
|         | OCT | 01406    |
| 1406ALM | TC  | ALARM    |
|         | OCT | 01406    |
|         | TCF | RATESTOP |

# \*\*\*\*\*  
# SPECIALIZED "PHASCHNG" SUBROUTINE  
# \*\*\*\*\*

July 12, 2016

935

```
FASTCHNG EBANK= PHSNAME2
 CA EBANK3 # SPECIALIZED 'PHASCHNG' ROUTINE
 XCH EBANK
 DXCH L
 TS PHSNAME3
 LXCH EBANK
 EBANK= E2DPS
 TC A
```

```

PARAMETER TABLE INDIRECT ADDRESSES

```

```
RDG = RBRFG
VDG = VBRFG
ADG = ABRFG
VDG2TTF = VBRFG*
ADG2TTF = ABRFG*
JDG2TTF = JBRFG*
```

```

LUNAR LANDING CONSTANTS

```

# Page 827

```
TABLTTFL ADRES TABLTTF +3 # ADDRESS FOR REFERENCING TTF TABLE
TTFSCALE = BIT12
TSCALINV = BIT4
-DEC103 DEC -103
+DEC99 DEC +99
TREDESCL DEC -.08
180DEGS DEC +180
1/2DEG DEC +.00278
PROJMAX DEC .42262 B-3 # SIN(25')/8 TO COMPARE WITH PROJ
PROJMIN DEC .25882 B-3 # SIN(15')/8 TO COMPARE WITH PROJ
V06N63 VN 0663 # P63
V06N64 VN 0664 # P64
V06N60 VN 0660 # P65, P66, P67
```

```
BANK 22
SETLOC LANDCNST
BANK
COUNT* $$/F2DPS
```

```
HIGHESTF 2DEC 4.34546769 B-12
GSCALE 2DEC 100 B-11
```

|          |      |          |
|----------|------|----------|
| 3/8DP    | 2DEC | .375     |
| 3/4DP    | 2DEC | .750     |
| DEPRCRIT | 2DEC | -.02 B-1 |

# Page 828

# \*\*\*\*\*  
# \*\*\*\*\*

This code is written to file `src/LUNAR-LANDING-GUIDANCE-EQUATIONS.s`.



## B.60 LUNAR LANDMARK SELECTION FOR CM

```

937 <src/LUNAR-LANDMARK-SELECTION-FOR-CM.s 937>≡
 # Copyright: Public domain.
 # Filename: LUNAR_LANDMARK_SELECTION_FOR_CM.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Jim Lawton <jim.lawton@gmail.com>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 936
 # Mod history: 2009-05-11 JVL Adapted from the Colossus249/ file
 # of the same name, using Comanche055 page
 # images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 936

 # *** END OF TROUBLE .043 ***

```

This code is written to file `src/LUNAR-LANDMARK-SELECTION-FOR-CM.s`.

## B.61 MAIN

```

938 <src/MAIN.s 938>≡
 # Copyright: Public domain.
 # Filename: MAIN.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC) Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo
 # Mod history: 2009-05-05 RSB Adapted from Colossus249/MAIN.agc.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A
 #
 # This file is a little different from the other Comanche055 files I'm providing,
 # in that it doesn't represent anything that appears directly in the original source
 # What I (RSB) have done for organizational purposes is to split the huge monolithic
 # source code into smaller, more manageable chunks--i.e., into individual source
 # files. Those files are rejoined within this file as "includes". It just makes
 # it a little easier to work with. The code chunks correspond to natural divisions
 # into sub-programs. In fact, these divisions are more-or-less specified by
 # the source code itself. Refer to the "SUBROUTINE CALLS" at the
 # very beginning of the file ASSEMBLY_AND_OPERATION_INFORMATION.agc.
 #
 # It may be reasonably asked why tens of thousands of lines of source are joined by
 # means of inclusion, rather than simply assembling the source files individually and
 # then linking them to form the executable. The answer is that the original
 # development team had no linker. The builds were monolithic just like this.
 # There was a big emphasis on reusability of the code in the original project,
 # apparently, but this reusability took the form of inserting your deck of

```

```

punch-cards at the appropriate position in somebody else's deck of punch-cards.
(Actually, I think the card-decks were turned into tape libraries, and the modules
were mixed-and-matched from the tape libraries, but the principle is the same.)
So, indeed, the method of file-inclusion is a very fair representation of the
methods used in the original development ... with the improvement, of course,
that you no longer have to worry about dropping the card deck. On the other hand,
I wasn't there at the time, so I may have no idea what I'm talking about.
#
Finally, note that the original Apollo AGC assembler (called "YUL") is no longer
available (as far as I can tell). Actually, it had already been replaced by another
assembler (called "GAP") by the time of Apollo 11, but GAP isn't available either.
The replacement assembler yaYUL accepts a slightly different format for the source
code from what YUL or GAP accepted, so the source code has been targeted for
assembly with yaYUL.

What follows is simply a bunch of file-includes for the individual code chunks.
I've marked the page numbers to make proof-reading easier. The page images also
contain a lot of interesting tables (cross-referenced to page numbers) created by GAP,
but not duplicated by yaYUL, so it's still valuable even if the source-files
listed below are in hand.

$CONTRACT_AND_APPROVALS.agc # p. 1
$ASSEMBLY_AND_OPERATION_INFORMATION.agc # pp. 2-26
$TAGS_FOR_RELATIVE_SETLOC.agc # pp. 27-35

 # p. 36 contains no code.

COMERASE
$ERASABLE_ASSIGNMENTS.agc # pp. 37-130

COMAID
$INTERRUPT_LEAD_INS.agc # pp. 131-132
$T4RUPT_PROGRAM.agc # pp. 133-169
$DOWNLINK_LISTS.agc # pp. 170-180
$FRESH_START_AND_RESTART.agc # pp. 181-210
$RESTART_TABLES.agc # pp. 211-221
$SXTMARK.agc # pp. 222-235
$EXTENDED_VERBS.agc # pp. 236-267
$PINBALL_NOUN_TABLES.agc # pp. 268-284
$CSM_GEOMETRY.agc # pp. 285-296
$IMU_COMPENSATION_PACKAGE.agc # pp. 297-306
$PINBALL_GAME_BUTTONS_AND_LIGHTS.agc # pp. 307-389
$R60_62.agc # pp. 390-398
$ANGLFIND.agc # pp. 399-411
$GIMBAL_LOCK_AVOIDANCE.agc # pp. 412-413
$KALCMANU_STEERING.agc # pp. 414-419
$SYSTEM_TEST_STANDARD_LEAD_INS.agc # pp. 420-422

```

|                                               |                 |
|-----------------------------------------------|-----------------|
| \$IMU_CALIBRATION_AND_ALIGNMENT.agc           | # pp. 423-455   |
| # COMEKISS                                    |                 |
| \$GROUND_TRACKING_DETERMINATION_PROGRAM.agc   | # pp. 456-459   |
| \$P34-35_P74-75.agc                           | # pp. 460-504   |
| \$R31.agc                                     | # pp. 505-510   |
| \$P76.agc                                     | # pp. 511-513   |
| \$R30.agc                                     | # pp. 514-524   |
| \$STABLE_ORBIT.agc                            | # pp. 525-532   |
| # TROUBLE                                     |                 |
| \$P11.agc                                     | # pp. 533-550   |
| \$TPI_SEARCH.agc                              | # pp. 551-561   |
| \$P20-P25.agc                                 | # pp. 562-634   |
| \$P30-P37.agc                                 | # pp. 635-648   |
| \$P32-P33_P72-P73.agc                         | # pp. 649-683   |
| \$P40-P47.agc                                 | # pp. 684-736   |
| \$P51-P53.agc                                 | # pp. 737-784   |
| \$LUNAR_AND_SOLAR_EPHEMERIDES_SUBROUTINES.agc | # pp. 785-788   |
| \$P61-P67.agc                                 | # pp. 789-818   |
| \$SERVICER207.agc                             | # pp. 819-836   |
| \$ENTRY_LEXICON.agc                           | # pp. 837-843   |
| \$REENTRY_CONTROL.agc                         | # pp. 844-882   |
| \$CM_BODY_ATTITUDE.agc                        | # pp. 883-889   |
| \$P37_P70.agc                                 | # pp. 890-933   |
| \$S-BAND_ANTENNA_FOR_CM.agc                   | # pp. 934-935   |
| \$LUNAR_LANDMARK_SELECTION_FOR_CM.agc         | # pp. 936       |
| # TVCDAPS                                     |                 |
| \$TVCINITIALIZE.agc                           | # pp. 937-944   |
| \$TVCEXECUTIVE.agc                            | # pp. 945-950   |
| \$TVCMASSPROP.agc                             | # pp. 951-955   |
| \$TVCRESTARTS.agc                             | # pp. 956-960   |
| \$TVCDAPS.agc                                 | # pp. 961-978   |
| \$TVCSTROKETEST.agc                           | # pp. 979-983   |
| \$TVCROLLDAP.agc                              | # pp. 984-998   |
| \$MYSUBS.agc                                  | # pp. 999-1001  |
| \$RCS-CSM_DIGITAL_AUTOPILOT.agc               | # pp. 1002-1024 |
| \$AUTOMATIC_MANEUVERS.agc                     | # pp. 1025-1036 |
| \$RCS-CSM_DAP_EXECUTIVE_PROGRAMS.agc          | # pp. 1037-1038 |
| \$JET_SELECTION_LOGIC.agc                     | # pp. 1039-1062 |
| \$CM_ENTRY_DIGITAL_AUTOPILOT.agc              | # pp. 1063-1092 |
| # CHIEFTAN                                    |                 |
| \$DOWN-TELEMETRY_PROGRAM.agc                  | # pp. 1093-1102 |
| \$INTER-BANK_COMMUNICATION.agc                | # pp. 1103-1106 |

|                                      |                                        |
|--------------------------------------|----------------------------------------|
| \$INTERPRETER.agc                    | # pp. 1107-1199                        |
| \$FIXED_FIXED_CONSTANT_POOL.agc      | # pp. 1200-1204                        |
| \$INTERPRETIVE_CONSTANTS.agc         | # pp. 1205-1206                        |
| \$SINGLE_PRECISION_SUBROUTINES.agc   | # p. 1207                              |
| \$EXECUTIVE.agc                      | # pp. 1208-1220                        |
| \$WAITLIST.agc                       | # pp. 1221-1235                        |
| \$LATITUDE_LONGITUDE_SUBROUTINES.agc | # pp. 1236-1242                        |
| \$PLANETARY_INERTIAL_ORIENTATION.agc | # pp. 1243-1251                        |
| \$MEASUREMENT_INCORPORATION.agc      | # pp. 1252-1261                        |
| \$CONIC_SUBROUTINES.agc              | # pp. 1262-1308                        |
| \$INTEGRATION_INITIALIZATION.agc     | # pp. 1309-1333                        |
| \$ORBITAL_INTEGRATION.agc            | # pp. 1334-1354                        |
| \$INFLIGHT_ALIGNMENT_ROUTINES.agc    | # pp. 1355-1364                        |
| \$POWERED_FLIGHT_SUBROUTINES.agc     | # pp. 1365-1372                        |
| \$TIME_OF_FREE_FALL.agc              | # pp. 1373-1388                        |
| \$STAR_TABLES.agc                    | # pp. 1389-1393                        |
| \$AGC_BLOCK_TWO_SELF-CHECK.agc       | # pp. 1394-1403                        |
| \$PHASE_TABLE_MAINTENANCE.agc        | # pp. 1404-1413                        |
| \$RESTARTS_ROUTINE.agc               | # pp. 1414-1419                        |
| \$IMU_MODE_SWITCHING_ROUTINES.agc    | # pp. 1420-1448                        |
| \$KEYRUPT_UPRUPT.agc                 | # pp. 1449-1451                        |
| \$DISPLAY_INTERFACE_ROUTINES.agc     | # pp. 1452-1484                        |
| \$SERVICE_ROUTINES.agc               | # pp. 1485-1492                        |
| \$ALARM_AND_ABORT.agc                | # pp. 1493-1496                        |
| \$UPDATE_PROGRAM.agc                 | # pp. 1497-1507                        |
| \$RT8_OP_CODES.agc                   | # pp. 1508-1516                        |
|                                      | # pp. 1517-1751: GAP-generated tables. |

This code is written to file `src/MAIN.s`.

## B.62 MEASUREMENT INCORPORATION

```

942 <src/MEASUREMENT-INCORPORATION.s 942>≡
Copyright: Public domain.
Filename: MEASUREMENT_INCORPORATION.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1252-1261
Mod history: 2009-05-14 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 1252
INCORP1 -- PERFORMS THE SIX DIMENSIONAL STATE VECTOR DEVIATION FOR POSITION
AND VELOCITY OR THE NINE-DIMENSIONAL DEVIATION OF POSITION, VELOCITY, AND
RADAR OR LANDMARK BIAS. THE OUTPUT OF THE BVECTOR ROUTINE ALONG WITH THE
ERROR TRANSITION MATRIX (W) ARE USED AS INPUT TO THE ROUTINE. THE DEVIATION
IS OBTAINED BY COMPUTING AN ESTIMATED TRACKING MEASUREMENT FROM THE
CURRENT STATE VECTOR AND COMPARING IT WITH AN ACTUAL TRACKING MEASUREMENT
AND APPLYING A STATISTICAL WEIGHTING VECTOR.
#
INPUT
DMENFLG = 0 (6-DIMENSIONAL BVECTOR), =1 (9-DIMENSIONAL)
W = ERROR TRANSITION MATRIX 6X6 OR 9X9
VARIANCE = VARIANCE (SCALAR)
DELTAQ = MEASURED DEVIATION (SCALAR)

```

```

BVECTOR = 6 OR 9 DIMENSIONAL BVECTOR
#
OUTPUT
DELTAX = STATE VECTOR DEVIATIONS 6 OR 9 DIMENSIONAL
ZI = VECTOR USED FOR THE INCORPORATION 6 OR 9 DIMENSIONAL
GAMMA = SCALAR
OMEGA = OMEGA WEIGHTING VECTOR 6 OR 9 DIMENSIONAL
#
CALLING SEQUENCE
L CALL INCORP1
#
NORMAL EXIT
L+1 OF CALLING SEQUENCE

 BANK 37
 SETLOC MEASINC
 BANK

 COUNT* $$/INCOR

 EBANK= W

INCORP1 STQ
 EGRESS
 AXT,1 SSP
 54D
 S1
 18D # IX1 = 54 S1= 18
 AXT,2 SSP
 18D
 S2
 6 # IX2 = 18 S2=6
Z123 VLOAD MXV*
 BVECTOR # BVECTOR (0)
 W +54D,1
 STORE ZI +18D,2
 VLOAD
 BVECTOR +6 # BVECTOR (1)
Page 1253
 MXV* VAD*
 W +108D,1
 ZI +18D,2
 STORE ZI +18D,2
 VLOAD
 BVECTOR +12D # BVECTOR (2)
 MXV* VAD*

```

```

W +162D,1
ZI +18D,2 # B(0)*W+B(1)*(W+54)+B(2)*(W+108) FIRST PASS
STORE ZI +18D,2 # ZI THEN Z2 THEN Z3
TIX,1
INCOR1 TIX,2 INCOR1
BON
Z123 # LOOP FOR Z1,Z2,Z3
DMENFLG
INCOR1A
VLOAD
ZEROVECS
STORE ZI +12D
SETPD VLOAD
0
ZI
VSQ RTB
TPMODE
PDVL VSQ
ZI +6
RTB TAD
TPMODE
PDVL VSQ
ZI +12D
RTB TAD
TPMODE
TAD AXT,2
VARIANCE
0
STORE TRIPA # ZI*2 + Z2*2 + Z3*2 + VARIANCE
TLOAD BOV
VARIANCE # CLEAR OVFIN
+1
STORE TEMPVAR # TEMP STORAGE FOR VARIANCE
BZE
INCOR1B SL2 INCOR1C
BOV
INCOR1C
STORE TEMPVAR
INCR,2 GOTO
DEC 1
INCOR1C TLOAD INCOR1B
ROUND
TRIPA
Page 1254
DMP SQRT
TEMPVAR

```



```

SL* TAD
 0,2
 TRIPA
NORM INCR,2
 X2
DEC -2
SXA,2 AXT,2
 NORMGAM # NORMALIZATION COUNT -2 FOR GAMMA
 162D
BDDV SETPD
 DP1/4TH
 0
STORE GAMMA
TLOAD NORM
 TRIPA
 X1
DLOAD PDDL # PD 0-1 = NORM (A)
 MPAC
 DELTAQ
NORM
 S1
XSU,1 SR1
 S1
DDV PUSH # PD 0-1 = DELTAQ/A
GOTO
 NEWZCOMP
-3 SSP
 S2
 54D
INCOR2 VLOAD VXM* # COMPUT OMEGA1,2,3
 ZI
 W +162D,2
PUSH VLOAD
 ZI +6
VXM* VAD
 W +180D,2
PUSH VLOAD
 ZI +12D
VXM* VAD
 W +198D,2
PUSH TIX,2 # PD 2-7=OMEGA1, 8-13=OMEGA2, 14-19=OMEGA3
 INCOR2
VLOAD STADR
STORE OMEGA +12D
VLOAD STADR
STORE OMEGA +6

```

```

Page 1255
VLOAD STADR
STORE OMEGA

BON VLOAD
 DMENFLG
 INCOR2AB
 ZEROVECS
STORE OMEGA +12D
INCOR2AB AXT,2 SSP
 18D
 S2
 6

INCOR3 VLOAD*
 OMEGA +18D,2
 VXSC VSL*
 0 # DELTAQ/A
 0,1
STORE DELTAX +18D,2
TIX,2 VLOAD
 INCOR3
 DELTAX +6
 VSL3
STORE DELTAX +6
GOTO
 EGRESS

```

```

Page 1256
INCORP2 -- INCORPORATES THE COMPUTED STATE VECTOR DEVIATIONS INTO THE
ESTIMATED STATE VECTOR. THE STATE VECTOR UPDATED MAY BE FOR EITHER THE
LEM OR THE CSM. DETERMINED BY FLAG VEHUPFLG. (ZERO = LEM) (1 = CSM)
#
INPUT
PERMANENT STATE VECTOR FOR EITHER THE LEM OR CSM
VEHUPFLG = UPDATE VEHICLE C=LEM 1=CSM
W = ERROR TRANSITION MATRIX
DELTAX = COMPUTED STATE VECTOR DEVIATIONS
DMENFLG = SIZE OF W MATRIX (ZERO=6X6) (1=9X9)
GAMMA = SCALAR FOR INCORPORATION
ZI = VECTOR USED IN INCORPORATION
OMEGA = WEIGHTING VECTOR
#
OUTPUT
UPDATED PERMANENT STATE VECTOR
#
CALLING SEQUENCE
L CALL INCORP2

```

```
#
NORMAL EXIT
L+1 OF CALLING SEQUENCE
#
```

|             |        |            |                                          |
|-------------|--------|------------|------------------------------------------|
|             | SETLOC | MEASINC1   |                                          |
|             | BANK   |            |                                          |
|             | COUNT* | \$\$/INCOR |                                          |
| INCRP2      | STQ    | CALL       |                                          |
|             |        | EGRESS     |                                          |
|             |        | INTSTALL   |                                          |
|             | VLOAD  | VXSC       | # CALC. GAMMA*OMEGA1,2,3                 |
|             |        | OMEGA      |                                          |
|             |        | GAMMA      |                                          |
|             | STOVL  | OMEGAM1    |                                          |
|             |        | OMEGA +6   |                                          |
|             | VXSC   |            |                                          |
|             |        | GAMMA      |                                          |
|             | STOVL  | OMEGAM2    |                                          |
|             |        | OMEGA +12D |                                          |
|             | VXSC   |            |                                          |
|             |        | GAMMA      |                                          |
|             | STORE  | OMEGAM3    |                                          |
|             | EXIT   |            |                                          |
|             | CAF    | 54DD       | # INITIAL IX 1 SETTING FOR W MATRIX      |
|             | TS     | WIXA       |                                          |
|             | TS     | WIXB       |                                          |
|             | CAF    | ZERO       |                                          |
|             | TS     | ZIXA       | # INITIAL IX 2 SETTING FOR Z COMPONENT   |
|             | TS     | ZIXB       |                                          |
| FAZA        | TC     | PHASCHNG   |                                          |
| # Page 1257 |        |            |                                          |
|             | OCT    | 04022      |                                          |
|             | TC     | UPFLAG     |                                          |
|             | ADRES  | REINTFLG   |                                          |
| FAZA1       | CA     | WIXB       | # START FIRST PHASE OF INCRP2            |
|             | TS     | WIXA       | # TO UPDATE 6 OR 9 DIM. W MATRIX IN TEMP |
|             | CA     | ZIXB       |                                          |
|             | TS     | ZIXA       |                                          |
|             | TC     | INTPRET    |                                          |
|             | LXA,1  | LXA,2      |                                          |
|             |        | WIXA       |                                          |
|             |        | ZIXA       |                                          |
|             | SSP    | DLOAD*     |                                          |

```

S1
6
ZI,2
DCOMP NORM # CALC UPPER 3X9 PARTITION OF W MATRIX
S2
VXSC XCHX,2
OMEGAM1
S2
LXC,2 XAD,2
X2
NORMGAM
VSL* XCHX,2
0,2
S2
VAD*
W +54D,1
STORE HOLDW
DLOAD* DCOMP # CALC MIDDLE 3X9 PARTITION OF W MATRIX
ZI,2
NORM VXSC
S2
OMEGAM2
XCHX,2 LXC,2
S2
X2
XAD,2 VSL*
NORMGAM
0,2
XCHX,2 VAD*
S2
W +108D,1
STORE HOLDW +6
BOFF
DMENFLG # BRANCH IF 6 DIMENSIONAL
FAZB
DLOAD* DCOMP # CALC LOWER 3X9 PARTITION OF W MATRIX
ZI,2
NORM VXSC
S2
OMEGAM3
XCHX,2 LXC,2
S2
X2
XAD,2 VSL*
NORMGAM

```

```

 0,2
 VAD*
 S2
 W +162D,1
 STORE HOLDW +12D
FAZB CALL
 GRP2PC
 EXIT
FAZB1 CA WIXA # START 2ND PHASE OF INCORP2 TO TRANSFER
 AD 6DD # TEMP REG TO PERM W MATRIX
 TS WIXB
 CA ZIXA
 AD MINUS2
 TS ZIXB
 TC INTPRET
 LXA,1 SSP
 WIXA
 S1
 6
 VLOAD
 HOLDW
 STORE W +54D,1
 VLOAD
 HOLDW +6
 STORE W +108D,1
 BOFF VLOAD
 DMENFLG
 FAZB5
 HOLDW +12D
 STORE W +162D,1
FAZB2 TIX,1 GOTO
 +2
 FAZC # DONE WITH W MATRIX. UPDATE STATE VECTOR
 RTB
 FAZA
FAZB5 SLOAD DAD
 ZIXB
 12DD
 BHIZ GOTO
 FAZC
 FAZB2
FAZC CALL
 GRP2PC
Page 1259
 VLOAD VAD # START 3RD PHASE OF INCORP2
 X789 # 7TH, 8TH, 9TH COMPONENTN OF STATE VECTOR

```

|        |        |             |                                   |
|--------|--------|-------------|-----------------------------------|
|        |        | DELTAX +12D | # INCORPORATION FOR X789          |
|        | STORE  | TX789       |                                   |
|        | BON    | RTB         |                                   |
|        |        | VEHUPFLG    |                                   |
|        |        | DOCSM       |                                   |
|        |        | MOVEPLEM    |                                   |
| FAZAB  | BOVB   | AXT,2       |                                   |
|        |        | TCDANZIG    |                                   |
|        |        | 0           |                                   |
|        | BOFF   | AXT,2       |                                   |
|        |        | MOONTHIS    |                                   |
|        |        | +2          |                                   |
|        |        | 2           |                                   |
|        | VLOAD  | VSR*        |                                   |
|        |        | DELTAX      | # B27 IF MOON ORBIT, B29 IF EARTH |
|        |        | 0 -7,2      |                                   |
|        | VAD    | BOV         |                                   |
|        |        | TDELTAV     |                                   |
|        |        | FAZAB1      |                                   |
|        | STOVL  | TDELTAV     |                                   |
|        |        | DELTAX +6   | # B5 IF MOON ORBIT, B7 IF EARTH   |
|        | VSR*   | VAD         |                                   |
|        |        | 0 -4,2      |                                   |
|        |        | TNUV        |                                   |
|        | BOV    |             |                                   |
|        |        | FAZAB2      |                                   |
|        | STCALL | TNUV        |                                   |
|        |        | FAZAB3      |                                   |
| FAZAB1 | VLOAD  | VAD         |                                   |
|        |        | RCV         |                                   |
|        |        | DELTAX      |                                   |
|        | STORE  | RCV         |                                   |
| FAZAB2 | VLOAD  | VAD         |                                   |
|        |        | VCV         |                                   |
|        |        | DELTAX +6   |                                   |
|        | STORE  | VCV         |                                   |
|        | SXA,2  | CALL        |                                   |
|        |        | PBODY       |                                   |
|        |        | RECTIFY     |                                   |
| FAZAB3 | CALL   |             |                                   |
|        |        | GRP2PC      |                                   |
|        | BON    | RTB         |                                   |
|        |        | VEHUPFLG    |                                   |
|        |        | DOCSM1      |                                   |
|        |        | MOVEALEM    |                                   |
|        | CALL   |             |                                   |

July 12, 2016

951

|             |        |            |                               |
|-------------|--------|------------|-------------------------------|
| FAZAB4      | CALL   | SVDWN2     | # STORE DOWNLINK STATE VECTOR |
| # Page 1260 |        |            |                               |
|             | BOFF   | GRP2PC     | # PHASE CHANGE                |
|             |        | VLOAD      |                               |
|             |        | DMENFLG    |                               |
|             |        | FAZAB5     | # 6 DIMENSIONAL               |
|             |        | TX789      | # 9 DIMENSIONAL               |
| FAZAB5      | STORE  | X789       |                               |
|             | LXA,1  | SXA,1      |                               |
|             |        | EGRESS     |                               |
|             |        | QPRET      |                               |
|             | EXIT   |            |                               |
|             | TC     | POSTJUMP   | # EXIT                        |
|             | CADR   | INTWAKE    |                               |
| DOCSM       | RTB    | GOTO       |                               |
|             |        | MOVEPCSM   |                               |
|             |        | FAZAB      |                               |
| DOCSM1      | RTB    | CALL       |                               |
|             |        | MOVEACSM   |                               |
|             |        | SVDWN1     | # STORE DOWNLINK STATE VECTOR |
|             | GOTO   |            |                               |
|             |        | FAZAB4     |                               |
| ZEROD       | =      | ZEROVECS   |                               |
| 54DD        | DEC    | 54         |                               |
| 6DD         | DEC    | -6         |                               |
| 12DD        | DEC    | 12         |                               |
|             | SETLOC | MEASINC2   |                               |
|             | BANK   |            |                               |
|             | COUNT* | \$\$/INCOR |                               |
| NEWZCOMP    | VLOAD  | ABVAL      |                               |
|             |        | ZI         |                               |
|             | STOVL  | NORMZI     |                               |
|             |        | ZI +6      |                               |
|             | ABVAL  | PUSH       |                               |
|             | DSU    | BMN        |                               |
|             |        | NORMZI     |                               |
|             |        | +3         |                               |
|             | DLOAD  | STADR      |                               |
|             | STORE  | NORMZI     |                               |
|             | VLOAD  | ABVAL      |                               |
|             |        | ZI +12D    |                               |
|             | PUSH   | DSU        |                               |
|             |        | NORMZI     |                               |
|             | BMN    | DLOAD      |                               |

|             |        |           |                 |
|-------------|--------|-----------|-----------------|
|             | STADR  | NEWZCMP1  |                 |
|             | STCALL | NORMZI    | # LARGEST ABVAL |
|             |        | NEWZCMP1  |                 |
|             | SETLOC | MEASINC3  |                 |
|             | BANK   |           |                 |
| # Page 1261 |        |           |                 |
| NEWZCMP1    |        | DLOAD     | SXA,1           |
|             |        | NORMZI    |                 |
|             |        | NORMZI    | # SAVE X1       |
|             | NORM   | INCR,1    |                 |
|             |        | X1        |                 |
|             | DEC    | 2         |                 |
|             | VLOAD  | VSL*      |                 |
|             |        | ZI        |                 |
|             |        | 0,1       |                 |
|             | STOVL  | ZI        |                 |
|             |        | ZI +6     |                 |
|             | VSL*   |           |                 |
|             |        | 0,1       |                 |
|             | STOVL  | ZI +6     |                 |
|             |        | ZI +12D   |                 |
|             | VSL*   | SXA,1     |                 |
|             |        | 0,1       |                 |
|             |        | NORMZI +1 | # SAVE SHIFT    |
|             | STORE  | ZI +12D   |                 |
|             | LXA,1  | XSU,1     |                 |
|             |        | NORMGAM   |                 |
|             |        | NORMZI +1 |                 |
|             | XSU,1  |           |                 |
|             |        | NORMZI +1 |                 |
|             | SXA,1  | LXC,1     |                 |
|             |        | NORMGAM   |                 |
|             |        | NORMZI +1 |                 |
|             | XAD,1  | SETPD     |                 |
|             |        | NORMZI    |                 |
|             |        | 2D        |                 |
|             | GOTO   |           |                 |
|             |        | INCOR2 -3 |                 |
| NORMZI      | =      | 36D       |                 |



July 12, 2016

953

This code is written to file `src/MEASUREMENT-INCORPORATION.s`.

## B.63 MYSUBS

```

954 <src/MYSUBS.s 954>≡
 # Copyright: Public domain.
 # Filename: MYSUBS.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 999-1001
 # Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
 # same name, using Comanche055 page images.
 # 2009-05-20 RSB Corrections: EBANK= changed from MPAC to KMPAC.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 999

 BANK 20
 SETLOC MYSUBS
 BANK

 EBANK= KMPAC
SPCOS1 EQUALS SPCOS
SPSIN1 EQUALS SPSIN
SPCOS2 EQUALS SPCOS
SPSIN2 EQUALS SPSIN

 COUNT 21/DAPMS

```

## # ONE AND ONE HALF PRECISION MULTIPLICATION ROUTINE

```

SMALLMP TS KMPTEMP # A(X+Y)
EXTEND
MP KMPAC +1
TS KMPAC +1 # AY
CAF ZERO
XCH KMPAC
EXTEND
MP KMPTEMP # AX
DAS KMPAC # AX+AY
TC Q

```

## # SUBROUTINE FOR DOUBLE PRECISION ADDITIONS OF ANGLES

# A AND L CONTAIN A DP(1S) ANGLE SCALED BY 180 DEGS TO BE ADDED TO KMPAC.

# RESULT IS PLACED IN KMPAC. TIMING = 6 MCT (22 MCT ON OVERFLOW)

```

DPADD DAS KMPAC
EXTEND
BZF TSK +1 # NO OVERFLOW
CCS KMPAC
TCF DPADD+ # + OVERFLOW
TCF +2
TCF DPADD- # - OVERFLOW
CCS KMPAC +1
TCF DPADD2+ # UPPER = 0, LOWER +
TCF +2
COM # UPPER = 0, LOWER -
AD POSMAX # LOWER = 0, A = 0
TS KMPAC +1 # CAN NOT OVERFLOW
CA POSMAX # UPPER WAS = 0
TSK TS KMPAC
TC Q

```

```

DPADD+ AD NEGMAX # KMPAC GREATER THAN 0
TCF TSK

```

# Page 1000

```

DPADD- COM
AD POSMAX # KMPAC LESS THAN 0
TCF TSK

```

```

DPADD2+ AD NEGMAX # CAN NOT OVERFLOW
TS KMPAC +1
CA NEGMAX # UPPER WAS = 0
TCF TSK

```

# Page 1001 (empty page)

This code is written to file `src/MYSUBS.s`.

## B.64 ORBITAL INTEGRATION

```

957 <src/ORBITAL-INTEGRATION.s 957>≡
Copyright: Public domain.
Filename: ORBITAL_INTEGRATION.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1334-1354
Mod history: 2009-05-14 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
2009-05-20 RSB Corrections: DAT -> DAD in one place,
BWM -> BMN, DEFEQCNT -> DIFEQCNT.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 1334
ORBITAL INTEGRATION

DELETE
#
BANK 13
SETLOC ORBITAL
BANK
COUNT 11/ORBIT

DELETE
#
KEPPREP LXA,2 SETPD
PBODY

```

|        |           |                                       |     |        |  |
|--------|-----------|---------------------------------------|-----|--------|--|
|        | 0         |                                       |     |        |  |
| DLOAD* | SQRT      | # SQRT(MU) (+18 OR +15)               | 0D  | PL 2D  |  |
|        | MUEARTH,2 |                                       |     |        |  |
| PDVL   | UNIT      | #                                     |     | PL 8D  |  |
|        | RCV       |                                       |     |        |  |
| PDDL   | NORM      | # NORM R (+29 OR +27 - N1)            | 2D  | PL 4D  |  |
|        | 36D       |                                       |     |        |  |
|        | X1        |                                       |     |        |  |
| PDVL   |           |                                       |     |        |  |
| DOT    | PDDL      | # F*SQRT(MU) (+7 OR +5)               | 4D  | PL 6D  |  |
|        | VCV       |                                       |     |        |  |
|        | TAU.      | # (+28)                               |     |        |  |
| DSU    | NORM      |                                       |     |        |  |
|        | TC        |                                       |     |        |  |
|        | S1        |                                       |     |        |  |
| SR1    |           |                                       |     |        |  |
| DDV    | PDDL      |                                       |     |        |  |
|        | 2D        |                                       |     |        |  |
| DMP    | PUSH      | # FS (+6 +N1-N2)                      | 6D  | PL 8D  |  |
|        | 4D        |                                       |     |        |  |
| DSQ    | PDDL      | # (FS)SQ (+12 +2(N1-N2))              | 8D  | PL 10D |  |
|        | 4D        |                                       |     |        |  |
| DSQ    | PDDL*     | # SSQ/MU (-2 OR +2(N1-N2))            | 10D | PL 12D |  |
|        | MUEARTH,2 |                                       |     |        |  |
| SR3    | SR4       |                                       |     |        |  |
| PDVL   | VSQ       | # PREALIGN MU (+43 OR +37)            | 12D | PL 14D |  |
|        | VCV       |                                       |     |        |  |
| DMP    | BDSU      | #                                     |     | PL 12D |  |
|        | 36D       |                                       |     |        |  |
| DDV    | DMP       | #                                     |     | PL 10D |  |
|        | 2D        | # -(1/R-ALPHA) (+12 +3N1-2N2)         |     |        |  |
| DMP    | SL*       |                                       |     |        |  |
|        | DP2/3     |                                       |     |        |  |
|        | 0         | -3,1 # 10L(1/R-ALPHA) (+13 +2(N1-N2)) |     |        |  |
| XSU,1  | DAD       | # 2(FS)SQ - ETCETERA                  |     | PL 8D  |  |
|        | S1        | # X1 = N2-N1                          |     |        |  |
| SL*    | DSU       | # -FS+2(FS)SQ ETC (+6 +N1-N2)         |     | PL 6D  |  |
|        | 8D,1      |                                       |     |        |  |
| DMP    | DMP       |                                       |     |        |  |
|        | 0D        |                                       |     |        |  |
|        | 4D        |                                       |     |        |  |
| SL*    | SL*       |                                       |     |        |  |
|        | 8D,1      |                                       |     |        |  |
|        | 0,1       | # S(-FS(1-2FS)-1/6...) (+17 OR +16)   |     |        |  |
| DAD    | PDDL      | #                                     |     | PL 6D  |  |

```

 XKEP
DMP SL* # S(+17 OR +16)
 OD
 1,1
BOVB DAD
 TCDANZIG
STADR
STORE XKEPNEW
STQ AXC,1
 KEPRTN
DEC 10
BON AXC,1
 MOONFLAG
 KEPLERN
DEC 2
GOTO KEPLERN

```

# Page 1336

FBR3

```

LXA,1 SSP
 DIFEQCNT
 S1
DEC -13
DLOAD SR
 DT/2
 9D
TIX,1 ROUND
 +1
PUSH DAD
 TC
STODL TAU.
DAD
 TET
STCALL TET
 KEPPREP

```

# Page 1337

# AGC ROUTINE TO COMPUTE ACCELERATION COMPONENTS.

```

ACCOMP LXA,1 LXA,2
 PBODY
 PBODY
VLOAD
 ZEROVEC
STOVL FV
 ALPHAV

```

```

VSL* VAD
 0 -7,2
 RCV
STORE BETAV
BOF XCHX,2
 DIMOFLAG
 +5
 DIFEQCNT
STORE VECTAB,2
XCHX,2
 DIFEQCNT
VLOAD UNIT
 ALPHAV
STODL ALPHAV
 36D
STORE ALPHAM
CALL
 GAMCOMP
VLOAD SXA,1
 BETAV
 S2
STODL ALPHAV
 BETAM
STORE ALPHAM
BOF DLOAD
 MIDFLAG
 OBLATE
 TET
CALL
 LSPOS
AXT,2 LXA,1
 2
 S2
BOF MOONFLAG
 +3
VCOMP AXT,2
 0
STORE BETAV
STOVL RPQV
 2D
STORE RPSV
BOF VLOAD
 DIMOFLAG
 GETRPSV

```



|         |        |             |                                        |
|---------|--------|-------------|----------------------------------------|
|         |        | ALPHAV      |                                        |
|         | VXSC   | VSR*        |                                        |
|         |        | ALPHAM      |                                        |
|         |        | 1,2         |                                        |
|         | VSU    | XCHX,2      |                                        |
|         |        | BETAV       |                                        |
|         |        | DIFEQCNT    |                                        |
|         | STORE  | VECTAB +6,2 |                                        |
|         | XCHX,2 |             |                                        |
|         |        | DIFEQCNT    |                                        |
| GETRPSV | VLOAD  | INCR,1      |                                        |
|         |        | RPQV        |                                        |
|         |        | 4           |                                        |
|         | CLEAR  | BOF         |                                        |
|         |        | RPQFLAG     |                                        |
|         |        | MOONFLAG    |                                        |
|         |        | +5          |                                        |
|         | VSR    | VAD         |                                        |
|         |        | 9D          |                                        |
|         |        | RPSV        |                                        |
|         | STORE  | RPSV        |                                        |
|         | CALL   |             |                                        |
|         |        | GAMCOMP     |                                        |
|         | AXT,2  | INCR,1      |                                        |
|         |        | 4           |                                        |
|         |        | 4           |                                        |
|         | VLOAD  |             |                                        |
|         |        | RPSV        |                                        |
|         | STCALL | BETAV       |                                        |
|         |        | GAMCOMP     |                                        |
|         | GOTO   |             |                                        |
|         |        | OBLATE      |                                        |
| GAMCOMP | VLOAD  | VSR1        |                                        |
|         |        | BETAV       |                                        |
|         | VSQ    | SETPD       |                                        |
|         |        | 0           |                                        |
|         | NORM   | ROUND       |                                        |
|         |        | 31D         |                                        |
|         | PDDL   | NORM        | # NORMED B SQUARED TO PD LIST          |
|         |        | ALPHAM      | # NORMALIZE (LESS ONE) LENGTH OF ALPHA |
|         |        | 32D         | # SAVING NORM SCALE FACTOR IN X1       |
|         | SR1    | PDVL        |                                        |
|         |        | BETAV       | # C(PDL+2) = ALMOST NORMED ALPHA       |
|         | UNIT   |             |                                        |
|         | STODL  | BETAV       |                                        |

```

 36D
STORE BETAM
NORM BDDV # FORM NORMALIZE QUOTIEN ALPHAM/BETAM
 33D
SR1R PUSH # C(PDL+2) = ALMOST NORMALIZE RHO.
DLOAD*
 ASCALE,1
STORE S1
XCHX,2 XAD,2
 S1
 32D
XSU,2 DLOAD
 33D
 2D
SR* XCHX,2
 0 -1,2
 S1
PUSH SR1R # RHO/4 TO 4D
PDVL DOT
 ALPHAV
 BETAV
SL1R BDSU # (RHO/4) - 2(ALPHAV/2.BETAV/2)
PUSH DMPR # TO PDL+6
 4
SL1
PUSH DAD
 DQUARTER
PUSH SQRT
DMPR PUSH
 10D
SL1 DAD
 DQUARTER
PDDL DAD # (1/4)+2((Q+1)/4) TO PD+14D
 10D
 HALFDP
DMPR SL1
 8D
DAD DDV
 THREE/8
 14D
DMPR VXSC
 6
 BETAV
PDVL VSR3 # (G/2)(C(PD+4))B/2 TO PD+16D
 ALPHAV
VAD PUSH # A12 + C(PD+16D) TO PD+16D

```

# Page 1340

GOBAQUE

```

DLOAD DMP
 0
 12D
NORM ROUND

 30D
BDDV DMP*
 2
 MUEARTH,2
DCOMP VXSC
XCHX,2 XAD,2
 S1
 S2
XSU,2 XSU,2
 30D
 31D
BOV # CLEAR OVIND
 +1
VSR* XCHX,2
 0 -1,2
 S1
VAD
 FV
STORE FV
BOV RVQ # RETURN IF NO OVERFLOW
 +1
VLOAD ABVAL
 TDELTA V
BZE
 INT-ABRT
DLOAD SR
 H
 9D
PUSH BDSU
 TC
STODL TAU.
 TET
DSU STADR
STCALL TET
 KEPPREP
CALL
 RECTIFY
SETGO
 RPQFLAG
 TESTLOOP

```

```

INT-ABRT EXIT
 TC P00D00
 OCT 00430

```

```

Page 1341

```

```

THE OBLATE ROUTINE COMPUTES THE ACCELERATION DUE TO OBLATENESS. IT USES THE UNIT C
POSITION VECTOR FOUND IN ALPHAV AND THE DISTANCE TO THE CENTER IN ALPHAM. THIS IS
DISTURBING ACCELERATIONS IN FV AND THE PROPER DIFEQ STAGE IS CALLED VIA X1.

```

```

OBLATE LXA,2 DLOAD
 PBODY
 ALPHAM
 SETPD DSU*
 0
 RDE,2
 BPL BOF # GET URPV
 NBRANCH
 MOONFLAG
 COSPHIE
 VLOAD PDDL
 ALPHAV
 TET
 PDDL CALL
 3/5
 R-TO-RP
 STORE URPV
 VLOAD VXV
 504LM
 ZUNIT
 VAD VXM
 ZUNIT
 MMATRIX
 UNIT # PROBABLY UNNECESSARY.
COMTERM STORE UZ
 DLOAD DMPR
 COSPHI/2
 3/32
 PDDL DSQ # P2/64 TO PD0
 COSPHI/2
 DMPR DSU
 15/16
 3/64
 PUSH DMPR # P3/32 TO PD2
 COSPHI/2
 DMP SL1R
 7/12

```

July 12, 2016

965

# Page 1342

```
PDDL DMPR
 0
 2/3
BDSU PUSH # P4/128 TO PD4
DMPR DMPR
 COSPHI/2 # BEGIN COMPUTING P5/1024
 9/16
PDDL DMPR
 2
 5/128

Page 1342

BDSU
DMP*
 J4REQ/J3,2
DDV DAD # -3
 ALPHAM # (((P5/256)B 2 /R+P4/32) /R+P3/8)ALPHAV
 4 # 4 3
DMPR* DDV
 2J3RE/J2,2
 ALPHAM
DAD VXSC
 2
 ALPHAV
STODL TVEC
DMP* SR1
 J4REQ/J3,2
DDV DAD
 ALPHAM
DMPR* SR3
 2J3RE/J2,2
DDV DAD
 ALPHAM
VXSC VSL1
 UZ
BVSU
 TVEC
STODL TVEC
 ALPHAM
NORM DSQ
 X1
DSQ NORM
 S1 # 4
PUSH BDDV* # NORMED R TO OD
 J2REQSQ,2
VXSC BOV
 TVEC
```

```

 +1 # (RESET OVERFLOW INDICATOR)
XAD,1 XAD,1
 X1
 X1
XAD,1 VSL*
 S1
 0 -22D,1
VAD BOV
 FV
 GOBAQUE
STCALL FV
 QUALITY1
QUALITY3 DSQ # J22 TERM X R**4 IN 2D. SCALED B61
 # AS VECTOR.
 # STORE COSPHI**2 SCALED B2 IN 8D.
Page 1343 PUSH DMP
 5/8 # 5 SCALED B3
PDDL SR2 # PUT 5 COSPHI**2, D5, IN 8D. GET
 # COSPHI**2 D2 FROM 8D
DAD BDSU # END UP WITH (1-7 COSPHI**2), B5
 8D # ADDING COSPHI**2 B4 SAME AS COSPHI**2
 # X 2 D5
 # 1 SCALED B5
DMP D1/32
 DMP
 URPV # X COMPONENT
 5/8 # 5 SCALED B3
VXSC VSL5 # AFTER SHIFT, SCALED B5
 URPV # VECTOR, B1.
PDDL 8D # VECTOR INTO 8D, 10D, 12D, SCALED B5.
 # GET 5 COSPHI**2 OUT OF 8D
DSU DAD
 D1/32 # 1 B5
 8D # X COMPONENT (SAME AS MULTIPLYING
 # BY UNITX)
STODL 8D
 URPV # X COMPONENT
DMP DMP
 URPV # Z COMPONENT
 5/8 # 5 B3 ANSWER B5
SL1 DAD # FROM 12D FOR Z COMPONENT (SL1 GIVES 10
 # INSTEAD OF 5 FOR COEFFICIENT)
PDDL NORM # BACK INTO 12D FOR Z COMPONENT
 ALPHAM # SCALED B27 FOR MOON
 X2
PUSH SLOAD # STORE IN 14D, DESTROYING URPV
 # X COMPONENT

```

```

E32C31RM
DDV VXSC # IF X2 = 0, DIVISION GIVES B53, VXSC
 # out of 8D B5 GIVES B58
VSL* VAD # SHIFT MAKES B61, FOR ADDITION OF
 # VECTOR IN 2D
 0 -3,2
VSL* V/SC # OPERAND FROM OD, B108 FOR X1 = 0
 0 -27D,1 # FOR X1 = 0, MAKES B88, GIVING B-20
 # FOR RESULT.
PDDL PDDL
 TET
 5/8 # ANY NON-ZERO CONSTANT
LXA,2 CALL # POSITION IN OD, TIME IN 6D. X2 LEFT
 # ALONE.
 PBODY
 RP-TO-R
VAD BOV # OVERFLOW INDICATOR RESET IN *RP-TO-R*
 FV
 GOBAQUE
STORE FV
Page 1344
NBRANCH SLOAD LXA,1
 DIFEQCNT
 MPAC
DMP CGOTO
 -1/12
 MPAC
 DIFEQTAB
COSPHE DLOAD
 ALPHAV +4
STOVL COSPHI/2
 ZUNIT
GOTO
DIFEQTAB COMTERM
CADR DIFEQ+0
CADR DIFEQ+1
CADR DIFEQ+2
TIMESTEP BOF CALL
 MIDFLAG
 RECTEST # SKIP ORIGIN CHANGE LOGIC
 CHKSWTCH
BMN
 DOSWITCH
RECTEST VLOAD ABVAL # RECTIFY IF

```

|             |       |          |   |                                        |
|-------------|-------|----------|---|----------------------------------------|
|             |       | TDELTA V |   |                                        |
|             | BOV   | CALLRECT |   |                                        |
|             | DSU   | BPL      | # | 1) EITHER TDELTA V OR TNUV EQUALS OR   |
|             |       | 3/4      | # | EXCEEDS 3/4 IN MAGNITUDE               |
|             |       | CALLRECT | # |                                        |
|             | DAD   | SL*      | # | OR                                     |
|             |       | 3/4      | # |                                        |
|             |       | 0 -7,2   | # | 2) ABVAL(TDELTA V) EQUALS OR EXCEEDS   |
|             | DDV   | DSU      | # | .01(ABVAL(RCV))                        |
|             |       | 10D      |   |                                        |
|             |       | RECRATIO |   |                                        |
|             | BPL   | VLOAD    |   |                                        |
|             |       | CALLRECT |   |                                        |
|             |       | TNUV     |   |                                        |
|             | ABVAL | DSU      |   |                                        |
|             |       | 3/4      |   |                                        |
|             | BOV   |          |   |                                        |
|             |       | CALLRECT |   |                                        |
|             | BMN   |          |   |                                        |
|             |       | INTGRATE |   |                                        |
| CALLRECT    | CALL  | RECTIFY  |   |                                        |
| INTGRATE    | VLOAD |          |   |                                        |
|             |       | TNUV     |   |                                        |
| # Page 1345 |       |          |   |                                        |
|             | STOVL | ZV       |   |                                        |
|             |       | TDELTA V |   |                                        |
|             | STORE | YV       |   |                                        |
|             | CLEAR |          |   |                                        |
|             |       | JSWITCH  |   |                                        |
| DIFEQ0      | VLOAD | SSP      |   |                                        |
|             |       | YV       |   |                                        |
|             |       | DIFEQCNT |   |                                        |
|             |       | 0        |   |                                        |
|             | STODL | ALPHA V  |   |                                        |
|             |       | DPZERO   |   |                                        |
|             | STORE | H        | # | START H AT ZERO. GOES 0(DELTA/2)DELTA. |
|             | BON   | GOTO     |   |                                        |
|             |       | JSWITCH  |   |                                        |
|             |       | DOW..    |   |                                        |
|             |       | ACCOMP   |   |                                        |
| CHKSWTCH    | STQ   | BOF      |   |                                        |
|             |       | ORIGEX   |   |                                        |
|             |       | RPQFLAG  |   |                                        |



July 12, 2016

969

```

 RPQOK # MOON POSITION IS AVAILABLE
 CALL
 TET
 LUNPOS # GET MOON POSITION
BOF VCOMP
 MOONFLAG
 +1
STORE RPQV

RPQOK LXA,2 VLOAD # RESTORE X2 AFTER USING LUNPOS
 PBODY
 TDELTA
 VSL* VAD # -
 0 -7,2 # -RQC|-RSPHERE WHEN OUTSIDE THE SPHERE.
 RCV # - - -
 BOF ABVAL # R = RDEVIATION + RCONIC
 MOONFLAG
 EARSPH
SR2 BDSU # INSIDE
 RSPHERE
GOTO
 ORIGEX
EARSPH VSU ABVAL # OUTSIDE
 RPQV
DSU GOTO
 RSPHERE
 ORIGEX

DOSWITCH CALL
 ORIGCHNG
GOTO
 INTEGRATE

Page 1346
ORIGCHNG STQ CALL
 ORIGEX
 RECTIFY
VLOAD VSL*
 RCV
 0,2
VSU VSL*
 RPQV
 2,2
STORE RRECT
STODL RCV
 TET
```

```

CALL LUNVEL
BOF VCOMP
 MOONFLAG
 +1
PDVL VSL*
 VCV
 0,2
VSU
VSL* 0 +2,2
STORE VRECT
STORE VCV
LXA,2 SXA,2
 ORIGEX
 QPRET
BON GOTO
 MOONFLAG
 CLRMOON
 SETMOON

```

# Page 1347

# THE RECTIFY SUBROUTINE IS CALLED BY THE INTEGRATION PROGRAM AND OCCASIONALLY BY THE  
 # ROUTINES TO ESTABLISH A NEW CONIC.

```

RECTIFY LXA,2 VLOAD
 PBODY
 TDELTA
VSL* VAD
 0 -7,2
 RCV
STORE RRECT
STOVL RCV
 TNUV
VSL* VAD
 0 -4,2
 VCV
MINIRECT STORE VRECT
 STOVL VCV
 ZEROVEC
 STORE TDELTA
 STODL TNUV
 ZEROVEC
 STORE TC
 STORE XKEP
 RVQ

```

# Page 1348

# THE THREE DIFEQ ROUTINES -- DIFEQ+0, DIFEQ+12, DIFEQ+24 -- ARE ENTERED TO PROCESS THE CONTRIE

# BEGINNING, MIDDLE, AND END OF THE TIMESTEP, RESPECTIVELY. THE UPDATING IS DONE BY THE NYSTRO

|         |        |          |                                 |
|---------|--------|----------|---------------------------------|
| DIFEQ+0 | VLOAD  | VSR3     |                                 |
|         |        | FV       |                                 |
|         | STCALL | PHIV     |                                 |
|         |        | DIFEQCOM |                                 |
| DIFEQ+1 | VLOAD  | VSR1     |                                 |
|         |        | FV       |                                 |
|         | PUSH   | VAD      |                                 |
|         |        | PHIV     |                                 |
|         | STOVL  | PSIV     |                                 |
|         | VSR1   | VAD      |                                 |
|         |        | PHIV     |                                 |
|         | STCALL | PHIV     |                                 |
|         |        | DIFEQCOM |                                 |
| DIFEQ+2 | DLOAD  | DMPR     |                                 |
|         |        | H        |                                 |
|         |        | DP2/3    |                                 |
|         | PUSH   | VXSC     |                                 |
|         |        | PHIV     |                                 |
|         | VSL1   | VAD      |                                 |
|         |        | ZV       |                                 |
|         | VXSC   | VAD      |                                 |
|         |        | H        |                                 |
|         |        | YV       |                                 |
|         | STOVL  | YV       |                                 |
|         |        | FV       |                                 |
|         | VSR3   | VAD      |                                 |
|         |        | PSIV     |                                 |
|         | VXSC   | VSL1     |                                 |
|         | VAD    |          |                                 |
|         |        | ZV       |                                 |
|         | STORE  | ZV       |                                 |
|         | BOFF   | CALL     |                                 |
|         |        | JSWITCH  |                                 |
|         |        | ENDSTATE |                                 |
|         |        | GRP2PC   |                                 |
|         | LXA,2  | VLOAD    |                                 |
|         |        | COLREG   |                                 |
|         |        | ZV       |                                 |
|         | VSL3   |          | # ADJUST W-POSITION FOR STORAGE |
|         | STORE  | W        | +54D,2                          |
|         | VLOAD  |          |                                 |
|         |        | YV       |                                 |

```

Page 1349
VSL3 BOV
STORE WMATEND
 W,2

CALL GRP2PC

LXA,2 SSP
 COLREG
 S2
 0
INCR,2 SXA,2
 6
 YV
TIX,2 CALL
 RELOADSV
 GRP2PC
LXA,2 SXA,2
 YV
 COLREG

NEXTCOL CALL
 GRP2PC
LXA,2 VLOAD*
 COLREG
 W,2

VSR3 # ADJUST W-POSITION FOR INTEGRATION
STORE YV
VLOAD* AXT,1
 W +54D,2
 0

VSR3 # ADJUST W-VELOCITY FOR INTEGRATION
STCALL ZV
 DIFEQO

ENDSTATE BOV VLOAD
 GOBAQUE
 ZV
STOVL TNUV
 YV
STORE TDELTA V
BON BOFF
 MIDAVFLG
 CKMID2 # CHECK FOR MID2 BEFORE GOING TO TIMEINC
 DIMOFLAG
 TESTLOOP

```

July 12, 2016

973

```

EXIT
TC PHASCHNG
OCT 04022 # PHASE 1
TC UPFLAG # PHASE CHANGE HAS OCCURRED BETWEEN
ADRES REINTFLG # INTSTALL AND INTWAKE
TC INTPRET
SSP
 QPRET
 AMOVED
BON GOTO
 VINTFLAG

Page 1350
 ATOPCSM
 ATOPLEM
AMOVED SET SSP
 JSWITCH
 COLREG
DEC -30
BOFF SSP
 D60R9FLG
 NEXTCOL
 COLREG
DEC -48
GOTO NEXTCOL

RELOADSV DLOAD # RELOAD TEMPORARY STATE VECTOR
 TDEC # FROM PERMANENT IN CASE OF
STCALL TDEC1
 INTEGRV2 # BY STARTING AT INTEGRV2.
DIFEQCOM DLOAD # INCREMENT H AND DIFEQCNT.
 DAD
 DT/2
 H
INCR,1 SXA,1
DEC -12
 DIFEQCNT # DIFEQCNT SET FOR NEXT ENTRY.
STORE H
VXSC VSR1
 FV
VAD VXSC
 ZV
 H
VAD
 YV
STORE ALPHAV
BON GOTO
```

```
Page 1352
ORBITAL ROUTINE FOR EXTRAPOLATION OF THE W MATRIX. IT COMPUTES THE SECOND DERIVATIVE
VECTOR OF THE MATRIX AND CALLS THE NYSTROM INTEGRATION ROUTINES TO SOLVE THE DIFFERENTIAL
EQUATIONS. IT ALSO CALLS THE NYSTROM ROUTINE TO COMPUTE THE FIRST DERIVATIVE OF THE W MATRIX.
USES A TABLE OF VEHICLE POSITION VECTORS COMPUTED DURING THE INTEGRATION OF THE VEHICLE
```

|        |        |           |      |
|--------|--------|-----------|------|
| DOW..  | LXA,2  | DLOAD*    |      |
|        |        | PBODY     |      |
|        |        | MUEARTH,2 |      |
|        | STCALL | BETAM     |      |
|        |        | DOW..1    |      |
|        | STORE  | FV        |      |
|        | BOF    | INCR,1    |      |
|        |        | MIDFLAG   |      |
|        |        | NBRANCH   |      |
|        | DEC    | -6        |      |
|        | LXC,2  | DLOAD*    |      |
|        |        | PBODY     |      |
|        |        | MUEARTH   | -2,2 |
|        | STCALL | BETAM     |      |
|        |        | DOW..1    |      |
|        | BON    | VSR6      |      |
|        |        | MOONFLAG  |      |
|        |        | +1        |      |
|        | VAD    |           |      |
|        |        | FV        |      |
|        | STCALL | FV        |      |
|        |        | NBRANCH   |      |
| DOW..1 | VLOAD  | VSR4      |      |

|             |        |                            |
|-------------|--------|----------------------------|
|             | PDVL*  | ALPHAV<br>UNIT<br>VECTAB,1 |
|             | PDVL   | VPROJ<br>ALPHAV            |
|             | VXSC   | VSU<br>3/4                 |
|             | PDDL   | NORM<br>36D<br>S2          |
|             | PUSH   | DSQ                        |
|             | DMP    |                            |
|             | NORM   | PDDL<br>34D<br>BETAM       |
|             | SR1    | DDV                        |
|             | VXSC   |                            |
|             | LXA,2  | XAD,2<br>S2<br>S2          |
|             | XAD,2  | XAD,2<br>S2<br>34D         |
|             | VSL*   | RVQ                        |
| # Page 1353 |        | 0 -8D,2                    |
|             | SETLOC | ORBITAL1                   |
|             | BANK   |                            |
| 3/5         | 2DEC   | .6 B-2                     |
| THREE/8     | 2DEC   | .375                       |
| .3D         | 2DEC   | .3 B-2                     |
| 3/64        | 2DEC   | 3 B-6                      |
| DP1/4       | 2DEC   | .25                        |
| DQUARTER    | EQUALS | DP1/4                      |
| POS1/4      | EQUALS | DP1/4                      |
| 3/32        | 2DEC   | 3 B-5                      |
| 15/16       | 2DEC   | 15. B-4                    |

|                                                            |        |                       |     |
|------------------------------------------------------------|--------|-----------------------|-----|
| 3/4                                                        | 2DEC   | 3.0 B-2               |     |
| 7/12                                                       | 2DEC   | .5833333333           |     |
| 9/16                                                       | 2DEC   | 9 B-4                 |     |
| 5/128                                                      | 2DEC   | 5 B-7                 |     |
| DPZERO                                                     | EQUALS | ZEROVEC               |     |
| DP2/3                                                      | 2DEC   | .6666666667           |     |
| 2/3                                                        | EQUALS | DP2/3                 |     |
| OCT27                                                      | OCT    | 27                    |     |
|                                                            | BANK   | 13                    |     |
|                                                            | SETLOC | ORBITAL2              |     |
|                                                            | BANK   |                       |     |
| # IT IS VITAL THAT THE FOLLOWING CONSTANTS NOT BE SHUFFLED |        |                       |     |
|                                                            | DEC    | -11                   |     |
|                                                            | DEC    | -2                    |     |
|                                                            | DEC    | -9                    |     |
|                                                            | DEC    | -6                    |     |
|                                                            | DEC    | -2                    |     |
|                                                            | DEC    | -2                    |     |
|                                                            | DEC    | 0                     |     |
|                                                            | DEC    | -12                   |     |
|                                                            | DEC    | -9                    |     |
|                                                            | DEC    | -4                    |     |
| ASCALE                                                     | DEC    | -7                    |     |
|                                                            | DEC    | -6                    |     |
| # Page 1354                                                |        |                       |     |
|                                                            | 2DEC*  | 1.32715445 E16 B-54*  | # S |
|                                                            | 2DEC*  | 4.9027780 E8 B-30*    | # M |
| MUEARTH                                                    | 2DEC*  | 3.986032 E10 B-36*    |     |
|                                                            | 2DEC   | 0                     |     |
| J4REQ/J3                                                   | 2DEC*  | .4991607391 E7 B-26*  |     |
|                                                            | 2DEC   | -176236.02 B-25       |     |
| 2J3RE/J2                                                   | 2DEC*  | -.1355426363 E5 B-27* |     |
|                                                            | 2DEC*  | .3067493316 E18 B-60* |     |



|          |        |                      |
|----------|--------|----------------------|
| J2REQSQ  | 2DEC*  | 1.75501139 E21 B-72* |
| 3J22R2MU | 2DEC*  | 9.20479048 E16 B-58* |
| 5/8      | 2DEC   | 5 B-3                |
| -1/12    | 2DEC   | -.1                  |
| MUM      | =      | MUEARTH -2           |
| RECRATIO | 2DEC   | .01                  |
| RSPHERE  | 2DEC   | 64373.76 E3 B-29     |
| RDM      | 2DEC   | 16093.44 E3 B-27     |
| RDE      | 2DEC   | 80467.20 E3 B-29     |
| RATT     | EQUALS | 00                   |
| VATT     | EQUALS | 6D                   |
| TAT      | EQUALS | 12D                  |
| RATT1    | EQUALS | 14D                  |
| VATT1    | EQUALS | 20D                  |
| MU(P)    | EQUALS | 26D                  |
| TDEC1    | EQUALS | 32D                  |
| URPV     | EQUALS | 14D                  |
| COSPHI/2 | EQUALS | URPV +4              |
| UZ       | EQUALS | 20D                  |
| TVEC     | EQUALS | 26D                  |

This code is written to file `src/ORBITAL-INTEGRATION.s`.

**B.65 P11**

```

978 <src/P11.s 978>≡
 # Copyright: Public domain.
 # Filename: P11.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 533-550
 # Mod history: 2009-05-13 HG Started adapting from the Colossus249/ file
 # of the same name, using Comanche055 page
 # images 0533.jpg - 0550.jpg.
 # 2009-05-20 RSB Corrections: ERTHALT -> EARTHALT,
 # STATSW -> SATSW.
 # 2009-05-23 RSB At end of RESCALES, corrected TC 0 to TC Q.
 # Added an SBANK= prior to a 2CADR.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A
 #
 # Page 533
 # EARTH ORBIT INSERTION MONITOR PROGRAM
 # *****
 #
 # PROGRAM DESCRIPTION -P11-
 #
 # MOD NO. 1
 # MOD BY ELIASSEN
 #
 # FUNCTIONAL DESCRIPTION

```

```

#
P11 IS INITIATED BY
#
A) GYROCOMPASS PRG P02 WHEN LIFTOFF DISCRETE IS RECEIVED OR
B) BACKUP THRU VERB 75 ENTER
#
PROGRAM WILL
1. ZERO CMC CLOCK AT LIFTOFF (OR UPON RECEIPT OF BACKUP)
2. UPDATE TEPHEM TO TIME CMC CLOCK WAS ZEROED
3. INITIATE SERVICER AT PREREAD1
4. CHANGE MAJOR MODE TO 11
5. CLEAR DSKY IN CASE OF V 75
6. STORE LIFTOFF IMU-CDU ANGLES FOR ATT. ERROR DISPLAY
7. TERMINATE GYROCOMPASSING
8. COMPUTE INITIAL VECTORS -- --
9. COMPUTE REFSMMAT FOR PRELAUNCH ALIGNMENT WHERE $\bar{U}_X, \bar{U}_Y, \bar{U}_Z$ ARE
#
$\bar{U} = (\text{UNIT}(-R))$ LOCAL VERTICAL AT TIME OF LIFTOFF
\bar{Z}
$\bar{U} = \text{UNIT}(A)$, $A = \text{HOR VECTOR AT LAUNCH AZIMUTH}$
$\bar{X} \quad \bar{Y} \quad \bar{Z}$
$\bar{U} = \bar{U} * \bar{U}$
$\bar{U} \quad \bar{Z} \quad \bar{X}$
10. SET REFSMMAT KNOWN FLAG
11. SET AVGEXIT IN SERVICER TO VHHDOT TO
COMPUTE AND DISPLAY NOUN 62 EVERY 2 SECONDS
#
R1 V1 -- INERTIAL VELOCITY MAGNITUDE IN FPS
R2 HDOT -- RATE OF CHANGE OF VEHICLE VEL IN FPS
R3 H -- VEHICLE ALTITUDE ABOVE PAD IN NM
#
12. DISPLAY BODY AXES ATT. ERRORS ON FDAI NEEDLES
#
A) FROM L.O. TO RPSTART (APPROX. 0 TO +10SECS AFTER L.O.)
DESIRED ATTITUDE IS AS STORED AT L.O.
B) FROM RPSTART TO POLYSTOP (APPROX. +10 TO +133SECS AFTER LO)
DESIRED ATTITUDE IS SPECIFIED BY CMC PITCH AND ROLL
POLYNOMIALS DURING SATURN ROLLOUT AND PITCHOVER
#
Page 534
#
THE DISPLAY IS RUN AS LOW PRIORITY JOB APPROX.
EVERY 1/2 SEC OR LESS AND IS DISABLED UPON OVFL0 OF TIME1
#
SUBROUTINES CALLED
#
2PHSCHNG BANKCALL CALCGRA CDUTRIG CLEANDSP DANZIG
DELAYJOB EARTHTR ENDOFJOB FINDVAC IBNKCALL

```

```

INTPRET LALOTORV NEEDLER NEWMODEX PHASCHNG
POSTJUMP POWRSERS PREREAD1 REGODSPR S11.1
SERVEXIT TASKOVER TCDANZIG V1STO2S WAITLIST
#
ASTRONAUT REQUESTS (IF ALTITUDE ABOVE 300,000 FT)
#
DSKY --
MONITOR DISPLAY OF TIME TO PERIGEE R1 HOURS
R2 MINUTES
#
DSKY --
MONITOR DISPLAY OF R1 APOGEE ALTITUDE IN NAUTICAL MILES
R2 PERIGEE ALTITUDE IN NAUTICAL MILES
R3 TFF IN MINUTES/SECS
#
IF ASTRONAUT HAS REQUESTED ANY OF THESE DISPLAYS HE MUST
HIT PROCEED TO RETURN TO NORMAL NOUN 62 DISPLAY.
NORMAL EXIT MODE
#
ASTRONAUT VERB 37 ENTER 00 ENTER
#
ALARM MODES -- NONE
#
ABORT EXIT MODES --
#
OUTPUT
#
TLIFTOFF (DP) TEPHEM (TP)
REFSMMAT
DSKY DISPLAY
FDAI DISPLAY
#
ERASABLE INITIALIZATION
#
AZO, AXO, -AYO
LATITUDE
PADLONG
TEPHEM
PGNCSALT
POLYNUM THRU POLYNUM +14D
RPSTART
POLYSTOP
FLAGS SET OR RESET
Page 535
SET REFSMFLG
SET DVMON IDLE FLAG

```

```

CLEAR ERADFLAG
#
DEBRIS
#
LIFTTEMP
POLYNUM THRU POLYNUM +7
SPOLYARG
BODY1, BODY2, BODY3
VMAG2, ALTI, HDOT
CENTRALS, CORE SET, AND VAC AREAS

COUNT 34/P11

BITS5-6 = SUPER011
 BANK 42
 SETLOC P11ONE
 BANK

P11 EBANK= TEPHEM
 CA EBANK3
 TS EBANK

 EXTEND
 DCA REP11S # DIRECT RESTARTS TO REP11
 DXCH -PHASE3
 CS ZERO
 ZL
 TS LIFTTEMP
 DXCH -PHASE5 # INACTIVE GROUP 5, PRELAUNCH PROTECTION
P11+7 EXTEND
 DCA REP11SA
 DXCH TLIFTOFF

 EXTEND
 DCA TIME2
 DXCH LIFTTEMP # FORE RESTARTS

 CA ZERO
 ZL
 DXCH TIME2
REP11A-2 DXCH TLIFTOFF
REP11A-1 DXCH -PHASE3 # RESET PHASE

REP11A INHINT
 EXTEND
 DCA TEPHEM +1

```

# Page 536

DXCH TEPHEM1 +1  
CA TEPHEM

XCH TEPHEM1

EXTEND

DCA TLIFTOFF  
DAS TEPHEM1 +1  
ADS TEPHEM1

# CORRECT FOR OVERFLOW

TC PHASCHNG  
OCT 05023  
OCT 22000

INHINT

EXTEND

DCA TEPHEM1  
DXCH TEPHEM  
CA TEPHEM1 +2  
XCH TEPHEM +2

CAF EBDVCNT  
TS EBANK  
EBANK= DVCNTR  
TC IBNKCALL  
CADR PREREAD1

# ZERO PIPS AND INITIALIZE AVERAGEG

TC PHASCHNG  
OCT 05023  
OCT 22000

# CONTINUE HERE ON RESTART

CAF .5SEC  
TC WAITLIST  
EBANK= BODY3  
2CADR ATERTASK

# START ATT ERROR DISPLAY  
# IN .5 SEC

TC NEWMODEX  
MM 11

# DISPLAY MM 11

TC UPFLAG  
ADRES NODOPO1

CA POWDNCOD  
TS DNLSTCOD

# SWITCH TO POWERED FLIGHT DOWNLIST

TC BANKCALL

```

Page 537
CADR CLEANDSP # CLEAR DSKY IN CASE OF V75

TC 2PHSCHNG
OCT 40514 # PROTECT ATERTASK
OCT 00073
CAF EBQPLACE

TS EBANK

EBANK= QPLACES
CA P11XIT # SET EXIT FROM PROUT IN EARTH
TS QPLACES
TC INTPRET
VLOAD MXV
 THETAN
 XSM
VSL1 VAD
 ERCOMP
STODL ERCOMP
 TLIFTOFF
SSP GOTO
 S2
CADR PROUT # RETURN FROM EARTH
 EARTH +3

MATRXJOB ZL # STORE DP GIMBAL ANGLES FOR ATTITUDE
CA CDUX # ERROR DISPLAY AFTER LIFTOFF
DXCH OGC
ZL
CA CDUY
DXCH IGC
ZL
CA CDUZ
DXCH MGC
TC INTPRET
 -
VLOAD VSR1 # SCALE OGC B-1
 OGC
STORE OGC
SSP
 RTX2 # ZERO RTX2
 0 # FOR
 0 # EARTH
DLOAD PDDL
 PGNCALT # ALTITUDE OF PGNC
 PADLONG # LONGITUDE
PDDL VDEF
 LATITUDE # GEODETIC LATITUDE
STODL LAT # LAT, LONG, ALT ARE CONSECUTIVE

```

# Page 538

```

 HI6ZEROS # TIME = 0
CLEAR CALL
 ERADFLAG
 LALOTORV # CONVERT TO POSITION VECTOR IN REF. COORDS.

STCALL RN1 #
 GETDOWN # RETURN WITH VECTOR FOR DOWN DIRECTION
VCOMP UNIT
STOVL REFSMMAT +12D # UNITZ = UNIT(GRAV)
 RN1
VXV VXSC

 UNITW # SCALED AT 1
 -ERTHRAT # V = EARTHRAE X R
VSL4 # SCALE TO 2(7) M/CS
STOVL VN1
 REFSMMAT +12D
VXV UNIT
 UNITW # (REF3 X UNITW) = EAST
PUSH VXV
 REFSMMAT +12D # (EAST X REF3) = -SOUTH
UNIT PDDL
 LAUNHAZ # COS(AZ)*SOUTH
COS VXSC
STADR
STODL REFSMMAT # TEMPORARY STORAGE
 LAUNHAZ
SIN VXSC # SIN(AZ)*EAST
VAD UNIT # SIN(AZ)*EAST - COS(AZ)*SOUTH = REF1
 REFSMMAT
STORE REFSMMAT

VXV UNIT # (REF1 X REF3) = -REF3
 REFSMMAT +12D
VCOMP
STORE REFSMMAT +6
DLOAD DSU
 DPHALF # 1/2 REV
 LAUNHAZ
DAD PDDL
 AZIMUTH
 SATRLRT # SET SATRLRT = -SATRLRT IF
SIGN STADR # (1/2REV -LAVNHAZ +AZIMUTH) IS NEGATIVE
STORE SATRLRT # FOR ROLL CALC IN FDAI ATT. ERROR DISPLAY
SET EXIT
 REFSMFLG # SET REFSMMAT KNOWN FLAG

```



July 12, 2016

985

```
TC PHASCHNG
OCT 04023

EXTEND
DCA P11SCADR
DXCH AVGEXIT # SET AVGEXIT

CA PRI031 # 2 SECONDS AT 2(+8)
TS 1/PIPADT

EBANK= RCSFLAGS
CA EBANK6
TS EBANK

INHINT

Page 539

CS ZERO
TS TBASE5 # RESTART READACCS 2 SECONDS AFTER LIFTOFF

CS TIME1
AD 2SECS # DO READACCS 2 SECONDS AFTER LIFTOFF

CCS A # CHECK TO INSURE DT IS POSITIVE
TCF +3 # TIME POSITIVE
TCF +2 # CANNOT GET HERE
CA ZERO # TIME NEGATIVE -- SET TO 1
AD ONE # RESTORE TIME -- OR MAKE POSITIVE

TC WAITLIST
EBANK= AOG
2CADR READACCS

TC 2PHSCHNG
OCT 00003 # TURN OFF GROUP 3
OCT 00025 # PROTECT NORMLIZE AND READACCS

TC POSTJUMP
CADR NORMLIZE # DO NORMLIZE AND ENDOFJOB

EBANK= TEPHEM
INHINT
CCS PHASE5
TC ENDOFJOB

REP11
```

```

 CCS LIFTTEMP
 TCF +4
 TCF +3
 TCF +2
 TCF P11+7

 CS TLIFTOFF
 EXTEND
 BZMF ENDREP11

 CCS TIME2 # ** TIME2 MUST BE NON-ZERO AT LIFTOFF **
 TCF REP11A -5 # T2,T1 NOT YET ZEROED, GO AND DO IT

 EXTEND # T2,T1 ZEROED, SET TLIFTOFF
 DCA LIFTTEMP
 TCF REP11A-2

ENDREP11 EXTEND
 DCA REP11SA
 TCF REP11A-1

#Page 540
REP11S 2OCT 7776600011

REP11SA 2OCT 7776400013

P11XIT GENADR P11OUT
-ERTHRAT 2DEC* -7.292115138 E-7 B18* # - EARTH RATE AT 2(18)

 EBANK= BODY3
P11SCADR 2CADR VHHDOT

POWDNCOD EQUALS THREE

 EBANK= BODY3
VHHDOT IS EXECUTED EVERY 2 SECONDS TO DISPLAY ON DSKY
VI INERTIAL VELOCITY MAGNITUDE
HDOT RATE OF CHANGE OF ALT ABOVE L PAD RADIUS
H ALTITUDE ABOVE L PAD RADIUS

VHHDOT TC INTPRET
 CALL # LOAD VMAGI, ALTI,
 S11.1 # HDOT FOR DISPLAY
 EXIT
 TC PHASCHNG
 OCT 00035

```

July 12, 2016

987

```
CAF V06N62 # DISPLAY IN R1 R2 R3
TC BANKCALL # VI HDOT H
CADR REGODSP

ATERTASK CAF PRI01 # ESTABLISH JOB TO DISPLAY ATT ERRORS
 TC FINDVAC # COMES HERE AT L.O. + .33 SEC
 EBANK= BODY3
 2CADR ATERJOB

 CS RCSFLAGS # SET BIT3 FOR
 MASK BIT3 # NEEDLER
 ADS RCSFLAGS # INITIALIZATION PASS
 TC IBNKCALL # AND GO
 CADR NEEDLER # DO IT
 CA BIT1 # SET SW
 TS SATSW # FOR DISPLAY
 TC TASKOVER
GETDOWN STQ SETPD
 INCORPEX
 OD
 DLOAD
 HI6ZEROS

#Page 541

 STODL 6D
 DPHALF
 STCALL 8D
 LALOTORV +5

THIS SECTION PROVIDES ATTITUDE ERROR DISPLAYS TO THE FDAI DURING SONE BOOST
#
COMPUTE DESIRED PITCH W.R.T. PAD LOCAL VERTICAL AT LIFTOFF
2 3 4 5 6
PITCH = A0+A1T+A2T +A3T +A4T +A5T +A6T
SCALED TO 32 REVS. -14
IF TL = TIME IN SECS FROM L.O., THEN T = 100(TL-RPSTART)2
WHERE TL GE RPSTART
TL LE (-POLYSTOP + RPSTART)
COMPUTE DESIRED ROLL WEHRE ROLL EQUALS ANGLE FROM
LAUNCHAZ TO -Z(S/C) AS SEEN FROM X(S/C).
ROLL = LAUNCHAZ-AZIMUTH-.5+SATRLRT*T IN REV
SATRLRT = RATE OF ROLL IN REV/CENTI-SEC
T, IN CENTI-SEC, IS DEFINED AS ABOVE, INCLUSIVE OF TIME RESTRICTIONS
#
FOR SIMPLICITY, LET P = 2*PI*PITCH
R = 2*PI*ROLL
#
CONSTRUCT THE TRANSFORMATION MATRIX, TSMV, GIVING DESIRED S/C AXES IN
```

```

TERMS OF SM COORDINATES. LET THE RESULTING ROWS EQUAL THE VECTORS XDC,
YDC, AND ZDC.
#
* (SIN(P) 0 -COS(P)) (XDC)
TSMV = (-SIN(R)*COS(P) -COS(R) -SIN(R)*SIN(P)) = (YDC)
(-COS(R)*COS(P) SIN(R) -COS(R)*SIN(P)) (ZDC)
#
XDC,YDC,ZDC ARE USED AS INPUT TO CALCGTA FOR THE EXTRACTION OF THE
EULER SET OF ANGLES WHICH WILL BRING THE SM INTO THE DESIRED
ORIENTATION. THIS EULER SET, OGC, IGC, AND MGC, MAY BE IDENTIFIED
AS THE DESIRED CDU ANGLES.
#
(XDC) (OGC)
(YDC) ---) CALCGTA ---) (IGC)
(ZDC) (MGC)
#
-
DEFINE THE VECTOR DELTACDU.
#
- (OGC) (CDUX)
DELTACDU = (IGC) - (CDUY)
(MGC) (CDUZ)
#
COMPUTE ATTITUDE ERRORS, A, WHERE - * -
A = TGSC*DELTACDU
#
* (1 SIN(CDUZ) 0) THE GIMBAL ANGLES
#
#Page 542
TGSC = (0 COS(CDUX)*COS(CDUZ) SIN(CDUX)) = TO SPACECRAFT AXES
(0 -SIN(CDUX)*COS(CDUZ) COS(CDUX)) CONVERSION MATRIX
#
-
THE ATTITUDE ERRORS, A, ARE STORED ONE HALF SINGLE PRECISION IN
THE REGISTERS AK, AK1, AK2 AS INPUT TO NEEDLER, THE FDAI ATTITUDE
ERROR DISPLAY ROUTINE.
#
#
ATERJOB CAE FLAGWRD6 # CHECK FLAGWRD6
MASK OCT60000 # BITS 14, 15
EXTEND
BZF +2 # OK -- CONTINUE
TC ENDOFJOB # STAURN STICK ON -- KILL JOB
CAF BIT10 # CHECK IF S/C CONTROL
EXTEND # OF SATURN PANEL
RAND CHAN30 # SWITCH IS ON
EXTEND
BZF STRSAT # IT IS -- GO STEER
CCS SATSW # IT IS NOT -- WAS IT ON LAST CYCLE
TC ATTDISP # NO -- CONTINUE
TC ATRESET # YES -- REINITIALIZE NEEDLER

```

```

 TC ATRESET # YES -- REINITIALIZE NEEDLER
ATTDISP CS RPSTART # PITCH/ROLL START TIME
 AD TIME1
 EXTEND
 BZMF NOPOLY # IF MINUS THEN ATTITUDE HOLD
 TS MPAC # MPAC = TIME1 - RPSTART
 TS SPOLYARG # SAVE FOR USE IN ROLL CALCULATION
 AD POLYSTOP # NEG PITCHOVER TIME IN CSECS
 EXTEND
 BZMF +2
 TC NOPOLY # GO TO ATTITUDE HOLD
 CA TIME2
 EXTEND
 BZMF +2
 TC NOPOLY # GO TO ATTITUDE HOLD
 CAE POLYNUM
 TS L
 CAF COEFPOLY # EVALUATE PITCH POLYNOMIAL
 TC POWRSERS # SCALED TO 32 REVOLUTIONS
 CA ZERO # RETURN WITH PITCH(32REV)
 TS MODE # STORED MPAC, MPAC +1
 TC INTPRET
 SETPD SL # 32(PITCH(32REV)) = PITCH(REV)
 0
 5
 PUSH
 GOTO
 ATTDISP1 # AROUND SETLOC

#Page 543
#
*
CONSTRUCT SM TO S/C MATRIX, TSMV

 SETLOC P11TWO
 BANK
 # 36 IN COL., 34 IN DISK

 COUNT 36/P11

ATTDISP1 COS DCOMP
 STODL 14D # -.5*COS(P)
 SIN
 STODL 10D # .5*SIN(P)
 ZEROVECS
 STORE 12D # 0

EVALUATE ROLL = LAUNCHAZ-AZIMUTH-.5+SATRLRT*T
 SLOAD DMP

```

```

 SPOLYARG # TIME1 - RSPSTART, CSECS B-14.
 SATRLRT
SL DSU
 14D
 DPHALF
DAD DSU # ASSUMING X(SM) ALONG LAUNCH AZIMUTH,
 LAUNHAZ # LAUNHAZ = ANGLE FROM NORTH TO X(SM).
 AZIMUTH # AZIMUTH = -ANGLE FROM NORTH TO Z(S/C)
RTB # DETERMINE IF ROLLOUT
 RLTST # IS COMPLETED
ATTDISPR PUSH COS # CONTINUE COMPUTING TSMV
 PUSH # LET R(RAD) = 2*PI*ROLL(REV)
 DMP SL1
 14D
STODL 22D # -.5*COS(R)*COS(P)
DCOMP
STORE 18D # -.5*COS(R)
DMP SL1
 10D
STODL 26D # -.5*COS(R)*SIN(P)
SIN PUSH
STORE 24D # .5*SIN(R)
DMP SL1
 14D
STODL 16D # -.5*SIN(R)*COS(P)
DCOMP
DMP SL1
 10D
STOVL 20D # -.5*SIN(R)*SIN(P)
 10D

FROM TSMV FIND THE HALF UNIT VECTORS XDC,YDC,ZDC = INPUT TO CALCGTA
Page 544
 UNIT
STOVL XDC # XDC = .5*UNIT(SIN(P),0,-COS(P))
 16D
 UNIT
STOVL YDC # YDC = .5*UNIT(-SIN(R)*COS(P),-COS(R),
 22D # -SIN(R)*SIN(P))
 UNIT
STCALL ZDC # ZDC = .5*UNIT(-COS(R)*COS(P),SIN(R),
 CALCGTA # -COS(R)*SIN(P))

CALL CALCGTA TO COMPUTE DESIRED SM ORIENTATION OGC,IGC, AND MGC
#
FIND DIFFERENCE VECTOR DELTACDU = OGC-CDUX

```

```

#
ENTER HERE IF ATTITUDE HOLD

NOPOLYM VLOAD PUSH # OGC IGC
 OGC # CHANGE IGC TO MGC FOR COMPATIBILITY
 PUSH CALL # MGC OGC
 CDUTRIG # WITH Y,Z,X ORDER OF CDUSPOT
 VLOAD RTB # _ DPHI OGC-CDUX , PD4
 2 # DELTACDU = DTHETA = IGC-CDUY , 0
 V1ST02S # DPSI MGC-CDUZ , 2
 STOVL BOOSTEMP
 ZEROVECS
 STOVL 0
 CDUSPOT
 RTB RTB
 V1ST02S
 DELSTOR
 STODL 10D
 SINCDUZ
 DMP SL1
 0
 DAD SR2 # CHANGE SCALE OF AK TO 2REVS
 4
 GOTO
 ATTDISP2

 SETLOC P11ONE
 BANK
 COUNT 34/P11

ATTDISP2 STODL 16D # 16D, .5(DPHI + DTHETA*SIN(CDUZ))
 COSCDUZ
 DMP PUSH
 0
 DMP SL1
 COSCDUX
 PDDL DMP

Page 545
 SINCDUX
 2
 DAD SL1
 STADR
 STODL 17D # 17D, .5(DTHETA*COS(CDUX)*COS(CDUZ)
 DMP SL1 #
 SINCDUX +DPSI*SIN(CDUX))
 PDDL DMP

```

```

COSCDUX
2
DSU SL1
STADR
STORE 18D # 18D, .5(-DTHETA*SIN(CDUX)*COS(CDUZ)
TLOAD #
 +DPSI*COS(CDUX))
 16D
STORE AK # STORE ATTITUDE ERRORS IN AK,AK1,AK2
EXIT

CA SATSW
CCS A # CHK TAKEOVER STATUS
TC SATOUT # POS -- DISPLAY ONLY
TC AKLOAD # 0 STORE BIAS
STEERSAT TC INTPRET # NEG STEER L/V
TLOAD TAD
 BIASAK
 AK
STORE AK # AKS = AKS - STORED BIAS
EXIT
CA AK
TC RESCALES
TS AK
CA AK1
TC RESCALES
TS AK1
CA AK2
TC RESCALES
TS AK2
DISPLAY ATTITUDE ERRORS ON FDAI VIA NEEDLER

SATOUT TC BANKCALL
 CADR NEEDLER
ATERSET CAF OCT31 # DELAY .25 SEC
 TC BANKCALL # EXECUTION + DELAY = .56SEC APPROX
 CADR DELAYJOB
 TC ATERJOB # END OF ATT ERROR DISPLAY CYCLE

AKLOAD CS AK # STORE AKS
 TS BIASAK # INTO BIAS
 CS AK1 # COMPLEMENTED
 TS BIASAK +1

Page 546
 CS AK2
 TS BIASAK +2
 CS BIT1 # SET SW

```



|          |        |          |                                       |
|----------|--------|----------|---------------------------------------|
|          | TS     | SATSW    | # TO STEER                            |
|          | TC     | STEERSAT | # GO STEER                            |
| STRSAT   | CA     | SATSW    | # CHECK IF NEDLER                     |
|          | EXTEND |          | # HAS BEEN INITIALIZED                |
|          | BZMF   | ATTDISP  | # YES -- CONTINUE                     |
| ATRESET  | CS     | RCSFLAGS | # NO -- SET                           |
|          | MASK   | BIT3     | # INITIALIZATION SW                   |
|          | ADS    | RCSFLAGS | # FOR NEDLER                          |
|          | TC     | BANKCALL | # AND GO                              |
|          | CADR   | NEEDLER  | # DO IT                               |
|          | CAF    | REVCNT   | # OCT 6                               |
|          | TC     | BANKCALL | # DELAY JOB                           |
|          | CADR   | DELAYJOB | # 60 MS --WAIT TILL IMUERRCNTR ZEROED |
|          | CCS    | SATSW    | # CHECK SW STATUS                     |
|          | TC     | TAKEON   | # POS STEER INIT.                     |
|          | TC     | +1       | # 0 RETURN TO DISPLAY                 |
|          | CA     | BIT1     | # NEG RETURN TO DISPLAY               |
|          | TS     | SATSW    | # SW = DISPLAY ON                     |
|          | CS     | BIT9     | # DISABLE                             |
|          | EXTEND |          | # SIVB                                |
|          | WAND   | CHAN12   | # TAKEOVER                            |
|          | TC     | SATOUT   | # DISPLAY                             |
| TAKEON   | CAF    | BIT9     | # ENABLE                              |
|          | EXTEND |          | # SIVB                                |
|          | WOR    | CHAN12   | # TAKEOVER                            |
|          | CA     | ZERO     | # INDICATE NEEDLER                    |
|          | TS     | SATSW    | # WAS INITIALIZED                     |
|          | TC     | SATOUT   |                                       |
| S11.1    | VLOAD  | ABVAL    |                                       |
|          |        | VN       |                                       |
|          | STOVL  | VMAGI    | # VI SCALED 2(7) IN METERS/CSEC       |
|          |        | RN       |                                       |
|          | UNIT   | DOT      |                                       |
|          |        | VN       |                                       |
|          | SL1    |          |                                       |
|          | STODL  | HDOT     |                                       |
|          |        | RPAD     |                                       |
|          | BOF    | VLOAD    |                                       |
|          |        | AMOONFLG |                                       |
|          |        | EARTHALT |                                       |
|          |        | RLS      |                                       |
|          | ABVAL  | SR2      |                                       |
| EARTHALT | BDSU   |          |                                       |
|          |        | 36D      |                                       |
|          | STORE  | ALTI     |                                       |
|          | RVQ    |          |                                       |

```

DELSTOR CA BOOSTEMP
Page 547

EXTEND
MSU MPAC
INDEX FIXLOC
TS 0
CA BOOSTEMP +1
EXTEND
MSU MPAC +1
INDEX FIXLOC
TS 2
CA BOOSTEMP +2
EXTEND
MSU MPAC +2
INDEX FIXLOC
TS 4
TCF DANZIG
RLTST CA MPAC # DETERMINE IF ROLLOUT
EXTEND # IS COMPLETED
MP SATRLRT +1
EXTEND
BZMF DANZIG # UNLIKE SIGNS STILL ROLLING
EXTEND # ROLLOUT COMPLETED
DCA MBDYTCTL +2 # ZERO OUT ROLL CONTRIBUTION
DXCH MPAC
TC DANZIG
NOPOLY TC INTPRET # COMES HERE IF
SETPD GOTO # ATTITUDE HOLD
 0
 NOPOLYM
COEFPOLY ADRES POLYLOC
V06N62 VN 0662
RESCALES EXTEND
 MP SATSCALE # RESCALE AK S FOR
 DDOUBL # NEW HARDWARE
 DDOUBL # SCALING FOR
 TC Q # STEERING
 # SATURN

SATURN TAKEOVER FUNCTION

#
PROGRAM DESCRIPTION
#
MOD NUMBER 1
MOD BY ELIASSEN
#

```

```

FUNCTIONAL DESCRIPTION
#
DURING THE COASTING PHASE OF SIVB ATTACHED, THE
ASTRONAUT MAY REQUEST SATURN TAKEOVER THROUGH
EXTENDED VERB 46 (BITS 13,14 OF DAPDATR1 SET).
THE CMC REGARDS RHC COMMANDS AS BODY-AXES RATE
COMMANDS AND IT TRANSMITS THESE TO SATURN AS DC
#Page 548
VOLTAGES. THE VALUE OF THE CONSTANT RATE COMMAND
IS 0.5 DEG/SEC. AN ABSENCE OF RHC ACTIVITY RE-
SULTS IN A ZERO RATE COMMAND.
#
THE FDAI ERROR NEEDLES WILL INDICATE THE VALUE
OF THE RATE COMMAND.
#
CALLING SEQUENCE
#
DAPFIG +9D TC POSTJUMP
CADR SATSTKON
#
SUBROUTINES CALLED
#
ENDEXT
IBNKCALL
STICKCHK
NEEDLER
T5RUPT
RESUME
#
ASTRONAUT REQUESTS
#
ENTRY -- VERB 46 ENTER
(CONDITION -- BITS 13, 14 OF DAPDATR1 SET)
#
EXIT -- VERB 48 ENTER (FLASH V06N46)
VERB 21 ENTER AXXXX ENTER WHERE A=0 OR 1
VERB 34 ENTER
VERB 46 ENTER
#
NORMAL EXIT MODE
#
VERB 46 ENTER (SEE ASTRONAUT ABOVE)
#
ALARM OR ABORT EXIT MODES
#
NONE

```

```

#
OUTPUT
#
SATURN RATES IN CDUXCMD, CDUYCMD, CDUZCMD
#
ERASABLE INITIALIZATION
#
DAPDATR1 (BITS 13, 14 MUST BE SET)
#
DEBRIS
#
CENTRALS
#Page 549
CDUXCMD, CDUYCMD, CDUZCMD

 BANK 43
 SETLOC EXTVERBS
 BANK

SATSTKON COUNT 23/STTKE
 EXTEND
 DCA 2REDOSAT
 INHINT
 DXCH T5LOC
 CAF POSMAX
 TS TIME5
 CS FLAGWRD6 # TURN ON BITS 15,14 OF
 MASK RELTAB11 # FLAGWRD6
 ADS FLAGWRD6 # SATSTICK CONTROL OF T5
 TC IBNKCALL # ZERO JET CHANNELS IN 14 MS AND THEN
 CADR ZEROJET # LEAVE THE T6 CLOCK DISABLED
 RELINT
 TC GOPIN # EXIT THUS BECAUSE WE CAME VIA V46

 SBANK= PINSUPER # Added RSB 2009
 EBANK= BODY3
 2REDOSAT 2CADR REDOSAT

 SBANK= LOWSUPER
 BANK 32
 SETLOC P11FOUR
 BANK

REDOSAT LXCH BANKRUPT # ALSO COMES HERE FOR RESTARTS
 EXTEND

```

```

QXCH QRUPT
CS RCSFLAGS # TURN ON BIT3 OF RCSFLAGX
MASK BIT3 # FOR
ADS RCSFLAGS # NEEDLER INITIALIZATION
TC IBNKCALL
CADR NEEDLER # DISABLE IMU ERR COUNTERS ETC.
CAF BIT9 # SIVB
EXTEND
WOR CHAN12 # ENABLE
EXTEND
DCA 2SATSTCK
DXCH T5LOC
CAF 100MST5 # IN 100 MSECS
TS TIME5
TCF RESUME # END OF SATURN STICK INITIALIZATION

```

# THIS SECTION IS EXECUTED EVERY 100 MSECS.

#Page 550

```

SATSTICK
 LXCH BANKRUPT
 EXTEND
 QXCH QRUPT

 CAF 2SATSTCK # SET UP RUPT
 TS T5LOC # LO ORDER LOC SET
 CAF 100MST5 # 100 MSECS
 TS TIME5
 CAF STIKBITS
 EXTEND
 RXOR CHAN31 # CHECK IF MAN ROT BITS SAME
 MASK STIKBITS
 TC IBNKCALL # SET RATE INDICES
 CADR STICKCHK # FOR PITCH YAW AND ROLL

 INDEX RMANNDX # SET SATURN RATES
 CA SATRATE
 TS AK # ROLL
 INDEX PMANNDX
 CA SATRATE
 TS AK1 # PITCH
 INDEX YMANNDX
 CA SATRATE
 TS AK2 # YAW

 TC IBNKCALL # FOR SATURN INTERFACE AND FDAI DISPLAY
 CADR NEEDLER
 TCF RESUME # END OF SATURN STICK CONTROL

```

|          |        |          |
|----------|--------|----------|
| STIKBITS | OCT    | 00077    |
| 100MST5  | DEC    | 16374    |
|          | EBANK= | BODY3    |
| 2SATSTCK | 2CADR  | SATSTICK |

This code is written to file `src/P11.s`.

**B.66 P12**

```

999 <src/P12.s 999>≡
 # Copyright: Public domain.
 # Filename: P12.agc
 # Purpose: Part of the source code for Luminary 1A build 099.
 # It is part of the source code for the Lunar Module's (LM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 838-842
 # Mod history: 2009-05-23 HG Transcribed from page images.
 #
 # This source code has been transcribed or otherwise adapted from
 # digitized images of a hardcopy from the MIT Museum. The digitization
 # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
 # the Museum. Many thanks to both. The images (with suitable reduction
 # in storage size and consequent reduction in image quality as well) are
 # available online at www.ibiblio.org/apollo. If for some reason you
 # find that the images are illegible, contact me at info@sandroid.org
 # about getting access to the (much) higher-quality images which Paul
 # actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
 # 16:27 JULY 14, 1969

 # Page 838

 BANK 24
 SETLOC P12
 BANK

 EBANK= DVCNTR
 COUNT* $$/P12

P12LM TC PHASCHNG
 OCT 04024

 TC BANKCALL
 CADR R02BOTH # CHECK THE STATUS OF THE IMU.

 TC UPFLAG
 ADRES MUNFLAG

```

# Page 839  
P12LMB

|        |          |                                             |
|--------|----------|---------------------------------------------|
| TC     | UPFLAG   | # INSURE 4-JET TRANSLATION CAPABILITY.      |
| ADRES  | ACC4-2FL |                                             |
| TC     | UPFLAG   | # PREVENT R10 FROM ISSUING CROSS-POINTER    |
| ADRES  | R10FLAG  | # OUTPUTS.                                  |
| TC     | CLRADMOD | # INITIALIZE RADMODES FOR R29               |
| TC     | DOWNFLAG | # CLEAR RENDEVOUS FLAG FOR P22              |
| ADRES  | RNDVZFLG |                                             |
| CAF    | THRESH2  | # INITIALIZE DVMON                          |
| TS     | DVTHRUSH |                                             |
| CAF    | FOUR     |                                             |
| TS     | DVCNTR   |                                             |
| CA     | ZERO     |                                             |
| TS     | TRKMKCNT | # SHOW THAT R29 DOWNLINK DATA IS NOT READY. |
| CAF    | V06N33A  |                                             |
| TC     | BANKCALL | # FLASH TIG                                 |
| CADR   | GOFLASH  |                                             |
| TCF    | GOTOPOOH |                                             |
| TCF    | +2       | # PROCEED                                   |
| TCF    | -5       | # ENTER                                     |
| TC     | PHASCHNG |                                             |
| OCT    | 04024    |                                             |
| TC     | INTPRET  |                                             |
| CALL   |          | # INITIALZE WM AND /LAND/                   |
|        | GUIDINIT |                                             |
| SET    | CALL     |                                             |
|        | FLPI     |                                             |
|        | P12INIT  |                                             |
| DLOAD  | (TGO)A   | # SET TGO TO AN INITIAL NOMINAL VALUE.      |
| STODL  | TGO      |                                             |
|        | TIG      |                                             |
| STCALL | TDEC1    |                                             |
|        | LEMPREC  | # ROTATE THE STATE VECTORS TO THE           |
| VLOAD  | MXV      | # IGNITION TIME.                            |
|        | VATT     |                                             |
|        | REFSMMAT |                                             |
| VSL1   |          |                                             |
| STOVL  | V1S      | # COMPUTE V1S = VEL(TIG)*2(-7) M/CS.        |



July 12, 2016

1001

```

 RATT
MXV VSL6
 REFSMMAT
STCALL R # COMPUTE R = POS(TIG)*2(-24) M.
 MUNGRAV # COMPUTE GDT1/2(TIG)*2(-T)M/CS.
VLOAD UNIT
 R
STCALL UNIT/R/ # COMPUTE UNIT/R/ FOR YCOMP.
 YCOMP
SR DCOMP
 5D
STODL XRANGE # INITIALIZE XRANGE FOR NOUN 76
 VINJNOM
STODL ZDOTD
 RDOTDNOM
STORE RDOTD
EXIT

TC PHASCHNG
OCT 04024

NEWLOAD CAF V06N76 # FLASH CROSS-RANGE, AND APOLUNE VALUES.
 TC BANKCALL
 CADR GOFLASH
 TCF GOTOPOOH
 TCF +2 # PROCEED
 TCF NEWLOAD # ENTER NEW DATA.

CAF P12ADRES
TS WHICH

TC PHASCHNG
OCT 04024

TC INTPRET
DLOAD SL
 XRANGE
 5D
DAD

 Y
STOVL YCO
 UNIT/R/
VXSC VAD
 49FPS
 V1S
```

# Page 840

1002

July 12, 2016

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | STORE  | V        | # V(TIPOVER) = V(IGN) + 57FPS (UNIT/R/)    |
|            | DOT    | SL1      |                                            |
|            |        | UNIT/R/  |                                            |
|            | STOVL  | RDOT     | # RDOT = 2(-7)                             |
|            |        | UNIT/R/  |                                            |
|            | VXV    | UNIT     |                                            |
|            |        | QAXIS    |                                            |
|            | STORE  | ZAXIS1   |                                            |
|            | SETGO  |          |                                            |
|            |        | FLVR     |                                            |
|            |        | ASCENT   |                                            |
| P12RET     | DLOAD  |          |                                            |
|            |        | ATP      | # ATP(2)*2(18)                             |
|            | DSQ    | PDDL     |                                            |
|            |        | ATY      | # ATY(2)*2(18)                             |
|            | DSQ    | DAD      |                                            |
|            | BZE    | SQRT     |                                            |
|            |        | YAWDUN   |                                            |
|            | SL1    | BDDV     |                                            |
|            |        | ATY      |                                            |
|            | ARCSIN |          |                                            |
| YAWDUN     | STOVL  | YAW      |                                            |
|            |        | UNFC/2   |                                            |
|            | UNIT   | DOT      |                                            |
|            |        | UNIT/R/  |                                            |
|            | SL1    | ARCCOS   |                                            |
|            | DCOMP  |          |                                            |
|            | STORE  | PITCH    |                                            |
|            | EXIT   |          |                                            |
|            | TC     | PHASCHNG |                                            |
|            | OCT    | 04024    |                                            |
|            | TC     | DOWNFLAG |                                            |
|            | ADRES  | FLPI     |                                            |
|            | INHINT |          |                                            |
|            | TC     | IBNKCALL |                                            |
|            | CADR   | PFLITEDB |                                            |
|            | RELINT |          |                                            |
|            | TC     | POSTJUMP |                                            |
|            | CADR   | BURNBABY |                                            |
| P12INIT    | DLOAD  |          | # INITIALIZE ENGINE DATA. USED FOR P12 AND |
| # Page 841 |        | (1/DV)A  | # P71.                                     |

July 12, 2016

1003

|          |        |          |                                            |
|----------|--------|----------|--------------------------------------------|
|          | STORE  | 1/DV3    |                                            |
|          | STORE  | 1/DV2    |                                            |
|          | STODL  | 1/DV1    |                                            |
|          |        | (AT)A    |                                            |
|          | STODL  | AT       |                                            |
|          |        | (TBUP)A  |                                            |
|          | STODL  | TBUP     |                                            |
|          |        | ATDECAY  |                                            |
|          | DCOMP  | SL       |                                            |
|          |        | 11D      |                                            |
|          | STORE  | TTO      |                                            |
|          | SLOAD  | DCOMP    |                                            |
|          |        | APSVEX   |                                            |
|          | SR2    |          |                                            |
|          | STORE  | VE       |                                            |
|          | BOFF   | RVQ      |                                            |
|          |        | FLAP     |                                            |
|          |        | COMMINIT |                                            |
| COMMINIT | DLOAD  | DAD      | # INITIALIZE TARGET DATA. USED BY P12, P70 |
|          |        | HINJECT  | # AND P71 IF IT DOES NOT FOLLOW P70.       |
|          |        | /LAND/   |                                            |
|          | STODL  | RCO      |                                            |
|          |        | HI6ZEROS |                                            |
|          | STORE  | TXO      |                                            |
|          | STORE  | YCO      |                                            |
|          | STORE  | RDOTD    |                                            |
|          | STOVL  | YDOTD    |                                            |
|          |        | VRECTCSM |                                            |
|          | VXV    | MXV      |                                            |
|          |        | RRECTCSM |                                            |
|          |        | REFSMMAT |                                            |
|          | UNIT   |          |                                            |
|          | STORE  | QAXIS    |                                            |
|          | RVQ    |          |                                            |
| P12ADRES | REMADR | P12TABLE |                                            |
|          | SETLOC | P12A     |                                            |
|          | BANK   |          |                                            |
|          | COUNT* | \$\$/P12 |                                            |
| GUIDINIT | STQ    | SETPD    |                                            |
|          |        | TEMPR60  |                                            |
|          |        | OD       |                                            |
|          | VLOAD  | PUSH     |                                            |
|          |        | UNITZ    |                                            |
|          | RTB    | PUSH     |                                            |

```
 LOADTIME
 CALL
 RP-TO-R
Page 842
 MXV VXSC
 REFSMMAT
 MOONRATE
 STOVL WM
 RLS
 ABVAL SL3
 STCALL /LAND/
 TEMPR60

49FPS 2DEC .149352 B-6 # EXPECTED RDOT AT TIPOVER
VINJNOM 2DEC 16.7924 B-7 # 5509.5 FPS(APO=30NM WITH RDOT=19.5FPS)
RDOTDNOM 2DEC .059436 B-7 # 19.5 FPS
```

This code is written to file `src/P12.s`.

## B.67 P20-P25

```

1005 <src/P20-P25.s 1005>≡
Copyright: Public domain.
Filename: P20-P25.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 562-534
Mod history: 2009-05-10 RSB Adapted from the Colossus249/ file
of the same name, using Comanche055 page
images.
2009-05-20 RSB Corrections: P20S -> P20S, ST0 -> STQ,
GOTOP00H -> GOTOP00H, a duplicated EXTEND
was fixed, P23.10 -> R23.10,
S22BOX44 -> S22BOX44, S22SUBSCL -> 22SUBSCL,
S22DPP -> S22DSPP, changed some instructions in
P23.152.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 562
RENDEZVOUS NAVIGATION PROGRAM 20
#
PROGRAM DESCRIPTION
#
MOD NO -- 1
MOD BY -- N. BRODEUR

```

```

#
FUNCTIONAL DESCRIPTION
#
TO CONTROL THE CSM ATTITUDE AND OPTICS TO ACQUIRE THE LEM IN THE S+T
FIELD AND TO POINT THE CSM TRANSPONDER AT THE LEM. TO UPDATE EITHER THE
LEM OR CSM STATE VECTOR (AS SPECIFIED BY THE ASTRONAUT BY THE DSKY
ENTRY) ON THE BASIS OF OPTICAL TRACKING DATA (REQUESTED BY DSKY)
#
CALLING SEQUENCE --
#
ASTRONAUT REQUEST THROUGH DSKY V37E20E
#
SUBROUTINES CALLED
#
R02BOTH (IMU STATUS CHECK) BANKCALL
FLAGUP 2PHCHNG LOADTIME
R61CSM (PREFERRED TRACKING ATTITUDE) FLAGDOWN
R52 (AUTO OPTICS POSITIONING ROUT) SETINTG
R22 (REND TRACK DATA PROC ROUT) PRIOCHNG
ENDOFJOB INTEGRV GRP2PC
INTPRET MKRLEES FINDVAC
#
NORMAL EXIT MODES --
#
P20 MAY BE TERMINATED IN TWO WAYS -- ASTRONAUT SELECTION OF IDLING
PROGRAM (P00) BY KEYING V37E00E OR BY KEYING IN V56E
#
ALARM OR ABORT EXIT MODES --
#
NONE DIRECTLY FROM P20
#
OUTPUT
#
TRKMKCNT = NO OF RENDEZVOUS TRACKING MARKS TAKEN (COUNTER)
VHFCNT = NO OF VHF RANGING MARKS INCORPORATED (COUNTER)
#
FLAGS SET + RESET
#
RNDVZFLG,VEHUPFLG,UPDATFLG,TRACKFLG,TARG1FLG
HOLDFLAG,WBODY,WBODY1,WBODY2,DELCDEX,DELCDEUY,DELCDEUZ
STIKFLAG,PRFTRKAT,VINTFLAG,DIMOFFLAG,R60FLAG,R61CNTR
#
BANK 33
SETLOC P20S
BANK

```

July 12, 2016

1007

```
EBANK= ESTROKER
COUNT* $$/P20

PROG20 TC BANKCALL
 CADR R02BOTH # IMU STATUS CHECK
 # BLOCKING OF UPLINK IS DONE BY UPLINK PRG

 CAF ZERO
 TS TRKMKCNT # ZERO REND TRACKING MARK COUNTER
 TS VHFCNT # ZERO REND VHF RNG MRK COUNTER
 TC UPFLAG # SET PREF TRACK ATT FLAG
 ADRES PRFTRKAT # BIT 10 FLAG 5
 TC DOWNFLAG # LEM TO BE UPDATED. VEHUPFLG RESET.
 ADRES VEHUPFLG # BIT 8 FLAG 1

Page 563

 TC UPFLAG # SET TRACKFLAG
 ADRES TRACKFLG # BIT 5 FLAG 1
 TC UPFLAG # SET UPDATFLG
 ADRES UPDATFLG # BIT 7 FLAG 1
 TC UPFLAG # SET RNDVZFLG
 ADRES RNDVZFLG # BIT 7 FLAG 0
 TC 2PHSCHNG
 OCT 4
 OCT 05022
 OCT 26000
 TC INTPRET
 RTB

 LOADTIME
 STCALL MARKTIME
 SETINTG # SET INTEGRATION FLAGS
 BOFF SET
 RENDWFLG
 P20.1
 DIMOFLAG # SET TO INTEGRATE THE W MATRIX
 P20.1 BON CLEAR
 VEHUPFLG
 P20.2
 VINTFLAG # SET FOR LM INTEGRATION
 P20.2 CALL
 INTEGRV
 CALL
 GRP2PC # GROUP 2 PHASE CHANGE
 CALL
 SETINTG # SET INTEGRATION FLAGS
 BOFF CLEAR
 VEHUPFLG
 P20.3
```

|            |        |          |                                  |
|------------|--------|----------|----------------------------------|
|            |        | VINTFLAG | # SET FOR LM INTEGRATION         |
| P20.3      | CALL   |          |                                  |
|            |        | INTEGRV  |                                  |
|            | EXIT   |          |                                  |
|            | CAF    | PRI026   |                                  |
|            | TC     | FINDVAC  |                                  |
|            | EBANK= | MRKBUF2  |                                  |
|            | 2CADR  | R22      |                                  |
|            | TC     | 2PHSCHNG |                                  |
|            | OCT    | 00072    |                                  |
|            | OCT    | 00111    |                                  |
| PIKUP20    | CAF    | PRI014   | # ALLOW HIGHER PRIO THAN LAMBERT |
|            | TC     | PRI0CHNG |                                  |
|            | CAF    | BIT5     | # IS TRACK FLAG SET              |
|            | MASK   | STATE +1 |                                  |
|            | EXTEND |          |                                  |
|            | BZF    | ENDOFJOB | # NO                             |
| # Page 564 |        |          |                                  |
|            | CAF    | BIT13    |                                  |
|            | MASK   | STATE +3 | # IS REFSMFLG SET                |
|            | EXTEND |          |                                  |
|            | BZF    | ENDOFJOB |                                  |
|            | CAF    | ZERO     |                                  |
|            | TS     | R61CNTR  | # INITIALIZE R61 COUNTER         |
|            | TC     | BANKCALL |                                  |
|            | CADR   | R61CSM   |                                  |
|            | EBANK= | QMIN     |                                  |
|            | CAF    | EBANK5   |                                  |
|            | TS     | EBANK    |                                  |
|            | TC     | UPFLAG   | # SET TARGET FLAG TO LEM         |
|            | ADRES  | TARG1FLG | # BIT 10 FLAG 1                  |
| P20R52JB   | TC     | INTPRET  |                                  |
|            | CALL   |          |                                  |
|            |        | R52      | # SET UP AUTO OPTICS JOB         |
|            | EXIT   |          |                                  |
|            | TC     | BANKCALL |                                  |
|            | CADR   | MKRLEES  |                                  |
|            | CAF    | ONE      | # HOLD PRESENT ATTITUDE          |
|            | TS     | HOLDFLAG |                                  |
|            | TC     | ENDOFJOB |                                  |
| OCT203     | OCT    | 00203    |                                  |
| FIRST3     | EQUALS | FURST3   |                                  |



```

BANK 31
SETLOC P20S1
BANK

EBANK= LANDMARK
COUNT* $$/P22

PROG22 TC DOWNFLAG # RESET RNDVZFLG BIT 7 FLAG 0
 ADRES RNDVZFLG
 TC BANKCALL
 CADR R02BOTH # IMU STATUS CHECK
 TC INTERPRET # COMPUTE ANGLE BETWEEN Y AND VXR SM
 RTB
 LOADTIME
 STCALL TDEC1
 CSMCONIC # INTEGRATE TO PRESENT TIME
 VLOAD VXV # CROSS PRODUCT BETWEEN V AND R
 VATT
 RATT
 UNIT DOT
 REFSMMAT +6
 ABS
 SL1 ARCCOS
 STORE +MGA
 CLEAR EXIT
 RENDWFLG
 CAF V06N45B
 TC BANKCALL
 CADR GOFLASHR
 TC GOTOP00H # TERM P22
 TC PROG22A # PROC
 TC -5 # ENTER
 CAF THREE
 TC BLANKET # BLANK OUT R1 + R2
 TC ENDOFJOB
 CS PRI07 # RESULT=70777 SET OFFSET NO.=0
 MASK LANDMARK
 TS LANDMARK
 TC INTERPRET
 CLEAR
 P22MKFLG
 SET BOFF
 ERADFLAG
 CMOONFLG
 PROG22B # EARTH

```

```

Page 566

 SET LUNAFLAG # MOON
 DLOAD CALL # MPAC=V05N70,MPAC+1=NONZERO FOR N70
 V05N7022
 S22N7071
 GOTO
 CALLR52
 CLEAR SET # EARTH ORBIT
 LUNAFLAG
 KNOWNFLG
 CALL # GET LAT/LONG/ALT FROM ASTRO
 P22SUBRB
 CALLR52 EXIT
 TC 2PHSCHNG
 OCT 00004
 OCT 05022
 OCT 13000
 CAF FIVE
 TS MARKINDX # SET MARK INDEX=5 FOR R52
 TC UPFLAG
 ADRES TARG2FLG # SET FOR SIGHTING LMK
 TC DOWNFLAG
 ADRES TARG1FLG # CLEAR FOR NON-LEM
 TC INTPRET
 CALL
 R52
 DOV5N71 SLOAD # MPAC=V05N71,MPAC+1=0 FOR N71
 CALL
 V05N7122
 S22N7071
 PROG22C LXC,2 SLOAD*
 MARKSTAT
 QPRET,2
 STCALL 8NN
 S22.1 # ESTABLISH LANDMARK -- COMPUTE ORBITAL
 P22OVER EXIT
 TC PHASCHNG
 OCT 04022
 TC PROG22A # POINT A ON GSOP
 V06N45B VN 0645
 V05N7022 VN 00570
 V05N7122 VN 00571
 SETLOC P20S
 BANK
 SETLOC P20S1

```

July 12, 2016

1011

```
BANK

S22LSITE RTB # CONVERT RLS FROM MOON-FIXED TO BASIC REF
 LOADTIME
 STOVL 6D # 6-7D= TIME
 RLS
 STODL OD # 0-5D= LANDING SITE VECTOR
 HIDPHALF # MPAC= ANY NON-ZERO FOR MOON
 CALL

Page 567

 RP-TO-R # RLS IN BASIC REF B-27 IN MPAC
 VSR2 # LUNAFLAG AND ERADFLAG SET ABOVE
 STORE ALPHA # SCALE RLS B-29 FOR LAT-LONG
 RTB
 LOADTIME # SET PRESENT TIME IN MPAC FOR LAT-LONG
 CALL
 LAT-LONG
 GOTO
 S22TOFF # EXIT
OBTAINLL CALL # GET LAT/LONG/ALT FROM ASTRO
 P22SUBRB
 GOTO
 S22TOFF # EXIT
P22SUBRB STQ EXIT # GET LAT/LONG/ALT FROM ASTRO
 S22TOFF +1
 CAF V06N89B
 TC BANKCALL
 CADR GOFLASH
 TC GOTOP00H # TERMINATE
 TC +2 # PROCEED
 TC -5 # ENTER OR RECYCLE
 TC INTPRET
 DLOAD ABS # TEST LAT/LONG GREATER THAN 90
 LANDLAT
 BDSU BMN # 1/4 REV - LAT
 DP1/4TH
 N89ERRX
 DLOAD ABS
 LANDLONG
 BDSU BPL
 DP1/4TH
 +4

N89ERRX EXIT
 TC FALTON
 TC P22SUBRB +2 # LONG GR. THAN 90 REDISPLAY
```

|            |        |            |                                            |
|------------|--------|------------|--------------------------------------------|
|            | CALL   | LLASRDA    |                                            |
|            | GOTO   |            |                                            |
| S22N7071   | STORE  | S22TOFF +1 | # EXIT                                     |
|            | STQ    | 8KK        | # 8KK=V05N71,V05N70 8KK+1=0 N71, NOT 0 N70 |
|            |        | EXIT       |                                            |
| S22DSPPA   |        | S22TOFF    |                                            |
|            | CA     | 8KK        | # V05N70 OR V05N71                         |
|            | TC     | BANKCALL   |                                            |
|            | CADR   | GOFLASHR   |                                            |
|            | TC     | GOTOPOOH   | # V34E TERMINATE                           |
|            | TC     | +5         | # V33E PROCEED                             |
|            | TC     | -5         | # V32E RECYCLE                             |
|            | CAF    | FIVE       | # IMMEDIATE RETURN BLANK OUT R1,R3         |
|            | TC     | BLANKET    |                                            |
| # Page 568 |        |            |                                            |
|            | TC     | ENDOFJOB   |                                            |
|            | CA     | LANDMARK   |                                            |
|            | MASK   | PRI07      | # 07000                                    |
|            | TS     | CXOFF      | # 08000 = OFFSET INDICATOR                 |
|            | CS     | PRI05      | # 5 8-5                                    |
|            | AD     | CXOFF      |                                            |
|            | EXTEND |            |                                            |
|            | BZMF   | +2         | # OFF=0 THRU 5 OK                          |
|            | TC     | S22DSPP    | # OFF=6.7 ILLEGAL REDISPLAY                |
|            | TC     | DOWNFLAG   |                                            |
|            | ADRES  | KNOWNFLG   | # CLEAR KNOWNFLG                           |
|            | CA     | LANDMARK   | # MASK A FROM ABCDE                        |
|            | MASK   | 13,14,15   |                                            |
|            | EXTEND |            |                                            |
|            | BZMF   | S22DSPP    | # A=0,4,5,6,7 ILLEGAL REDISPLAY            |
|            | TS     | 22SUBSCL   | # TEMP                                     |
|            | CS     | BIT14      |                                            |
|            | AD     | 22SUBSCL   |                                            |
|            | CCS    | A          |                                            |
| S22DSPP    | TC     | FALTON     | # + A=3 ILLEGAL REDISPLAY                  |
|            | TC     | S22DSPPA   | # COMMON ERROR EXIT BACK TO DISPLAY        |
|            | TC     | +2         | # - A=1 KNOWN LMK CHECK DE                 |
|            | TC     | S22ABDE    | # -0 A=2 UNKNOWN LMK, DE MEANINGLESS       |
|            | TC     | UPFLAG     |                                            |
|            | ADRES  | KNOWNFLG   | # SET KNOWNFLG                             |
|            | CS     | HIGH9      | # RESULT= 00077                            |
|            | MASK   | LANDMARK   |                                            |
|            | TS     | 22SUBSCL   | # 000DE                                    |
|            | CS     | BIT1       |                                            |
|            | AD     | 22SUBSCL   |                                            |

July 12, 2016

1013

```
EXTEND
BZMF S22ABDE # DE=0,1 OK FOR BOTH N70,N71
CA 8KK +1 # =0 FOR N71, NOT =0 FOR N70
EXTEND
BZF S22DSPP # N71 REDISPLAY DE MUST= 0 OR 1
CA BIT5 # N70 TEST DE= 50-57 OCTAL FOR ADV. ORBIT
AD OCT50 # SUM=00070
MASK 22SUBSCL # 00D0
CS A
AD OCT50
EXTEND
BZF DE-GR-50 # D=5 OR DE=50-57, OK FOR N70
TC S22DSPP # DE NOT 50-57 ILLEGAL, REDISPLAY
S22ABDE TC INTPRET
BOFF SLOAD
KNOWNFLG # UNKNOWN LMK, DE MEANINGLESS
S22TOFF
22SUBSCL # =0 GET LLA FROM ASTRO, NOT=0 USE RLS
BHIZ GOTO
OBTAINLL # GET LAT/LONG/ALT FROM ASTRO N89
Page 569
DE-GR-50 S22LSITE # GET LAT/LONG/ALT FROM RLS
TC 2PHSCHNG
OCT 00004
OCT 05022
OCT 13000
CA FIVE
TS MARKINDX
TC UPFLAG
ADRES TARG2FLG
TC DOWNFLAG
ADRES TARG1FLG
TC INTPRET
CALL
ADVORB
GOTO
DOV5N71
```

```
Page 570
PROGRAM NAME: OPTICS CALIBRATION ROUTINE
MOD NO: 1
MOD BY: TOM KNATT
#
FUNCTIONAL DESCRIPTION: TO MEASURE THE EFFECT OF SOLAR RADIATION ON
THE SXT TRUNNON ANGLE AND STORE THE MEASURED TRUNNION BIAS FOR P23
#
```

```

CALLING SEQUENCE: CALL
R57
#
SUBROUTINES CALLED: DISPLAY ROUTINES
#
NORMAL EXIT MODES: VIA EGRESS
#
ALARMS: NONE
#
ABORT MODES: P23 ABORT IF MARKING SYSTEM OR EXTENDED VERB ACTIVE
#
INPUT: NONE REQUIRES, NORMALLY CALLED BY P23
#
OUTPUT: TRUNNION BIAS ANGLE: ANGLE DETERMINED WHEN SHAFT LINE OF SIGHT
(SLOS) AND LANDMARK LINE OF SIGHT (LLOS) ARE SUPERIMPOSED. THIS ANGLE
MAY NOT BE EXACTLY ZERO BECAUSE OF UNEVEN HEATING OF THE OPTICS, FOR
EXAMPLE.
#
ERASABLE INITIALIZATION REQUIRED: MRKBUF1, EXTVBACT
#
DEBRIS: RUPTREGS USED BY MARKRUPT AND ERASABLES USED BY DISPLAYS.

BANK 33
SETLOC P20S
BANK
COUNT* $$/R57
EBANK= MRKBUF1
R57 STQ EXIT
 EGRESS
CAF EBANK7
TS EBANK
CAF SIX # BIT2 = MARKING SYSTEM IN USE
MASK EXTVBACT # BIT3 = EXTENDED VERB IN PROGRESS
CCS A
TC P23ABRT # SET, THEREFORE ABORT
CAF BIT2 # NOT SET
ADS EXTVBACT # SET IT
R57A TC UPFLAG # SET V59FLAG (BIT 12 FLAG 5 TO INDICATE
 ADRES V59FLAG # CALIBRATION MARK
CAF V59NB
TC BANKCALL
CADR GOMARKFR
TC GOTOPOOH # TERMINATE
TC ENDR57
TC ENDR57
CAF SEVEN

```

# Page 571

|                               |        |            |                                       |
|-------------------------------|--------|------------|---------------------------------------|
|                               | TC     | BLANKET    | # BLANK OUT R1,R2,R3                  |
|                               | TC     | ENDOFJOB   |                                       |
| # STORE TRUNNION ANGLE (OCDU) |        |            |                                       |
| MARKDISP                      | CAF    | V06N87NB   |                                       |
|                               | TC     | BANKCALL   |                                       |
|                               | CADR   | GOMARKFR   |                                       |
|                               | TC     | GOTOP00H   | # TERMINATE                           |
|                               | TC     | R57B       | # PROCEED                             |
|                               | TC     | R57A       | # ENTER (RECYCLE)                     |
|                               | CAF    | FIVE       |                                       |
|                               | TC     | BLANKET    | # BLANK OUT R1,R3                     |
|                               | TC     | ENDOFJOB   |                                       |
| R57B                          | CA     | 19.77DEG   | # PUT FIXED INTO ERASABLE FOR MSU     |
|                               | TS     | L          | # INSTRUCTION COMING UP               |
|                               | CA     | MRKBUF1 +5 | # CONTAINS TRUNNION                   |
|                               | EXTEND |            |                                       |
|                               | MSU    | L          | # CONNECTS TRUNBIAS FROM 2'S TO 1'S   |
|                               | TS     | TRUNBIAS   |                                       |
| ENDR57                        | TC     | DOWNFLAG   | # RESET V59FLAG                       |
|                               | ADRES  | V59FLAG    | # BIT 12 FLAG 5                       |
|                               | CAF    | EBANK5     |                                       |
|                               | TS     | EBANK      |                                       |
|                               | CAF    | PRI014     |                                       |
|                               | TC     | NOVAC      | # THIS JOB CLEARS BIT IN              |
|                               | EBANK= | MARKSTAT   |                                       |
|                               | 2CADR  | ENDMARK    | # MARKING IN R57 SO R53 CAN TAKE OVER |
|                               | TC     | INTPRET    |                                       |
|                               | GOTO   |            |                                       |
|                               |        | EGRESS     |                                       |
| P23ABRT                       | TC     | BAILOUT    |                                       |
|                               | OCT    | 01211      |                                       |
| V06N87NB                      | VN     | 0687       |                                       |
| V59NB                         | VN     | 5900       |                                       |
| V51NB                         | VN     | 5100       |                                       |
| 19.77DEG                      | OCT    | 61740      |                                       |

# Page 572

# PROGRAM DESCRIPTION

# MOD NO: 1

# MOD BY: N. BRODEUR

#

# FUNCTIONAL DESCRIPTION

#

# TO PERFORM SIGHTING MARKS IN CONJUNCTION WITH THE RENDEZVOUS NAVIG-

```

ATION PROGRAM. CALLED BY ASTRONAUT VIA EXTENDED VERB.
#
CALLING SEQUENCE:
#
R21 VIA V57
R23 VIA V 54
#
SUBROUTINES CALLED:
#
FLAGUP FLAGDOWN BANKCALL
ENDOFJOB GOMARK2 GOMARKF
INTPRET GENTRAN KLEENEX
ENDMARK
#
NORMAL EXIT MODES:
#
MARKRUPT USED BY SXTMARK HAS BEEN MODIFIED TO STORE MARK IN MRKBUF2
FOR USE BY R22. WHEN ASTRONAUT IS FINISHED TAKING MARKS, HE HITS AN
PROCEED, R21 IS TERMINATED THUS CAUSING THE FINAL MARK TO BE TRANSFRD
TO MRKBUF2 FOR PROCESSING BY R22
#
ALARM OR ABORT EXIT MODES:
#
NONE
#
OUTPUT:
#
7 REGISTER MRKBUF2 CONTAINING TIME2,TIME1,CDUY,OPTICS X,CDUZ, OPTICS Y,
CDUX.
#
ERASABLE INTIALIZATION REQUIRED
#
FLAGS SET AND RESET
R21MARK (COMMUNICATION TO MARKRUPT TO STORE MARKS IN MRKBUF1 +2)
R23FLG INDICATES COAS MARKING
#
DEBRIS
#
EBANK= MRKBUF1
SETLOC RENDEZ
BANK
#
COUNT* $$/R21
#
R21CSM TC UPFLAG # SET R21MARK
 ADRES R21MARK # BIT 14 FLAG 2

```



July 12, 2016

1017

```
R23CSM CA NEGONE
 TS MRKBUF1
 TS MRKBUF2
 CA FLAGWRD1
 MASK R23BIT # TEST R23FLG
 EXTEND
 BZF R21C1 # NOT SET REGULAR R21 MARKING
 CAF V0694 # R23 BACKUP MARKING
 TC BANKCALL # DISPLAY SHAFT + TRUNNION
 CADR GOMARKF
 TC R21END # TERM
 TC +2 # PROC

Page 573
 TC -5 # ENTER
R23CSM1 CAF V53 # PERFORM ALT LOST SIGHT MARK
 TC BANKCALL
 CADR GOMARK2
 TC R21END # V34: TERMINATE R23
 TCF R21CSMA # PROCEED: END BACK UP MARKING (R23)
 CAF SIX # TRANSFER MRKBUF1 TO MRKBUF2
 TC GENTRAN
 ADRES MRKBUF1
 ADRES MRKBUF2
 EXTEND
 DCA TIME2
 DXCH MRKBUF1 # READ TIME
 CA CDUY # READ CDU ANGLES
 TS MRKBUF1 +2
 CA CDUZ
 TS MRKBUF1 +4
 CA CDUX
 TS MRKBUF1 +6
 RELINT
 TC R23CSM1
R21C1 CAF V51NB
 TC BANKCALL
 CADR GOMARK2
 TC R21END # V34: TERMINATE R21
 TCF R21CSMA # PROCEED: END R21
 TCF R21C1 # RECYCLE

R21CSMA CA MRKBUF1 # IF -1 NO MARK
 AD ONE
 EXTEND
 BZF R21END # ZERO = NO MARK
 CAF SIX # MARK THEREFORE TRANSFER IT TO MRKBUF2
```

```

R21CSM1 TC GENTRAN # TRANSFER MRKBUF1 TO MRKBUF2
 ADRES MRKBUF1
 ADRES MRKBUF2
 RELINT
R21END TC BANKCALL
 CADR KLEENEX
 TC DOWNFLAG # RESET R21MARK
 ADRES R21MARK # BIT 14 FLAG 2
 TC ENDMARK # END MARKING AND ENDJOB
V53 VN 5300
V0694 VN 0694

```

```

Page 574
PREFERRED TRACKING ATTITUDE ROUTINE R61CSM
#
PROGRAM DESCRIPTION
MOD NO: 2
MOD BY: N. BRODEUR
#
FUNCTIONAL DESCRIPTION:
#
TO COMPUTE THE PREFERRED TRACKING ATTITUDE OF THE CSM TO ENABLE OPTICS
TRACKING OF THE LM AND TO PERFORM THE MANEUVER TO THE PREFERRED
OR X-AXIS TRACKING ATTITUDE.
#
CALLING SEQUENCE:
#
TC BANKCALL
CADR R61CSM
#
SUBROUTINE CALLED
#
MAKECADR BANKCALL
INTPRET FLAGUP FLAGDOWN
BANKJUMP CRS61.1 R60CSM
PHASCHNG
#
NORMAL EXIT MODES:
#
NORMAL RETURN TO CALLER + 1
#
OUTPUT:
#
SEE OUTPUT FOR CRS61.1 & ATTITUDE MANEUVER ROUTINE (R60CSM)
#
ERASABLE INITIALIZATION REQUIRED

```

```

#
GENRET USED TO SAVE Q FOR RETURN
R61CNTR MUST BE PRESET TO ZERO
#
FLAGS SET + RESET
#
3-AXIS FLAG
#
DEBRIS
#
SEE SUBROUTINES

EBANK= GENRET
COUNT* $$/R61 # ROUTINES -- NAVIGATION -- PREF. TR. 9TT=

R61CSM CAF EBANK6 # SWITCH TO EBANK 6
 XCH EBANK
 TS SAVBNK # SAVE EBANK
 TC MAKECADR
 TS GENRET
 CCS R61CNTR # TEST R61DNTR
 TC DECRM61 # NOT READY TO DO R61
 TC +2 # DO R61
 TC DECRM61 +1
 TC INTPRET
 CALL
 CRS61.1 # LOS DETERMINATION + VEH ATTITUDE
 EXIT
 INDEX MPAC
 TC +1
 TC R61END # SUBROUTINE DRIVING DAP (EXIT R61)
 # OR AUTO MODE NOT SET (EXIT R61)
 # OR SKIKFLAG SET (EXIT R61)
R61C1 TC DOWNFLAG # RESET 3-AXIS FLAG
 ADRES 3AXISFLG # BIT 6 FLAG 5

Page 575
 CS ONE # SET R61CNTR NEG. TO INDICATE KALCMANU
 TS R61CNTR

 TC UPFLAG # SET FLAG FOR PRIORITY DISPLAYS FOR R60
 ADRES PDSPFLAG # BIT 12 FLAG 4
 TC BANKCALL
 CADR R60CSM
 TC DOWNFLAG # RESET FLAG FOR PRIORITY DISPLAYS IN R60
 ADRES PDSPFLAG # BIT 12 FLAG 4
 TC PHASCHNG

```

```

OCT 00111
CAF ZERO
TC DECRM61
R61END CAF THREE

DECRM61 TS R61CNTR

CAE GENRET
LXCH A # RETURN IS IN L
CA SAVBNK # RESTORE EBANK
XCH EBANK
LXCH A # RETURN IS NOW BACK IN A
TC BANKJUMP # EXIT R61
BANK 13
SETLOC P20S2
BANK

EBANK= MRKBUF1

```

```

Page 576

```

```

BVECTOR PERFORMS COMPUTATIONS FOR

```

```

#

```

```

DELTAQ, THE MEASURED DEVIATION BASED ON THE DIFFERENCE BETWEEN THE CSM-LEM
STATE VECTOR ESTIMATES AND THE ACTUAL TRACKING MEASUREMENT.

```

```

#

```

```

US, THE MODIFIED FICTITIOUS STAR DIRECTION VECTOR
GEOMETRY VECTORY B ASSOCIATED WITH EACH TRACKING MEASUREMENT.

```

```

#

```

```

INPUT

```

```

#

```

```

UM, 1/2 UNIT VECTOR ALONG THE CSM-LM LINE-OF-SIGHT (BASIC REF. SYSTEM)

```

```

#

```

```

USTAR, FICTITIOUS STAR DIRECTION (1/2 UNIT VECTOR)

```

```

#

```

```

RCLP, RELATIVE CSM TO LM POSITION VECTOR

```

```

#

```

```

OUTPUT

```

```

#

```

```

USTAR, MODIFIED FICTITIOUS STAR DIRECTION (1/2 UNIT VECTOR)

```

```

#

```

```

BVECTOR = 9 DIMENTIONAL BVECTOR (1/2 UNIT VEC.)

```

```

#

```

```

DELTAQ = MEASURED DEVIATION

```

```

#

```

```

CALLING SEQUENCE

```

```

#

```

```

L CALL BVECTORS

```

July 12, 2016

1021

```
#
NORMAL EXIT
#
L+1 OF CALLING SEQUENCE

COUNT 23/20SUB

BVECTORS STQ
 EGRESS
 VLOAD UNIT
 RCLP # RELATIVE POSITION VECTOR
 STODL 26D # RCLP UNIT VEC
 36D # RCLP ABS VALUE
 STOVL TEMPOR1 # MOVE TO SAFE LOCATION
 USTAR
 VXV UNIT
 26D # USTAR = UNIT(US X UCL)
 STCALL BVECTOR
 GRP2PC # PHASE CHANGE
 VLOAD
 BVECTOR
 STORE USTAR
 DOT SL1
 UM # USTAR DOT UM
 ACOS DSU
 DP1/4TH
 NORM DMP
 X1
 PI/4.0
 DMP SRR*
 TEMPOR1 # RCLP ABS VALUE
 0 -3,1 # ADJUST SCALING
 STOVL DELTAQ
 ZEROVECS
 STORE BVECTOR +6
 STORE BVECTOR +12D
 GOTO

Page 577

 EGRESS
PI/4.0 2DEC .785398164
```

```
Page 578
GETUM: DETERMINES THE LINE OF SIGHT UNIT VECTOR UM IN THE BASIC REFERENCE
COORDINATE SYSTEM FROM THE OPTICS SHAFT AND TRUNNION ANGLES AND THE IMU
GIMBAL ANGLES.
#
```

```

INPUT
#
MARKDATA, BASE ADDRESS OF MARK DATA
REFSMMAT, ROTATION MATRIX FROM STABLE MEMBER TO BASIC REF. COORD. SYSTEM
#
SUROUTINES CALLED:
#
SXTNB -- SEXT. ANGULAR READINGS TO NAV. BASE COOR.
NBSM -- TRANSFORM FROM NAV. BASE TO STABLE MEMBER
#
OUTPUT
#
MPAC = LINE OF SIGHT 1/2 UNIT VECTOR IN BASIC REFERENCE SYSTEM
#
CALLING SEQUENCE
#
L CALL GETUM
#
NORMAL EXIT
#
L+1 OF CALLING SEQUENCE

GETUM STQ SETPD
 EGRESS
 0
 LXC,1 VLOAD*
 MARKDATA # CONTAINS ADDRESS OF MARK DATA
 1,1
 STODL* MARKDOWN +1 # TRANSFER DATA FROM WORKING STORAGE
 0,1 # TO MARKDOWN ARRAY FOR DOWNLINK
 STORE MARKDOWN
 AXT,2
 2
 XSU,2 SXA,2
 X1 # X1 = MARKDATA
 S1 # S1 = MARKDATA(ADR) +2
 CALL
 SXTNB # SEXT. ANGULAR READINGS TO NAV. BASE COOR.
 CALL
 NBSM # TRANSFORM FROM NAV. BASE TO STABLE MEM.
 VXM VSL1
 REFSMMAT
 GOTO # MPAC = (UM) LINE OF SIGHT VECTOR
 EGRESS # EXIT

```

```

RENDEZVOUS TRACKING DATA PROCESSING ROUTINE (R22)
#
PURPOSE (1) TO PROCESS RENDEZVOUS SIGHTING MARK DATA TO UPDATE THE STATE VECTOR OF EITHER
DEFINED BY THE RENDEZVOUS NAVIGATION PROGRAM (P20).
#
ASSUMPTIONS (1) THIS ROUTINE IS MANUALLY SELECTED BY THE ASTRONAUT BY V55E WHENEVER RENDEZVOUS
ARE DESIRED. ITS SELECTION, HOWEVER, IS LIMITED TO PERIODS WHEN THE CMC IS HOLDING
DATA DISPLAY. THIS ROUTINE RETURNS TO THE ORIGINAL PROGRAM AT THE INTERRUPTED
#
BANK 34
SETLOC P20S3
BANK
#
COUNT 34/R22
#
R22 CAF PRI026
 TS PHSPRDT2
 TC PRIOCHNG
 CA NEG3
 TS MRKBUF2
 TC INTPRET
 RTB
 LOADTIME
 STORE VHFTIME # PRESENT TIME
REND1 CALL
 GRP2PC
 CALL
 WAITONE
REND1A EXIT
 CA MRKBUF2
 EXTEND
 BZF REND2
 EXTEND
 BZMF REND3A
REND2 CAF SIX
 TC GENTRAN
 ADRES MRKBUF2
 ADRES MARKTIME # MARKTIME MUST BE CONTIGUOUS WITH VTEMP
 CAF NEG3 # NEG VALUE TO INDICATE VALUES USED
 TS MRKBUF2
 RELINT
 TC INTPRET
 CLEAR CALL
 SOURCFLG # 0 = OPTICS DATA
 GRP2PC
 SSP GOTO

```

```

REND3A MARKDATA
REND3 ECADR VTEMP -2
Page 580 REND4
 TC INTPRET
 CALL

 GRP2PC
 CALL
 WAITONE
 BOFF
 VHFRFLAG
 REND1A
 RTB
 LOADTIME # PRESENT TIME
 DSU DSU
 60SECDP # 1 MIN
 VHFTIME # LAST READING OF RADAR
 BMN CALL
 REND1A
 RANGERD # READ RADAR RANGE
 DLOAD
 MARKTIME
 STORE VHFTIME # FOR DOWNLINK
REND4 CALL
 SETINTG # SET INTEGRV FLAGS
 BON CALL
 VEHUPFLG
 CSMUPP # BRANCH IF CSM UPDATE
 INTEGRV
 CALL
 GRP2PC # PHASE CHANGE
 CALL
 SETINTG # SET INTEGRV FLAGS
 CLEAR
 VINTFLAG # SET INTEGRATION VEHICLE TO LM
REND5 BOFF SET
 RENDWFLG
 REND5A # DO NOT INTEGRATE W IF FLAG = 0
 DIMOFLAG
REND5A CALL
 INTEGRV
 CALL
 SHIFTNDX # SET EARTH MOON SCALING INDEX
 CALL
 CMPOS # SET CSM POSITION
 SET CALL

```



July 12, 2016

1025

```
Page 581
CLEAR INCORFLG # SET FOR 1ST PASS
LMPOS # SET LM POSITION
BON
ORBWFLAG # CLEAR FOR ORBITAL AND CISLUNAR
RENDWFLG
REND6
DLOAD
WRENDPOS
STCALL 0 # 0 = WRENDPOS 1 = WRENDVEL
INITIALW # INTIIALIZE W MATRIX

REND6 DLOAD
ZEROVECS
STORE VHFCNT # ZERO OUT VHFCNT AND TRKMKCNT
SET
RENDWFLG
VLOAD VSU
LEMPOS
CSMPOS
STORE RCLP # LM - CSM
UNIT BON
SOURCFLG
REND14 # BRANCH IF DATA IS RADAR
STORE UCL
BOFF CALL
INCORFLG
REND9
GETUM # CALCULATE UM LINE OF SIGHT
STOVL UM
UCL
VXV BOV
UM # UCL X UM
REND8
UNIT BOV
REND3 # BRANCH IF OVERFLOW IGNORE MARK
STORE USTAR
REND9 CALL
BVECTORS
BON VLOAD
VEHUPFLG
REND9A
BVECTOR
VCOMP
STORE BVECTOR
REND9A CALL
GRP2PC
```

```

BON
R23FLG
REND15 # BRANCH IF BACKUP OPTICS (R23 WORKING)
DLOAD DAD
SXTVAR
IMUVAR
REND10 STOVL VARIANCE # TEMP STORAGE FOR VARIANCE CALC.
RCLP
ABVAL NORM
X1
DSQ DMP
VARIANCE
XAD,1 CALL
X1
SHIFTNDX # GET EARTH MOON SCALING INDEX
Page 582
XAD,1 XAD,1
X2
X2
SR* TLOAD
0 -2,1 # ADJUST SCALING TO B-40
MPAC
STORE VARIANCE
SLOAD SR
INTVAR # INTEGRATION VARIANCE SCALED B-15
25D # SCALE IT B-40
TAD RTB
VARIANCE
TPMODE
STORE VARIANCE
BOFF TAD
SOURCFLG # BRANCH IF NOT VHF RADAR
REND10A
RVARMIN # VHF RADAR MIN. VARIANCE
BPL TLOAD
REND10A
RVARMIN
ABS # MIN. VALUE WAS STORED AS NEG.
STORE VARIANCE # STORE MIN. VALUE
REND10A CLEAR CALL
DMENFLG # CLEAR FOR 6 X 6 W MATRIX
INCORP1 # CALCULATE UPDATE
CALL
GRP2PC
BOFF CALL
INCORFLG

```

July 12, 2016

1027

```

 REND12
 SHIFTNDX # GET EARTH MOON SCALING INDEX
VLOAD ABVAL
 DELTAX +6
SR*
 0,2
STOVL N49DISP +2
 DELTAX
ABVAL SR*
 0,2
STORE N49DISP
SLOAD
 RMAX
SR DSU
 10D
 N49DISP
BMN SLOAD
 RENDISP # BRANCH IF POS UP. GREATER THAN MAX.
 VMAX
DSU BMN
 N49DISP +2
 RENDISP # BRANCH IF VEL. UPDATE GREATER THAN MAX.
REND12 CALL
 INCORP2 # INCORPORATE UPDATE VALUES INTO STATE VEC
BON BOFF
 SOURCFLG
 REND16 # BRANCH IF DATA IS RADAR
 INCORFLG
 REND17
CALL
 SHIFTNDX # GET EARTH MOON SCALING INDEX
BON CALL
 VEHUPFLG
 REND18 # BRANCH IF CSM UPDATE
 LMPOS # GET LM POSITION
REND13 CALL
 GRP2PC # PHASE CHANGE
VLOAD VSU
 LEMPOS
 CSMPOS
STORE RCLP # LM - CSM
CLRGO
 INCORFLG
 REND7 # BRANCH FOR 2ND PASS THIS OPTICS MARK
CSMUPP CLEAR CALL
```

|            |       |              |                                 |
|------------|-------|--------------|---------------------------------|
|            |       | VINTFLAG     | # SET INTEGRATION VEHICLE EQ LM |
|            |       | INTEGRV      |                                 |
|            | CALL  | GRP2PC       | # PHASE CHANGE                  |
|            | CALL  | SETINTG      | # SET FLAGS FOR INTEGRATION     |
|            | GOTO  | REND5        |                                 |
| REND14     | STOVL | BVECTOR      | # VHF RADAR BVECTOR             |
|            |       | ZEROVECS     |                                 |
|            | STORE | BVECTOR +6   |                                 |
|            | STOVL | BVECTOR +12D |                                 |
|            |       | RCLP         |                                 |
|            | UNIT  | DLOAD        |                                 |
|            |       | VHFRANGE     | # VHFRANGE SCALED B-27          |
|            | BON   | SR2          |                                 |
|            |       | MOONTHIS     |                                 |
|            |       | +1           |                                 |
|            | DSU   | SET          |                                 |
|            |       | 36D          | # ABVAL (RCLP)                  |
|            |       | INCORFLG     |                                 |
|            | STORE | DELTAQ       |                                 |
|            | BOFF  | VLOAD        |                                 |
|            |       | VEHUPFLG     |                                 |
|            |       | REND14A      |                                 |
| # Page 584 |       | BVECTOR      |                                 |
|            | VCOMP |              |                                 |
|            | STORE | BVECTOR      |                                 |
| REND14A    | CALL  |              |                                 |
|            |       | GRP2PC       |                                 |
|            | DLOAD | GOTO         |                                 |
|            |       | RVAR         |                                 |
|            |       | REND10       |                                 |
| REND15     | SLOAD | DAD          | # GET ALT LOS VARIANCE          |
|            |       | ALTVAR       | # BACKUP OPTICS                 |
|            |       | IMUVAR       | # IMU VARIANCE                  |
|            | GOTO  |              |                                 |
|            |       | REND10       |                                 |
| REND16     | LXA,1 | INCR,1       |                                 |
|            |       | VHFCNT       | # VHF RADAR UPDATE COUNT        |
|            | DEC   | 1            |                                 |
|            | SXA,1 | GOTO         |                                 |
|            |       | VHFCNT       | # UPDATE COUNT                  |
|            |       | REND1        |                                 |
| REND17     | LXA,1 | INCR,1       |                                 |

July 12, 2016

1029

```

 TRKMKCNT # OPTICS MARK COUNT
 1
DEC
SXA,1 GOTO
 TRKMKCNT # UPDATE COUNT
 REND3
REND18 CALL
 CMPOS # GET CSM POSITION
 GOTO
 REND13
CMPOS VLOAD VSR*
 DELTACSM
 7,2
 VAD
 RCVCSM
 STORE CSMPOS # CSM POSITION SCALED B-27 OR B-29
 RVQ
LMPOS VLOAD VSR*
 DELTALEM
 7,2
 VAD
 RCVLEM
 STORE LEMPOS # LM POSITION SCALED B-27 OR B-29
 RVQ
RENDISP EXIT
 CA FLAGWRD9
 MASK SOURCBIT
 EXTEND
 BZF +3
 CA BIT2
 TC +2
Page 585
 CA BIT1
 TS N49DISP +4
 CAF ZERO # SET TEMPOR1 > ZERO TO INDICATE
 TS TEMPOR1 # V06 N49 DISPLAY HASN'T BEEN ANSWERED
 TC PHASCHNG
 OCT 04022
 CAF PRI027 # SET UP DISPLAY JOB WITH HIGHER PRIORITY
 TC NOVAC
 EBANK= MRKBUF1 # THAN PRESENT JOB
 2CADR RENDISP2
RENDISP7 TC INTPRET
 STORE MPAC
 SLOAD BZE
 TEMPOR1
```

|            |        |               |                                          |
|------------|--------|---------------|------------------------------------------|
|            |        | RENDISP7 +1   | # DISPLAY HAS NOT BEEN ANSWERED YET      |
|            | BMN    | GOTO          |                                          |
|            |        | REND12        | # NEG INDICATES PROCEED                  |
|            |        | RENDISP3      | # POS INDICATES RECYCLE                  |
| RENDISP2   | CAF    | V06N49        |                                          |
|            | TC     | BANKCALL      |                                          |
|            | CADR   | PRIODSP       |                                          |
|            | TC     | GOTOV56       | # TERM EXIT P20 VIA V56                  |
|            | CS     | ONE           | # NEG INDICATES PROCEED RENDISP7 JOB     |
|            | TS     | TEMPOR1       | # POS INDICATES RECYCLE RENDISP7 JOB     |
|            | TC     | ENDOFJOB      | # GO COMPLETE ABOVE JOB                  |
| RENDISP3   | BON    |               |                                          |
|            |        | SOURCFLG      |                                          |
|            |        | REND1         | # DATA WAS RADAR GO LOOK FOR OPTICS NEXT |
|            | EXIT   |               |                                          |
|            | EBANK= | MRKBUF1       |                                          |
|            | INHINT |               |                                          |
|            | CAF    | BUFBANK       |                                          |
|            | TS     | BBANK         |                                          |
|            | CA     | NEGONE        |                                          |
|            | TS     | MRKBUF1       | # ERASE MARK ONE BUFFER                  |
|            | TS     | MRKBUF2       | # ERASE MARK TWO BUFFER                  |
|            | RELINT |               |                                          |
| RENDISP4   | TC     | INTPRET       |                                          |
|            | GOTO   |               |                                          |
|            |        | REND3         |                                          |
| SXTVAR     | 2DEC   | 0.04 E-6 B+16 | # SXT ERROR VARIANCE = .04 (MR)SQ        |
| IMUVAR     | 2DEC   | 0.04 E-6 B+16 | # IMU ERROR VARIANCE = .04 (MR)SQ        |
| V06N49     | VN     | 0649          |                                          |
|            | EBANK= | MRKBUF1       |                                          |
| BUFBANK    | BBCON  | RENDISP3      |                                          |
|            | BANK   | 31            |                                          |
|            | SETLOC | R22S1         |                                          |
| # Page 586 |        |               |                                          |
|            | BANK   |               |                                          |
| SETINTG    | STQ    | CALL          |                                          |
|            |        | EGRESS        |                                          |
|            |        | INTSTALL      | # RESERVE INTEGRATION                    |
|            | DLOAD  | SET           |                                          |
|            |        | MARKTIME      |                                          |
|            |        | STATEFLG      |                                          |
|            | STORE  | TDEC1         | # MARKTIME                               |
|            | CLEAR  | CLEAR         |                                          |
|            |        | INTYPFLG      | # PRECISION INTEGRATION                  |

July 12, 2016

1031

```

 DIMOFLAG
 SET CLRG0
 VINTFLAG # SET VEHICLE EQ. CSM
 D6OR9FLG # SET W MATRIX DIM. EQ 6
 EGRESS # EXIT
CNTCHK STQ
 POINTEX
CONTCHK BOFF
 REFSMFLG # BRANCH TO END OF JOB IF REFSMMAT NO GOOD
 ENDPLAC
 SLOAD BMN
 R61CNTR
 WAITONE1
 BON BOFF # IS TRACK FLAG SET
 UPDATFLG
 POINTEX
 TRACKFLG
 ENDPLAC
 EXIT
REDOR22 TC PHASCHNG
 OCT 00132
 CAF PRI026
 TC PRIOCHNG
 TC WAITONE +3
WAITONE STQ
 POINTEX
WAITONE1 EXIT
 CAF 4SECS # WAIT 4 SECS.
 TC BANKCALL
 CADR DELAYJOB
 TC INTPRET
 GOTO
 CONTCHK # CHECK AGAIN NOW
RANGERD EXIT
 INHINT
 CS OCT17
 EXTEND
 WAND CHAN13 # ZERO OUT BITS 1-4 OF CHANNEL 13
 CAF OCT11
 EXTEND
Page 587
 WOR CHAN13 # GENERAGE SHIFT PULSES TO RADR, SET R. BIT
 RELINT
 EXTEND
 DCA TIME2
 DXCH MARKTIME # READ PRESENT TIME
```

|          |        |               |                                            |
|----------|--------|---------------|--------------------------------------------|
|          | TC     | DOWNFLAG      |                                            |
|          | ADRES  | SKIPVHF       |                                            |
|          | TC     | BANKCALL      |                                            |
|          | CADR   | RADSTALL      | # WAIT FOR RANGE COMPLETE                  |
|          | TC     | LIGHTON       | # BAD DATA GOOD BIT                        |
|          | TC     | TRFAILOF      | # TURN TRACKER LIGHT OFF                   |
| RANGERD1 | CCS    | RM            | # 15 BIT UNSIGNED RANGE                    |
|          | TC     | RANGERD4      | # GR + 0                                   |
|          | TC     | LIGHTON +4    | # = + 0                                    |
|          | TC     | RANGERD3      | # L - 0                                    |
|          | TC     | RANGERD3      | # = - 0                                    |
| RANGERD4 | TC     | INTPRET       |                                            |
|          | SLOAD  | DMP           |                                            |
|          |        | RM            |                                            |
|          |        | CONVRNGE      | # CONVERT RANGE TO METERS B-27             |
| RANGERD2 | STORE  | VHFRANGE      |                                            |
|          | SET    | RVQ           |                                            |
|          |        | SOURCFLG      | # SOURCE OF DATA TO VHF RADAR              |
| RANGERD3 | CA     | RM            |                                            |
|          | MASK   | POSMAX        |                                            |
|          | TS     | MPAC          | # MASK OUT NEG. SIGN BIT                   |
|          | TC     | INTPRET       |                                            |
|          | DMP    |               |                                            |
|          |        | CONVRNGE      | # CONVERT FROM NM TO METERS AND SCALE B-27 |
|          | DAD    | GOTO          |                                            |
|          |        | RANGEB14      | # VALUE IN METERS OF SIGN BIT SCALED B-27  |
|          |        | RANGERD2      |                                            |
| LIGHTON  | CA     | VHFRANGE      |                                            |
|          | EXTEND |               |                                            |
|          | BZF    | +2            |                                            |
|          | TC     | TRFAILON      |                                            |
|          | TC     | INTPRET       |                                            |
|          | DLOAD  |               |                                            |
|          |        | MARKTIME      |                                            |
|          | STORE  | VHFTIME       |                                            |
|          | GOTO   |               |                                            |
|          |        | REND1         |                                            |
| RANGEB14 | 2DEC   | 303431.7 B-27 | # 16384 X 18.52 SCALED B-27                |
| OCT17    | OCT    | 00017         |                                            |
| OC40200  | OCT    | 40200         |                                            |
| CONVRNGE | 2DEC   | 18.52 B-13    | # VHF INPUT RANGE CONV. FROM .01 NM TO M   |
| VHFREAD  | EXTEND |               |                                            |
|          | ROR    | SUPERBNK      | # MUST SAVE SBANK BECAUSE OF RUPT          |



July 12, 2016

1033

|          |        |          |                                        |
|----------|--------|----------|----------------------------------------|
|          | TS     | BANKRUPT | # EXITS VIA TASKOVER BADEND OR GOODEND |
|          | CS     | ZERO     |                                        |
|          | TS     | RUPTAGN  |                                        |
|          | EXTEND |          |                                        |
|          | QXCH   | QRUPT    |                                        |
|          | CS     | STATE +2 |                                        |
|          | MASK   | SKIPVBIT | # SKIPVHF FLAG                         |
|          | EXTEND |          |                                        |
|          | BZF    | TASKOVER | # BRANCH IF VHF RESTART BIT SET        |
|          | CAF    | UPDATBIT |                                        |
|          | MASK   | STATE +1 | # UPDATEFLG                            |
|          | EXTEND |          |                                        |
|          | BZF    | BYPASS1  |                                        |
|          | CS     | STATE +4 |                                        |
|          | MASK   | PDSPFBIT | # PDSPFLAG                             |
|          | EXTEND |          |                                        |
|          | BZF    | BYPASS1  |                                        |
|          | CA     | RNRAD    |                                        |
|          | TS     | RM       | # SAVE RANGE                           |
|          | CAF    | BIT2     |                                        |
|          | EXTEND |          |                                        |
|          | RAND   | CHAN33   | # READ DATA GOOD BIT                   |
|          | EXTEND |          |                                        |
|          | BZF    | VHFGOOD  | # BRANCH IF DATA GOOD BIT EQUALS GOOD  |
| BYPASS   | TS     | VHFRANGE | # STORE NON ZERO VALUE                 |
|          | CAF    | TWO      |                                        |
|          | TC     | POSTJUMP |                                        |
|          | CADR   | BADEND   |                                        |
| BYPASS1  | CAF    | ZERO     |                                        |
|          | TC     | BYPASS   |                                        |
| VHFGOOD  | CAF    | TWO      |                                        |
|          | TC     | POSTJUMP |                                        |
|          | CADR   | GOODEND  |                                        |
| SHIFTNDX | AXT,2  | BON      |                                        |
|          |        | 0        |                                        |
|          |        | VEHUPFLG |                                        |
|          |        | SHIFTA   | # VEHICLE IS CSM                       |
|          | BON    | RVQ      |                                        |
|          |        | LMOONFLG |                                        |
|          |        | +1       |                                        |
|          | INCR,2 | RVQ      |                                        |
|          | DEC    | -2       |                                        |
| SHIFTA   | BON    | RVQ      |                                        |
|          |        | CMOONFLG |                                        |
|          |        | +1       | # MOON ORB.                            |
|          | INCR,2 | RVQ      |                                        |

```

INITIALW DEC -2
 AXT,1 SSP
 36D
 S1
Page 589
 6
 VLOAD
 ZEROVECS
INITA STORE W +36D,1 # CLEAR 0 - 35
 TIX,1 AXT,1
 INITA
 36D
INITB STORE W +90D,1 # CLEAR 54 - 89
 TIX,1 SLOAD
 INITB
 0 # POSITION VALUE
 STORE W # INITIALIZE DIAGONAL W POSITION
 STORE W +8D
 STORE W +16D
 SLOAD
 1 # VELOCITY VALUE
 STORE W +72D # INITIALIZE DIAGONAL W VELOCITY
 STORE W +80D
 STORE W +88D
 RVQ
Page 590
CRS61.1 R/10/68
#
TO COMPUTE THE PREFERRED TRACKING ATTITUDE OF THE CSM WHICH ENABLES
OPTICS TRACKING OF THE LEM AND LM TRACKING OF THE CSM RADAR TRANSPONDER
AND TO COMPUTE THE X-AXIS TRACKING ATTITUDE OF THE CSM WHICH ENABLES
COAS TRACKING OF THE LM.
#
TO PERFORM THE MANEUVER TO THE SELECTED TRACKING ATTITUDE IF THE
MANEUVER IS LESS THAN 10 DEGREES BUT TO CALL R60 IF THE MANEUVER IS
GREATER THAN 10 DEGREES BUT TO CALL R60 IF THE MANEUVER IS
GREATER THAN 10 DEGREES OR IF THE R60 FLAG IS SET.
#
(1) EXTRAPOLATE LM AND CSM STATE VECTORS TO PRESENT TIME USING
CONIC EQUATIONS.
#
(2) CALCULATE LOS FROM CSM TO LM = RL - RC.
#
(3) THE PREFERRED TRACKING ATTITUDE IS DEFINED AS FOLLOWS:
THE TRACK AXIS (I) IS ALIGNED ALONG THE LOS TO THE LM. THE
TRACK AXIS (I) IS DEFINED AS:

```

```

#
UNIT(I)=UNIT(Z)COS55 & UNIT(X)SIN55
- -SC -SC
#
(4) COMPUTE DESIRED CDU ANGLES, USING VECPOINT.
#
(Sorry, I don't know where (5) and (6) are. --- RSB 2009.)
(7) FORM DIFFERENCE BETWEEN DESIRED AND ACTUAL CDUS.
IF ANY OF THE THREE ANGLE DIFFERENCES EXCEEDS 10 DEGREES,
GROSS MANEUVER IS REQUIRED. SIGNAL R61 (SET MPAC=1) TO
OPERATE KALCMANU AND EXIT CRS61.1.
IF ALL DIFFERENCES ARE LESS THAN 10 DEGREES, CONTINUE.
#
(8) CALCULATE ORTHOGONAL LOS RATE IN REF COORDS AS
#
OMEGATH = (UNITLOS(B1) X UNITDV(B1))(ABSDV(B7)/ABSLOS(B29))
#
CONVERSION FACTOR OF 100/2PI (B4) REV CSEC PER RAD SEC IS
APPLIED TO YIELD UNITS OF REVS/SEC. SCALE IS CARRIED AS
B+1+1+7-29+4+1 PLUS RESULTS OF NORMALIZING ABSDV, ABSLOS.
THE EXTRA B+1 RESULTS FROM RESCALING ABSDV B8 AFTER NORM
TO AVOID OVFLOW ON DIVIDE.
#
UNITLOS = UNIT(RL - RC) B1.
UNITDY = UNIT(VL - VC) B1.
ABSLOS = LENGTH OF LOS, METERS B29.
ABSDV = LENGTH OF DV, METERS/CSEC B7.
#
(9) OBTAIN RATE IN SM COORDS.
#
OMETATHSM = (REFSMMAT)(OMEGATH).
#
(10) OBTAIN GIMBAL ANGLE INCREMETNS FOR 0.1 SECOND.
#
DTHETASM = (0.1)(OMEGATHSM)
#
(11) OBTAIN DELCDUX,Y,Z USING SUBR SMCDURES.
Page 591
INPUT CONSISTS OF
#
(A) VECTOR OF ANGULAR INCREMENTS, DTHETASM, STORED
IN V(DTHETASM).
(B) SIN,COS CDUX,Y,Z FROM SUBR CDUTRIG.
#
TRANSFER OUTPUT OF SMCDURES FROM V(DCDU) TO VAC14D.
#

```

```

(12) CALCULATE ANG LOS RATE IN BODY(NB) COORDS USING SUBR SMNB.
#
OMEGANB = (SMNB)(OMETATHSM)
#
SUBR SMNB REQUIRES OMEGATHSM IN V(VAC32D) AND ACTUAL CDUS
(Y,X,Z ORDER) IN V(VAC20D) WITH S1 OF VAC = BASE ADDRESS
OF CDUS (FIXLOC + 20D).
#
(13) CALCULATE ANG LOS RATE IN CONTROL COORDS AS FOLLOWS
#
WBODY = (MBDYCTL)(OMEGANB) UNITS=REVS/SEC(B0) (?).
#
(0.5 0 0) BODY TO
MBDYCTL(B1) = (0 COS(7.25)B1 -SIN(7.25)B1) = CONTROL
(0 SIN(7.25)B1 COS(7.25)B1) AXES
CONVERSION
MATRIX
#
(14) RESCALE WBODY TO UNITS OF 460 DEG/SEC BY APPLYING FACTOR
OF 0.8 TO REVS/SEC.
#
(15) ADDRESS LIVE AUTOPILOT REGISTERS IN BASIC (UNDER INHINT).
#
TRANSFER DESIRED CDUS, SCALED 180 DEGREES, FROM T(SAVEDCDU)
TO V(CDUXD).
#
TRANSFER DELCDUS, SCALED 180 DEG, FROM V(VAC14D)
TO V(DELCDUX).
#
TRANSFER OMEGA CONTROL, SCALED 450 DEG/SEC, FROM V(MPAC)
TO V(WBODY).
#
RELINT, SET MPAC=0, EXIT CRS61.1.
#
CALL: L CALL CRS61.1
#
RETURNS: ALL TO L+1.
#
(1) S(MPAC)=0. NORMAL EXIT. 3 SETS OF INPUTS FED TO DAP.
(2) S(MPAC)=1. CALCULATED DESIRED CDUS,SP, SET IN T(CPHI)
FOR KALCMANU. ABS(ACDU - DCDU) EXCEEDS 10 DEGREES.
(3) S(MPAC)=2. GNCS AUTO MODE NOT SELECTED (BIT10=1).
(4) S(MPAC)=3. DAP HOLD FLAG (HOLDFLAG) NOT EQUAL -1.
#
Page 592
#
INPUT: (1) TIME2,TIME1. COMPUTER CLOCK TIME,DP, CENTISEC B28.

```

```

(2) CDUX,Y,Z. PRESENT CDU ANGLES,SP,2S COMPL HALF-REVS BO.
(3) M(REFSMMAT), STABLE MEMBER COORDS B1.
#
OUTPUT: NORMAL. EXIT WITH S(MPAC) = 0.
#
(1) CDUXD,CDUYD,CDUZD, DESIRED OUTER, INNER, MIDDLE CDU ANGLES,
DP, IS COMPL, SCALED 180 DEGREES (HALF-REVS BO).
(2) DELCDUX,DELCDUY,DELCDUZ. 0.1 SEC DCU ANGULAR INCREMENTS,
DP, IS COMPL, SCALED 180 DEG.
(3) WBODY,WBODY1,WBODY2. LOS ANGULAR RATE IN CONTROL COORDS,
DP, IS COMPL, SCALED 450 DEG/SEC.
#
SPECIAL. EXIT WITH S(MPAC) = 1.
#
(1) CPHI,CTHETA,CPSI. DESIRED OUTER, INNER, MIDDLE CDU ANGLES,
SP, 2'S COMPL, SCALED 180 DEGREES.
#
EXTERNAL SUBROUTINES USED (B)=BASIC
#
(1) CALCGA (5) LOADTIME(B) (9) SMNB
(2) CDUTRIG (6) MATMOVE
(3) CSMCONIC (7) RCDUS(B)
(4) LEMCONIC (8) SMCDURES
#
ERASABLE
#
(1) S(Q611),EBANK7 CRS61.1 EXIT.
(2) S(Q6111),EBANK7 CALCDCDU EXIT.
(3) T(SAVEDCDU),E6 SP VECTOR OF CDUDS.
(4) V(SAVEPOS),E7 CSM POS VEC AND D(SAVEPOS)= LENGTH OF LOS.
(5) V(SAVEVEL),E7 CSM VEL VEC.
#
FLAGWDS: HOLDFLAG. USED, NOT SET.
#
MISC: (1) ERASABLE ITEMP1 USED TO TEMP STORE EBANK UNDER INHINT.
(2) ERASABLE P21TIME USED AS TEMP STORE DURING CRS61.1
(3) ERAS DTHETASM USED AS TEMP STORE DURING EARLY CRS61.1
#
DEBRIS -- CURRENT VAC AREA, CRS61.1 ERASABLES, ITEMP1, P21TIME
#
BANK 24
SETLOC P20S4
BANK
#
EBANK= CDUXD
COUNT* $/CRS61

```

```

CRS61.1 STQ SETPD
 Q611
 0
 RTB
Page 593
 LOADTIME # LOAD CLOCK TIME2,1 INTO MPAC.

STORT STCALL P21TIME # STORE CLOCK TIME FOR SUBR R63
 R63 # SUBR TO CALC DCU (T=PRESENT,PASS1)
 TLOAD
 THETAD # SAVE DCU(T) FROM CALCDCU FOR STEP4.
 STORE SAVEDCDU

 EXIT
 TC STEP2CK
 CAF PRI030
 EXTEND
 RXOR CHAN31
 MASK FURST3
 EXTEND
 BZF DAPCK # AUTO MODE SELECTED (BITS 15-13=011)
 TC ASET # YES -- CONTINUE.

DAPCK CS FLAGWRD1 # IS STIKFLAG SET (I.E., IS SOMEONE ON RHC)
 MASK STIKBIT
 CCS A
 TC STEP3CK
 CAF ZERO
 TS MPAC
 TC INTPRET # EXIT CRS61.1
 GOTO Q611

STEP2CK TC BANKCALL
 CADR UPACTOFF

 CAF TWO # SET TEMPORARY INDEX DTHETASM = 2
 TS DTHETASM
 INDEX DTHETASM
 CA CDUX # SET A = ACTUAL CDU (ACDU).
 EXTEND
 INDEX DTHETASM # SET INDEX TO ACCESS DESIRED CDU (DCDU).
 MSU THETAD # A = DIFF = ACDU - DCU.
 TS MPAC # RETURN TO INTERPRETER FOR 10 DEGREE CK.
 TC INTPRET # (DP APPROX SP OK FOR ROUGH CHECK)

```

July 12, 2016

1039

```

ABS DSU
 DEGREE10 # IS (ACDU - DCDU) MORE THAN 10 DEGREES.
BPL EXIT # NO -- OK, CONTINUE CHECKING OTHER ANGLES.
 STKTEST # TEST STICK FLAG
CCS DTHETASM # HAVE ALL 3 ANGLE DIFFS BEEN CHECKED.
TC CDULOOP # NO -- DIM COUNT, CHECK NEXT ANGLE DIFF.
TC AUTOCK
STKTEST EXIT
CS FLAGWRD1
MASK STIKBIT

Page 594
CCS A
TC MANUEXIS # STIKFLAG IS NOT SET (DO R63)
CAF BIT3
EXTEND
WOR DSALMOUT # STIKFLG IS SET
 # TURN ON UPACTY LIGHT

STEP3CK TC ASET # EXIT AND SET R61CNTR
 TC INTPRET
 SETPD
 0 # *
 # NOW HAVE DCDUS STORED IN T(SAVEDCDU).
 # GO CALC OTHER DAP INPUTS (DELCDU,WBODY)

CRS61.2 VLOAD VSU
 DCDU
 SAVEVEL # DV = VL - VC
UNIT VCOMP # V(MPAC)=-UNITDV. VAC36D=ABSDV.
VXV VXSC # (-UNITDV)CROSS(UNITLOS).
 SAVEPOS
 RVCS/RDS # (UNITLOS B1)(UNITDV B1)(CONST B4)=CROSS.
PUSH # HOLD CROSS IN PUSHLIST0. SCALED B6.
DLOAD NORM # OBTAIN ABS VALUE OF LOS.
 P21TIME # P21TIME IS TEMP STORE FOR ABSLOS.
 X1
PUSH # NORM ABSLOS(DENOM) AND HOLD IN PUSH1.

 DLOAD NORM
 36D # NORM ABS VALUE OF DV(NUM).
 S1

XSU,1 SR1 # X1 = X1(N DENOM)-S1(N NUM).
 S1 # SR1 TO AVOID OFLOW ON DOV.
DDV VXSC # ABSDV(MPAC)/ABSLOS(PUSH1) = QUOT.
SXA,1 # QUOT(MPAC) X CROSS(PUSH0)
 Q6111 # SAVE SCALE OF RESULT (R-15,1X).
 # X1= NORM OF QUOT. QTUOT SCALE B7-B29=B-22
```

```

CROSS IS SCALED B6. NEED SL1 TO RECOVER
SR1 SO THAT -22+6+1=-15. MPAC NOW HOLDS
ORTHO LOS RATE (OMEGA TH, B-15,X1).
OBTAIN RATE IN SM COORDS (OMEGTHSM) AND
ADJUST FOR REFSMMAT SCALE OF B1.
OMEGTHSM = VAC20D
DELTA THETA SM = OMEGTHSM * .1B-3.

MXV VSL1
 REFSMMAT
STORE 20D

VXSC

 TENTH
STORE DTHETASM
CALL

 CDUTRIG
SETPD CALL
 0
 SMCDURES

STORE SM INCREM ANGLES FOR SMCDURES.
OBTAIN SIN,COSCDUS FOR SMCDURES.
SMCDURES USES PUSH
OBTAIN DELCDU IN V(DCDU).

Page 595

LXA,1

 Q6111
VLOAD VSL*
 DCDU
 0 -17D,1
STORE 14D

RELOAD X1
RECOVER SCALE.
(B-15,X1) + TENTH(B-3) + HALFREVS(B1)
EQUALS B-17D,1 TO OBTAIN HALFREVS BO.
HOLD DELS IN V(VAC14D) FOR AUTOPILOT.

CALL

 CDUTRIG
VLOAD CALL
 20D
 SMNB

COMPUTES SINES AND COSINES FOR *SMNB*
LOAD VECTOR AND CALL TRANSFORMATION
VECTOR FOR TRG*SMNB INTO MPAC
OBTAIN ANG. RATE REFERRED TO NB (BODY)

MXV

 MBDYCTL
VXSC

 POINT8
LXA,1 VSL*
 Q6111
 0 -14D,1

CONVERT RATE(OMEGA) TO CONTROL COORDS.
MULT. BY 0.8 TO RESCALE REVS TO 450 DEG.
RECOVER SCALE.
RELOAD X1 TO RECOVER NORMALIZ.
(B-15,X1) + MBDYCTL(B1) = B-14D,1 TO
OBTAIN REVS SCALED AT 450 DEGREES.

CRS61.2A EXIT
 INHINT
CAF ZERO
TS CDUXD +1
TS CDUYD +1
TS CDUZD +1
CA SAVEDCDU
TS CDUXD
CA SAVEDCDU +1
TS CDUYD

```



July 12, 2016

1041

```
CA SAVEDCDU +2
TS CDUZD

EXTEND
DCA MPAC
DXCH WBODY
EXTEND
DCA MPAC +3
DXCH WBODY1
EXTEND
DCA MPAC +5
DXCH WBODY2

EXTEND
INDEX FIXLOC
DCA 14D
DXCH DELCDUX
EXTEND
INDEX FIXLOC
DCA 16D

Page 596
DXCH DELCDUY
EXTEND
INDEX FIXLOC
DCA 18D
DXCH DELCDUZ
CS ONE
TS HOLDFLAG
RELINT
TC ASET
MANUEXIS TC INTPRET
MANUEXIT TLOAD
 SAVEDCDU
STORE CPHI
SLOAD GOTO
 LOONE
 Q611

R63 STQ DLOAD
 Q6111
 P21TIME
 STCALL TDEC1
 CSMCONIC

HOLDATT VLOAD
 RATT
 STOVL SAVEPOS

TRANSFER OMEGA CONTROL (ANG. LOS RATE)
FROM V(MPAC) TO V(WBODY)

TRANSFER CDU INCREMENTS
FROM V(VAC14D) TO V(DELCDUX)

NOW DAP VARIABLES LOADED. SET HOLDFLAG.
TO -1.

ENTER FROM STEP2. ACDU-DCDU EXCEEDS
10 DEG. STORE DCDU(T) IN CPHI,CTHETA,
CPSI FOR KALCMANU.
SPECIAL RETURN (MPAC+0 = 1)
OCTAL 00001

SUBR TO CALC DCDUS(T)

HOLD EXTRAPOLATED CSM POSITION AND
VELOCITY
```

|            |        |           |                                        |
|------------|--------|-----------|----------------------------------------|
|            |        | VATT      |                                        |
|            | STORE  | SAVEVEL   |                                        |
| CALCLEM    | DLOAD  |           | # EXTRAPOLATE LEM STATE VECTOR TO SAVE |
|            |        | P21TIME   | # TIME AS CSM USING LEMCONIC           |
|            | STCALL | TDEC1     |                                        |
|            |        | LEMCONIC  |                                        |
|            | VLOAD  |           |                                        |
|            |        | VATT      |                                        |
|            | STOVL  | DCDU      | # STORE VATT IN DCDU TEMPORARILY       |
|            |        | RATT      | # LOS = RL RC                          |
|            | VSU    | UNIT      |                                        |
|            |        | SAVEPOS   |                                        |
|            | STORE  | SAVEPOS   | # SAVE UNITLOS FOR CRS61.2 RATE CALC.  |
|            | MXV    | VSL1      |                                        |
|            |        | REFSMMAT  | # CONVERT TO STABLE MEMBER             |
|            | STODL  | POINTVSM  |                                        |
|            |        | 36D       | # HOLD ABS VAL OF LOS (VAC 36D)        |
|            | STORE  | P21TIME   | # IN D(P21TIME) FOR CRS61.2 RATE CALC. |
|            | VLOAD  |           |                                        |
|            |        | UNITX     |                                        |
|            | STCALL | SCAXIS    | # TRACK AXIS UNIT VECTOR               |
|            |        | VECPOINT  | # FOR +X-AXIS TRACKING ATTITUDE        |
|            | STORE  | CPHIX     | # STORE ANGLES FOR N96 DISPLAY         |
|            | VLOAD  |           |                                        |
|            |        | PRFUNIT   |                                        |
| # Page 597 |        |           |                                        |
|            | STCALL | SCAXIS    |                                        |
|            |        | VECPOINT  |                                        |
|            | STORE  | PRAXIS    | # STORE ANGLES FOR N95 DISPLAY         |
|            | BOFF   |           |                                        |
|            |        | PRFTRKAT  |                                        |
|            |        | CRSTOR1   |                                        |
| CRSTOR     | STORE  | THETAD    | # STORE ANGLES FOR N18 DISPLAY         |
|            | GOTO   |           |                                        |
|            |        | Q6111     |                                        |
| CRSTOR1    | VLOAD  |           |                                        |
|            |        | UNITX     |                                        |
|            | STORE  | SCAXIS    |                                        |
|            | TLOAD  | GOTO      |                                        |
|            |        | CPHIX     |                                        |
|            |        | CRSTOR    |                                        |
| PRFUNIT    | 2DEC   | .40957602 | # 55 DEG TRACK AXIS UNIT VECTOR        |
|            | 2DEC   | 0.0       | # FOR USE WITH VECPOINT                |
|            | 2DEC   | .28678822 |                                        |

July 12, 2016

1043

|          |        |                |                                            |                        |
|----------|--------|----------------|--------------------------------------------|------------------------|
| DEGREE10 | DEC    | .05556         | # 10 DEG IN REVS                           | STEP2                  |
| RVCS/RDS | 2DEC   | 15.915494 B-4  | # 100/2PI REV-CSEC/RAD-SEC.                |                        |
| TENTH    | 2DEC   | .1 B+3         | # .1 B-3 (TO SCALE ANG. RATE TO .1 INREMS) |                        |
| MAT1B1   | 2DEC   | 1.0 B-1        |                                            |                        |
| MBDYTCTL | 2DEC   | .5             | #                                          | 7.25 DEG NEGATIVE      |
|          | 2DEC   | 0              | #                                          | X-AXIS ROTATION MATRIX |
|          | 2DEC   | 0              | #                                          | CONVERTS BODY TO CTL   |
|          | 2DEC   | 0              | #                                          | AXES. CAME AS QUADROT  |
|          | 2DEC   | .99200495 B-1  | # COS7.25 B1                               | BUT SCALED B           |
|          | 2DEC   | -.12619897 B-1 | # -SIN7.25 B1                              |                        |
|          | 2DEC   | 0              |                                            |                        |
|          | 2DEC   | .12619897 B-1  | # SIN7.25 B1                               |                        |
|          | 2DEC   | .99200495 B-1  | # COS7.25 B1                               |                        |
| LOONE    | OCT    | 00001          | # TO SET MPAC = 00001 FOR SPECIAL EXIT.    |                        |
| FURST3   | EQUALS | 13,14,15       | # CONSTANT FOR AUTOCK (OCT 70000).         |                        |

# Page 598

# ..... S22.1 ORBITAL NAVIGATION ROUTINE

# MOD 1

#

# FUNCTIONAL DESCRIPTION

# 1. UPDATE CSM STATE VECTOR

# 2. UPDATE LANDMARK POSITION

# 3. CONVERT W MATRIX FROM 9 TO 6 DIMENSIONS

#

# SUBROUTINES CALLED

# INTSTALL,INTEGRV,GETNUM,SETRE,R-TO-TP,RP-TO-R,BVECTORS,INCP1,INCP2

# LALOTORV,S22F2410,LAT-LONG,ROWDOT

#

# ERASABLE INITIALIZATION

# W=9X9 MATRIX

# ORBFLAG=0 FOR INVALID W MATRIX, =1 FOR VALID W MATRIX

# ASTRONAUT ENTRY OF KNOWN,L,OFF

```

8NN= NUMBER OF MARKS, DECIMAL INTEGER B-14
REFSMMAT= TRANSFORMATION MATRIX
MARKSTAT= ADDRESS OF START OF MARK DATA (MARK DATA OF EACH MARK IS
STORED AS FOLLOWS: TIME,AIG,SA,AMG,PA,AOG) TIME IS IN DOUBLE
PRECISION, ALL OTHERS ARE IN SINGLE PRECISION
CSM STATE VECTOR
#
OUTPUT
UPDATED CSM STATE VECTOR
UPDATED LANDMARK POSITION
NEW 6 DIMENSIONAL W MATRIX
#
DEBRIS
PUSH LIST,CSMPOS,ALPHAV,ERADM,UM,RCLP,USTAR,VARIANCE,X789,BVECTOR,8KK,
S22LOC,SVMRKDAT TABLE,22SUBSCL,LANDMARK,CXOFF,S22C,LAT,LOG,ALT,
TEMPOR1,S22TOFF,S221OFF,DSPTM1,S22EORM,S22TPRIM

 BANK 13

 SETLOC P20S6
 BANK

 EBANK= LANDMARK
 COUNT 35/LUORB

S22.1 STQ SSP
 S22RTNEX
 S1
 DEC 6
 SSP SSP # SET I=1 ITEM 8KK IS I
 8KK
 DEC 1
 S22LOC
 ECADR SVMRKDAT # SET MARK DATA ADDRESS INTO S22LOC

Page 599
 LXC,2 AXT,1
 MARKSTAT
 DEC 36
S22.111 VLOAD* # MOVE MARK DATA (5 SETS FROM ADDR. IN
 0,2 # MARKSTAT TO SVMRKDAT TABLE TO AVOID LOSS
 STORE SVMRKDAT +36D,1 # IF RESTART OCCURS
 INCR,2 TIX,1
 DEC -6
 S22.111
 SET EXIT
 P22MKFLG # DOWNLINKED SVMRKDAT HOLDS PRESENT MARKS

```

July 12, 2016

1045

```
TC BANKCALL # RELEASE VAC AREA WHERE MARK DATA WAS
CADR MKRELEAS
TC 2PHSCHNG
OCT 00004
OCT 05022
OCT 13000
TC INTPRET
AXT,1 BOFF
 OD
 CMOONFLG # =0 EARTH, =1 MOON
 S22SHIFT
S22SHIFT INCR,1
 DEC -2
 SXA,1 SETPD
 S22EORM # SET =0 EARTH, =-2 MOON FOR SHIFTING
 OD
FIG2EXIT CALL
 INTSTALL
 CALL
 S22FLGS
FLOWCHART D=0 THEN DIMOFLAG=0, D609FLG NOT TESTED
FLOWCHART D=6 THEN DIMOFLAG=1, D60R9FLG=0
FLOWCHART D=9 THEN DIMOFLAG=1, D60R9FLG=1
BOFF CLRG0
 ORBWFLAG
 SETWW5D # BRANCH TO SET W0-W5, ORBWFLAG, D
 D60R9FLG # FLOWCHART D=6 PATH
 SETVANDI
SETWW5D CLEAR
 DIMOFLAG # FLOWCHART D=0 PATH
 AXT,1 SSP
 DEC 108
 S1
 DEC 6
 CLEAR VLOAD
 RENDWFLG # GSOP CHANGE 8/18/67
 ZEROVECS
CLEARWW5 STORE W +108D,1
Page 600
 TIX,1 SLOAD
 CLEARWW5
 WORBPOS
 STORE W # SET DIAGONALS OF W0
 STORE W +8D
 STORE W +16D
```

```

 SLOAD
 WORBVEL
 STORE W +72D # SET DIAGONALS OF W4
 STORE W +80D
 STORE W +88D
SETVANDI CLEAR
 DMENFLG # 0=6X6W, 1=9X9W
S22NXTIN CALL
 GETTF
 STCALL TDEC1
 INTEGRV
 CALL
 S22CALRC # CALC. RC B-29 OR B-27 (CSMPOS)
 LXA,1 SXA,1
 S22LOC # SETUP ADDR. OF MARK DATA FOR GETUM SUBR.
 MARKDATA
 CALL # COMPUTE UM
S2GETUM GETUM
 STORE UM
DMPINTEG SLOAD PUSH # TEST OFF=I
 8KK
 SLOAD SR3 # CXOFF SCALED B-5, MUST MOVE TO B-14
 CXOFF # BEFORE SUBT.
 SR3 SR3
 DSU
 BHIZ BON
 S22OFF=I # BRANCH HERE IF OFF=I
 DMENFLG # 0=6X6W, 1=9X9W
 S22D=9
 CALL
 GRP2PC
 SET
 ORBWFLAG
 SET SET
 DMENFLG # =0 ON FIRST PASS THRU HERE FOR D=0, OR 6
 22DSPFLG # =1 TO DISPLAY DR,DV ON FIRST PASS
 SET BON
 ERADFLAG # =1 TO COMPUTE FISCHER RADIUS
 KNOWNFLG
 S22BOX22
 VLOAD UNIT # UNIT ALSO PUTS ABVAL(RC) IN 36D
 CSMPOS
 STORE ALPHAV # ALPHAV +4=SINL FOR SETRE
 CLEAR BOFF
 LUNAFLAG

```

July 12, 2016

1047

```

 CMOONFLG
 S22C=I
S22C=I SET LUNAFLAG
 CALL # ERADM= R0 METERS B-29 BOTH EARTH/MOON
 SETRE
 CALL # COMPUTE RL FROM EQUATION 2.4.10
 S22F2410 # STORED IN X789,MPAC B-27,B-29
 BOFF # SCALE RL B-29 FOR BOTH EARTH/MOON
 VSR2
 CMOONFLG
 +1
 STORE S22RL
 DOT SL1
 UM
 STOVL S22D # D=UM RL B-29
 ZEROVECS
 SETPD PUSH
 OD
 PUSH PDDL # SET 0-18D = I BACKWARDS
 HIDPHALF # PD 18
 SR2 # B-3
 STORE 4D
 STORE 8D
 STOVL 12D
 UM # B-1
 STOVL S223X1
 S22RL # B-29
 CALL # (UM)(RL T) B-30 STORED IN S22UMRL THRU
 S2231X13 # S22UMRL +17D
 AXT,1 SSP
 DEC 18
 S1
 DEC 6
S22NXTU VLOAD* VSR2 # (UM)(RL T) B-32
 S22UMRL +18D,1
 V/SC
 S22D # D B-29
 BVSU STADR # SUBTRACT FROM I B-3
 STORE S22UMRL +18D,1 # U MATRIX B-3
 TIX,1 AXT,1 # PD 0 AFTER TIX
 S22NXTU
 DEC 36 # S1 STILL 6 FROM ABOVE
S22NXTWI VLOAD* MXV
 W +36D,1 # B-19
 S22UMRL # B-3
 VSL3
```

# Page 602

|         |        |                |                                      |
|---------|--------|----------------|--------------------------------------|
|         | STORE  | W +144D,1      | # W(I+18)= UW(I) B-19                |
|         | TIX,1  | DLOAD          |                                      |
|         |        | S22NXTWI       |                                      |
|         |        | S22RHO         | # B-28,B-30                          |
|         | BOFF   | SR2            | # MAKE RHO B-30                      |
|         |        | CMOONFLG       |                                      |
|         |        | +1             |                                      |
|         | NORM   | XAD,2          |                                      |
|         |        | X2             |                                      |
|         |        | X2             |                                      |
|         | DSQ    | DMP            |                                      |
|         |        | SCTVAR         | # B+16                               |
|         | SR1    |                | # ACCOUNTS FOR 1/2 IN NEXT FORMULA   |
|         | STORE  | S22RHO         | # 1/2(RHO SQ)(VARSC)                 |
|         | AXT,1  |                |                                      |
|         | DEC    | 18             | # S1 STILL 6 FROM ABOVE              |
| S22NXXA | VLOAD* | MXV            |                                      |
|         |        | S22UMRL +18D,1 | # B-3                                |
|         |        | S22UMRL        | # B-3                                |
|         | VXSC   | VSR*           |                                      |
|         |        | S22RHO         |                                      |
|         |        | 0 -12D,2       | # WITH VARRP SCALED B-28             |
|         | STORE  | S22UUT +18D,1  | # 1/2(RHO SQ)(VARSC)(U)(U T)         |
|         | TIX,1  | VLOAD          |                                      |
|         |        | S22NXXA        |                                      |
|         |        | UM             |                                      |
|         | STCALL | S223X1         | # UM ALSO IN MPAC FOR S2231X13 SUBR. |
|         |        | S2231X13       | # (UM)(UM T) B-2 IN S22UMRL,P17D     |
|         | DLOAD  | SR3            |                                      |
|         |        | ERADM          | # B0 B-29                            |
|         | DDV    | DSQ            |                                      |
|         |        | S22D           | # B-29                               |
|         | DMP    |                |                                      |
|         |        | RPVAR          | # ***** METERS SQ                    |
|         | STORE  | S22RHO         | # TEMP (VARRP)(R0/D)                 |
|         | AXT,1  |                |                                      |
|         | DEC    | 18             | # S1 STILL 6 FROM ABOVE              |
| S22NXXB | VLOAD* | VXSC           |                                      |
|         |        | S22UMRL +18D,1 | # (UM)(UM T) B-2                     |
|         |        | S22RHO         |                                      |
|         | VAD*   |                |                                      |
|         |        | S22UUT +18D,1  |                                      |
|         | STORE  | S22UUT +18D,1  | # SMALL E MATRIX                     |
|         | VLOAD  |                |                                      |
|         |        | ZEROVECS       |                                      |



# Page 603

|         |       |             |                            |
|---------|-------|-------------|----------------------------|
|         | STORE | W +162D,1   | # CLEAR W8                 |
|         | TIX,1 | BOV         |                            |
|         |       | S22NXXB     |                            |
|         |       | +1          |                            |
|         | DLOAD | BMN         |                            |
|         |       | S22UUT +16D | # E5                       |
|         |       | S22W76X     |                            |
|         | SQRT  | BZE         |                            |
|         |       | S22W76X     |                            |
|         | STODL | W +148D     | # W74= SQ ROOT E5          |
|         |       | S22UUT +14D | # E4                       |
|         | DDV   | BOV         |                            |
|         |       | W +148D     |                            |
|         |       | S22W72X     |                            |
| S22W72X | STORE | W +146D     | # W73= E4/W74              |
|         | DLOAD | DDV         |                            |
|         |       | S22UUT +12D | # E3                       |
|         |       | W +148D     |                            |
|         | BOV   |             |                            |
|         |       | S22W76X     |                            |
| S22W76X | STORE | W +144D     | # W72= E4/W74              |
|         | DLOAD | DSQ         |                            |
|         |       | W +146D     | # W73                      |
|         | BDSU  | BMN         |                            |
|         |       | S22UUT +8D  | # E2                       |
|         |       | S22W78X     |                            |
|         | SQRT  | BZE         |                            |
|         |       | S22W78X     |                            |
|         | STODL | W +152D     | # W76= SQ ROOT (E2-W73 SQ) |
|         |       | W +144D     | # W72                      |
|         | DMP   | BDSU        |                            |
|         |       | W +146D     | # W73                      |
|         |       | S22UUT +6D  | # E1                       |
|         | DDV   | BOV         |                            |
|         |       | W +152D     | # W76                      |
|         |       | S22W78X     |                            |
| S22W78X | STORE | W +150D     | # W75= (E1-W72W73)/W76     |
|         | DLOAD | DSQ         |                            |
|         |       | W +150D     |                            |
|         | PDDL  | DSQ         |                            |
|         |       | W +144D     | # W72                      |
|         | DAD   |             |                            |
|         | BDSU  | BMN         |                            |
|         |       | S22UUT      | # E0                       |
|         |       | S22SCLW     |                            |

1050

July 12, 2016

|            |        |            |                                            |
|------------|--------|------------|--------------------------------------------|
|            | SQRT   |            |                                            |
|            | STORE  | W +156D    | # W78= SQ RT(E0-W72 SQ-W75 SQ)             |
| S22SCLW    | VLOAD  | VSR1       | # SCALE W6 METERS B-19                     |
|            |        | W +144D    |                                            |
|            | STOVL  | W +144D    |                                            |
|            |        | W +150D    |                                            |
|            | VSR1   |            |                                            |
|            | STOVL  | W +150D    |                                            |
|            |        | W +156D    |                                            |
|            | VSR1   |            |                                            |
|            | STORE  | W +156D    |                                            |
| S22SAVET   | CALL   |            |                                            |
|            |        | GETTF      |                                            |
| # Page 604 |        |            |                                            |
|            | STORE  | S22TPRIM   | # SAVE PRESENT TIME FOR PIOS               |
| S22I=N     | EXIT   |            | # TEST I=N                                 |
|            | TC     | PHASCHNG   |                                            |
|            | OCT    | 04022      |                                            |
|            | CS     | 8KK        |                                            |
|            | AD     | 8NN        |                                            |
|            | EXTEND |            |                                            |
|            | BZMF   | S22F244X   | # EXIT TO FIGURE 2.4-4                     |
|            | CA     | 8KK        | # I=I+1                                    |
|            | AD     | ONE        |                                            |
|            | TS     | TEMPOR1    |                                            |
|            | CA     | S22LOC     | # ADD 7 TO LOC TO GET ADDR. OF NEXT MARK   |
|            | AD     | SEVEN      |                                            |
|            | TS     | TEMPOR1 +1 |                                            |
|            | TC     | PHASCHNG   |                                            |
|            | OCT    | 04022      |                                            |
|            | CA     | TEMPOR1    |                                            |
|            | TS     | 8KK        |                                            |
|            | CA     | TEMPOR1 +1 |                                            |
|            | TS     | S22LOC     |                                            |
|            | TC     | INTPRET    |                                            |
|            | CALL   |            | # FOR ALL INTEGRATIONS OTHER THAN FIRST    |
| S2INTS1    |        | INTSTALL   |                                            |
|            | CALL   |            |                                            |
|            |        | S22FLGS    |                                            |
|            | BON    | CLEAR      |                                            |
|            |        | DMENFLG    |                                            |
|            |        | S22NXTIN   | # RETURN ALWAYS EXCEPT OFFSET POINT MARK 1 |
|            |        | DIMOFLEG   |                                            |
|            | BOFF   | SET        |                                            |
|            |        | ORBWFLAG   |                                            |
|            |        | S22NXTIN   | # OFFSET POINT MARK 1, NO W INTEGRATION    |

July 12, 2016

1051

```

 DIMOFLAG
 CLRGO
 D6OR9FLG
 S22NXTIN # OFFSET POINT MARK 1, INTEGRATE W 6X6
S220FF=I CALL
 GETTF
 STOVL S22TOFF # TIME SUB OFF
 UM
 STCALL S22UOFF # U SUB OFF
 S22I=N # TEST I=N
S22D=9 VLOAD # D=9 PATH
 X789
 STODL OD # CALL PIOS TWICE TO TRANSFORM RL TO TIME
 S22TPRIM # T(SUB F) FROM TIME T PRIME
 STORE 6D
 SLOAD CALL
 S22EORM # 0=EARTH, NON-ZERO=MOON
 R-TO-RP
S2RTRP
Page 605 PUSH CALL # R-TO-RP LEAVES PUSHLOC AT 0
 GETTF
 STORE 6D
 SLOAD CALL
 S22EORM
 RP-TO-R
S2RPTR
S22BOX32 STORE X789
 SET BOV
 INCORFLG # FLAG=1
 +1 # CLEAR OVERFLOW
 VSU
 CSMPOS
 STORE RCLP # RCL=RL-RC
 UNIT VXV # USTAR=UNIT(UNIT(RCL)XUM)
 UM
 UNIT BOV
 S22SAVET # COMPUTATION OVERFLOW, SAVE TF
 STORE USTAR
S22BOX12 SET SET
 DMENFLG # =1 FOR 9X9 W
 VEHUPFLG # =1 FOR CSM
 DLOAD DAD
 SCTVAR # B+18
 IMUVARR # B+18
 STOVL VARIANCE
 RCLP # B-29 OR B-27
 ABVAL NORM
```

|            |        |              |                                            |
|------------|--------|--------------|--------------------------------------------|
|            |        | X1           |                                            |
|            | DSQ    | DMP          |                                            |
|            |        | VARIANCE     |                                            |
|            | XAD,1  | XAD,1        |                                            |
|            |        | X1           | # DOUBLE NORM SHIFT SINCE RCLP WAS SQUARED |
|            |        | S22EORM      | # DOUBLE EARTH OR MOON SHIFT, SAME REASON  |
|            | XAD,1  | SR*          |                                            |
|            |        | S22EORM      |                                            |
|            |        | 0,1          | # SCALE VARIANCE B-40 FOR BOTH EARTH, MOON |
|            | TLOAD  |              | # CHANGE MODE TO TRIPLE                    |
|            |        | MPAC         |                                            |
| S2BVTRS    | STCALL | VARIANCE     | # CALC B0,B1,DELTAQ, NEW USTAR             |
|            |        | BVECTORS     |                                            |
|            | VLOAD  | VCOMP        |                                            |
|            |        | BVECTOR      |                                            |
| S2INCP1    | STCALL | BVECTOR +12D | # B2=-B0                                   |
|            |        | INCRP1       |                                            |
|            | CALL   |              |                                            |
|            |        | GRP2PC       |                                            |
|            | BOFF   | CLEAR        |                                            |
|            |        | 22DSPFLG     | # =1 DISPLAY DELTA R,V      =0 DO NOT      |
|            |        | S22BOX42     |                                            |
|            |        | 22DSPFLG     |                                            |
| # Page 606 |        |              |                                            |
|            | CALL   |              |                                            |
|            |        | GRP2PC       |                                            |
|            | VLOAD  | ABVAL        |                                            |
|            |        | DELTAX       | # DELTA R                                  |
|            | LXA,1  | SR*          |                                            |
|            |        | S22EORM      | # SCALE DELTA R ALWAYS METERS B-29         |
|            |        | 0,1          |                                            |
|            | STOVL  | N49DISP      |                                            |
|            |        | DELTAX +6    | # DELTA V                                  |
|            | ABVAL  | SR*          | # DELTA V=METERS/CSEC B-7 ALWAYS           |
|            |        | 0,1          |                                            |
|            | STORE  | N49DISP +2   |                                            |
|            | EXIT   |              |                                            |
|            | CAF    | V06N49EE     |                                            |
|            | TC     | BANKCALL     |                                            |
|            | CADR   | GOFLASHR     |                                            |
|            | TC     | GOTOP00H     | # V34E TERMINATE                           |
|            | TC     | +5           | # INCORPORATE CHANGES                      |
|            | TC     | S22EXEX      | # V32E RECYCLE                             |
|            | CAF    | BIT3         |                                            |
|            | TC     | BLANKET      |                                            |
|            | TC     | ENDOFJOB     |                                            |

July 12, 2016

1053

|            |        |           |                                         |
|------------|--------|-----------|-----------------------------------------|
|            | TC     | INTPRET   |                                         |
| S22BOX42   | CALL   |           |                                         |
|            |        | INCORP2   |                                         |
|            | CALL   |           | # CSMPOS=RC B-29 OR B-27                |
|            |        | S22CALRC  |                                         |
| DMPINCP2   | BOFF   | CALL      |                                         |
|            |        | INCORFLG  |                                         |
|            |        | S22SAVET  | # SAVE TF AND TEST I=N                  |
|            |        | GRP2PC    |                                         |
|            | CLEAR  | VLOAD     |                                         |
|            |        | INCORFLG  | # FLAG=0                                |
|            |        | X789      |                                         |
|            | VSU    |           |                                         |
|            |        | CSMPOS    |                                         |
|            | STCALL | RCLP      | # RCL=RL-RC                             |
|            |        | S22BOX12  |                                         |
| S22BOX22   | AXT,1  | SSP       | # CLEAR W6,W7,W8. (27 ELEMENTS 54 REGS) |
|            | DEC    | 54        |                                         |
|            |        | S1        |                                         |
|            | DEC    | 6         |                                         |
|            | VLOAD  |           |                                         |
|            |        | ZEROVECS  |                                         |
| CLRW678    | STORE  | W +162D,1 |                                         |
|            | TIX,1  | SLOAD     |                                         |
|            |        | CLRW678   |                                         |
|            |        | S22WSUBL  |                                         |
|            | STORE  | W +144D   |                                         |
|            | STORE  | W +152D   |                                         |
| # Page 607 |        |           |                                         |
|            | STORE  | W +160D   |                                         |
|            | CLEAR  | BOFF      | # SET LUNAFLAG, TIME FOR LALOTORV       |
|            |        | LUNAFLAG  | # ERADFLAG,LAT,LONG,ALT SET PREVIOUSLY  |
|            |        | CMOONFLG  | # CHECK SCALING OF ITEMS,ALT INPUT AND  |
|            |        | S22BX22A  | # RL OUTPUT IN ALPHAV BOTH B-29         |
|            | SET    |           |                                         |
|            |        | LUNAFLAG  |                                         |
| S22BX22A   | CALL   |           |                                         |
|            |        | GETTF     |                                         |
|            | CALL   |           | # COMPUTE RL                            |
|            |        | LALOTORV  |                                         |
|            | VLOAD  | BOFF      |                                         |
|            |        | ALPHAV    | # RL B-29                               |
|            |        | CMOONFLG  |                                         |
|            |        | S22BX22B  |                                         |
|            | VSL2   |           | # SCALE RL B-27 FOR MOON                |
| S22BX22B   | GOTO   |           |                                         |

|            |        |          |                                        |
|------------|--------|----------|----------------------------------------|
| S22F244X   | TC     | S22BOX32 |                                        |
| S22F244    | SLOAD  | INTPRET  |                                        |
|            |        | BHIZ     | # FIG 2.4-4 TEST OFF=0                 |
|            |        | CXOFF    |                                        |
|            |        | S22BOX44 |                                        |
|            | SR     |          | # SCALE OFFSET B-14 THEN GET GR. 8NN   |
|            |        | 9D       |                                        |
|            | STORE  | ALPHAV   | # TEMP                                 |
|            | SLOAD  | DSU      |                                        |
|            |        | 8NN      |                                        |
|            |        | ALPHAV   |                                        |
|            | BMN    | CALL     | # OFFSET GR. NO. MARKS. FORGET IT      |
|            |        | S22BOX44 |                                        |
|            |        | GRP2PC   | # GROUP 2 PHASE CHANGE                 |
|            | DLOAD  |          |                                        |
|            |        | S22TOFF  |                                        |
|            | STCALL | TDEC1    | # CALC RC AT OFFSET TIME               |
|            |        | CSPREC   |                                        |
|            | VLOAD  |          |                                        |
|            |        | RATT1    | # RC METERS B-29 OR B-27               |
|            | STOVL  | CSPPOS   |                                        |
|            |        | S22UOFF  |                                        |
|            | STOVL  | UM       | # U=UOFF                               |
|            |        | X789     |                                        |
|            | ABVAL  | BOFF     |                                        |
|            |        | CMOONFLG |                                        |
|            |        | +2       |                                        |
|            | SR2    |          | # SCALE MOON RO B-29 FOR S22F2410 SUBR |
|            | STCALL | ERADM    |                                        |
|            |        | S22F2410 |                                        |
|            | GOTO   |          |                                        |
|            |        | S22BX44A |                                        |
| S22BOX44   | CALL   |          |                                        |
| # Page 608 |        |          |                                        |
|            |        | GETTF    |                                        |
|            | STORE  | S22TOFF  | # PRESENT TIME FOR LAT-LONG SETUP      |
| S22BX44A   | CLEAR  | VLOAD    |                                        |
|            |        | LUNAFLAG |                                        |
|            |        | X789     |                                        |
|            | BOFF   | SET      |                                        |
|            |        | CMOONFLG |                                        |
|            |        | S22BX44B |                                        |
|            |        | LUNAFLAG | # SET = 1 FOR LAT-LONG                 |
|            | VSR2   |          | # SCALE RL MOON B-29 FOR LAT-LONG      |
| S22BX44B   | STODL  | ALPHAV   | # RL SCALED B-29 FOR LAT-LONG          |
|            |        | S22TOFF  | # EITHER PRESENT OR OFFSET TIME        |

July 12, 2016

1055

|            |       |          |                                           |                         |
|------------|-------|----------|-------------------------------------------|-------------------------|
|            | CALL  |          |                                           |                         |
|            |       | LAT-LONG | # **** ALT OUTPUT ALWAYS B-29             |                         |
|            | CALL  |          | # DISPLAY LAT/LONG/ALT                    |                         |
|            |       | LLASRD   |                                           |                         |
|            | EXIT  |          |                                           |                         |
|            | CAF   | V06N89B  |                                           |                         |
|            | TC    | BANKCALL |                                           |                         |
|            | CADR  | GOFLASH  |                                           |                         |
|            | TC    | S22GTP   | # V34E TERMINATE                          |                         |
|            | TC    | +2       | # PROCEED                                 | SAVE LANDING SITE COORD |
|            | TC    | S22.981X | # RECYCLE                                 | POINT A IN GSOP         |
|            | TC    | INTPRET  |                                           |                         |
|            | DLOAD |          |                                           |                         |
|            |       | S22TOFF  | # EITHER PRESENT OR OFFSET TIME           |                         |
|            | STOVL | 6D       | # 6-7D= LANDING SITE TIME FOR R-TO-RP     |                         |
|            |       | X789     |                                           |                         |
|            | STORE | OD       | # 0-5D= LANDING SITE VEC FOR R-TO-RP      |                         |
|            | SLOAD | CALL     |                                           |                         |
|            |       | HIDPHALF | # ANY NON-ZERO FOR MOON                   |                         |
|            |       | R-TO-RP  | # CONVERT RLS TO MOON-FIXED COORD         |                         |
|            | STORE | RLS      | # LANDING SITE VECTOR                     |                         |
|            | EXIT  |          |                                           |                         |
| S22.981X   | TC    | INTPRET  |                                           |                         |
|            | CALL  |          |                                           |                         |
|            |       | 9DWT06DW |                                           |                         |
|            | EXIT  |          | # GO TO POINT A IN CHAPTER 5              |                         |
| S22EXEX    | TC    | INTPRET  | # WITHOUT CONVERTING W                    |                         |
|            | GOTO  |          |                                           |                         |
|            |       | S22RTNEX |                                           |                         |
|            |       |          |                                           |                         |
| S22GTP     | TC    | INTPRET  | # CONVERT W BEFORE TC GOTOP00H            |                         |
|            | CALL  |          |                                           |                         |
|            |       | 9DWT06DW |                                           |                         |
|            | EXIT  |          |                                           |                         |
|            | TC    | GOTOP00H |                                           |                         |
| S22F2410   | SETPD | VLOAD    | # COMPUTE FORMULA 2.4.10                  |                         |
|            |       | OD       |                                           |                         |
|            |       | CSMPOS   | # RC B-29 EARTH, B-27 MOON                |                         |
| # Page 609 |       |          |                                           |                         |
|            | UNIT  | DOT      | # UNIT ALSO SETS 36D=ABVAL(RC) USED BELOW |                         |
|            |       | UM       |                                           |                         |
|            | SL1   | DCOMP    | # GSOP CHANGE 8/18/67                     |                         |
|            | PUSH  |          | # PD 2D 8D=COSA=-(UM.RC)/ABVAL(RC)        | B-1                     |
|            | DSQ   | BDSU     |                                           |                         |
|            |       | DEC1B2   |                                           |                         |
|            | PDDL  | BOFF     | # PD 4D 2D=1-COSA SQ=SINA SQ              | B-2                     |

|                     |          |                                            |
|---------------------|----------|--------------------------------------------|
|                     | ERADM    | # R0 ALWAYS B-29 FROM SETRE                |
|                     | CMOONFLG |                                            |
|                     | +2       |                                            |
|                     | SL2      | # SCALE R0 B-27 FOR MOON                   |
|                     | SR1R     | # (R0/RC) B-1                              |
|                     | 36D      |                                            |
|                     | DSQ      | # PD 2D (RP/RC) SQ - SINA SQ B-2           |
|                     | SQRT     | # PD OD COSA-SQRT((R0/RC)SQ-SINA SQ) B-1   |
|                     | DMP      | # DMP RESULT B-28 MOON, B-30 EARTH         |
|                     | 36D      | # VXSC RESULT B-29 MOON, B-31 EARTH        |
|                     | STORE    | # RHO FOR W INIT. OF UNKNOWN LMK B-28,B-30 |
|                     | VXSC     |                                            |
|                     | UM       |                                            |
|                     | VSL2     | # SCALE B-27 MOON, B-29 EARTH AND ADD RC   |
|                     | VAD      |                                            |
|                     | CSMPOS   |                                            |
|                     | STORE    | X789                                       |
|                     | RVQ      | # B-27 FOR EARTH OR B-29 FOR MOON          |
| S22CALRC            | LXA,1    | # COMPUTE RC B-29 OR B-27                  |
|                     | VLOAD    | # =0 FOR EARTH, -2 FOR MOON                |
|                     | S22EORM  |                                            |
|                     | DELTACSM |                                            |
|                     | VSR*     | VAD                                        |
|                     |          | 7,1                                        |
|                     |          | RCVCSM                                     |
|                     | STORE    | CSMPOS                                     |
|                     | RVQ      |                                            |
|                     | SETLOC   | P22S                                       |
|                     | BANK     |                                            |
| S2231X13            | STORE    | S221X3                                     |
|                     | SSP      | AXT,2                                      |
|                     |          | S2                                         |
|                     | DEC      | 2                                          |
|                     | DEC      | 6                                          |
|                     | AXT,1    |                                            |
|                     | DEC      | 18                                         |
| S2231NXT            | VLOAD    | VXSC*                                      |
|                     |          | S221X3                                     |
|                     |          | S223X1 +6,2                                |
|                     | STORE    | S22UMRL +18D,1                             |
|                     | INCR,1   | TIX,2                                      |
|                     | DEC      | -6                                         |
|                     |          | S2231NXT                                   |
|                     | RVQ      |                                            |
| # Page 610<br>GETTF | LXC,1    | DLOAD*                                     |
|                     |          | S22LOC                                     |
|                     |          | # SET MPAC= TF                             |



July 12, 2016

1057

```

 0,1
S22FLGS RVQ
 SET SET # INTEGRATION FLAGS
 DIMOFLAG
 D6OR9FLG
 SET SET
 VINTFLAG
 STATEFLG
 CLEAR RVQ
 INTYPFLG

SUBROUTINE TO MODIFY ALT AND STORE LAT TO LAT+5 IN LANDLAT TO LANDLAT+5
PRIOR TO DISPLAY.

LLASRD DLOAD # ALT, LANDALT METERS B-29
 ALT
 STODL LANDALT
 LONG
 SR1
 STORE LANDLONG
 RVQ

SUBROUTINE TO MODIFY LANDALT AND STORE LANDALT TO LANDALT+5 IN LAT TO
LAT+5 AFTER LMK DATA LOADED BY ASTRONAUT.

LLASRDA DLOAD # ALT, LANDALT METERS B-29
 LANDALT
 STODL ALT
 LANDLONG
 SL1
 STORE LONG
 RVQ
 SETLOC P20S6
 BANK

9DWT06DW STQ SETPD
 9DWXX
 OD
 VLOAD PUSH # CLEAR WORKING AREA OF PUSHLIST
 HI6ZEROS # INCLUDING P
 PUSH PUSH # PD 18D
 SSP
 9DWJ # J=29 USE 2*29 FOR DP WORDS
 DEC 58
9DWI=J LXA,1 SXA,1 # SET I=J
 9DWJ
```

```

9DWEPCAL CALL 9DWI
Page 611

 ROWDOT
 # P VARIES 0-20 INSTEAD OF 20-0
 9DWP
 EMATRIX +40D,1
 9DWP
 SXA,1
 2
 9DWP
 SLOAD BHIZ # TEST I=0
 9DWI
 9DWTESTJ
 DSU # I=I-1
 9DWID
 STORE 9DWI
 DSU BHIZ # TEST I=26
 9DW26D
 9DWSETI2
 GOTO # NEXT E SUB P
9DWSETI2 SSP 9DWEPCAL
 GOTO # I=2
 9DWI
 DEC 4
 9DWEPCAL
9DWTESTJ SLOAD BHIZ # TEST J=0
 9DWJ
 9DWFFIG6
 DSU
 9DWID
 STORE 9DWJ # J=J-1
 DSU BHIZ # TEST J=26
 9DW26D
 9DWSETJ2
 GOTO
9DWSETJ2 SSP 9DWI=J
 GOTO # SET J=2
 9DWJ
 DEC 4
 9DWI=J
9DWFFIG6 CALL
 GRP2PC
 SSP VLOAD # START OF FIGURE 2.4-6
 9DWJ # J=29
 DEC 58
 HI6ZEROS

```

```

STORE 9DWP # P,N,I=0
AXT,1 SSP
DEC 108 # CLEAR W0 TO W54
 S1
 6
CLEARW54 STORE W +108D,1
TIX,1
Page 612
9DWI=JA LXA,1 CLEARW54
 SXA,1 # I=J
 9DWJ
 9DWI
CALL
 ROWDOT
LXA,1 BDSU*
 9DWP
 EMATRIX +40D,1
INCR,1 SXA,1 # -(P+1)
 2
 9DWP
LXC,1 XSU,1 # -(I+N)
 9DWI
 9DWN
BPL DLOAD # TEST WSQ LTE 0
 9DWAAA
 HI6ZEROS # W=0
GOTO
 9DWAAB
9DWAAA SQRT # W= SQRT(WSQ)
9DWAAB STORE W,1
STODL WORKW
 9DWJ # TEST J=0
BHIZ
 9DWEXITX # EXIT
TST2I=0 SLOAD BHIZ # TEST I=0
 9DWI
 9DWN=N+3
DSU
 9DWID
STORE 9DWI # I=I-1
DSU BHIZ # TEST I=26
 9DW26D
 9DWAAC
GOTO
 9DWNEXEP
9DWAAC SSP # I=2

```

```

 9DWI
 4
9DWNEXEP CALL ROWDOT
 LXA,1 BDSU* # (EP-ROWI*ROWJ)/W
 9DWP
 EMATRIX +40D,1
 DDV INCR,1 # P=P+1
 WORKW
 2
 SXA,1 LXC,1
 9DWP
Page 613
 9DWI
 XSU,1 BOV # -(I+N)
 9DWN
 SETWIN=0
 GOTO
 9DWSETWX
SETWIN=0 DLOAD # W(I+N)=0
 HI6ZEROS
9DWSETWX STORE W,1
 GOTO
 TST2I=0
9DWN=N+3 LXA,1 INCR,1 # N=N+3
 9DWN
 6
 SXA,1 SLOAD # J=J-1
 9DWN
 9DWJ
 DSU
 9DWID
 STORE 9DWJ
 DSU BHIZ # TEST J=26
 9DW26D
 SETJ=2A
 GOTO
 9DWI=JA
SETJ=2A SSP GOTO # J=2
 9DWJ
 4
 9DWI=JA
9DWEXITX CALL
 GRP2PC
 AXT,1 SSP # CLEAR W6,W7,W8 USED TEMP FOR EMATRIX
 DEC 54

```

|            |        |           |      |
|------------|--------|-----------|------|
|            |        | S1        |      |
|            |        | 6         |      |
|            | VLOAD  |           |      |
|            |        | HI6ZEROS  |      |
| 9DWEXXA    | STORE  | W +162D,1 |      |
|            | TIX,1  | GOTO      |      |
|            |        | 9DWEXXA   |      |
|            |        | 9DWXX     |      |
| ROWDOT     | SSP    | BOV       |      |
|            |        | XTMP1     |      |
|            | OCT    | 377       |      |
|            |        | +1        |      |
|            | LXC,1  | LXC,2     |      |
|            |        | 9DWI      |      |
|            |        | 9DWJ      |      |
|            | DLOAD  | PUSH      |      |
|            |        | HI6ZEROS  |      |
| # Page 614 |        |           |      |
| ROWDOT1    | DLOAD* | DMPR*     |      |
|            |        | W,1       |      |
|            |        | W,2       |      |
|            | DAD    | PUSH      |      |
|            | BOV    | INCR,1    |      |
|            |        | ROWDOT3   |      |
|            | DEC    | -6        |      |
|            | INCR,2 | SLOAD     |      |
|            | DEC    | -6        |      |
|            |        | XTMP1     |      |
|            | BHIZ   | SR1       |      |
|            |        | ROWDOT2   |      |
|            | STORE  | XTMP1     |      |
|            | GOTO   |           |      |
|            |        | ROWDOT1   |      |
| ROWDOT2    | DLOAD  |           |      |
|            | RVQ    |           |      |
| ROWDOT3    | CLRGO  |           |      |
|            |        | ORBWFLAG  |      |
|            |        | ROWDOT2   |      |
| WORKW      | =      | 0D        |      |
| XTMP1      | =      | 6D        |      |
| 9DWP       | =      | 8D        | # P  |
| 9DWI       | =      | 10D       | # I  |
| 9DWN       | =      | 12D       | # N  |
| 9DWJ       | =      | 14D       | # J  |
| 9DWXX      | =      | S22UOFF   |      |
| S22UMRL    | =      | BVECTOR   | # 18 |

|          |        |               |      |           |
|----------|--------|---------------|------|-----------|
| S22UUT   | =      | DELTAX        | # 18 |           |
| S223X1   | =      | 18D           | # 6  |           |
| S221X3   | =      | 24D           | # 6  |           |
| S22D     | =      | 30D           | # 2  |           |
| S22RHO   | =      | 32D           | # 2  |           |
| S22RL    | =      | W +156D       | # 6  |           |
| 9DW26D   | 2DEC   | 52 B-14       |      |           |
| 9DWID    | 2DEC   | 2 B-14        |      |           |
| SCTVAR   | 2DEC   | 1.0 E-6 B+18  |      |           |
| IMUVARR  | 2DEC   | 0.04 E-6 B+18 |      |           |
| DEC1B2   | 2DEC   | 1 B-2         |      |           |
| V06N49EE | VN     | 00649         |      |           |
| V06N89B  | VN     | 00689         |      |           |
| S22UOFF  | =      | LEMP0S        | # 6  | U SUB OFF |
|          | SETLOC | P20S2         |      |           |
|          | BANK   |               |      |           |

# Page 615

# Nothing on this page. --- RSB 2009.

# Page 616

# SUBROUTINE NAME: V89CALL

# MOD NO: 0

DATE: 8 FEB 1968

# MOD BY: DIGITAL DEVEL GROUP

LOG SECTION: P20-P25

#

# FUNCTIONAL DESCRIPTION:

#

# CALLED BY VERB 89 ENTER DURING P00. Prio 10 USED. CALCULATES AND  
 # DISPLAYS FINAL GIMBAL ANGLES TO POINT CSM +X AXIS OR PREFERRED AXIS  
 # (UNIT(Z)COS55 DEG + UNIT(X)SIN55 DEG) AT LM.

#

# 1. KEY IN V89 E ONLY IF IN PROG 00. IF NOT IN P00, OPERATOR ERROR AND  
 # EXIT R63, OTHERWISE CONTINUE.

#

# 2. IF IN P00, DO IMU STATUS CHECK (R02BOTH). IF IMU ON AND ITS  
 # ORIENTATION KNOWN TO CGC, CONTINUE.

#

# 3. FLASH DISPLAY V 04 N 06. R2 INDICATES WHICH SPACECRAFT AXIS IS TO  
 # BE POINTED AT LM. INITIAL CHOICE IS PREFERRED AXIS. (R2=1).  
 # ASTRONAUT CAN CHANGE TO (+X) AXIS (R2 NOT= 1) BY V22 E 2 E. CONTINUE  
 # AFTER KEYING IN PROCEED.

#

July 12, 2016

1063

```
4. SET PREFERRED ATTITUDE FLAG ACCORDING TO OPTION DESIRED. SET FLAG
FOR PREFERRED AXIS. RESET FLAG FOR X AXIS.
#
5. CURRENT TIME IS STORED AND R63COMP IS CALLED
#
R63COMP JOB:
#
UPDATE CSM AND LM STATE VECTORS USING CONIC EQUATIONS
#
CALCULATES BOTH PREFERRED AND X AXIS TRACKING ATT FROM CSM TO LM.
#
DESIRED GIMBAL ANGLES AS INDICATED BY PREFERRED ATTITUDE FLAG
ARE STORED FOR LATER R60CSM CALL.
#
6. FLASH DISPLAY V 06 N18 AND AWAIT RESPONSE.
#
7. RECYCLE: RETURN TO STEP 5.
TERMINATE: EXIT R63 ROUTINE
PROCEED: RESET 3AXISFLG AND CALL R60CSM FOR ATTITUDE MANEUVER.
#
CALLING SEQUENCE: V 89 E
#
SUBROUTINES CALLED: CHKP00H, R02BOTH, GOXDSPF, R63COMP, R60CSM
#
ALARMS 1. OPERATOR ERROR IF NOT IN P00
2. PROGRAM ALARM IF IMU IS OFF
3. PROGRAM ALARM IF IMU ORIENTATION IS UNKNOWN
#
Page 617
#
ERASABLE INITIALIZATION REQUIRED: NONE
#
DEBRIS: OPTION1, OPTION1+1, PRFTEXAT(PREF ATT FLAG), P21TIME, 3AXISFLG

DP1MIN 2DEC 6000

 EBANK= P21TIME
 BANK 34
 SETLOC P20S4
 BANK
 COUNT* $$/R63

V89CALL TC BANKCALL # IMU STATUS CHECK. RETURNS IF ORIENTATION
 CADR R02BOTH # KNOWN. ALARMS IF NOT.
 CAF THREE # ALLOW ASTRONAUT TO SELECT DESURED
 TS OPTION1 # TRACKING ATTITUDE AXIS
 CAF ONE
```

|            |        |            |                                            |
|------------|--------|------------|--------------------------------------------|
|            | TS     | OPTION1 +1 |                                            |
|            | CAF    | VB04N06    | # V 04 N 06                                |
|            | TC     | BANKCALL   |                                            |
|            | CADR   | GOFLASH    |                                            |
|            | TC     | ENDEXT     | # TERMINATE                                |
|            | TC     | +2         | # PROCEED                                  |
|            | TC     | -5         | # DATA IN. OPTION1 +1 = 1 FOR PREF AXIS    |
|            |        |            | # = 2 FOR X AXIS                           |
|            | CS     | OPTION1 +1 | # 1 FOR PREF AXIS. 2 FOR X AXIS.           |
|            | AD     | ONE        |                                            |
|            | EXTEND |            |                                            |
|            | BZF    | SETPAF     |                                            |
| RSTPAF     | TC     | DOWNFLAG   | # RESET PREF ATT FLAG FOR R63COMP          |
|            | ADRES  | RNGSCFLG   | # TO DO X AXIS. RESET BIT 10 FLAG 5        |
| V89RECL    | TC     | INTPRET    |                                            |
|            | RTB    | DAD        |                                            |
|            |        | LOADTIME   | # READ PRESENT TIME                        |
|            |        | DP1MIN     | # INTEGRATE TO 1 MIN FROM NOW              |
|            | STCALL | P21TIME    | # STORE TIME FOR CALL TO R63COMP. R63COMP  |
|            |        | R63COMP    | # LEAVES DESIRED GIM ANG IN THETAD, LOS IN |
|            | EXIT   |            | # POINTVSM, AND SELECTED AXIS IN SCAXIS.   |
|            | CAF    | VB06N18    | # V 06 N 18                                |
|            | TC     | BANKCALL   | # NOUN 18 REFERS TO THE DESIRED GIMBAL     |
|            | CADR   | GOFLASH    |                                            |
|            | TC     | ENDEXT     | # TERMINATE                                |
|            | TC     | +2         | # PROCEED                                  |
|            | TC     | V89RECL    | # RECYCLE                                  |
|            | TC     | DOWNFLAG   | # RESET 3 AXIS FLAG                        |
|            | ADRES  | 3AXISFLG   | # RESET BIT 6 FLAG 5                       |
| # Page 618 |        |            |                                            |
|            | TC     | BANKCALL   | # PERFORMS CSM MANEUVER TO ALIGN SELECTED  |
|            | CADR   | R60CSM     | # SPACECRAFT AXIS TO LOS.                  |
|            | TCF    | ENDEXT     |                                            |
|            |        |            |                                            |
| SETPAF     | TC     | UPFLAG     | # SET PREFERRED ATT FLAG FOR R63COMP       |
|            | ADRES  | RNGSCFLG   | # TO DO PREF AXIS. SET BIT 10 FLAG 5.      |
|            | TC     | V89RECL    |                                            |
|            |        |            |                                            |
| VB04N06    | VN     | 0406       |                                            |
| VB06N18    | VN     | 0618       |                                            |
|            |        |            |                                            |
| R63COMP    | EQUALS | R63        |                                            |

# Page 619

# PROGRAM NAME: P23 CISLUNAR MIDCOURSE NAVIGATION

# MOD NO:



```

MOD BY: TOM KNATT
#
FUNCTIONAL DESCRIPTION: DO MIDCOURSE NAVIGATION BY INCORPORATION OF
STAR/EARTH AND STAR/MOON OPTICAL MEASUREMENTS.
#
CALLING SEQUENCE: ASTRONAUT OPERATED
#
SUBROUTINES CALLED: R52,R53,R57,R60,ORBITAL INTEGRATION (INTEGRV)
INCORP1,INCORP2,LALOTORV,LUNLMKLD, AND DISPLAY INTERFACE ROUTINES.
#
NORMAL EXIT MODES: VIA R00
#
ALARMS: NONE
#
ABORT MODES: NONE
#
ERASABLE INITIALIZATION REQUIRED: PAD-LOADED ERASABLES, ORBWFLAG RESET,
REFSMFLG=0 IF IMU OFF AND REFSMFLG=1 IF IMU ONE
#
INPUTS BY USER REQUIRED: STAR NUMBER, LANDMARK LAT, LONG/2, ALT OR ID NUMB.
IF LANDMARK IS USED, NEAR OR FAR HORIZON IF HORIZON IS USED, AND
BODY TO BE MARKED ON (EARTH OR MOON). SEE GSOP CHAPT 4.
#
OUTPUT: UPDATED CMC STATE VECTOR. VECTOR FROM S/C TO HORIZON OR LANDMARK
IN POINTAXS. POINTAXS CAN BE USED TO GENERATE THIS VECTOR APART FROM
P23 IF DESIRED.
#
DEBRIS: NO USABLE DEBRIS IS GENERATED. RENDWFLG IS RESET FOR P20 UPON
COMPLETION OF P23. RUPTREGS AND ERASABLES USED BY DISPLAYS ARE DEBRIS.

P23 BANK 31
 SETLOC RT23
 BANK
 COUNT 31/S23
 EBANK= W
 TC DOWNFLAG
 ADRES RNDVZFLG

 TC 2PHSCHNG
 OCT 00004 # LEAVE GROUP 4
 OCT 00012 # ENTER GROUP 2
 CAF PRI013
 TS PHSPRDT2
 TC INTPRET
 SSP CLEAR
 MARKINDX

```

```

1
TARG2FLG # TARGET FLAG USED R52 AND R53
CLEAR SSP
TARG1FLG
STARIND
0
SSP CLEAR
BESTI
0
R57FLG # SET = DO NOT REPERFORM R57
CLEAR EXIT
V94FLG # SET = ALLOW V94
TC INTPRET

P23.00
Page 620

BON CALL
REFSMFLG
P23.05 # SET NOW AS INPUT, NORMALLY EXTERNAL CONT
R57 # WHEN ALIGNED, PERFORM MEASUREMENT
 # DO OPTICS CALIBRATION IF IMU NOT ALIGNED.
CALL
R53
GOTO
P23.60
P23.05 CLEAR EXIT
SAVECFLG # USED TO SAVE SPACE IN P23.65
P23.06 CAF V05N70
TC BANKCALL # IDENTIFICATION: STAR, HOR IDENT.
CADR GOFLASH
TC GOTOP00H # TERMINATE
TC P23.15
TC -5
P23.15 CA LANDMARK # IF C=2, LUNAFLAG=1. IF C=1, LUNAFLAG=0
EXTEND
BZF P23.151
CA HORIZON
EXTEND
BZF +2
TC R23.10 # OPERATOR DSKY ERROR
CA LANDMARK
TC P23.152
P23.151 CA HORIZON
EXTEND
BZF R23.10
P23.152 MASK BITS7-9 # IS C EQUAL TO 1 OR 2
AD NEG100
EXTEND
BZF P23.16

```

|            |        |          |                                           |
|------------|--------|----------|-------------------------------------------|
|            | AD     | NEG100   |                                           |
|            | EXTEND |          |                                           |
|            | BZF    | +2       |                                           |
|            | TC     | R23.10   |                                           |
|            | TC     | UPFLAG   |                                           |
|            | ADRES  | LUNAFLAG |                                           |
|            | TCF    | +3       |                                           |
| P23.16     | TC     | DOWNFLAG |                                           |
|            | ADRES  | LUNAFLAG |                                           |
|            | CA     | STARCODE | # IS STARCODE GREATER THAN OR             |
|            | EXTEND |          | # EQUAL TO 0 AND LESS THAN 37             |
|            | BZF    | P23.176  |                                           |
|            | EXTEND |          |                                           |
|            | BZMF   | R23.10   |                                           |
|            | AD     | NEG37    |                                           |
|            | EXTEND |          |                                           |
|            | BZMF   | +2       |                                           |
|            | TC     | R23.10   |                                           |
| # Page 621 |        |          |                                           |
|            | TC     | INTPRET  |                                           |
| P23.17     | SLOAD  | BZE      |                                           |
|            |        | STARCODE |                                           |
|            |        | P23.175  |                                           |
|            | PUSH   |          |                                           |
|            | SLOAD  | DMP      |                                           |
|            |        | SPSIX    |                                           |
|            | LXA,1  | SXA,1    |                                           |
|            |        | MPAC +1  |                                           |
|            |        | BESTI    | # BESTI = 6 X STAR NUMBER                 |
|            | CALL   |          |                                           |
|            |        | LOWMEMRY | # NEEDED TO RETRIEVE STAR VECTOR FROM LOW |
|            | STORE  | STARSAV2 | # STORE FOR R53,P23. US(IN P23)=STARSAV2  |
| P23.175    | EXIT   |          |                                           |
| P23.176    | CA     | HORIZON  |                                           |
|            | EXTEND |          |                                           |
|            | BZF    | P23.20   |                                           |
|            | MASK   | BITS4-6  |                                           |
|            | AD     | -OCT10   |                                           |
|            | EXTEND |          |                                           |
|            | BZF    | P23.18   |                                           |
|            | AD     | -OCT10   |                                           |
|            | EXTEND |          |                                           |
|            | BZF    | +2       |                                           |
|            | TC     | R23.10   |                                           |
|            | TC     | UPFLAG   |                                           |

|                       |        |          |                                   |
|-----------------------|--------|----------|-----------------------------------|
|                       | ADRES  | NORFHOR  |                                   |
|                       | TC     | P23.30   |                                   |
| P23.18                | TC     | DOWNFLAG |                                   |
|                       | ADRES  | NORFHOR  |                                   |
|                       | TC     | P23.30   |                                   |
| P23.20                | TC     | INTPRET  |                                   |
|                       | CALL   |          |                                   |
|                       |        | P22SUBRB |                                   |
|                       | EXIT   |          |                                   |
| P23.30                | TC     | INTPRET  |                                   |
|                       | SLOAD  | BZE      |                                   |
|                       |        | STARCODE |                                   |
|                       |        | LDPLANET |                                   |
| P23.31                | BON    | EXIT     |                                   |
|                       |        | SAVECFLG |                                   |
|                       |        | P23.85   |                                   |
|                       | CAF    | V50N25P  |                                   |
|                       | TC     | BANKCALL |                                   |
|                       | CADR   | GOPERF1  | # GOPERF1 BLANKS OUT R2 AND R3    |
|                       | TC     | GOTOP00H |                                   |
|                       | TC     | V94ENTER | # PROCEED. AUTOCONTROL CMC        |
| P23.55                | TC     | INTPRET  |                                   |
|                       | GOTO   |          |                                   |
|                       |        | P23.56   |                                   |
| # Page 622            |        |          |                                   |
| # VERB 94 BEGINS HERE |        |          |                                   |
| V94ENTER              | TC     | INTPRET  |                                   |
|                       | RTB    |          |                                   |
|                       |        | LOADTIME | # READ CLOCK                      |
|                       | STCALL | MARKTIME |                                   |
|                       |        | POINTAXS | # RETURN LOS IN RCLL AND MPAC     |
|                       | MXV    | UNIT     |                                   |
|                       |        | REFSMMAT |                                   |
|                       | STOVL  | POINTVSM |                                   |
|                       |        | JCAXIS   |                                   |
|                       | STORE  | SCAXIS   |                                   |
|                       | EXIT   |          |                                   |
|                       | TC     | DOWNFLAG | # CLEAR AND GO TO VECPOINT IN R60 |
|                       | ADRES  | 3AXISFLG | # BIT 6 FLAG 5                    |
|                       | CAF    | R60ADRS  |                                   |
|                       | TS     | TEMPFLSH |                                   |
|                       | TC     | PHASCHNG |                                   |
|                       | OCT    | 00012    |                                   |
| R60CALL               | TC     | BANKCALL |                                   |
|                       | CADR   | R60CSM   |                                   |

July 12, 2016

1069

|            |        |             |                        |
|------------|--------|-------------|------------------------|
|            | TC     | PHASCHNG    |                        |
|            | OCT    | 04022       |                        |
|            | TC     | INTPRET     |                        |
|            | BON    |             |                        |
|            |        | R57FLAG     |                        |
|            |        | P23.57      | # DO NOT REPERFORM R57 |
| P23.56     | CALL   |             |                        |
|            |        | R57         |                        |
| P23.57     | SET    | SET         |                        |
|            |        | V94FLAG     |                        |
|            |        | R57FLAG     |                        |
|            | CALL   |             |                        |
|            |        | R52         |                        |
|            | CLEAR  | CLEAR       |                        |
|            |        | V94FLAG     |                        |
|            |        | R57FLAG     |                        |
| P23.60     | EXIT   |             |                        |
|            | INHINT |             |                        |
|            | CA     | MARKSTAT    |                        |
|            | MASK   | LOW10       |                        |
|            | TS     | MARKDATA    |                        |
|            | EXTEND |             |                        |
|            | INDEX  | MARKDATA    |                        |
|            | DCA    | 0           |                        |
|            | DXCH   | MARKTIME    |                        |
|            | INDEX  | MARKDATA    |                        |
|            | CA     | 5           |                        |
|            | XCH    | TRUNION     |                        |
|            | RELINT |             |                        |
|            | TC     | INTPRET     |                        |
| # Page 623 |        |             |                        |
|            | LXC,1  | VLOAD*      |                        |
|            |        | MARKDATA    |                        |
|            |        | 1,1         |                        |
|            | STODL* | MARKDOWN +1 |                        |
|            |        | 0,1         |                        |
|            | STORE  | MARKDOWN    |                        |
|            | EXIT   |             |                        |
|            | CAF    | V05N71      |                        |
|            | TC     | BANKCALL    |                        |
|            | CADR   | GOFLASH     |                        |
|            | TC     | GOTOP00H    | # TERMINATE            |
|            | TC     | P23.65      | # STORE DATA           |
|            | TC     | -5          | # REDISPLAY            |
| P23.65     | TC     | INTPRET     |                        |
|            | SET    | EXIT        |                        |

```

 SAVECFLG
 P23.15
P23.85 TC CLEAR CALL
 RENDWFLG
 POINTAXS
 GOTO
 R23.55

WE BEGIN CALCULATIONS HERE
POINTAXIS SUBROUTINE

POINTAXS STQ
R23.05 BON POINTEX
 DLOAD
 ORBWFLAG
 R23.1
 WMIDPOS
 STCALL 0
 INITIALW # INITIALIZE W-MATRIX FIRST PASS IN P23
R23.1 CALL SETINTG # SETUP FOR CSM INTEGRATION
 SET
 BOF ORBWFLAG
 R23.2
 DIMOFLAG
R23.2 SET CALL
 ORBWFLAG
 INTEGRV # INTEGRATE CSM STATE VEC. TO MARKTIME
 EXIT
 TC PHASCHNG
 OCT 04022
 TC INTPRET
 CALL
 RECT.1 # PICKUP CSM STATE VECTOR FROM PERM
 BOFF
 ZMEASURE # IN SPHERE OF INFLUENCE OF PRIMARY BODY
 R23.3
Page 624
 DLOAD CALL
 MARKTIME
 LUNPOS
 BON VCOMP
 CMOONFLG
 +1
 VAD
 RZC

```

July 12, 2016

1071

```
R23.3 STORE RZC
 SLOAD BHIZ
 LANDMARK # IF LANDMARK = 0, USE HORIZ SUBR
 R23.4
 SET
 ERADFLAG
 DLOAD CALL
 MARKTIME
 LALOTORV
 GOTO
 R23.5
R23.4 CALL
 HORIZ
R23.5 VSU SETPD
 RZC
 0
 GOTO
 POINTEX
Page 625
R23.55 UNIT PUSH # RCLL IS IN MPAC
 VLOAD
 34D # RCLL * RCLL
 STOVL 30D # PUSH 30-31 =RCLL*RCLL 32-33=ABVAL RCLL
 VZC
 VXSC VSR
 ONE/C
 15D
 VAD # PUSH UP RCLL(UNIT)
 UNIT
 STOVL UCLSTAR
 VZC
 VSR2 VSU
 VES0
 VXSC VSR
 ONE/C
 13D
 VAD UNIT
 US
 STORE USSTAR
 DOT SL1
 UCLSTAR
 PUSH VLOAD # PD 0,1 = USSTAR(DOT)UCLSTAR
 UCLSTAR
 VXSC VCOMP
 VSL1 VAD
 USSTAR
```

```

UNIT
STOVL BVECTOR # USSTAR - COSQ(UCLSTAR)
 ZEROVECS
STORE BVECTOR +6
STODL BVECTOR +12D
 0
ACOS DCOMP
PUSH DLOAD
 ZEROVECS

EXIT
CA VARSUBL # PUT FIXED INTO ERASABLE FOR MSU
TS L # INSTRUCTION COMING UP
CA TRUNION # REQUIRED TO CHANGE 2'S COMPLEMENT
EXTEND
MSU L # TRUNION (2'S)-00000 CONVERTS TRUNION TO
TS MPAC # 1'S. VARSUBL=00000.
TC INTPRET
PUSH SLOAD # PUSH IS DP. WHEN BDSU IS EXECUTED, 2ND
 TRUNBIAS # HALF OF PUSHLIST IS GUARANTEED ZERO FROM
 DLOAD ZEROVECS ABOVE
BDSU
SR3 DAD
DAD DMP

Page 626

 TRUN19
 32D
DMP SL3
 PI/4.0
BOFF SL2
 CMOONFLG
 R23.51
R23.51 STODL DELTAQ
 30D # RCLL * RCLL
DMP RTB
 TRUNVAR
 TPMODE
TAD
 VARSUBL
STORE VARIANCE
CLEAR CALL
 DMENFLG
 INCORP1
CALL
 GRP2PC
VLOAD ABVAL
 DELTAX +6
BOF SR2 # DISPLAY IS 2-27 IF IN LUNAR SPHERE.

```



July 12, 2016

1073

|            |       |            |                             |
|------------|-------|------------|-----------------------------|
|            |       | CMOONFLG   |                             |
|            |       | R23.52     |                             |
| R23.52     | STOVL | N49DISP +2 |                             |
|            |       | DELTAX     |                             |
|            | ABVAL |            |                             |
|            | BOF   | SR2        |                             |
|            |       | CMOONFLG   |                             |
|            |       | R23.53     |                             |
| R23.53     | STORE | N49DISP    |                             |
|            | EXIT  |            |                             |
| R23.6      | CAF   | V6N49      |                             |
|            | TC    | BANKCALL   |                             |
|            | CADR  | GOFLASHR   |                             |
|            | TC    | GOTOP00H   |                             |
|            | TC    | R23.7      | # INCORPORATE DATA          |
|            | TC    | GOTOP00H   |                             |
|            | CAF   | BIT3       | # BLAN OUT R3               |
|            | TC    | BLANKET    |                             |
|            | TC    | PHASCHNG   |                             |
|            | OCT   | 00012      |                             |
|            | TC    | ENDOFJOB   |                             |
| R23.7      | TC    | INTPRET    |                             |
| R23.8      | SET   | CALL       |                             |
|            |       | VEHUPFLG   |                             |
|            |       | INCORP2    |                             |
|            | EXIT  |            |                             |
| R23.END    | TC    | GOTOP00H   |                             |
| # Page 622 |       |            |                             |
| R23.10     | TC    | FALTON     |                             |
|            | TC    | P23.06     |                             |
| HORIZ      | STQ   | SETPD      |                             |
|            |       | SRRETURN   |                             |
|            |       | 0          |                             |
|            | DLOAD | PDDL       | # PUSH 0-1 = -AYO SCALED B0 |
|            |       | -AYO       |                             |
|            |       | AX0        |                             |
|            | PDDL  | PDVL       | # PUSH 2-3 = +AX SCALED B0  |
|            |       | DPPOS MAX  |                             |
|            |       | US         |                             |
|            | VXV   | UNIT       |                             |
|            |       | RZC        |                             |
|            | STOVL | UBAR2      |                             |
|            | VXV   | UNIT       | # PUSH UP                   |
|            |       | UBAR2      |                             |
|            | STOVL | UBAR0      |                             |

```

 UBAR2
VXV UNIT
 UBAR0
STORE UBAR1
BON DOT
 LUNAFLAG
 HORIZ.6
 0 # UBAR1 DOT UZ
STCALL ALPHAV +4
 GETERAD
DAD PDDL # MPAC HAS RADIUS OF FISCHER ELLIPSOID
 HORIZALT
 AEARTH # PUSH 0-1 = BH SCALED B29
DAD PUSH # PUSH 2-3 = AH B29
 HORIZALT
HORIZ.1 VLOAD MXV
 RZC # B29
 UBAR0 # B1
VSL1 PDVL # PUSH 4-9 = RH(XH,YH,ZH) B29
 US
MXV VSL1
 UBAR0
PDDL # PUSH 10-15 = USH B1
 2 # AH
STODL 34D
 4 # XH
CALL
 DIVIDE
SR* DMP
 8D,1 # NOW SCALED B9
 MPAC
STODL 30D
 0
Page 628
STODL 34D
 6 # YH
CALL
 DIVIDE
SR* DMP
 8D,1 # B9
 MPAC # B18
DAD PUSH # PUSH 16-17 =A SCALED B18
 30D
DSU SQRT
 1.0B18
PDDL # PUSH 18-19 SQRT(A-1) B9

```

```

16D
STODL 34D
4 # XH
CALL
DIVIDE
SR* PDDL
17D,1 # PUSH 20-21 = XH/A B29
6 # YH
CALL
DIVIDE
SR* PDDL
17D,1 # PUSH 22-23 = YH/A B29
16D # A
STODL 34D
18D # SQRT(A-1)
CALL
DIVIDE
SR* 8D,1
STODL 28D
0 # BH
STODL 34D
2 # AH
CALL
DIVIDE
SR* DMP # AH/BH SCALED B1
0,1
28D # SQRT(A-1)/A
DMP SL1
6 # YH
PDDL
2 # AH
STODL 34D
0
CALL
DIVIDE
SR* DMP # BH/AH SCALED B1
0,1
28D # SQRT (A-1)/A
DMP SL1
4 # XH
PDDL DAD
20D # XH/A
24D # ALPHA
PDDL DSU

```

|            |       |          |                                  |
|------------|-------|----------|----------------------------------|
|            |       | 22D      | # YH/A                           |
|            |       | 26D      | # BETA                           |
|            | PUSH  | SETPD    |                                  |
|            |       | 16D      |                                  |
|            | DLOAD | DSU      |                                  |
|            |       | 20D      | # XH/A                           |
|            |       | 24D      | # ALPHA                          |
|            | PDDL  | DAD      |                                  |
|            |       | 22D      | # YH/A                           |
|            |       | 26D      | # BETA                           |
|            | PDDL  | PUSH     |                                  |
|            |       | ZEROVECS |                                  |
|            | STOVL | 32D      | # ZERO THIRD COMP. OF T-0 VECTOR |
|            |       | 28D      |                                  |
|            | VSU   | UNIT     |                                  |
|            |       | 4        | # RH VECTOR                      |
|            | DOT   | PDVL     | # PUSH 22-23 A-SUB-ZERO          |
|            |       | 10D      | # USH VECTOR                     |
|            |       | 16D      | # T1 VECTOR                      |
|            | VSU   | UNIT     |                                  |
|            |       | 4        | # RH VECTOR                      |
|            | DOT   | PUSH     | # PUSH 24-25 A-SUB-ONE           |
|            |       | 10D      |                                  |
|            | BDSU  | BMN      |                                  |
|            |       | 22D      | # A-SUB-ZERO                     |
|            |       | HORIZ.3  |                                  |
|            | BON   |          |                                  |
|            |       | NORFHOR  |                                  |
|            |       | HORIZ.4  |                                  |
| HORIZ.2    | VLOAD | GOTO     |                                  |
|            |       | 28D      | # T-0 VECTOR                     |
|            |       | HORIZ.5  |                                  |
| HORIZ.3    | BON   | GOTO     |                                  |
|            |       | NORFHOR  |                                  |
|            |       | HORIZ.2  |                                  |
|            |       | HORIZ.4  |                                  |
| HORIZ.4    | VLOAD |          |                                  |
|            |       | 16D      | # T1 VECTOR                      |
| HORIZ.5    | VXM   | VSL1     |                                  |
|            |       | UBARO    |                                  |
|            | GOTO  |          |                                  |
|            |       | SRRETURN |                                  |
| HORIZ.6    | DLOAD | PUSH     |                                  |
| # Page 630 |       |          |                                  |
|            |       | RADMOON  |                                  |
|            | PUSH  | GOTO     |                                  |

July 12, 2016

1077

|        |       |                      |                                           |
|--------|-------|----------------------|-------------------------------------------|
|        |       | HORIZ.1              |                                           |
| DIVIDE | NORM  | SR1                  |                                           |
|        |       | X1                   |                                           |
|        | STODL | 36D                  |                                           |
|        |       | 34D                  |                                           |
|        | NORM  | BDDV                 |                                           |
|        |       | S1                   |                                           |
|        |       | 36D                  |                                           |
|        | XSU,1 | RVQ                  |                                           |
|        |       | S1                   |                                           |
| RECT.1 | BOFF  | AXT,2                | # SR TO SET ZMEASURE = 0 IF MEASUREMENT   |
|        |       | CMOONFLG             | # PLANET AND PRIMARY PLANET ARE THE SAME. |
|        |       | RECT.3               | # OTHERWISE = 1                           |
|        | DEC   | -2                   |                                           |
|        | BOFF  |                      | # VEC. AND SCALE B29 AND B7               |
|        |       | LUNAFLAG             |                                           |
|        |       | RECT.4               |                                           |
| RECT.2 | CLEAR | GOTO                 |                                           |
|        |       | ZMEASURE             |                                           |
|        |       | RECT.5               |                                           |
| RECT.3 | AXT,2 | BOFF                 |                                           |
|        |       | 0                    |                                           |
|        |       | LUNAFLAG             |                                           |
|        |       | RECT.2               |                                           |
| RECT.4 | SET   |                      |                                           |
|        |       | ZMEASURE             |                                           |
| RECT.5 | VLOAD | VSR7                 |                                           |
|        |       | DELTACSM             | # SCALED B22 OR B18                       |
|        | VSR*  | VAD                  |                                           |
|        |       | 0,2                  |                                           |
|        |       | RCVCSM               | # SCALED B29 OR B27                       |
|        | VSR*  |                      |                                           |
|        |       | 0,2                  |                                           |
|        | STOVL | RZC                  | # NOW SCALED B29                          |
|        |       | NUVCSM               | # SCALED B3 OR B-1                        |
|        | VSR4  | VSR*                 |                                           |
|        |       | 0,2                  |                                           |
|        | VAD   | VSR*                 |                                           |
|        |       | VCVCSM               | # SCALED B7 OR B5                         |
|        |       | 0,2                  |                                           |
|        | STORE | VZC                  | # NOW SCALED B7                           |
|        | RVQ   |                      |                                           |
| ONE/C  | 2DEC* | .333564049 E-6 B+21* |                                           |
| AEARTH | 2DEC  | 6378166 B-29         | # A AXIS OF EARTH (METERS B-29)           |

```

RADMOON 2DEC 1738090 B-29 # RADIUS MOON IN METERS

Page 631

TRUN19 OCT 01604
TRUN19A OCT 00000
1.0B18 2DEC 1.0 B-18

VARSUBL DEC 0
VARSUBL3 2DEC* 3.4299040 E+6 B-26*

TRUNVAR 2DEC 2.5 E-9 B+18

V6N49 VN 0649
V05N70 VN 0570
V05N71 VN 0571
OCT00077 OCT 00077
V50N25P OCT 00202
SPSIX OCT 00006
JCAXIS 2DEC .2688190620 # 1/2(SIN 32.523 DEG) TRACK AXIS

2DEC 0

2DEC .4215878460 # 1/2(COS 32.523 DEG)

R60ADRS CADR R60CALL +3
NEG37 DEC -37
BITS7-9 OCT 700
BITS4-6 OCT 70
SETLOC RT53
BANK
LOWMEMRY VLOAD* RVQ
 CATLOG,1
BANK 37
SETLOC P23S1
BANK
LDPLANET EXIT
CAF VNPLAN23 # KEEP THIS OPEN SUBROUTINE IN EBANK=5
TC BANKCALL # BECAUSE STAR IS EBANK=5
CADR GOFLASH # LDPLANET ALLOWS VECTOR TO PLANET TO BE
TC GOTOP00H # STORED IN STARS2 IF STORED STARS ARE
TC +2 # NOT VISIBLE
TC -5
TC INTPRET
VLOAD
 STARS2

```

July 12, 2016

1079

|            |        |            |                                        |
|------------|--------|------------|----------------------------------------|
|            | VXSC   | UNIT       |                                        |
|            |        | 1/SQR3     |                                        |
|            | STORE  | STARSAV2   |                                        |
|            | GOTO   |            |                                        |
|            |        | P23.31     |                                        |
| VNPLAN23   | VN     | 0688       |                                        |
|            | BLOCK  | 02         |                                        |
| GOTOV56    | EXTEND |            | # P20 TERMINATES BY GOTOV56 INSTEAD OF |
| # Page 632 |        |            |                                        |
|            | DCA    | VB56CADR   | #                                      |
|            | TCF    | SUPDXCHZ   | GOTOP00H                               |
| VB56CADR   | EBANK= | WHOCARES   |                                        |
|            | 2CADR  | TRACKTRM   |                                        |
|            | SETLOC | FFTAG2     |                                        |
|            | BANK   |            |                                        |
|            | COUNT* | \$\$/P20   |                                        |
|            | BANK   | 40         |                                        |
|            | SETLOC | ENDPINS1   |                                        |
|            | BANK   |            |                                        |
|            | COUNT* | \$\$/EXTVB |                                        |
| V67CALL    | TC     | INTPRET    |                                        |
|            | CALL   |            |                                        |
|            |        | V67WW      |                                        |
|            | EXIT   |            |                                        |
| V06N99DS   | CAF    | V06N99A    |                                        |
|            | TC     | BANKCALL   |                                        |
|            | CADR   | GOXDSPF    |                                        |
|            | TCF    | ENDEXT     |                                        |
|            | TC     | V06N9933   |                                        |
|            | TC     | V06N99DS   |                                        |
| V06N9933   | TC     | INTPRET    |                                        |
|            | SLOAD  | BHIZ       | # IF R3 OF V67 = 0 EXIT                |
|            |        | WWOPT      |                                        |
|            |        | +3         |                                        |
|            | GOTO   |            |                                        |
|            |        | V6N99INP   |                                        |
|            | EXIT   |            |                                        |
|            | TCF    | ENDEXT     |                                        |
| V6N99INP   | LXA,1  | LXA,2      |                                        |
|            |        | WWPOS      |                                        |
|            |        | WWVEL      |                                        |
|            | SLOAD  | DSU        |                                        |
|            |        | WWOPT      |                                        |
|            |        | V67DEC2    |                                        |
|            | BHIZ   | BPL        |                                        |

```

 V67WORB
 V67WMID
 SXA,1 SXA,2
 WRENDPOS
 WRENDVEL
 GOTO
 V67EXITX
V67WORB SXA,1 SXA,2
 WORBPOS
 WORBVEL
 GOTO
 V67EXITX
V67WMID SXA,1 SXA,2
Page 633
 WMIDPOS
 WMIDVEL
V67EXITX CLEAR CLEAR
 ORBWFLAG
 RENDWFLG
 EXIT
 TCF ENDEXT
V67WW STQ BOV
 S2
 +1
 CALL
 INTSTALL
 SSP DLOAD
 S1
 DEC 6
 ZEROVECS
 STORE WWPOS
 STORE WWVEL
 STORE WWOPT
 AXT,1
 DEC 36
 VLOAD* VSQ
 W +36D,1
 DAD
 WWPOS
 STORE WWPOS
 VLOAD* VSQ
 W +90D,1
 DAD
 WWVEL
 STORE WWVEL
 TIX,1 SQRT

```



```

 NXPOSVEL
 WWVEL
 WWPOS
 SQRT
 STORE WWPOS
 BOV GOTO
 +2
 V67XXX
 DLOAD
 DPPOS MAX
 STORE WWPOS
 STORE WWVEL
 DLOAD DSU
 WWPOS
 FT99999
 BMN DLOAD
 +3
 FT99999
Page 634
 STORE WWPOS
 LXA,1 SXA,1
 S2
 QPRET
 EXIT
 TC POSTJUMP
 CADR INTWAKE
 = RANGE
 = RRATE
 = RTHETA
 V06N99A VN 0699
 FT99999 2DEC 30479 B-19

 V67DEC2 2DEC 2 B-14

 SBANK= LOWSUPER

```

This code is written to file `src/P20-P25.s`.

**B.68 P30 P37**

```

1082 <src/P30-P37.s 1082>≡
Copyright: Public domain.
Filename: P30_P37.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 614-617
Mod history: 2009-05-17 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
2009-06-05 RSB Removed 4 lines of code that shouldn't
have survived from Luminary 131.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 614
PROGRAM DESCRIPTION P30 DATE 3-6-67
#
MOD.1 BY RAMA AIYAWAR
#
FUNCTIONAL DESCRIPTIONS
ACCEPT ASTRONAUT INPUTS OF TIG,DELV(LV)
CALL IMU STATUS CHECK ROUTINE (R02)
DISPLAY TIME TO GO, APOGEE, PERIGEE, DELV(MAG), MGA AT IGN
REQUEST BURN PROGRAM
#
CALLING SEQUENCE VIA JOB FROM V37
#

```

```

EXIT VIA V37 CALL OR TO GOTOP00H (V34E)
#
SUBROUTINE CALLS -- FLAGUP, PHASCHNG, BANKCALL, ENDOFJOB, GOFLASH, GOFLASHR
GOPERF3R, INTPRET, BLANKET, GOTOP00H, R02BOTH, S30.1,
TIG/N35, MIDGIM, DISPMGA
#
ERASABLE INITIALIZATION -- STATE VECTOR
#
OUTPUT -- RINIT, VINIT, +MGA, VTIG, RTIG, DELVSIN, DELVSAB, DELVSLV, HAPO,
HPER, TTOGO
#
DEBRIS -- A, L, MPAC, PUSHLIST

 BANK 32
 SETLOC P30S
 BANK
 EBANK= +MGA
 COUNT* $$/P30
P30 TC UPFLAG # SET UPDATE FLAG
 ADRES UPDATFLG
 TC UPFLAG # SET TRACK FLAG
 ADRES TRACKFLG

P30N33 CAF V06N33 # T OF IGN
 TC VNP00H # RETURN ON PROCEED, POOH ON TERMINATE

 CAF V06N81 # DISPLAY DELTA V (LV)
 TC VNP00H # REDISPLAY ON RECYCLE

 TC DOWNFLAG # RESET UPDATE FLAG
 ADRES UPDATFLG
 TC INTPRET
 CALL
 S30.1
 SET EXIT
 UPDATFLG
PARAM30 CAF V06N42 # DISPLAY APOGEE,PERIGEE,DELTA V
 TC VNP00H

Page 615

 TC INTPRET
 SETGO
 XDELVFLG # FOR P40'S: EXTERNAL DELTA-V GUIDANCE.
 REVN1645 # TRKMKCNT, T60, +MGA DISPLAY

V06N33 VN 0633

```

V06N42                      VN                      0642

```
Page 616
PROGRAM DESCRIPTION S30.1 DATE 9NOV66
MOD NO 1 LOG SECTION P30,P37
MOD BY RAMA AIYAWAR **
#
FUNCTIONAL DESCRIPTION
BASED ON STORED TARGET PARAMETERS (R OF IGNITION (RTIG), V OF
IGNITION (VTIG), TIME OF IGNITION (TIG)), COMPUTE PERIGEE ALTITUDE
APOGEE ALTITUDE AND DELTAV REQUIRED (DELVSIN).
#
CALLING SEQUENCE
L CALL
L+1 s30.1
#
NORMAL EXIT MODE
AT L+2 OR CALLING SEQUENCE (GOTO L+2)
#
SUBROUTINES CALLED
LEMPREC
PERIAPO
#
ALARM OR ABORT EXIT MODES
NONE
#
ERASABLE INITIALIZATION REQUIRED
TIG TIME OF IGNITION DP B28CS
DELVSLV SPECIFIED DELTA-V IN LOCAL VERT.
COORDS. OF ACTIVE VEHICLE AT
TIME OF IGNITION VECTOR B+7 METERS/CS
#
OUTPUT
RTIG POSITION AT TIG VECTOR B+29 METERS
VTIG VELOCITY AT TIG VECTOR B+29 METERS/CS
PDL 4D APOGEE ALTITUDE DP B+29 M, B+27 METERS.
HAPO APOGEE ALTITUDE DP B+29 METERS
PDL 8D PERIGEE ALTITUDE DP B+29 M, B+27 METERS.
HPER PERIGEE ALTITUDE DP B+29 METERS
DELVSIN SPECIFIED DELTA-V IN INTERTIAL
COORD. OF ACTIVE VEHICLE AT
TIME OF IGNITION VECTOR B+7 METERS/CS
DELVSAB MAG. OF DELVSIN VECTOR B+7 METERS/CS
#
DEBRIS QTEMP TEMP.ERASABLE
QPRET, MPAC
```

July 12, 2016

1085

```
PUSHLIST

 SETLOC P30S1
 BANK

 COUNT* $$/S30S

S30.1 STQ DLOAD
 QTEMP
 TIG # TIME IGNITION SCALED AT 2(+28)CS
 STCALL TDEC1
 LEMPREC # ENCKE ROUTINE FOR LEM

 VLOAD SXA,2

Page 617

 RATT
 RTX2
 STORE RTIG # RADIUS VECTOR AT IGNITION TIME
 UNIT VCOMP
 STOVL DELVSIN # ZRF/LV IN DELVSIN SCALED AT 2
 VATT # VELOCITY VECTOR AT TIG, SCALED 2(7) M/CS
 STORE VTIG
 VXV UNIT
 RTX1
 SETPD SXA,1
 0
 RTX1
 PUSH VXV # YRF/LV PDL 0 SCALED AT 2
 DELVSIN
 VSL1 PDVL
 PDVL PDVL # YRF/LV PDL 6 SCALED AT 2
 DELVSIN # ZRF/LV PDL 12D SCALED AT 2
 DELVSLV
 VXM VSL1
 0
 STORE DELVSIN # DELTAV IN INERT. COOR. SCALED TO B+7M/CS
 ABVAL
 STOVL DELVSAB # DELTA V MAG.
 RTIG # (FOR PERIAPO)
 PDVL VAD # VREQUIRED = VTIG + DELVSIN (FOR PERIAPO)
 VTIG
 DELVSIN
 CALL
 PERIAPO1
 CALL
 SHIFTR1 # RESCALE IF NEEDED
```

1086

July 12, 2016

```
CALL # LIMIT DISPLAY TO 9999.9 N. MI.
 MAXCHK
STODL HPER # PERIGEE ALT 2(29) METERS FOR DISPLAY
 4D
CALL
 SHIFTR1 # RESCALE IF NEEDED
CALL # LIMIT DISPLAY TO 9999.9 N. MI.
 MAXCHK
STCALL HAPO # APOGEE ALT 2(29) METERS FOR DISPLAY
 QTEMP
```

This code is written to file `src/P30--P37.s`.

**B.69 P30-P37**

```

1087 <src/P30-P37.s 1087>≡
 # Copyright: Public domain.
 # Filename: P30-P37.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 635-648
 # Mod history: 2009-05-10 RSB Adapted from the Colossus249/ file
 # of the same name, using Comanche055 page
 # images.
 # 2009-05-20 RSB Corrected BDV -> BOV.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 635

 BANK 32

 SETLOC P30S1
 BANK

 EBANK= +MGA

 COUNT 35/P34

DISPMGA STQ EXIT # USED IN P30

```

|         |        |                   |                                        |
|---------|--------|-------------------|----------------------------------------|
|         | TC     | RGEXIT<br>COMPTGO |                                        |
| DISP45  | CAF    | V16N45            |                                        |
|         | TC     | BANKCALL          |                                        |
|         | CADR   | GOFLASHR          |                                        |
|         | TC     | GOTOPOOH          |                                        |
|         | TC     | END45             |                                        |
|         | TC     | DISP45            |                                        |
| P30PHSI | TC     | PHASCHNG          |                                        |
|         | OCT    | 14                |                                        |
|         | TCR    | ENDOFJOB          |                                        |
| END45   | TC     | INTPRET           |                                        |
|         | CLEAR  | GOTO              |                                        |
|         |        | TIMRFLAG          |                                        |
|         |        | RGEXIT            |                                        |
| COMPTGO | EXTEND |                   | # USED TO COMPUTE TTOGO                |
|         | QXCH   | PHSPRDT6          | # ** GROUP 6 TEMPORARY USED, BEWARE ** |
|         | TC     | UPFLAG            | # SET TIMRFLAG                         |
|         | ADRES  | TIMRFLAG          | # BIT 11 FLAG 7                        |
|         | CAF    | ZERO              |                                        |
|         | TS     | NVWORD1           |                                        |
|         | CAF    | ONE               |                                        |
|         | TC     | WAITLIST          |                                        |
|         | EBANK= | TIG               |                                        |
|         | 2CADR  | CLOKTASK          |                                        |
|         | TC     | 2PHSCHNG          |                                        |
|         | OCT    | 40036             | # 6.3SPOT FOR CLOKTASK                 |
|         | OCT    | 05024             | # GROUP 4 CONTINUES HERE               |
|         | OCT    | 13000             |                                        |
|         | TC     | PHSPRDT6          |                                        |

# Page 636

# PROGRAM DESCRIPTION P30           DATE 3-6-67

# MOD. I BY S. ZELDIN: TO ADD P31 AND ADAPT P30 FOR P31 USE.   22DEC67

#

# FUNCTIONAL DESCRIPTION

#       P30       (EXTERNAL DELTA-V TARGETING PROGRAM)

#                ACCEPTS ASTRONAUT INPUTS OF TIG,DELV(LV) AND COMPUTES, FOR DISPLAY,

#                APOGEE, PERIGEE, DELV(MAG), MGA ASSOCIATED WITH DESIRED MANEUVER.

#       P31       (GENERAL LAMBERT AIMPOINT GUIDANCE)

#                A GROUND RULE FOR P31 IS THE ANGLE BETWEEN THE TARGET VECTOR AND



```

POSITION VECTOR AT TIG IS NOT 165-195 DEGREES APART
BASED ON STORED INPUT OF OFFSET TARGET(B+29) AND DELTA T TRANS, AND
ASTRONAUT ENTRY OF TIG, P31 COMPUTES REQUIRED VELOCITY FOR MANEUVER
AND, FOR DISPLAY, APOGEE, PERIGEE, DELV(7AG), +MGA ASSOCIATED WITH
DESIRED MANEUVER.
#
THE FOLLOWING SUBROUTINES ARE USED IN P30 AND P31
S30.1 (P30 ONLY)
S31.1 (P31 ONLY)
P30/P31 -- DISPLAYS TIG
CNTUP30 -- DISPLAYS DELV(LV)
PARAM30 -- DISPLAYS APOGEE, PERIGEE, DELV(MAG), MGA, TIME FROM TIG,
MARKS SINCE LAST THRUSTING MANEUVER
#
CALLING SEQUENCE VIA JOB FROM V37
#
EXIT VIA V37 OR GOTOP00H
#
OUTPUT FOR POWERED FLIGHT
VTIG X
RTIG X SEE S30.1
DELVSIN X
VGDISP
RTARG X
TPASS4 X SEE S31.1
X
#
COUNT 35/P30
#
P30 TC P30/P31
TC CNTNUP30
TC DOWNFLAG # RESET UPDATFLG
ADRES UPDATFLG # BIT 7 FLAG 1
TC INTPRET
CALL
S30.1
EXIT
TC PARAM30
TC UPFLAG
#
Page 637
ADRES XDELVFLG # SET XDELVFLG BIT 8 FLAG 2
TCF GOTOP00H
#
P31 TC P30/P31
TC DOWNFLAG
ADRES UPDATFLG # RESET UPDATFLG BIT 7 FLAG 1

```

|            |       |             |                              |
|------------|-------|-------------|------------------------------|
|            | TC    | DOWNFLAG    |                              |
|            | ADRES | NORMSW      | # RESET NORMSW BIT 10 FLAG 7 |
|            | TC    | INTPRET     |                              |
|            | CALL  |             |                              |
|            |       | S31.1       |                              |
|            | EXIT  |             |                              |
|            | TC    | CNTNUP30    |                              |
|            | TC    | PARAM30     |                              |
|            | TC    | DOWNFLAG    |                              |
|            | ADRES | XDELVFLG    | # BIT 8 FLAG 2.              |
|            | TCF   | GOTOP00H    |                              |
| P30/P31    | XCH   | Q           |                              |
|            | TS    | P30/31RT    |                              |
|            | TC    | UPFLAG      |                              |
|            | ADRES | UPDATFLG    | # SET UPDATFLG BIT 7 FLAG 1  |
|            | TC    | UPFLAG      |                              |
|            | ADRES | TRACKFLG    | # SET TRACKFLG BIT 5 FLAG 1  |
|            | CAF   | V06N33      | # T OF IGN                   |
|            | TC    | BANKCALL    |                              |
|            | CADR  | GOFLASHR    |                              |
|            | TCF   | GOTOP00H    |                              |
|            | TC    | P30/31RT    |                              |
|            | TCF   | P30/P31 +4  |                              |
|            | TC    | PHASCHNG    |                              |
|            | OCT   | 00014       |                              |
|            | TC    | ENDOFJOB    |                              |
| CNTNUP30   | XCH   | Q           |                              |
|            | TS    | P30/RET     |                              |
|            | CAF   | V06N81      |                              |
|            | TC    | BANKCALL    |                              |
|            | CADR  | GOFLASH     |                              |
|            | TCF   | GOTOP00H    |                              |
|            | TC    | P30/RET     |                              |
|            | TCF   | CNTNUP30 +2 |                              |
| PARAM30    | XCH   | Q           |                              |
|            | TS    | P30/31RT    |                              |
|            | CAF   | V06N42      |                              |
|            | TC    | BANKCALL    |                              |
|            | CADR  | GOFLASH     |                              |
|            | TC    | GOTOP00H    | # ON TERMINATION GOTOP00H    |
|            | TCF   | REFTEST     | # ON PROCEED GO DO REFTTEST  |
| # Page 638 |       |             |                              |
|            | TCF   | PARAM30 +2  |                              |
| REFTEST    | CAF   | BIT13       |                              |

July 12, 2016

1091

```

 MASK STATE +3 # REFSMFLAG
 EXTEND
 BZF NOTSET # REFSMFLAG =0, THEN BRANCH TO NOTSET
 TC INTPRET
 VLOAD PUSH
 DELVSIN
 CALL
 GET+MGA
 GOTO
 FLASHMGA
NOTSET EXTEND
 DCS MARSDP
 DXCH +MGA # +MGA, +MGA+1 CONTAINS (-00001)
 TC INTPRET
FLASHMGA CALL
 DISPMGA
 EXIT
 TC P30/31RT
MARSDP OCT 00000 # (00000) (16440) = (+00001)
 OCT 35100
 # (.01) DEGREES IN THE LOW ORDER REGISTER

V06N33 VN 0633
V06N42 VN 0642
V16N35 VN 1635
V06N45 VN 0645
```

```

Page 639
PROGRAM DESCRIPTION S30.1 DATE 9NOV66
MOD NO 1 LOG SECTION P30,P37
MOD BY RAMA AIYAWAR **
MOD.2 BY S.ZELDIN -- TO CORRECT MOD.1 FOR COLOSSUS 29DEC67
#
FUNCTIONAL DESCRIPTION
BASED ON STORED TARGET PARAMETERS (R OF IGNITION (RTIG), V OF
IGNITION (VTIG), TIME OF IGNITION (TIG)), DELV(LV), COMPUTE PERIGEE ALTITUDE
APOGEE ALTITUDE AND DELTA-V REQUIRED IN REF. COORDS. (DELVSIN).
#
CALLING SEQUENCE
L CALL
L+1 S30.1
#
NORMAL EXIT MODE
AT L+2 OR CALLING SEQUENCE (GOTO L+2)
#
SUBROUTINES CALLED
```

```

THISPREC
PERIAPO
#
ALARM OR ABORT EXIT MODES
NONE
#
ERASABLE INITIALIZATION REQUIRED
TIG TIME OF IGNITION DP B28CS
DELVSLV SPECIFIED DELTA-V IN LOCAL VERT.
COORDS. OF ACTIVE VEHICLE AT
TIME OF IGNITION VCT. B+7 M/CS
#
OUTPUT
RTIG POSITION AT TIG VCT. B+29 M
VTIG VELOCITY AT TIG VCT. B+7 M
HAPO APOGEE ALT. DP B+29 M
HPER PERIGEE ALT. DP B+29 M
DELVSIN DELVSLV IN REF COORDS VCT. B+7 M/CS
VGDISP MAG. OF DELVSIN DP B+7 M/CS
#
DEBRIS QTEMP TEMP.ERASABLE
QPRET, MPAC
PUSHLIST
#
SETLOC P30S1A
BANK
#
COUNT 35/S30S
#
S30.1 STQ DLOAD
QTEMP
TIG # TIME IGNITION SCALED AT 2(+28)CS
STCALL TDEC1
THISPREC # ENCKE ROUTINE FOR
#
VLOAD SXA,2
VATT
RTX2
STOVL VTIG
#
Page 640
#
RATT
STORE RTIG
STORE RACT3
VXV UNIT
VTIG
STCALL UNRM

```

```

 LOMAT
VLOAD VXM
 DELVSLV
 0
VSL1 SXA,1
 RTX1
STORE DELVSIN
ABVAL
STOVL VGDISP # MAG DELV
 RTIG
PDVL VAD
 DELVSIN
 VTIG
CALL
 PERIAPO1
CALL
 SHIFTR1
CALL
 MAXCHK
STODL HPER # PERIGEE ALT B+29
 4D
CALL
 SHIFTR1
CALL
 MAXCHK
STCALL HAPO # APOGEE ALT B+29
 QTEMP

```

```

Page 641
S31.1 PROGRAM DESCRIPTION 28DEC67
MOD.1 BY S.ZELDIN
#
S31.1 COMPUTES DELV IN REF AND LV COORDS,MAG OF DELV,INTERCEPT TIME,
APOGEE AND PERIGEE ALT FOR REQUIRED MANEUVER
#
CALLING SEQUENCE
L CALL
L+1 S31.1
#
NORMAL EXIT MODE
AT L +2 OF CALLING SEQUENCE (GOTO L+2)
#
SUBROUTINES CALLED
AGAIN
PERIAPO1
SHIFTR1

```

```

MIDGIM
#
NO ALARM OR ABORT MODES
#
INPUT
DELLT4 DP +28
TIG DP +28
RTARG VCT +29
#
OUTPUT
DELVLVC VCT +7
VGDISP DP +7
HAPO DP +29
HPER DP +29
TPASS4 DP +28
#
DEBRIS -- QTEMP

S31.1 STQ DLOAD
 QTEMP
 TIG
 STCALL TDEC1
 AGAIN # RETURNS RTX2,RTX1,RATT,VATT,VIPRIME
 VLOAD PDVL # DELVEET3
 RTIG
 VIPRIME
 CALL
 PERIAP01
 CALL
 SHIFTR1
 CALL
 MAXCHK
 STODL HPER # B29
 4D
 CALL
 SHIFTR1
 CALL
 MAXCHK
 STOVL HAPO # B29
Page 642
 DELVEET3
 STORE 0
 SET CALL
 AVFLAG
 MIDGIM # GET DELVLVC B7 FOR DISPLAY
 ABVAL

```

July 12, 2016

1095

```

 STODL VGDISP # B+7 FOR DISPLAY
 DELLT4
 DAD
 TIG
 STCALL TPASS4 # FOR S40.1
 QTEMP

Page 643
SUBROUTINE NAME: DELRSPL (CONTINUATION OF V 82 IN CSM IF P11 ACTIVE)
TRANSFERRED COMPLETELY FROM SUNDISK, P30S REV 33. 9 SEPT 67.
MOD NO: 0 MOD BY: ZELDIN DATE:
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 11 APR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 12 MAY 67 ADD UR.RT CALC WHEN BELOW 300K
MOD NO: 2.1 MOD BY: RR BAIRNSFATHER DATE: 5 JULY 67 FIX ERROR ON MOD. 2
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 12 JUL 67 CHANGE SIGN OF DISPLAYED ERROR.
MOD 4 MOD BY S.ZELDIN DATE: 3 APRIL 68 CHANGE EQUATIONS FOR L/D=.16 WH
#
FUNCTION: CALCULATE (FOR DISPLAY ON CALL) AN APPROXIMATE MEASURE OF IN-PLANE SPLASH DOWN
ERROR. IF THE FREE-FALL TRANSFER ANGLE TO 300K FT ABOVE PAD RADIUS IS POSITIVE
SPLASH ERROR= -RANGE TO TARGET + FREE-FALL TRANSFER ANGLE + ESTIMATED ENTRY ANG
THE TARGET LOCATION AT ESTIMATED TIME OF IMPACT IS USED. IF THE FREE-FALL TRAN
ANGLE IS NEGATIVE: SPASH ERROR= -RANGE TO TARGET.
THE PRESENT TARGET LOCATION IS USED.
#
CALLING SEQUENCE: CALLED AFTER SR30.1 IF IN CSM AND IF P11 OPERATING (UNDER CONTROL OF V82)
#
SUBROUTINES CALLED: VGAMCALC, TFF/TRIG, LALOTORV.
#
EXIT: RETURN DIRECTLY TO V 82 PROG. AT SPLRET
#
ERASABLE INITIALIZATION: LEFT BY SR30.1 AND V82GON1
#
OUTPUT: RSP-RREC RANGE IN REVOLUTIONS. DSKY DISPLAY IN N. MI.
#
DEBRIS: QPRET, PDL0 ... PDL7, PDL10.

THETA(1)

 BANK 32
 SETLOC DELRSPL1
 BANK
 COUNT* $$/P30 # PROGRAMS: P30 EXTERNAL DELTA V

DELRSPL STORE 8D
 BPL DSU
 CANTDO # GONE PAST 300K FT ALT
```

```

 1BITDP
BOV CALL
 CANTDO
 VGAMCALC
 # POSMAX INDICATES NO 300K FT SOLUTION.
 # +GAMMA(REV) IN PMAC,V300 MAG(B-7)=PDL 0
PUSH CALL
 TFF/TRIG
CALL CALL
 AUGEKUGL
PDDL ACOS
 # T ENTRY PDL 6
 CDELF/2
DAD
 4
GETARG STOVL THETA(1)
 LAT(SPL)
STODL LAT
 HI6ZEROS
STODL ALT
 # ALT=0 = LAT +4
 PIPTIME
Page 644
BON DLOAD
 V37FLAG
 +2
 TSTART82
DSU DAD
 8D
CLEAR CALL
 ERADFLAG
 LALOTORV
 # R RECOV. IN ALPHAV AND MPAC
UNIT PDVL
 RONE
UNIT DOT
SL1 ARCCOS
BDSU
 # ERROR = THETA EST - THETA TARG
 # NEGATIVE NUMBER SIGNIFIES THAT WILL FALL SH
 # POSITIVE NUMBER SIGNIFIES THAT WILL OVERSHO
DELRDONE STCALL THETA(1)
 RSP-RREC
 # DOWNRANGE RECOVERY RANGE ERROR /360
 INTWAKEO
CALL
 SPLRET
CANTDO DLOAD PDDL
 # INITIALIZE ERASE TO DOT TARGET AND UR
 # FOR RANGE ANGLE.
 # TO PDL 0 FOR DEN INDDV.
 HIDPHALF
 # ZERO TO PDL 2 FOR PHI ENTRY
 HI6ZEROS
PUSH

```



```

 STCALL 8D
 GETARG # GO SET RSP-RREC =0

AUGEKUGL VLOAD
 X1CON -2
 STODL X1 -2
 0
 DSU BMN
 V(21K)
 LOOPSET
 XSU,1 XCHX,2
 S1
 X1
 XCHX,2 DSU
 S1
 V(3K)
 BMN XCHX,2
 LOOPSET
 S1
 DSU BMN
 V(4K)
 LOOPSET
 XCHX,2 XCHX,2

Page 645
 S1
 X1
 DSU BMN
 V(400)
 LOOPSET
 SXA,1
 S1
 LOOPSET INCR,1 GOTO
 DEC 1
 K1K2LOOP
 K2CALC SXA,1
 S1
 K1K2LOOP DLOAD DSU*
 0
 V(32K) +1,1
 DMP* DAD*
 YK1K2 +1,1
 CK1K2 +1,1
 PDDL TIX,1
 2
 K2CALC
 DSU BDDV

```

|         |       |           |                                 |
|---------|-------|-----------|---------------------------------|
|         | PUSH  | BOV       |                                 |
|         |       | MAXPHI    |                                 |
|         | BMN   | DSU       |                                 |
|         |       | MAXPHI    |                                 |
|         |       | MAXPHIC   |                                 |
|         | BPL   |           |                                 |
|         |       | MAXPHI    |                                 |
| PHICALC | DLOAD | DSU       | # PHI ENTRY PDL 4D              |
|         |       | 0         |                                 |
|         |       | V(26K)    |                                 |
|         | BPL   | DLOAD     |                                 |
|         |       | TGR26     |                                 |
|         |       | TLESS26   |                                 |
|         | DDV   |           |                                 |
|         |       | 0         |                                 |
| TENT    | DMP   | RVQ       |                                 |
|         |       | 4D        |                                 |
| TGR26   | DLOAD | GOTO      |                                 |
|         |       | TGR26CON  |                                 |
|         |       | TENT      |                                 |
| MAXPHI  | DLOAD | PDDL      |                                 |
|         |       | MAXPHIC   |                                 |
|         | GOTO  |           |                                 |
|         |       | PHICALC   |                                 |
| MAXPHIC | 2DEC  | .09259298 | # 2000 NM FOR MAXIMUM PHI ENTRY |

# Page 646

COUNT\* \$\$/P30

#  
# \*\*\*\*\* TABLE IS INDEXED. KEEP IN ORDER \*\*\*\*\*

# Page 641

|       |      |                |        |
|-------|------|----------------|--------|
|       | 2DEC | 7.07304526 E-4 | # 5500 |
|       | 2DEC | 3.08641975 E-4 | # 2400 |
|       | 2DEC | 3.08641975 E-4 | # 2400 |
|       | 2DEC | -8.8888888 E-3 | # -3.2 |
|       | 2DEC | 2.7777777 E-3  | # 1    |
| CK1K2 | 2DEC | 6.6666666 E-3  | # 2.4  |

July 12, 2016

1099

|          |       |                     |             |
|----------|-------|---------------------|-------------|
|          | 2DEC  | 0                   | # 0         |
|          | 2DEC* | -1.86909989 E-5 B7* | # -.443     |
|          | 2DEC  | 0                   |             |
|          | 2DEC* | 1.11639691 E-3 B7*  | # .001225   |
|          | 2DEC* | 9.56911636 E-4 B7*  | # .00105    |
| YK1K2    | 2DEC* | 2.59733157 E-4 B7*  | # .000285   |
| V(400)   | 2DEC  | 1.2192 B-7          |             |
| V(28K)   | 2DEC  | 85.344 B-7          |             |
| V(3K)    | 2DEC  | 9.144 B-7           |             |
| V(24K)   | 2DEC  | 73.152 B-7          |             |
|          | 2DEC  | 85.344 B-7          |             |
| V(32K)   | 2DEC  | 97.536 B-7          |             |
| V(4K)    | 2DEC  | 12.192 B-7          |             |
| V(21K)   | 2DEC  | 64.000 B-7          |             |
| TLESS26  | 2DEC* | 5.70146688 E7 B-35* | # 8660PHI/V |
| TGR26CON | 2DEC  | 7.2 E5 B-28         | # PHI/3     |
| V(26K)   | 2DEC  | 79.248 B-7          | # 26000     |

# Page 647

|       |     |    |
|-------|-----|----|
| X1CON | DEC | 10 |
|       | DEC | 8  |
|       | DEC | 6  |

# \*\*\*\*\* TABLE IS INDEXED. KEEP IN ORDER \*\*\*  
# ABOVE

# Page 648  
# \*\*\*\*\* AVFLAG/P \*\*\*\*\*  
#

```

SUBROUTINES USED
#
UPFLAG
DOWNFLAG

 SETLOC P30SUBS
 BANK
 EBANK= SUBEXIT
AVFLAGA EXTEND # AVFLAG = CSM
 QXCH SUBEXIT
 TC DOWNFLAG
 ADRES AVFLAG # BIT 5 FLAG 2
 CAF EBANK7
 TS EBANK
 EBANK= ECSTEER
 CAF BIT13
 TS ECSTEER # SET ECSTEER = 1
 CAF EBANK4
 TS EBANK
 EBANK= SUBEXIT
 TC SUBEXIT
AVFLAGP EXTEND # AVFLAG = LEM
 QXCH SUBEXIT
 TC UPFLAG
 ADRES AVFLAG # BIT 5 FLAG 2
 TC SUBEXIT
P20FLGON EXTEND
 QXCH SUBEXIT
 TC UPFLAG
 ADRES TRACKFLG
 TC UPFLAG
 ADRES UPDATFLG
 TC SUBEXIT # DP B4

```

This code is written to file `src/P30-P37.s`.

**B.70 P32-P33 P72-P73**

```

1101 <src/P32-P33-P72-P73.s 1101>≡
Copyright: Public domain.
Filename: P32-P33_P72-P73.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 649-683
Mod history: 2009-05-09 RSB Adapted from the Luminary131/ file
P32-P35_P72-P75.agc and Comanche055 page
images.
2009-05-20 RSB Corrected CSI/COM3 -> CSI/CDH3,
CSI/CDHI -> CSI/CDH1, CDHTAB -> CDHTAG,
changed a SETLOC from CSI/CDH to CSI/CDH1,
a SETLOC CSI/CDH1 to CSIPROG.
2009-05-21 RSB Changed a P32/P72D to P32/P72E in
P32/P72D. DP1/4TH changed to DP1/4 in
CDHMR.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 649
COELLIPTIC SEQUENCE INITIATION (CSI) PROGRAMS (P32 AND P72)
#
MOD NO -1 LOG SECTION -- P32-P35, P72-P75
MOD BY WHITE, P. DATE 1 JUNE 67
#

```

## # PURPOSE

- # (1) TO CALCULATE PARAMETERS ASSOCIATED WITH THE FOLLOWING  
# CONCENTRIC FLIGHT PLAN MANEUVERS -- THE CO-ELLIPTIC SEQUENCE  
# INITIATION (CSI) MANEUVER AND THE CONSTANT DELTA ALTITUDE  
# (CDH) MANEUVER.
- # (2) TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA  
# APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.
- # (3) TO DISPLAY TO THE ASTRONAUT AND THE GROUND DEPENDENT VARIABLES  
# ASSOCIATED WITH THE CONCENTRIC FLIGHT PLAN MANEUVERS FOR  
# APPROVAL BY THE ASTRONAUT/GROUND.
- # (4) TO STORE THE CSI TARGET PARAMETERS FOR USE BY THE DESIRED  
# THRUSTING PROGRAM.

## # ASSUMPTIONS

- # (1) AT A SELECTED TPI TIME THE LINE OF SIGHT BETWEEN THE ACTIVE  
# AND PASSIVE VEHICLES IS SELECTED TO BE A PRESCRIBED ANGLE (E)  
# FROM THE HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE  
# POSITION.
- # (2) THE TIME BETWEEN CSI IGNITION AND CDH IGNITION MUST BE  
# COMPUTED TO BE GREATER THAN 10 MINUTES FOR SUCCESSFUL  
# COMPLETION OF THE PROGRAM.
- # (3) THE TIME BETWEEN CDH IGNITION AND TPI IGNITION MUST BE  
# COMPUTED TO BE GREATER THAN 10 MINUTES FOR SUCCESSFUL  
# COMPLETION OF THE PROGRAM.
- # (4) CDH DELTA V IS SELECTED TO MINIMIZE THE VARIATION OF THE  
# ALTITUDE DIFFERENCE BETWEEN THE ORBITS.
- # (5) CSI BURN IS DEFINED SUCH THAT THE IMPULSIVE DELTA V IS IN THE  
# HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION AT CSI  
# IGNITION.
- # (6) THE PERICENTER ALTITUDE OF THE ORBIT FOLLOWING CSI AND CDH  
# MUST BE GREATER THAN 35,000 FT (LUNAR ORBIT) OR 85 NM (EARTH  
# ORBIT) FOR SUCCESSFUL COMPLETION OF THIS PROGRAM.
- # (7) THE CSI AND CDH MANEUVERS ARE ORIGINALLY ASSUMED TO BE  
# PARALLEL TO THE PLANE OF THE CSM ORBIT. HOWEVER, CREW

# Page 650

- # MODIFICATION OF DELTA V (LV) COMPONENTS MAY RESULT IN AN  
# OUT-OF-PLANE CSI MANEUVER
- # (8) STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC  
# STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION 10).
- # (9) COMPUTED VARIABLES MAY BE STORED FOR LATER VERIFICATION BY  
# THE GROUND. THESE STORAGE CAPABILITIES ARE NORMALLY LIMITED  
# ONLY TO THE PARAMETERS FOR ONE THRUSTING MANEUVER AT A TIME  
# EXCEPT FOR CONCENTRIC FLIGHT PLAN MANEUVER SEQUENCES.
- # (10) THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM  
# OR CSM STATE VECTORS FOR THIS PROGRAM. IF RADAR USE IS  
# DESIRED THE RADAR WAS TURNED ON AND LOCKED BY THE CSM BY

```

PREVIOUS SELECTION OF P20. RADAR SIGHTING MARKS WILL BE MADE
AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE
TRACK AND UPDATE FLAGS (SEE P20). THE RENDEZVOUS TRACKING
MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH
THRUSTING MANEUVER.
(11) THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.
(12) THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
CALCULATES THE MANEUVER PARAMETERS. SET AT THE START OF
EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
COMPLETED THE FINAL MANEUVER COMPUTATION AND DISPLAY
CYCLE.
#
EXTERNAL DELTA V STEERING FLAG -- DESIGNATES THE TYPE OF
STEERING REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE
THRUSTING PROGRAM SELECTED AFTER COMPLETION OF THIS
PROGRAM.
#
(13) IT IS NORMALLY REQUIRED THAT THE ISS BE ON FOR 1 HOUR PRIOR TO
A THRUSTING MANEUVER.
#
(14) THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY
#
P32 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
P72 IF THIS VEHICLE IS THE PASSIVE VEHICLE.
#
INPUT
(1) TCSI TIME OF THE CSI MANEUVER
Page 651
(2) NN NUMBER OF APSIDAL CROSSINGS THRU WHICH THE ACTIVE
VEHICLE ORBIT CAN BE ADVANCED TO OBTAIN THE CDH
MANEUVER POINT.
(3) ELEV DESIRED LOS ANGLE AT TPI
(4) TTPI TIME OF THE TPI MANEUVER
#
OUTPUT
(1) TRKMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO
(3) +MGA MIDDLE GIMBAL ANGLE
(4) DIFFALT DELTA ALTITUDE AT CDH
(5) T1TOT2 DELTA TIME FROM CSI TO CDH

```

```

(6) T2TOT3 DELTA TIME FROM CDH TO TPI
(7) DELVLVC DELTA VELOCITY AT CSI -- LOCAL VERTICAL COORDINATES
(8) DELVLVC DELTA VELOCITY AT CDH -- LOCAL VERTICAL COORDINATES
#
DOWNLINK
(1) TCSI TIME OF THE CSI MANEUVER
(2) TCDH TIME OF THE CDH MANEUVER
(3) TTPI TIME OF THE TPI MANEUVER
(4) TIG TIME OF THE CSI MANEUVER
(5) DELVEET1 DELTA VELOCITY AT CSI -- REFERENCE COORDINATES
(6) DELVEET2 DELTA VELOCITY AT CDH -- REFERENCE COORDINATES
(7) DIFFALT DELTA ALTITUDE AT CDH
(8) NN NUMBER OF APSIDAL CROSSINGS THRU WHICH THE ACTIVE
VEHICLE ORBIT CAN BE ADVANCED TO OBTAIN THE CDH
MANEUVER POINT
(9) ELEV DESIRED LOS ANGLE AT TPI
#
COMMUNICATION TO THRUSTING PROGRAMS
(1) TIG TIME OF THE CSI MANEUVER
(2) RTIG POSITION OF ACTIVE VEHICLE AT CSI -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE
(3) VTIG VELOCITY OF ACTIVE VEHICLE AT CSE -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE
(4) DELVSIN DELTA VELOCITY AT CSI -- REFERENCE COORDINATES
(5) DELVSAB MAGNITUDE OF DELTA VELOCITY AT CSI
(6) XDELVFLG SET TO INDICATE EXTERNAL DELTA V VG COMPUTATION
#
SUBROUTINES USED
AVFLAGA
AVFLAGP
P20FLGON
VARALARM
BANKCALL
GOFLASH
GOTOPOOH
Page 652
VNPOOH
GOFLASHR
BLANKET
ENDOFJOB
SELECTMU
ADVANCE
INTINT
PASSIVE
CSI/A
S32/33.1

```



July 12, 2016

1105

```
DISDVLVC
VN1645

 BANK 35
 SETLOC CSI/CDH1
 BANK
 EBANK= SUBEXIT
 COUNT 35/P3272
P32 TC AVFLAGA
 TC P32STRT
P72 TC AVFLAGP
P32STRT TC INTPRET
 DLOAD
 ZEROVEC
 STORE CENTANG
 EXIT
 TC P32/P72A
ALMXITA SXA,2
 CSIALRM
ALMXIT LXC,1
 CSIALRM
 SLOAD* EXIT
 ALARM/TB -1,1
 CA MPAC
 TC VARALARM
 CAF V05N09
 TC BANKCALL
 CADR GOFLASH
 TC GOTOP00H
 TC -4
P32/P72A TC P20FLGON
 TC INTPRET
 DLOAD
 ZEROVEC
 STORE NN
 EXIT
 CAF V06N11 # TCSI
 TC VNP00H
 CAF V06N55
Page 653
 TC BANKCALL
 CADR GOFLASH
 TC GOTOP00H
 TC +2
 TC -5
 CAF V06N37 # TTPI
```

|            |        |          |
|------------|--------|----------|
|            | TC     | VNPOOH   |
|            | TC     | INTPRET  |
|            | DLOAD  |          |
|            |        | TCSI     |
|            | STCALL | TIG      |
|            |        | SELECTMU |
| P32/P72B   | CALL   |          |
|            |        | ADVANCE  |
|            | SETPD  | VLOAD    |
|            |        | OD       |
|            |        | VPASS1   |
|            | PDVL   | PDDL     |
|            |        | RPASS1   |
|            |        | TCSI     |
|            | PDDL   | PDDL     |
|            |        | TTPI     |
|            |        | 2PISC    |
|            | SL2    | PUSH     |
|            | CALL   |          |
|            |        | INTINT   |
|            | CALL   |          |
|            |        | PASSIVE  |
|            | CALL   |          |
|            |        | CSI/A    |
| P32/P72C   | BON    | SET      |
|            |        | FINALFLG |
|            |        | P32/P72D |
|            |        | UPDATFLG |
| P32/P72D   | DLOAD  | GOTO     |
|            |        | T1TOT2   |
|            |        | P32/P72E |
|            | SETLOC | CSI/CDH3 |
|            | BANK   |          |
| P32/P72E   | STORE  | T1TOT2   |
|            | DSU    | BPL      |
|            |        | 60MIN    |
|            |        | P32/P72E |
|            | DLOAD  | GOTO     |
|            |        | T2TOT3   |
|            |        | P32/P72F |
|            | SETLOC | CSI/CDH1 |
|            | BANK   |          |
| P32/P72F   | STORE  | T2TOT3   |
|            | DSU    | BPL      |
| # Page 654 |        | 60MIN    |

|        |          |
|--------|----------|
|        | P32/P72F |
| EXIT   |          |
| CAF    | V06N75   |
| TC     | VNP00H   |
| TC     | INTPRET  |
| VLOAD  | CALL     |
|        | DELVEET1 |
|        | S32/33.1 |
| STOVL  | DELVEET1 |
|        | RACT2    |
| STOVL  | RACT1    |
|        | DELVEET2 |
| AXT,1  | CALL     |
| VN     | 0682     |
|        | DISDVLVC |
| DLOAD  |          |
|        | TTPI     |
| STCALL | TTPIO    |
|        | VN1645   |
| GOTO   |          |
|        | P32/P72B |

```
Page 655
CONSTANT DELTA HEIGHT (CDH) PROGRAMS (P33 AND P73)
MOD NO -1 LOC SECTION -- P32-P35, P72-P75
MOD BY WHITE, P. DATE: 1 JUNE 67
#
PURPOSE
#
(1) TO CALCULATE PARAMETERS ASSOCIATED WITH THE CONSTANT DELTA
ALTITUDE MANEUVER (CDH).
#
(2) TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA
APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.
#
(3) TO DISPLAY TO THE ASTRONAUT AND THE GROUND DEPENDENT VARIABLES
ASSOCIATED WITH THE CDH MANEUVER FOR APPROVAL BY THE
ASTRONAUT/GROUND.
#
(4) TO STORE THE CDH TARGET PARAMETERS FOR USE BY THE DESIRED
THRUSTING PROGRAM.
#
ASSUMPTIONS
#
(1) THIS PROGRAM IS BASED UPON PREVIOUS COMPLETION OF THE
CO-ELLIPTIC SEQUENCE INITIATION (CSI) PROGRAM (P32/P72).
```

# THEREFORE --

#

# (A) AT A SELECTED TPI TIME (NOW IN STORAGE) THE LINE OF SIGHT

# BETWEEN THE ACTIVE AND PASSIVE VEHICLES WAS SELECTED TO BE

# A PRESCRIBED ANGLE (E) (NOW IN STORAGE) FROM THE

# HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION.

#

# (B) THE TIME BETWEEN CSI IGNITION AND CDH IGNITION WAS

# COMPUTED TO BE GREATER THAN 10 MINUTES.

#

# (C) THE TIME BETWEEN CDH IGNITION AND TPI IGNITION WAS

# COMPUTED TO BE GREATER THAN 10 MINUTES.

#

# (D) THE VARIATION OF THE ALTITUDE DIFFERENCE BETWEEN THE

# ORBITS WAS MINIMIZED.

#

# (E) CSI BURN WAS DEFINED SUCH THAT THE IMPULSIVE DELTA V WAS

# IN THE HORIZONTAL PLANE DEFINED BY ACTIVE VEHICLE

# POSITION AT CSI IGNITION.

#

# (F) THE PERICENTER ALTITUDES OF THE ORBITS FOLLOWING CSI AND

# CDH WERE COMPUTED TO BE GREATER THAN 35,000 FT FOR LUNAR

# ORBIT OR 85 NM FOR EARTH ORBIT.

#

# (G) THE CSI AND CDH MANEUVERS WERE ASSUMED TO BE PARALLEL TO

# THE PLANE OF THE PASSIVE VEHICLE ORBIT. HOWEVER, CREW

# Page 656

# MODIFICATION OF DELTA V (LV) COMPONENTS MAY HAVE RESULTED

# IN AN OUT-OF-PLANE MANEUVER.

#

# (2) STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC

# STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION 4).

#

# (3) COMPUTED VARIABLES MAY BE STORED FOR LATER VERIFICATION BY

# THE GROUND. THESE STORAGE CAPABILITIES ARE NORMALLY LIMITED

# ONLY TO THE PARAMETERS FOR ONE THRUSTING MANEUVER AT A TIME

# EXCEPT FOR CONCENTRIC FLIGHT PLAN MANEUVER SEQUENCES.

#

# (4) THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM.

# OR CSM STATE VECTORS FOR THIS PROGRAM. IF RADAR USE IS

# DESIRED THE RADAR WAS TURNED ON AND LOCKED ON THE CSM BY

# PREVIOUS SELECTION OF P20. RADAR SIGHTING MARKS WILL BE MADE

# AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE

# TRACK AND UPDATE FLAGS (SEE P20). THE RENDEZVOUS TRACKING

# MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH

# THRUSTING MANEUVER.

```

#
(5) THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.
#
(6) THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
CALCULATES THE MANEUVER PARAMETERS. SET AT THE START OF
EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
COMPLETED THE FINAL MANEUVER COMPUTATION AND DISPLAY
CYCLE.
#
EXTERNAL DELTA V STEERING FLAG -- DESIGNATES THE TYPE OF
STEERING REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE
THRUSTING PROGRAM SELECTED AFTER COMPLETION OF THIS
PROGRAM.
#
(7) IT IS NORMALLY REQUIRED THAT THE ISS BE ON FOR 1 HOUR PRIOR TO
A THRUSTING MANEUVER.
#
(8) THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY.
#
P33 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
P73 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
INPUT
#
(1) TTPIO TIME OF THE TPI MANEUVER -- SAVED FROM P32/P72
Page 657
(2) ELEV DESIRED LOS ANGLE AT TPI -- SAVED FROM P32/P72
(3) TCDH TIME OF THE CDH MANEUVER
#
OUTPUT
#
(1) TRKMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO
(3) +MGA MIDDLE GIMBAL ANGLE
(4) DIFFALT DELTA ALTITUDE AT CDH
(5) T2TOT3 DELTA TIME FROM CDH TO COMPUTED TPI
(6) NOMTPI DELTA TIME FROM NOMINAL TPI TO COMPUTED TPI
(7) DELVLC DELTA VELOCITY AT CDH -- LOCAL VERTICAL COORDINATES
#
DOWNLINK

```

```

#
(1) TCDH TIME OF THE CDH MANEUVER
(2) TTPI TIME OF THE TPI MANEUVER
(3) TIG TIME OF THE CDH MANEUVER
(4) DELLVEET2 DELTA VELOCITY AT CDH -- REFERENCE COORDINATES
(5) DIFFALT DELTA ALTITUDE AT CDH
(6) ELEV DESIRED LOS ANGLE AT TPI
#
COMMUNICATION TO THRUSTING PROGRAMS
#
(1) TIG TIME OF THE CDH MANEUVER
(2) RTIG POSITION OF ACTIVE VEHICLE AT CDH -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE.
(3) VTIG VELOCITY OF ACTIVE VEHICLE AT CDH -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE.
(4) DELVSIN DELTA VELOCITY AT CDH -- REFERENCE COORDINATES.
(5) DELVSAB MAGNITUDE OF DELTA VELOCITY AT CDH.
(6) XDELVFLG SET TO INDICATE EXTERNAL DELTA V VG COMPUTATION.
#
SUBROUTINES USED
#
AVFLAGA
AVFLAGP
P20FLGON
VNPOOH
SELECTMU
ADVANCE
CDHMVR
INTINT3P
ACTIVE
PASSIVE
S33/S34.1
ALARM
BANKCALL
GOFLASH
GOTOPOOH
S32/33.1
Page 658
VN1645

```

COUNT 35/P3373

```

P33 TC AVFLAGA
 TC P33/P73A
P73 TC AVFLAGP
P33/P73A TC P20FLGON

```

July 12, 2016

1111

|            |        |          |        |
|------------|--------|----------|--------|
|            | CAF    | V06N13   | # TCDH |
|            | TC     | VNP00H   |        |
|            | TC     | INTPRET  |        |
|            | DLOAD  |          |        |
|            |        | TTPIO    |        |
|            | STODL  | TTPI     |        |
|            |        | TCDH     |        |
|            | STCALL | TIG      |        |
|            |        | SELECTMU |        |
| P33/P73B   | CALL   |          |        |
|            |        | ADVANCE  |        |
|            | CALL   |          |        |
|            |        | CDHMVR   |        |
|            | SETPD  | VLOAD    |        |
|            |        | OD       |        |
|            |        | VACT3    |        |
|            | PDVL   | CALL     |        |
|            |        | RACT2    |        |
|            |        | INTINT3P |        |
|            | CALL   |          |        |
|            |        | ACTIVE   |        |
|            | SETPD  | VLOAD    |        |
|            |        | OD       |        |
|            |        | VPASS2   |        |
|            | PDVL   | CALL     |        |
|            |        | RPASS2   |        |
|            |        | INTINT3P |        |
|            | CALL   |          |        |
|            |        | PASSIVE  |        |
|            | DLOAD  | SET      |        |
|            |        | ZEROVEC  |        |
|            |        | ITSWICH  |        |
|            | STCALL | NOMTPI   |        |
|            |        | S33/34.1 |        |
|            | BZE    | EXIT     |        |
|            |        | P33/P73C |        |
|            | TC     | ALARM    |        |
|            | OCT    | 611      |        |
|            | CAF    | V05N09   |        |
|            | TC     | BANKCALL |        |
|            | CADR   | GOFLASH  |        |
|            | TC     | GOTOP00H |        |
| # Page 659 |        |          |        |
|            | TC     | +2       |        |
|            | TC     | P33/P73A |        |
|            | TC     | INTPRET  |        |

|          |        |          |
|----------|--------|----------|
|          | DLOAD  | ZEROVEC  |
|          | STCALL | NOMTPI   |
|          |        | P33/P73C |
|          | SETLOC | CSI/CDH2 |
|          | BANK   |          |
| P33/P73C | BON    | SET      |
|          |        | FINALFLG |
|          |        | P33/P73D |
|          |        | UPDATFLG |
| P33/P73D | DLOAD  | DAD      |
|          |        | NOMTPI   |
|          |        | TTPI     |
|          | STORE  | TTPI     |
|          | DSU    | GOTO     |
|          |        | TCDH     |
|          |        | P33/P73E |
|          | SETLOC | CSI/CDH1 |
|          | BANK   |          |
| P33/P73E | DSU    | BPL      |
|          |        | 60MIN    |
|          |        | P33/P73E |
|          | DAD    |          |
|          |        | 60MIN    |
|          | STODL  | T1TOT2   |
|          |        | TTPI     |
|          | DSU    | PUSH     |
|          |        | TTPIO    |
| P33/P73F | ABS    | DSU      |
|          |        | 60MIN    |
|          | BPL    | DAD      |
|          |        | P33/P73F |
|          |        | 60MIN    |
|          | SIGN   | STADR    |
|          | STORE  | T2TOT3   |
|          | EXIT   |          |
|          | CAF    | V06N75   |
|          | TC     | VNP00H   |
|          | TC     | INTPRET  |
|          | VLOAD  | CALL     |
|          |        | DELVEET2 |
|          |        | S32/33.1 |
|          | STCALL | DELVEET2 |
|          |        | VN1645   |



July 12, 2016

1113

GOTO  
# Page 660  
P33/P73B

# Page 661  
# \*\*\*\*\* AVFLAGA/P \*\*\*\*\*

# Page 662  
# \*\*\*\*\* DISDVLVC \*\*\*\*\*  
#  
# SUBROUTINES USED  
#  
# S32/33.X  
# VNPOOH

|          |        |          |
|----------|--------|----------|
|          | SETLOC | CDHTAG3  |
|          | BANK   |          |
| DISDVLVC | STORE  | DELVLVC  |
|          | STQ    | CALL     |
|          |        | NORMEX   |
|          |        | S32/33.X |
|          | VLOAD  | MXV      |
|          |        | DELVLVC  |
|          |        | OD       |
|          | VSL1   | SXA,1    |
|          |        | VERBNOUN |
|          | STORE  | DELVLVC  |
|          | EXIT   |          |
|          | CA     | VERBNOUN |
|          | TC     | VNPOOH   |
|          | TC     | INTPRET  |
|          | GOTO   |          |
|          |        | NORMEX   |
|          | SETLOC | FFTAG12  |
|          | BANK   |          |
| V06N11   | VN     | 0611     |
| V06N13   | VN     | 0613     |
| V06N75   | VN     | 0675     |
| V06N50   | VN     | 0650     |

# Page 663  
# \*\*\*\*\* CSI/A \*\*\*\*\*

```

#
SUBROUTINES USED
#
VECSHIFT
TIMETHET
PERIAP0
SHIFTR1
INTINT2C
CDHMVR
PERIAP01
INTINT
ACTIVE

 BANK 34
 SETLOC CSIPROG
 BANK
 EBANK= SUBEXIT
 COUNT 34/CSI

60MIN 2DEC 360000

ALARM/TB OCT 00600 # NO 1
 OCT 00601 # 2
 OCT 00602 # 3
 OCT 00603 # 4
 OCT 00604 # 5
 OCT 00605 # 6
 OCT 00606 # 7

LOOPMX 2DEC 16

INITST 2DEC .03048 B-7 # INITIAL DELDV = 10 FPS

DVMAX1 2DEC 3.0480 B-7 # MAXIMUM DV1 = 1000 FPS

DVMAX2 2DEC 3.014472 B-7 # 989 FPS

1DPB2 2DEC 1.0 B-2

1DPB28 2DEC 1

EPSILN1 2DEC .0003048 B-7 # .1 FPS

FIFPSDP 2DEC -.152400 B-7 # 5 FPS

DELMAX1 2DEC .6096000 B-7 # 200 FPS

```

|            |        |             |                                             |                  |
|------------|--------|-------------|---------------------------------------------|------------------|
|            | SETLOC | CSI/CDH     |                                             |                  |
|            | BANK   |             |                                             |                  |
| PMINE      | 2DEC   | 157420 B-29 | # 84 NM -- MUST BE 8 WORDS BEFORE PMINM     |                  |
| # Page 664 |        |             |                                             |                  |
| NICKELDP   | 2DEC   | .021336 B-7 | # 7 FPS                                     |                  |
| INITST1    | 2DEC   | .03048 B-7  | # INITIAL DELDV = 10 FPS                    |                  |
| ONETHTH    | 2DEC   | .0001 B-3   |                                             |                  |
| PMINM      | 2DEC   | 10668 B-29  | # 35000 FT -- MUST BE 8 WORDS AFTER PMINE   |                  |
|            | SETLOC | CSIPROG     |                                             |                  |
|            | BANK   |             |                                             |                  |
| CSI/A      | CLEAR  | SET         | # INITIALIZE INDICATORS                     |                  |
|            |        | S32.1F1     | # DVT1 HAS EXCEEDED MAX INDICATOR           |                  |
|            |        | S32.1F2     | # FIRST PASS FOR NEWTON ITERATION INDICATOR |                  |
|            | CLEAR  | SET         |                                             |                  |
|            |        | S32.1F3A    | # 00=1ST 2 PASSES 2ND CYCLE, 01=FIRST CYCLE |                  |
|            |        | S32.1F3B    | # 10=2ND CYCLE, 11=50 FPS STAGE 2ND CYCLE   |                  |
|            | DLOAD  |             |                                             |                  |
|            |        | ZEROVEC     |                                             |                  |
|            | STORE  | LOOPCT      |                                             |                  |
|            | STORE  | CSIALRM     |                                             |                  |
| CSI/B      | SETPD  | VLOAD       |                                             |                  |
|            |        | OD          |                                             |                  |
|            |        | RACT1       |                                             |                  |
|            | ABVAL  | PUSH        | # RA1                                       | B29 PL02D        |
|            | NORM   | SR1         |                                             |                  |
|            |        | X2          | #                                           | B29-N2+ B1 PL04D |
|            | PDVL   | ABVAL       |                                             |                  |
|            |        | RPASS3      |                                             |                  |
|            | NORM   | BDDV        | # RA1/RP3                                   | B1 PL02D         |
|            |        | X1          |                                             |                  |
|            | XSU,2  | SR*         | #                                           | B2               |
|            |        | X1          |                                             |                  |
|            |        | 1,2         |                                             |                  |
|            | DAD    | DMP         | # (1+(RA1/RP3))RA1                          | B29+B2=B31 PL00D |
|            |        | 1DPB2       |                                             |                  |
|            | NORM   | PDDL        | #                                           | PL02D            |
|            |        | X1          |                                             |                  |
|            |        | RTMU        |                                             |                  |

|            |        |          |                                  |                |
|------------|--------|----------|----------------------------------|----------------|
|            | SR1    | DDV      | #                                | B38-B31= B7 PI |
|            | SL*    | SQRT     | #                                | B7             |
|            |        | 0 -7,1   |                                  |                |
|            | PDVL   | UNIT     | #                                | PI             |
|            |        | RACT1    |                                  |                |
|            | PDVL   | VXV      |                                  |                |
|            |        | UP1      |                                  |                |
|            | UNIT   |          | # UNIT(URP1 X UVP1 X URA1) = UH1 |                |
|            | DOT    | SL1      | # VA1 . UH1                      | B7             |
|            |        | VACT1    |                                  |                |
|            | BDSU   | STADR    | #                                | PI             |
| # Page 665 |        |          |                                  |                |
|            | STODL  | DELVCSI  |                                  |                |
|            |        | INITST   | # 10 FPS                         |                |
|            | STORE  | DELDV    |                                  |                |
| CSI/B1     | DLOAD  | DAD      | # IF LOOPCT = 16                 |                |
|            |        | LOOPCT   |                                  |                |
|            |        | 1DPB28   |                                  |                |
|            | STORE  | LOOPCT   |                                  |                |
|            | DSU    | AXT,2    |                                  |                |
|            |        | LOOPMX   |                                  |                |
|            |        | 6        |                                  |                |
|            | BPL    | GOTO     |                                  |                |
|            |        | SCNDSOL  |                                  |                |
|            |        | CSI/B2   |                                  |                |
|            | SETLOC | CSIPROG2 |                                  |                |
|            | BANK   |          |                                  |                |
| CSI/B2     | SETPD  |          |                                  |                |
|            |        | OD       |                                  |                |
|            | DLOAD  | ABS      |                                  |                |
|            |        | DELVCSI  |                                  |                |
|            | DSU    | BMN      |                                  |                |
|            |        | DVMAX1   |                                  |                |
|            |        | CSI/B23  |                                  |                |
|            | AXT,2  | BON      |                                  |                |
|            |        | 7        |                                  |                |
|            |        | S32.1F1  |                                  |                |
|            |        | SCNDSOL  |                                  |                |
|            | BOFF   | BON      |                                  |                |
|            |        | S32.1F3A |                                  |                |
|            |        | CSI/B22  | # FLAG 3 NEQ 3                   |                |
|            |        | S32.1F3B |                                  |                |
|            |        | SCNDSOL  |                                  |                |
| CSI/B22    | SET    | DLOAD    |                                  |                |

July 12, 2016

1117

|            |        |          |                                   |
|------------|--------|----------|-----------------------------------|
|            |        | S32.1F1  |                                   |
|            |        | DVMAX2   |                                   |
|            | SIGN   |          |                                   |
|            |        | DELVCSI  |                                   |
|            | STCALL | DELVCSI  |                                   |
|            |        | CSI/B23  |                                   |
|            | SETLOC | CSIPROG3 |                                   |
|            | BANK   |          |                                   |
| CSI/B23    | VLOAD  | PUSH     |                                   |
|            |        | RACT1    |                                   |
|            | UNIT   | PDVL     |                                   |
|            |        | UP1      |                                   |
|            | VXV    | UNIT     | # UNIT (URP1 X UVP1 X URA1) = UH1 |
|            | VXSC   | VSL1     |                                   |
| # Page 666 |        |          |                                   |
|            |        | DELVCSI  |                                   |
|            | STORE  | DELVEET1 |                                   |
|            | VAD    | BOV      |                                   |
|            |        | VACT1    |                                   |
|            |        | CSI/B23D |                                   |
| CSI/B23D   | STCALL | VACT4    |                                   |
|            |        | VECSHIFT |                                   |
|            | STOVL  | VVEC     |                                   |
|            | SET    |          |                                   |
|            |        | RVS      |                                   |
|            | STOVL  | RVEC     |                                   |
|            |        | SN359+   |                                   |
|            | STCALL | SNTH     | # ALSO Csth                       |
|            |        | TIMETHET |                                   |
|            | SR1    | LXA,1    |                                   |
|            |        | RTX1     |                                   |
|            | STCALL | HAFPA1   |                                   |
|            |        | PERIAPO  |                                   |
|            | CALL   |          |                                   |
|            |        | SHIFTR1  |                                   |
|            | STODL  | POSTCSI  |                                   |
|            |        | CENTANG  |                                   |
|            | BZE    | GOTO     |                                   |
|            |        | +2       |                                   |
|            |        | CIRCL    |                                   |
|            | DLOAD  |          |                                   |
|            |        | ECC      |                                   |
|            | DSU    | BMN      |                                   |
|            |        | ONETHTH  |                                   |

|            |       |          |                        |                       |
|------------|-------|----------|------------------------|-----------------------|
|            |       | CIRCL    |                        |                       |
|            | DLOAD | CALL     |                        |                       |
|            |       | R1       |                        |                       |
|            |       | SHIFTR1  |                        |                       |
|            | SETPD | NORM     |                        |                       |
|            |       | 2D       |                        |                       |
|            |       | X1       |                        |                       |
|            | PDVL  | DOT      | #                      | PL04D                 |
|            |       | RACT1    |                        |                       |
|            |       | VACT4    |                        |                       |
|            | ABS   | DDV      |                        |                       |
|            |       | 02D      | # (/RDOTV/)/R1         | B36-B29= B7           |
|            | SL*   | DSU      |                        |                       |
|            |       | 0,1      |                        |                       |
|            |       | NICKELDP |                        |                       |
|            | BMN   | DLOAD    |                        |                       |
|            |       | CIRCL    |                        |                       |
|            |       | P        |                        |                       |
|            | SL2   | DSU      |                        |                       |
|            |       | 1RTEB2   | # 1.B.2                |                       |
|            | STODL | 14D      |                        |                       |
| # Page 667 |       | RTSR1/MU |                        |                       |
|            | SR1   | DDV      | # (1/ROOTMU)/R1        | B-16-B29 = B-45 PL02D |
|            | PDDL  | DMP      |                        |                       |
|            |       | P        |                        |                       |
|            |       | R1       |                        |                       |
|            | CALL  |          |                        |                       |
|            |       | SHIFTR1  |                        |                       |
|            | SL4   | SL1      |                        |                       |
|            | SQRT  | DMP      | # ((P/MU)**.5)/R1      | B14+B-14 = B-31 BL02D |
|            | BOFF  | SL3      |                        |                       |
|            |       | CMOONFLG |                        |                       |
|            |       | CSI/B3   |                        |                       |
| CSI/B3     | PDVL  | DOT      |                        |                       |
|            |       | RACT1    |                        |                       |
|            |       | VACT4    |                        |                       |
|            | STORE | RDOTV    |                        |                       |
|            | ABS   |          |                        |                       |
|            | NORM  | DMP      | # ((P/MU)**.5)RDOTV/R1 | PL02D                 |
|            |       | X2       |                        |                       |
|            | XSU,1 | SL*      | #                      | B-31+B36-B3 = B2      |
|            |       | X2       |                        |                       |
|            |       | 3,1      |                        |                       |
|            | STODL | 12D      |                        |                       |
|            |       | ZEROVECS |                        |                       |

July 12, 2016

1119

|            |        |              |                          |
|------------|--------|--------------|--------------------------|
|            | STORE  | 16D          |                          |
|            | VLOAD  | UNIT         |                          |
|            |        | 12D          |                          |
|            | STOVL  | SNTH         | # ALSO STORES CSTH AND 0 |
|            |        | RACT1        |                          |
|            | PDVL   | SIGN         |                          |
|            |        | VACT4        |                          |
|            |        | RDOTV        |                          |
|            | VCOMP  | CALL         |                          |
|            |        | VECSHIFT     |                          |
|            | STOVL  | VVEC         |                          |
|            | SETGO  |              |                          |
|            |        | RVSW         |                          |
|            |        | CSINEXT      |                          |
| SN359+     | 2DEC   | - .000086601 |                          |
| CS359+     | 2DEC   | + .499999992 |                          |
|            | SETLOC | CSIPROG4     |                          |
|            | BANK   |              |                          |
| CSINEXT    | STCALL | RVEC         |                          |
|            |        | TIMETHET     |                          |
|            | PDDL   | BPL          |                          |
|            |        | RDOTV        |                          |
| # Page 668 |        |              |                          |
|            |        | NTP/2        |                          |
|            | DLOAD  | DSU          |                          |
|            |        | HAFPA1       |                          |
|            | PUSH   | GOTO         |                          |
|            |        | NTP/2        |                          |
| CIRCL      | SETPD  | DLOAD        |                          |
|            |        | OOD          |                          |
|            |        | ZEROVECS     |                          |
|            | PUSH   |              |                          |
| NTP/2      | DLOAD  | DMP          |                          |
|            |        | NN           |                          |
|            |        | HAFPA1       |                          |
|            | SL     | DSU          |                          |
|            |        | 14D          |                          |
|            | DAD    |              |                          |
|            |        | TCSI         |                          |
|            | STORE  | TCDH         |                          |
|            | BDSU   | AXT,2        |                          |
|            |        | TTPI         |                          |

```

 5D
BMN SETPD
 SCNDSOL
 OD
VLOAD PDVL
 VACT4
 RACT1
CALL
 INTINT2C
STOVL RACT2
 VATT
STOVL VACT2
 VPASS1
SETPD PDVL
 OD
 RPASS1
GOTO
 CSINEXT1

SETLOC CSIPROG5
BANK

CSINEXT1 CALL
 INTINT2C
STOVL RPASS2
 VATT
STCALL VPASS2
 CDHMVR
VLOAD SETPD
 RACT2
 OD
Page 669 PDVL CALL
 VACT3
 PERIAP01
CALL
 SHIFTR1
STOVL POSTCDH
 VACT3
SETPD PDVL
 OD
 RACT2
PDDL PDDL
 TCDH
 TTPI
PDDL SL2

```



July 12, 2016

1121

```

 2PISC
PUSH CALL
 INTINT
CALL
 ACTIVE
DLOAD
 ELEV
SETPD SINE
 6D
PDVL UNIT
 RACT3
STORE OOD # URA3 AT OOD
PDVL VXV # PL14D, PL08D
 UP1
UNIT
PDDL COSINE # UNIT(URA3 X UVA3 X URA3) = UH3 B1 PL14D
 ELEV
VXSC STADR # (COSLOS)(UH3) B2 PL08D
STCALL 18D # PLUS
 CSINEXT2

SETLOC CSIPROG6
BANK

CSINEXT2 DLOAD VXSC # (SINLOS)(URA3) = U B2 PL00D
 VAD VSL1
 18D # B1
PUSH DOT # PL06D
 RACT3 # (U . RA3) = TEMP1 B1 + B29 = B30
SL1 PUSH # B29 PL08D
DSQ TLOAD # TEMP1**2 B58
 MPAC
PDVL DOT # PL11D
 RACT3
 RACT3
TLOAD DCOMP # RA3 . RA3

Page 670
 MPAC
PDVL DOT # RP3 . RP3 B58 PL14D
 RPASS3
 RPASS3 # PL11D
TAD TAD # TEMP1**2 + RA3.RA3 + RP3.RP3 = TEMP2 PL08D
BPL DLOAD
 K10RK2
 LOOPCT
DSU AXT,2
```

|            |        |          |                            |          |
|------------|--------|----------|----------------------------|----------|
|            |        | 1DPB28   |                            |          |
|            |        | 1D       |                            |          |
|            | BZE    |          |                            |          |
|            |        | ALMXITA  |                            |          |
|            | DLOAD  | SR1      |                            |          |
|            |        | DELDV    |                            |          |
|            | STORE  | DELDV    |                            |          |
|            | BDSU   |          |                            |          |
|            |        | DVPREV   |                            |          |
|            | STCALL | DELVCSI  |                            |          |
|            |        | CSI/B1   |                            |          |
| K10RK2     | SQRT   | PUSH     | # TEMP3 = TEMP2** .5       | B29 PL10 |
|            | DCOMP  | DSU      |                            |          |
|            |        | 06D      | # -TEMP1-TEMP3 = K2 AT 10D |          |
|            | STODL  | 10D      | #                          | PL08     |
|            | DSU    | STADR    | #                          | PL06     |
|            | STORE  | 12D      | # -TEMP1+TEMP3 = K1 AT 12D |          |
|            | ABS    |          |                            |          |
|            | STODL  | 14D      |                            |          |
|            |        | 10D      |                            |          |
|            | ABS    | DSU      |                            |          |
|            |        | 14D      |                            |          |
|            | BMN    | DLOAD    |                            |          |
|            |        | K2.      |                            |          |
|            |        | 12D      |                            |          |
|            | STCALL | 10D      | # K EQUALS K1              |          |
|            |        | K2.      |                            |          |
|            | SETLOC | CSIPROG7 |                            |          |
|            | BANK   |          |                            |          |
| K2.        | DLOAD  |          |                            |          |
|            |        | 10D      |                            |          |
|            | VXSC   | VSL1     |                            |          |
|            | VAD    | UNIT     | # V = RA3 + KU UNIT        | B1       |
|            |        | RACT3    |                            |          |
|            | PDVL   | UNIT     |                            |          |
|            |        | RPASS3   | #                          | PL06     |
|            | PDVL   | UNIT     |                            |          |
|            |        | VPASS3   | #                          | PL12     |
|            | VXV    | PDVL     | # UVP3 X URP3              | PL18     |
| # Page 671 |        | 06D      |                            |          |
|            |        | 06D      |                            |          |
|            | VXV    | DOT      |                            |          |
|            |        | 00D      |                            |          |

July 12, 2016

1123

|         |          |                                       |              |
|---------|----------|---------------------------------------|--------------|
| STADR   |          | #                                     | PL12D        |
| STOVL   | 12D      | # (URP3 X V).(UVP3 X URP3)=TEMP       | PL06D        |
| DOT     | SL1      | #                                     | PL00D        |
| ARCCOS  | SIGN     |                                       |              |
|         | 12D      | #                                     | B0           |
| SR1     | PUSH     | # GAMMA = SIGN(TEMP)ARCOS(UNITV.URP3) | PL02D        |
| BON     | DLOAD    |                                       |              |
|         | S32.1F2  |                                       |              |
|         | FRSTPAS  |                                       |              |
|         | OOD      | # NOT THE FIRST PASS OF A CYCLE       |              |
| DSU     | PDDL     | # GAMMA-GAMPREV                       | B1 PL04D     |
|         | GAMPREV  |                                       |              |
|         | DELVCSI  |                                       |              |
| DSU     | NORM     | #                                     | B7           |
|         | DVPREV   |                                       |              |
|         | X1       |                                       |              |
| BDDV    | PDDL     | # (GAM-GAMPREV)/(DV-DVPREV)           | B-6+N1 PL06D |
|         | O2D      | # = SLOPE                             |              |
|         | DELVCSI  |                                       |              |
| STORE   | DVPREV   |                                       |              |
| BOFF    | BOFF     |                                       |              |
|         | S32.1F3A |                                       |              |
|         | THRDCHK  |                                       |              |
|         | S32.1F3B |                                       |              |
|         | THRDCHK  |                                       |              |
| DLOAD   | DMP      |                                       |              |
|         | O2D      |                                       |              |
|         | GAMPREV  |                                       |              |
| BPL     | DLOAD    |                                       |              |
|         | FIFTYFPS |                                       |              |
|         | INITST1  |                                       |              |
| SIGN    |          |                                       |              |
|         | DELDV    |                                       |              |
| STORE   | DELDV    |                                       |              |
| SET     | CLEAR    |                                       |              |
|         | S32.1F3A |                                       |              |
|         | S32.1F3B |                                       |              |
| FRSTPAS | DLOAD    |                                       |              |
|         | OOD      |                                       |              |
| STODL   | GAMPREV  |                                       |              |
|         | DELVCSI  |                                       |              |
| STCALL  | DVPREV   |                                       |              |
|         | CSINEXT3 |                                       |              |
| SETLOC  | CSIPROG8 |                                       |              |
| BANK    |          |                                       |              |

# Page 672

|          |        |                                               |
|----------|--------|-----------------------------------------------|
| CSINEXT3 | DSU    | CLEAR<br>DELDV<br>S32.1F2                     |
|          | STCALL | DELVCSI<br>CSI/B1                             |
| THRDCHK  | BON    | BON<br>S32.1F3A<br>NEWTN<br>S32.1F3B<br>NEWTN |
| FIFTYFPS | DLOAD  | SIGN<br>FIFPSDP<br>04D                        |
|          | SIGN   | GAMPREV                                       |
|          | STORE  | DELDV                                         |
|          | DCOMP  | DAD<br>DELVCSI                                |
|          | STODL  | DELVCSI<br>OOD                                |
|          | SET    | SET<br>S32.1F3B<br>S32.1F3A                   |
|          | STCALL | GAMPREV<br>CSI/B2                             |
| NEWTN    | DLOAD  | NORM<br>04D<br>X2                             |
|          | BDDV   | XSU,1<br>OOD<br>X2                            |
|          | SR*    | 0,1                                           |
|          | STODL  | DELDV<br>OOD                                  |
|          | STORE  | GAMPREV                                       |
|          | DLOAD  | ABS<br>DELDV                                  |
|          | PUSH   | DSU<br>EPSILN1                                |
|          | BMN    | DLOAD<br>CSI/SOL                              |
|          | DSU    | BMN<br>DELMAX1                                |

#

PL081

|            |        |            |
|------------|--------|------------|
|            |        | CSISTEP    |
|            | DLOAD  | SIGN       |
|            |        | DELMAX1    |
|            |        | DELDV      |
|            | STORE  | DELDV      |
| CSISTEP    | DLOAD  | DSU        |
| # Page 673 |        |            |
|            |        | DELVCSI    |
|            |        | DELDV      |
|            | STCALL | DELVCSI    |
|            |        | CSI/B1     |
| CSI/SOL    | DLOAD  | AXT,2      |
|            |        | POSTCSI    |
|            |        | 2          |
|            | LXA,1  | GOTO       |
|            |        | RTX1       |
|            |        | CSINEXT4   |
|            | SETLOC | CSIPROG9   |
|            | BANK   |            |
| CSINEXT4   | DSU*   | BMN        |
|            |        | PMINE -2,1 |
|            |        | SCNDSOL    |
|            | AXT,2  | DLOAD      |
|            |        | 3          |
|            |        | POSTCDH    |
|            | DSU*   | BMN        |
|            |        | PMINE -2,1 |
|            |        | SCNDSOL    |
|            | DLOAD  | DSU        |
|            |        | TCDH       |
|            |        | TCSI       |
|            | STORE  | T1TOT2     |
|            | AXT,2  | DSU        |
|            |        | 4          |
|            |        | 600SEC     |
|            | BMN    | AXT,2      |
|            |        | SCNDSOL    |
|            |        | 5          |
|            | DLOAD  | DSU        |
|            |        | TTPI       |
|            |        | TCDH       |
|            | STORE  | T2TOT3     |
|            | DSU    | BPL        |
|            |        | 600SEC     |

|         |       |                                                              |
|---------|-------|--------------------------------------------------------------|
| SCNDSOL | BON   | P32/P72C<br>BOFF<br>S32.1F3A<br>ALMXIT<br>S32.1F3B<br>ALMXIT |
|         | SXA,2 | DLOAD<br>CSIALRM<br>ZEROVECS                                 |
|         | CLEAR | SET<br>S32.1F1                                               |

# Page 674

|  |        |                               |
|--|--------|-------------------------------|
|  |        | S32.1F2                       |
|  | CLEAR  | CLEAR<br>S32.1F3A<br>S32.1F3B |
|  | STCALL | LOOPCT<br>CSI/B               |

# Page 675

# \*\*\*\*\* ADVANCE \*\*\*\*\*

#

# SUBROUTINES USED

# PRECSET

# ROTATE

|        |         |
|--------|---------|
| SETLOC | CDHTAG3 |
| BANK   |         |

|         |        |                             |
|---------|--------|-----------------------------|
| ADVANCE | STQ    | DLOAD<br>SUBEXIT<br>TIG     |
|         | STCALL | TDEC1<br>PRECSET            |
|         | SET    | VLOAD<br>XDELVFLG<br>VPASS3 |
|         | STORE  | VPASS2                      |
|         | STOVL  | VPASS1<br>RPASS3            |
|         | STORE  | RPASS2                      |
|         | STORE  | RPASS1                      |
|         | UNIT   | VXV<br>VPASS1               |
|         | UNIT   |                             |
|         | STOVL  | UP1                         |

|  |        |                         |
|--|--------|-------------------------|
|  | STCALL | RACT3<br>RTIG<br>ROTATE |
|  | STORE  | RACT2                   |
|  | STOVL  | RACT1<br>VACT3          |
|  | STCALL | VTIG<br>ROTATE          |
|  | STORE  | VACT2                   |
|  | STCALL | VACT1<br>SUBEXIT        |

# Page 676

# \*\*\*\*\* ROTATE \*\*\*\*\*

|        |        |                    |
|--------|--------|--------------------|
|        | SETLOC | CDHTAG             |
|        | BANK   |                    |
| ROTATE | PUSH   | PUSH               |
|        | DOT    | VXSC<br>UP1<br>UP1 |
|        | VSL2   | BVSU               |
|        | UNIT   | PDVL               |
|        | ABVAL  | VXSC               |
|        | VSL1   | RVQ                |

# Page 677

# \*\*\*\*\* INTINTNA \*\*\*\*\*

|          |        |                                |
|----------|--------|--------------------------------|
|          | SETLOC | CDHTAG2                        |
|          | BANK   |                                |
| INTINT2C | PDDL   | PDDL<br>TCSI<br>TCDH           |
|          | PDDL   | PUSH<br>TWOPI                  |
|          | GOTO   |                                |
| INTINT3P | PDDL   | INTINT<br>PDDL<br>TCDH<br>TTPI |
|          | PDDL   | PUSH<br>ZEROVECS               |
|          | GOTO   |                                |

## INTINT

```
Page 678
***** S32/33.1 *****
#
SUBROUTINES USED
S32/33.X
```

|          |        |          |
|----------|--------|----------|
|          | SETLOC | CSI/CDH  |
|          | BANK   |          |
| S32/33.1 | STQ    | AXT,1    |
|          |        | SUBEXIT  |
|          | VN     | 0681     |
|          | CALL   |          |
|          |        | DISDVLVC |
|          | CALL   |          |
|          |        | S32/33.X |
|          | VLOAD  | VXM      |
|          |        | DELVLVC  |
|          |        | OD       |
|          | VSL1   |          |
|          | STORE  | DELVSIN  |
|          | PUSH   | ABVAL    |
|          | STOVL  | DELVSAB  |
|          | GOTO   |          |
|          |        | SUBEXIT  |

```
Page 679
***** S32/33.X *****
```

|          |        |         |
|----------|--------|---------|
|          | SETLOC | CDHTAGS |
|          | BANK   |         |
| S32/33.X | SETPD  | VLOAD   |
|          |        | 6D      |
|          |        | UP1     |
|          | VCOMP  | PDVL    |
|          |        | RACT1   |
|          | UNIT   | VCOMP   |
|          | PUSH   | VXV     |
|          |        | UP1     |
|          | VSL1   |         |
|          | STORE  | OD      |
|          | RVQ    |         |



# Page 680

# \*\*\*\*\* CDHMVR \*\*\*\*\*

#

# SUBROUTINES USED

# VECSHIFT

# TIMETHET

# SHIFTR1

SETLOC CDHTAG  
BANK

CDHMVR

STQ VLOAD  
SUBEXIT  
RACT2

PUSH UNIT

STOVL UNVEC  
RPASS2

# UR SUB A

UNIT DOT  
UNVEC

PUSH SL1

STODL CSTD

DSQ PDDL  
DP1/4

SR2 DSU

SQRT SL1

PDVL VCOMP

VXV  
RPASS2DOT PDDL  
UP1

SIGN STADR

STOVL SNTH  
RPASS2PDVL CALL  
VPASS2

VECSHIFT

STOVL VVEC

CLEAR

RVSW

STCALL RVEC

TIMETHET

LXA,2 VSL\*

RTX2

0,2

STORE 18D

DOT SL1R

# Page 681

|       |         |                                        |                    |     |
|-------|---------|----------------------------------------|--------------------|-----|
| PDVL  | UNVEC   |                                        |                    |     |
| SL*   | ABVAL   | # OD = V SUB PV                        |                    |     |
|       | PDVL    |                                        |                    |     |
|       | 0,2     |                                        |                    |     |
|       | RACT2   |                                        |                    |     |
| ABVAL | PDDL    | # 2D = LENGTH OF R SUB A               |                    |     |
| DSU   |         |                                        |                    |     |
|       | 02D     |                                        |                    |     |
| STODL | DIFFALT | # DELTA H IN METERS                    | B+29               |     |
|       | R1A     |                                        |                    |     |
| NORM  | PDDL    | # 2 - R V**/MU                         |                    | 04D |
|       | X1      |                                        |                    |     |
|       | R1      |                                        |                    |     |
| CALL  |         |                                        |                    |     |
|       | SHIFTR1 |                                        |                    |     |
| SR1R  | DDV     |                                        |                    |     |
| SL*   | PUSH    |                                        |                    |     |
|       | 0 -5,1  |                                        |                    |     |
| DSU   | PDDL    | # A SUB A                              | B+29               | 04D |
|       | DIFFALT |                                        |                    |     |
| SR2   | DDV     | # A SUB P                              | B+31               |     |
|       | 04D     | #                                      | B+2                |     |
| PUSH  | SQRT    | # A SUB P/A SUB A                      |                    | 06D |
| DMPR  | DMP     |                                        |                    |     |
|       | 06D     |                                        |                    |     |
|       | 00D     |                                        |                    |     |
| SL3R  | PDDL    | # V SUB A V METERS/CS                  | B+7                | 08D |
|       | 02D     | # R SUB A MAGNITUDE                    | B+29               |     |
| NORM  | PDDL    |                                        |                    |     |
|       | X1      |                                        |                    |     |
|       | RTMU    |                                        |                    |     |
| SR1   | DDV     | # 2MU                                  | B+38               |     |
| SL*   | PDDL    | # 2 MU/R SUBAA                         | B+14               | 10D |
|       | 0 -5,1  |                                        |                    |     |
|       | 04D     | # ASUBA                                | B+29               |     |
| NORM  | PDDL    |                                        |                    |     |
|       | X2      |                                        |                    |     |
|       | RTMU    |                                        |                    |     |
| SR1   | DDV     |                                        |                    |     |
| SL*   | BDSU    |                                        |                    |     |
|       | 0 -6,2  | # 2U/R - U/A                           | B+14 (METERS/CS)SQ |     |
| PDDL  | DSQ     | #                                      |                    | 10D |
|       | 08D     |                                        |                    |     |
| BDSU  | SQRT    |                                        |                    |     |
| PDVL  | VXV     | # SQRT(MU(2/R SUB A-1/A SUB A)-VSUBA2) |                    | 10D |

July 12, 2016

1131

```

 UP1
 UNVEC
UNIT VXSC
 10D
PDVL VXSC
 UNVEC
 08D
VAD VSL1
STADR
Page 682
STORE VACT3
VSU
 VACT2
STCALL DELVEET2
 SUBEXIT
 # DELTA VCDH -- REFERENCE COORDINATES

Page 683
***** COMPTGO *****
#
SUBROUTINES USED
CLOKTASK
2PHSCHNG

BANK 35
SETLOC CSI/CDH
BANK

EBANK= RTRN

COUNT* $$/P3575
```

This code is written to file `src/P32-P33-P72-P73.s`.

## B.71 P32-P35 P72-P75

```

1132 <src/P32-P35-P72-P75.s 1132>≡
 # Copyright: Public domain.
 # Filename: P32-P35_P72-P75.agc
 # Purpose: Part of the source code for Luminary 1A build 099.
 # It is part of the source code for the Lunar Module's (LM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 618-650
 # Mod history: 2009-05-18 RSB Adapted from the Luminary 131 file of the
 # same name, as corrected from Luminary 099
 # page images.
 #
 # This source code has been transcribed or otherwise adapted from
 # digitized images of a hardcopy from the MIT Museum. The digitization
 # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
 # the Museum. Many thanks to both. The images (with suitable reduction
 # in storage size and consequent reduction in image quality as well) are
 # available online at www.ibiblio.org/apollo. If for some reason you
 # find that the images are illegible, contact me at info@sandroid.org
 # about getting access to the (much) higher-quality images which Paul
 # actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
 # 16:27 JULY 14, 1969
 #
 # Page 618
 # COELLIPTIC SEQUENCE INITIATION (CSI) PROGRAMS (P32 AND P72)
 #
 # MOD NO -1 LOG SECTION -- P32-P35, P72-P75
 # MOD BY WHITE, P. DATE 1 JUNE 67
 #
 # PURPOSE
 #
 # (1) TO CALCULATE PARAMETERS ASSOCIATED WITH THE TIME FOLLOWING
 # CONCENTRIC FLIGHT PLAN MANEUVERS -- THE CO-ELLIPTIC SEQUENCE
 # INITIATION (CSI) MANEUVER AND THE CONSTANT DELTA ALTITUDE
 # (CDH) MANEUVER.
 #
 # (2) TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA
 # APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.

```

# (3) TO DISPLAY TO THE ASTRONAUT AND THE GROUND DEPENDENT VARIABLES  
# ASSOCIATED WITH THE CONCENTRIC FLIGHT PLAN MANEUVERS FOR  
# APPROVAL BY THE ASTRONAUT/GROUND.

# (4) TO STORE THE CSI TARGET PARAMETERS FOR USE BY THE DESIRED  
# THRUSTING PROGRAM.  
#

# ASSUMPTIONS

# (1) AT A SELECTED TPI TIME THE LINE OF SIGHT BETWEEN THE ACTIVE  
# AND PASSIVE VEHICLES IS SELECTED TO BE A PRESCRIBED ANGLE (E)  
# FROM THE HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE  
# POSITION.

# (2) THE TIME BETWEEN CSI IGNITION AND CDH IGNITION MUST BE  
# COMPUTED TO BE GREATER THAN 10 MINUTES FOR SUCCESSFUL  
# COMPLETION OF THE PROGRAM.

# (3) THE TIME BETWEEN CDH IGNITION AND TPI IGNITION MUST BE  
# COMPUTED TO BE GREATER THAN 10 MINUTES FOR SUCCESSFUL  
# COMPLETION OF THE PROGRAM.

# (4) CDH DELTA V IS SELECTED TO MINIMIZE THE VARIATION OF THE  
# ALTITUDE DIFFERENCE BETWEEN THE ORBITS.

# (5) CSI BURN IS DEFINED SUCH THAT THE IMPULSIVE DELTA V IS IN THE  
# HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION AT CSI  
# IGNITION.

# (6) THE PERICENTER ALTITUDE OF THE ORBIT FOLLOWING CSI AND CDH  
# MUST BE GREATER THAN 35,000 FT (LUNAR ORBIT) OR 85 NM (EARTH  
# ORBIT) FOR SUCCESSFUL COMPLETION OF THIS PROGRAM.

# (7) THE CSI AND CDH MANEUVERS ARE ORIGINALLY ASSUMED TO BE  
# PARALLEL TO THE PLANE OF THE CSM ORBIT. HOWEVER, CREW  
# Page 619  
# MODIFICATION OF DELTA V (LV) COMPONENTS MAY RESULT IN AN  
# OUT-OF-PLANE CSI MANEUVER

# (8) STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC  
# STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION 10).

# (9) COMPUTED VARIABLES MAY BE STORED FOR LATER VERIFICATION BY  
# THE GROUND. THESE STORAGE CAPABILITIES ARE NORMALLY LIMITED  
# ONLY TO THE PARAMETERS FOR ONE THRUSTING MANEUVER AT A TIME

```

EXCEPT FOR CONCENTRIC FLIGHT PLAN MANEUVER SEQUENCES.

(10) THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM
OR CSM STATE VECTORS FOR THIS PROGRAM. IF RADAR USE IS
DESIRED THE RADAR WAS TURNED ON AND LOCKED BY THE CSM BY
PREVIOUS SELECTION OF P20. RADAR SIGHTING MARKS WILL BE MADE
AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE
TRACK AND UPDATE FLAGS (SEE P20). THE RENDEZVOUS TRACKING
MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH
THRUSTING MANEUVER.

(11) THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.

(12) THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
CALCULATES THE MANEUVER PARAMETERS. SET AT THE START OF
EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
COMPLETED THE FINAL MANEUVER COMPUTATION AND DISPLAY
CYCLE.
#
EXTERNAL DELTA V STEERING FLAG -- DESIGNATES THE TYPE OF
STEERING REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE
THRUSTING PROGRAM SELECTED AFTER COMPLETION OF THIS
PROGRAM.

(13) IT IS NORMALLY REQUIRED THAT THE ISS BE ON FOR 1 HOUR PRIOR TO
A THRUSTING MANEUVER.

(14) THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY
#
P32 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
P72 IF THIS VEHICLE IS THE PASSIVE VEHICLE.
#
INPUT

(1) TCSI TIME OF THE CSI MANEUVER

Page 620
(2) NN NUMBER OF APSIDAL CROSSINGS THRU WHICH THE ACTIVE
VEHICLE ORBIT CAN BE ADVANCED TO OBTAIN THE CDH
MANEUVER POINT.

```

July 12, 2016

1135

```
(3) ELEV DESIRED LOS ANGLE AT TPI
(4) TTPI TIME OF THE TPI MANEUVER
#
OUTPUT
#
(1) TRKMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO
(3) +MGA MIDDLE GIMBAL ANGLE
(4) DIFFALT DELTA ALTITUDE AT CDH
(5) T1TOT2 DELTA TIME FROM CSI TO CDH
(6) T2TOT3 DELTA TIME FROM CDH TO TPI
(7) DELVLVC DELTA VELOCITY AT CSI -- LOCAL VERTICAL COORDINATES
(8) DELVLVC DELTA VELOCITY AT CDH -- LOCAL VERTICAL COORDINATES
#
DOWNLINK
#
(1) TCSI TIME OF THE CSI MANEUVER
(2) TCDH TIME OF THE CDH MANEUVER
(3) TTPI TIME OF THE TPI MANEUVER
(4) TIG TIME OF THE CSI MANEUVER
(5) DELVEET1 DELTA VELOCITY AT CSI -- REFERENCE COORDINATES
(6) DELVEET2 DELTA VELOCITY AT CDH -- REFERENCE COORDINATES
(7) DIFFALT DELTA ALTITUDE AT CDH
(8) NN NUMBER OF APSIDAL CROSSINGS THRU WHICH THE ACTIVE
VEHICLE ORBIT CAN BE ADVANCED TO OBTAIN THE CDH
MANEUVER POINT
(9) ELEV DESIRED LOS ANGLE AT TPI
#
COMMUNICATION TO THRUSTING PROGRAM
#
(1) TIG TIME OF THE CSI MANEUVER
(2) RTIG POSITION OF ACTIVE VEHICLE AT CSI -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE
(3) VTIG VELOCITY OF ACTIVE VEHICLE AT CSE -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE
(4) DELVSIN DELTA VELOCITY AT CSI -- REFERENCE COORDINATES
(5) DELVSAB MAGNITUDE OF DELTA VELOCITY AT CSI
(6) XDELVFLG SET TO INDICATE EXTERNAL DELTA V VG COMPUTATION
#
SUBROUTINES USED
#
AVFLAGA
AVFLAGP
P20FLGON
VARALARM
BANKCALL
```

```

GOFLASH
GOTOP00H
Page 621
VNPO0H
GOFLASHR
BLANKET
ENDOFJOB
SELECTMU
ADVANCE
INTINT
PASSIVE
CSI/A
S32/33.1
DISDVLVC
VN1645

```

```

 BANK 35
 SETLOC CSI/CDH
 BANK
 EBANK= SUBEXIT
 COUNT* $$/P3272
P32 TC AVFLAGA
 TC P32STRT
P72 TC AVFLAGP
P32STRT EXTEND
 DCA P30ZERO
 DXCH CENTANG
 TC P32/P72A
ALMXITA SXA,2
 CSIALRM
ALMXIT LXC,1
 CSIALRM
 SLOAD* EXIT
 ALARM/TB -1,1
 CA MPAC
 TC VARALARM
 CAF V05N09
 TC BANKCALL
 CADR GOFLASH
 TC GOTOP00H
 TC -4
P32/P72A TC P20FLGON
 CAF P30ZERO
 TS NN +1
 TS TCSI
 TS TCSI +1

```



July 12, 2016

1137

|            |        |          |                   |
|------------|--------|----------|-------------------|
| VN0611     | CAF    | V06N11   | # TCSI            |
|            | TC     | VNP00H   |                   |
|            | TC     | INTPRET  |                   |
|            | DLOAD  | DCOMP    |                   |
|            |        | TCSI     |                   |
|            | BMN    | DLOAD    |                   |
|            |        | VN0655   |                   |
| # Page 622 |        |          |                   |
|            |        | TETLEM   |                   |
|            | STCALL | TDEC1    |                   |
|            |        | PRECSET  |                   |
|            | VLOAD  | VSR*     |                   |
|            |        | RACT3    |                   |
|            |        | 0,2      |                   |
|            | STOVL  | RVEC     |                   |
|            |        | VACT3    |                   |
|            | VSR*   | SET      |                   |
|            |        | 0,2      |                   |
|            |        | RVSW     |                   |
|            | STODL  | VVEC     |                   |
|            |        | DPPOSMAX |                   |
|            | STCALL | RDESIRED |                   |
|            |        | TIMERAD  |                   |
|            | DAD    |          |                   |
|            |        | TDEC2    |                   |
|            | STORE  | TCSI     |                   |
|            | EXIT   |          |                   |
|            | TC     | VN0611   |                   |
| VN0655     | EXIT   |          |                   |
|            | CAF    | V06N55   | # NN, ELEV(RGLOS) |
|            | TC     | BANKCALL |                   |
|            | CADR   | GOFLASH  |                   |
|            | TC     | GOTOP00H |                   |
|            | TC     | +2       |                   |
|            | TC     | -5       |                   |
|            | CAF    | V06N37   | # TTPI            |
|            | TC     | VNP00H   |                   |
|            | TC     | INTPRET  |                   |
|            | DLOAD  |          |                   |
|            |        | TCSI     |                   |
|            | STCALL | TIG      |                   |
|            |        | SELECTMU |                   |
| P32/P72B   | CALL   |          |                   |
|            |        | ADVANCE  |                   |
|            | SETPD  | VLOAD    |                   |
|            |        | OD       |                   |

|            |       |          |
|------------|-------|----------|
|            |       | VPASS1   |
|            | PDVL  | PDDL     |
|            |       | RPASS1   |
|            |       | TCSI     |
|            | PDDL  | PDDL     |
|            |       | TTPI     |
|            |       | TWOPI    |
|            | PUSH  | CALL     |
|            |       | INTINT   |
|            | CALL  |          |
|            |       | PASSIVE  |
|            | CALL  |          |
| # Page 623 |       |          |
|            |       | CSI/A    |
| P32/P72C   | BON   | SET      |
|            |       | FINALFLG |
|            |       | P32/P72D |
|            |       | UPDATFLG |
| P32/P72D   | DLOAD |          |
|            |       | T1TOT2   |
| P32/P72E   | STORE | T1TOT2   |
|            | DSU   | BPL      |
|            |       | 60MIN    |
|            |       | P32/P72E |
|            | DLOAD |          |
|            |       | T2TOT3   |
| P32/P72F   | STORE | T2TOT3   |
|            | DSU   | BPL      |
|            |       | 60MIN    |
|            |       | P32/P72F |
|            | EXIT  |          |
|            | CAF   | V06N75   |
|            | TC    | VNP00H   |
|            | TC    | INTPRET  |
|            | VLOAD | CALL     |
|            |       | DELVEET1 |
|            |       | S32/33.1 |
|            | STOVL | DELVEET1 |
|            |       | RACT2    |
|            | STOVL | RACT1    |
|            |       | DELVEET2 |
|            | AXT,1 | CALL     |
|            | VN    | 0682     |
|            |       | DISDVLVC |
|            | DLOAD |          |
|            |       | TTPI     |

July 12, 2016

1139

STCALL TTPIO  
VN1645  
GOTO  
P32/P72B

# Page 624

# CONSTANT DELTA HEIGHT (CDH) PROGRAMS (P33 AND P73)

# MOD NO -1 LOC SECTION -- P32-P35, P72-P75

# MOD BY WHITE, P. DATE: 1 JUNE 67

#

# PURPOSE

#

# (1) TO CALCULATE PARAMETERS ASSOCIATED WITH THE CONSTANT DELTA  
# ALTITUDE MANEUVER (CDH).

#

# (2) TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA  
# APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.

#

# (3) TO DISPLAY TO THE ASTRONAUT AND THE GROUND DEPENDENT VARIABLES  
# ASSOCIATED WITH THE CDH MANEUVER FOR APPROVAL BY THE  
# ASTRONAUT/GROUND.

#

# (4) TO STORE THE CDH TARGET PARAMETERS FOR USE BY THE DESIRED  
# THRUSTING PROGRAM.

#

# ASSUMPTIONS

#

# (1) THIS PROGRAM IS BASED UPON PREVIOUS COMPLETION OF THE  
# CO-ELLIPTIC SEQUENCE INITIATION (CSI) PROGRAM (P32/P72).  
# THEREFORE --

#

# (A) AT A SELECTED TPI TIME (NOW IN STORAGE) THE LINE OF SIGHT  
# BETWEEN THE ACTIVE AND PASSIVE VEHICLES WAS SELECTED TO BE  
# A PRESCRIBED ANGLE (E) (NOW IN STORAGE) FROM THE  
# HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION.

#

# (B) THE TIME BETWEEN CSI IGNITION AND CDH IGNITION WAS  
# COMPUTED TO BE GREATER THAN 10 MINUTES.

#

# (C) THE TIME BETWEEN CDH IGNITION AND TPI IGNITION WAS  
# COMPUTED TO BE GREATER THAN 10 MINUTES.

#

# (D) THE VARIATION OF THE ALTITUDE DIFFERENCE BETWEEN THE  
# ORBITS WAS MINIMIZED.

#

# (E) CSI BURN WAS DEFINED SUCH THAT THE IMPULSIVE DELTA V WAS

# IN THE HORIZONTAL PLANE DEFINED BY ACTIVE VEHICLE  
# POSITION AT CSI IGNITION.  
#  
# (F) THE PERICENTER ALTITUDES OF THE ORBITS FOLLOWING CSI AND  
# CDH WERE COMPUTED TO BE GREATER THAN 35,000 FT FOR LUNAR  
# ORBIT OR 85 NM FOR EARTH ORBIT.  
#  
# (G) THE CSI AND CDH MANEUVERS WERE ASSUMED TO BE PARALLEL TO  
# THE PLANE OF THE PASSIVE VEHICLE ORBIT. HOWEVER, CREW  
# Page 625  
# MODIFICATION OF DELTA V (LV) COMPONENTS MAY HAVE RESULTED  
# IN AN OUT-OF-PLANE MANEUVER.  
#  
# (2) STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC  
# STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION 4).  
#  
# (3) COMPUTED VARIABLES MAY BE STORED FOR LATER VERIFICATION BY  
# THE GROUND. THESE STORAGE CAPABILITIES ARE NORMALLY LIMITED  
# ONLY TO THE PARAMETERS FOR ONE THRUSTING MANEUVER AT A TIME  
# EXCEPT FOR CONCENTRIC FLIGHT PLAN MANEUVER SEQUENCES.  
#  
# (4) THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM.  
# OR CSM STATE VECTORS FOR THIS PROGRAM. IF RADAR USE IS  
# DESIRED THE RADAR WAS TURNED ON AND LOCKED ON THE CSM BY  
# PREVIOUS SELECTION OF P20. RADAR SIGHTING MARKS WILL BE MADE  
# AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE  
# TRACK AND UPDATE FLAGS (SEE P20). THE RENDEZVOUS TRACKING  
# MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH  
# THRUSTING MANEUVER.  
#  
# (5) THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.  
#  
# (6) THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --  
#  
# ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS  
# DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH  
# CALCULATES THE MANEUVER PARAMETERS. SET AT THE START OF  
# EACH RENDEZVOUS PRE-THRUSTING PROGRAM.  
#  
# FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS  
# COMPLETED THE FINAL MANEUVER COMPUTATION AND DISPLAY  
# CYCLE.  
#  
# EXTERNAL DELTA V STEERING FLAG -- DESIGNATES THE TYPE OF  
# STEERING REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE  
# THRUSTING PROGRAM SELECTED AFTER COMPLETION OF THIS

```

PROGRAM.
#
(7) IT IS NORMALLY REQUIRED THAT THE ISS BE ON FOR 1 HOUR PRIOR TO
A THRUSTING MANEUVER.
#
(8) THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY.
#
P33 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
P73 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
INPUT
#
(1) TTPIO TIME OF THE TPI MANEUVER -- SAVED FROM P32/P72
Page 626
(2) ELEV DESIRED LOS ANGLE AT TPI -- SAVED FROM P32/P72
(3) TCDH TIME OF THE CDH MANEUVER
#
OUTPUT
#
(1) TRMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO
(3) +MGA MIDDLE GIMBAL ANGLE
(4) DIFFALT DELTA ALTITUDE AT CDH
(5) T2TOT3 DELTA TIME FROM CDH TO COMPUTED TPI
(6) NOMTPI DELTA TIME FROM NOMINAL TPI TO COMPUTED TPI
(7) DELVLVC DELTA VELOCITY AT CDH -- LOCAL VERTICAL COORDINATES
#
DOWNLINK
#
(1) TCDH TIME OF THE CDH MANEUVER
(2) TTPI TIME OF THE TPI MANEUVER
(3) TIG TIME OF THE CDH MANEUVER
(4) DELLVEET2 DELTA VELOCITY AT CDH -- REFERENCE COORDINATES
(5) DIFFALT DELTA ALTITUDE AT CDH
(6) ELEV DESIRED LOS ANGLE AT TPI
#
COMMUNICATION TO THRUSTING PROGRAMS
#
(1) TIG TIME OF THE CDH MANEUVER
(2) RTIG POSITION OF ACTIVE VEHICLE AT CDH -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE.
(3) VTIG VELOCITY OF ACTIVE VEHICLE AT CDH -- BEFORE ROTATION
INTO PLANE OF PASSIVE VEHICLE.
(4) DELVSIN DELTA VELOCITY AT CDH -- REFERENCE COORDINATES.
(5) DELVSAB MAGNITUDE OF DELTA VELOCITY AT CDH.

```

```

(6) XDELVFLG SET TO INDICATE EXTERNAL DELTA V VG COMPUTATION.
#
SUBROUTINES USED
#
AVFLAGA
AVFLAGP
P20FLGON
VNPOOH
SELECTMU
ADVANCE
CDHMVR
INTINT3P
ACTIVE
PASSIVE
S33/S34.1
ALARM
BANKCALL
GOFLASH
GOTOP00H
S32/33.1
Page 627
VN1645

```

```

COUNT* $$/P3373
P33 TC AVFLAGA
 TC P33/P73A
P73 TC AVFLAGP
P33/P73A TC P20FLGON
 CAF V06N13 # TCDH
 TC VNPOOH
 TC INTPRET
 DLOAD
 TPIO
 STODL TTPI
 TCDH
 STCALL TIG
 SELECTMU
P33/P73B CALL
 ADVANCE
 CALL
 CDHMVR
 SETPD VLOAD
 OD
 VACT3
 PDVL CALL
 RACT2

```

|            |        |          |
|------------|--------|----------|
|            |        | INTINT3P |
|            | CALL   | ACTIVE   |
|            | SETPD  | VLOAD    |
|            |        | OD       |
|            |        | VPASS2   |
|            | PDVL   | CALL     |
|            |        | RPASS2   |
|            |        | INTINT3P |
|            | CALL   | PASSIVE  |
|            | DLOAD  | SET      |
|            |        | P30ZERO  |
|            |        | ITSWICH  |
|            | STCALL | NOMTPI   |
|            |        | S33/34.1 |
|            | BZE    | EXIT     |
|            |        | P33/P73C |
|            | TC     | ALARM    |
|            | OCT    | 611      |
|            | CAF    | V05N09   |
|            | TC     | BANKCALL |
|            | CADR   | GOFLASH  |
|            | TC     | GOTOP00H |
|            | TC     | +2       |
| # Page 628 |        |          |
|            | TC     | P33/P73A |
|            | TC     | INTPRET  |
|            | DLOAD  | P30ZERO  |
|            | STORE  | NOMTPI   |
| P33/P73C   | BON    | SET      |
|            |        | FINALFLG |
|            |        | P33/P73D |
|            |        | UPDATFLG |
| P33/P73D   | DLOAD  | DAD      |
|            |        | NOMTPI   |
|            |        | TTPI     |
|            | STORE  | TTPI     |
|            | DSU    |          |
|            |        | TCDH     |
| P33/P73E   | DSU    | BPL      |
|            |        | 60MIN    |
|            |        | P33/P73E |
|            | DAD    |          |
|            |        | 60MIN    |

```

 STODL T1TOT2
 TTPI
 DSU PUSH
 TTPIO
P33/P73F ABS DSU
 60MIN
 BPL DAD
 P33/P73F
 60MIN
 SIGN STADR
 STORE T2TOT3
 EXIT
 CAF V06N75
 TC VNP00H
 TC INTPRET
 VLOAD CALL
 DELVEET2
 S32/33.1
 STCALL DELVEET2
 VN1645
 GOTO
 P33/P73B

```

```

Page 629
***** ADFLAG/P *****
#
SUBROUTINES USED
#
UPFLAG
DOWNFLAG

```

```

AVFLAGA EXTEND # AVFLAG = LEM
 QXCH SUBEXIT
 TC UPFLAG
 ADRES AVFLAG
 TC SUBEXIT
AVFLAGP EXTEND # AVFLAG = CSM
 QXCH SUBEXIT
 TC DOWNFLAG
 ADRES AVFLAG
 TC SUBEXIT
P20FLGON EXTEND
 QXCH SUBEXIT
 TC UPFLAG
 ADRES UPDATFLG # SET UPDATFLG
 TC UPFLAG

```



July 12, 2016

1145

|                         | ADRES    | TRACKFLG    | # SET TRACKFLG |
|-------------------------|----------|-------------|----------------|
|                         | TC       | SUBEXIT     |                |
| # Page 630              |          |             |                |
| # ***** DISDVLVC *****  |          |             |                |
| #                       |          |             |                |
| # SUBROUTINES USED      |          |             |                |
| #                       |          |             |                |
| #                       | S32/33.X |             |                |
| #                       | VNPOOH   |             |                |
| DISDVLVC                |          |             |                |
|                         | STORE    | DELVLVC     |                |
|                         | STQ      | CALL        |                |
|                         |          | NORMEX      |                |
|                         |          | S32/33.X    |                |
|                         | VLOAD    | MXV         |                |
|                         |          | DELVLVC     |                |
|                         |          | OD          |                |
|                         | VSL1     | SXA,1       |                |
|                         |          | VERBNOUN    |                |
|                         | STORE    | DELVLVC     |                |
|                         | EXIT     |             |                |
|                         | CA       | VERBNOUN    |                |
|                         | TC       | VNPOOH      |                |
|                         | TC       | INTPRET     |                |
|                         | GOTO     |             |                |
|                         |          | NORMEX      |                |
| # Page 631              |          |             |                |
| # ***** CONSTANTS ***** |          |             |                |
| V06N11                  | VN       | 0611        |                |
| V06N13                  | VN       | 0613        |                |
| V06N75                  | VN       | 0675        |                |
| SN359+                  | 2DEC     | -.000086601 |                |
| CS359+                  | 2DEC     | +.499999992 |                |
| P30ZERO                 | 2DEC     | 0           |                |
| 60MIN                   | 2DEC     | 360000      |                |
| ALARM/TB                | OCT      | 00600       | # NO 1         |
|                         | OCT      | 00601       | # 2            |
|                         | OCT      | 00602       | # 3            |
|                         | OCT      | 00603       | # 4            |

|     |       |   |   |
|-----|-------|---|---|
| OCT | 00604 | # | 5 |
| OCT | 00605 | # | 6 |
| OCT | 00606 | # | 7 |

```
Page 632
***** CSI/A *****
#
SUBROUTINES USED
#
VECSHIFT
TIMETHET
PERIAP0
SHIFTR1
INTINT2C
CDHMVR
PERIAP01
INTINT
ACTIVE
```

|          |        |              |                                           |
|----------|--------|--------------|-------------------------------------------|
|          | BANK   | 34           |                                           |
|          | SETLOC | CSI/CDH1     |                                           |
|          | BANK   |              |                                           |
|          | EBANK= | SUBEXIT      |                                           |
|          | COUNT* | \$\$/CSI     |                                           |
| LOOPMX   | 2DEC   | 16           |                                           |
| INITST   | 2DEC   | .03048 B-7   | # INITIAL DELDV = 10 FPS                  |
| DVMAX1   | 2DEC   | 3.0480 B-7   | # MAXIMUM DV1 = 1000 FPS                  |
| DVMAX2   | 2DEC   | 3.014472 B-7 | # 989 FPS                                 |
| 1DPB2    | 2DEC   | 1.0 B-2      |                                           |
| 1DPB28   | 2DEC   | 1            |                                           |
| PMINE    | 2DEC   | 157420 B-29  | # 85 NM -- MUST BE 8 WORDS BEFORE PMINM   |
| EPSILN1  | 2DEC   | .0003048 B-7 | # .1 FPS                                  |
| NICKELDP | 2DEC   | .021336 B-7  | # 7 FPS (CHANGED FROM .05 FPS)            |
| FIFPSDP  | 2DEC   | -.152400 B-7 | # 50 FPS                                  |
| PMINM    | 2DEC   | 10668 B-29   | # 35000 FT -- MUST BE 8 WORDS AFTER PMINE |

July 12, 2016

1147

|            |       |              |                                             |                   |
|------------|-------|--------------|---------------------------------------------|-------------------|
| DELMAX1    | 2DEC  | .6096000 B-7 | # 200 FPS                                   |                   |
| ONETHTH    | 2DEC  | .0001 B-3    |                                             |                   |
| TMIN       | 2DEC  | 60000        | # 10 MIN                                    |                   |
| CSI/A      | CLEAR | SET          | # INITIALIZE INDICATORS                     |                   |
|            |       | S32.1F1      | # DVT1 HAS EXCEEDED MAX INDICATOR           |                   |
|            |       | S32.1F2      | # FIRST PASS FOR NEWTON ITERATION INDICATOR |                   |
| # Page 633 |       |              |                                             |                   |
|            | CLEAR | SET          |                                             |                   |
|            |       | S32.1F3A     | # 00=1ST 2 PASSES 2ND CYCLE, 01=FIRST CYCLE |                   |
|            |       | S32.1F3B     | # 10=2ND CYCLE, 11=50 FPS STAGE 2ND CYCLE   |                   |
|            | DLOAD |              |                                             |                   |
|            |       | P30ZERO      |                                             |                   |
|            | STORE | LOOPCT       |                                             |                   |
|            | STORE | CSIALRM      |                                             |                   |
| CSI/B      | SETPD | VLOAD        |                                             |                   |
|            |       | OD           |                                             |                   |
|            |       | RACT1        |                                             |                   |
|            | ABVAL | PUSH         | # RA1                                       | B29 PL02D         |
|            | NORM  | SR1          |                                             |                   |
|            |       | X2           | #                                           | B29-N2+ B1 PL04D  |
|            | PDVL  | ABVAL        |                                             |                   |
|            |       | RPASS3       |                                             |                   |
|            | NORM  | BDDV         | # RA1/RP3                                   | B1 PL02D          |
|            |       | X1           |                                             |                   |
|            | XSU,2 | SR*          | #                                           | B2                |
|            |       | X1           |                                             |                   |
|            |       | 1,2          |                                             |                   |
|            | DAD   | DMP          | # (1+(RA1/RP3))RA1                          | B29+B2=B31 PL00D  |
|            |       | 1DPB2        |                                             |                   |
|            | NORM  | PDDL         | #                                           | PL02D             |
|            |       | X1           |                                             |                   |
|            |       | RTMU         |                                             |                   |
|            | SR1   | DDV          | #                                           | B38-B31= B7 PL00D |
|            | SL*   | SQRT         | #                                           | B7                |
|            |       | 0            | -7,1                                        |                   |
|            | PDVL  | UNIT         | #                                           | PL02D             |
|            |       | RACT1        |                                             |                   |
|            | PDVL  | VXV          |                                             |                   |
|            |       | UP1          |                                             |                   |
|            | UNIT  |              | # UNIT(URP1 X UVP1 X URA1) = UH1            |                   |
|            | DOT   | SL1          | # VA1 . UH1                                 | B7                |
|            |       | VACT1        |                                             |                   |
|            | BDSU  | STADR        | #                                           | PL00D             |

|            |       |          |                                   |
|------------|-------|----------|-----------------------------------|
|            | STODL | DELVCSI  |                                   |
|            |       | INITST   | # 10 FPS                          |
|            | STORE | DELDV    |                                   |
| CSI/B1     | DLOAD | DAD      | # IF LOOPCT = 16                  |
|            |       | LOOPCT   |                                   |
|            |       | 1DPB28   |                                   |
|            | STORE | LOOPCT   |                                   |
|            | DSU   | AXT,2    |                                   |
|            |       | LOOPMX   |                                   |
|            |       | 6        |                                   |
|            | BPL   |          |                                   |
|            |       | SCNDSOL  |                                   |
| CSI/B2     | SETPD |          |                                   |
|            |       | OD       |                                   |
| # Page 634 |       |          |                                   |
|            | DLOAD | ABS      |                                   |
|            |       | DELVCSI  |                                   |
|            | DSU   | BMN      |                                   |
|            |       | DVMAX1   |                                   |
|            |       | CSI/B23  |                                   |
|            | AXT,2 | BON      |                                   |
|            |       | 7        |                                   |
|            |       | S32.1F1  |                                   |
|            |       | SCNDSOL  |                                   |
|            | BOFF  | BON      |                                   |
|            |       | S32.1F3A |                                   |
|            |       | CSI/B22  | # FLAG 3 NEQ 3                    |
|            |       | S32.1F3B |                                   |
|            |       | SCNDSOL  |                                   |
| CSI/B22    | SET   | DLOAD    |                                   |
|            |       | S32.1F1  |                                   |
|            |       | DVMAX2   |                                   |
|            | SIGN  |          |                                   |
|            |       | DELVCSI  |                                   |
|            | STORE | DELVCSI  |                                   |
| CSI/B23    | VLOAD | PUSH     |                                   |
|            |       | RACT1    |                                   |
|            | UNIT  | PDVL     |                                   |
|            |       | UP1      |                                   |
|            | VXV   | UNIT     | # UNIT (URP1 X UVP1 X URA1) = UH1 |
|            | VXSC  | VSL1     |                                   |
|            |       | DELVCSI  |                                   |
|            | STORE | DELVEET1 |                                   |
|            | VAD   | BOV      |                                   |
|            |       | VACT1    |                                   |
|            |       | CSI/B23D |                                   |

July 12, 2016

1149

CSI/B23D

STCALL VACT4  
VECSHIFT  
STOVL VVEC  
SET

RVSW  
STOVL RVEC  
SN359+

STCALL SNTH  
TIMETHET

# ALSO Csth

SR1 LXA,1  
RTX1

STCALL HAFPA1  
PERIAPO

CALL  
SHIFTR1

STODL POSTCSI  
CENTANG

BZE GOTO  
+2

# Page 635

CIRCL  
DLOAD

ECC  
DSU BMN  
ONETHTH  
CIRCL

DLOAD CALL  
R1  
SHIFTR1

SETPD NORM  
2D

X1  
PDVL DOT

#

PL04D

RACT1  
VACT4

ABS DDV  
02D

# (/RDOTV/)/R1

B38-B29= B7

SL\* DSU  
0,1  
NICKELDP

BMN DLOAD  
CIRCL  
P

SL2 DSU  
1DPB2

STODL 14D

|            |        |          |                          |                       |
|------------|--------|----------|--------------------------|-----------------------|
|            |        | RTSR1/MU |                          |                       |
|            | SR1    | DDV      | # (1/ROOTMU)/R1          | B-16-B29 = B-45 PL02D |
|            | PDDL   | DMP      |                          |                       |
|            |        | P        |                          |                       |
|            |        | R1       |                          |                       |
|            | CALL   |          |                          |                       |
|            |        | SHIFTR1  |                          |                       |
|            | SL4    | SL1      |                          |                       |
|            | SQRT   | DMP      | # ((P/MU)**.5)/R1        | B14+B-14 = B-31 PL02D |
|            | BOFF   | SL3      |                          |                       |
|            |        | CMOONFLG |                          |                       |
|            |        | CSI/B3   |                          |                       |
| CSI/B3     | PDVL   | DOT      |                          |                       |
|            |        | RACT1    |                          |                       |
|            |        | VACT4    |                          |                       |
|            | STORE  | RDOTV    |                          |                       |
|            | ABS    |          |                          |                       |
|            | NORM   | DMP      | # ((P/MU)**.5)RDOTV/R1   | PL02D                 |
|            |        | X2       |                          |                       |
|            | XSU,1  | SL*      | #                        | B-31+B36-B3 = B2      |
|            |        | X2       |                          |                       |
|            |        | 3,1      |                          |                       |
|            | STODL  | 12D      |                          |                       |
|            |        | P30ZERO  |                          |                       |
| # Page 636 |        |          |                          |                       |
|            | STORE  | 16D      |                          |                       |
|            | VLOAD  | UNIT     |                          |                       |
|            |        | 12D      |                          |                       |
|            | STOVL  | SNTH     | # ALSO STORES CSTH AND 0 |                       |
|            |        | RACT1    |                          |                       |
|            | PDVL   | SIGN     |                          |                       |
|            |        | VACT4    |                          |                       |
|            |        | RDOTV    |                          |                       |
|            | VCOMP  | CALL     |                          |                       |
|            |        | VECSHIFT |                          |                       |
|            | STOVL  | VVEC     |                          |                       |
|            | SET    |          |                          |                       |
|            |        | RVS      |                          |                       |
|            | STCALL | RVEC     |                          |                       |
|            |        | TIMETHET |                          |                       |
|            | PDDL   | BPL      |                          |                       |
|            |        | RDOTV    |                          |                       |
|            |        | NTP/2    |                          |                       |
|            | DLOAD  | DSU      |                          |                       |
|            |        | HAFPA1   |                          |                       |
|            | PUSH   | GOTO     |                          |                       |

July 12, 2016

1151

|       |               |                                  |
|-------|---------------|----------------------------------|
| CIRCL | SETPD         | NTP/2<br>DLOAD<br>OOD<br>P30ZERO |
| NTP/2 | PUSH<br>DLOAD | DMP<br>NN<br>HAFPA1              |
|       | SL            | DSU<br>14D                       |
|       | DAD           |                                  |
|       | STORE         | TCSI                             |
|       | BDSU          | TCDH<br>AXT, 2<br>TTPI<br>5D     |
|       | BMN           | SETPD<br>SCNDSOL<br>OD           |
|       | VLOAD         | PDVL<br>VACT4<br>RACT1           |
|       | CALL          |                                  |
|       | STOVL         | INTINT2C<br>RACT2<br>VATT        |
|       | STOVL         | VACT2<br>VPASS1                  |
|       | SETPD         | PDVL                             |
|       |               | OD<br>RPASS1                     |
|       | CALL          |                                  |
|       | STOVL         | INTINT2C<br>RPASS2<br>VATT       |
|       | STCALL        | VPASS2<br>CDHMVR                 |
|       | VLOAD         | SETPD<br>RACT2<br>OD             |
|       | PDVL          | CALL<br>VACT3<br>PERIAP01        |
|       | CALL          | SHIFTR1                          |

# Page 637

|       |         |                                        |               |
|-------|---------|----------------------------------------|---------------|
| STOVL | POSTCDH |                                        |               |
|       | VACT3   |                                        |               |
| SETPD | PDVL    |                                        |               |
|       | OD      |                                        |               |
|       | RACT2   |                                        |               |
| PDDL  | PDDL    |                                        |               |
|       | TCDH    |                                        |               |
|       | TTPI    |                                        |               |
| PDDL  | PUSH    |                                        |               |
|       | TWOPI   |                                        |               |
| CALL  |         |                                        |               |
|       | INTINT  |                                        |               |
| CALL  |         |                                        |               |
|       | ACTIVE  |                                        |               |
| DLOAD |         |                                        |               |
|       | ELEV    |                                        |               |
| SETPD | SINE    |                                        |               |
|       | 6D      |                                        |               |
| PDVL  | UNIT    |                                        |               |
|       | RACT3   |                                        |               |
| STORE | OOD     | # URA3 AT OOD                          |               |
| PDVL  | VXV     | # PL14D, PL08D                         |               |
|       | UP1     |                                        |               |
| UNIT  |         |                                        |               |
| PDDL  | COSINE  | # UNIT(URA3 X UVA3 X URA3) = UH3       | B1 PI         |
|       | ELEV    |                                        |               |
| VXSC  | STADR   | # (COSLOS)(UH3)                        | B2 PI         |
| STORE | 18D     | # PLUS                                 |               |
| DLOAD | VXSC    | # (SINLOS)(URA3) = U                   | B2 PI         |
| VAD   | VSL1    |                                        |               |
|       | 18D     | #                                      | B1            |
| PUSH  | DOT     | #                                      | PI            |
|       | RACT3   | # (U . RA3) = TEMP1                    | B1 +B29 = B30 |
| SL1   | PUSH    | #                                      | B29 PI        |
| DSQ   | TLOAD   | # TEMP1**2                             | B58           |
|       | MPAC    |                                        |               |
| PDVL  | DOT     | #                                      | PI            |
|       | RACT3   |                                        |               |
|       | RACT3   |                                        |               |
| TLOAD | DCOMP   | # RA3 . RA3                            |               |
|       | MPAC    |                                        |               |
| PDVL  | DOT     | # RP3 . RP3                            | B58 PI        |
|       | RPASS3  |                                        |               |
|       | RPASS3  | #                                      | PI            |
| TAD   | TAD     | # TEMP1**2 + RA3.RA3 + RP3.RP3 = TEMP2 | PI            |



|            |        |         |                            |           |
|------------|--------|---------|----------------------------|-----------|
|            | BPL    | DLOAD   |                            |           |
|            |        | K10RK2  |                            |           |
|            |        | LOOPCT  |                            |           |
|            | DSU    | AXT,2   |                            |           |
|            |        | 1DPB28  |                            |           |
|            |        | 1D      |                            |           |
|            | BZE    |         |                            |           |
|            |        | ALMXITA |                            |           |
|            | DLOAD  | SR1     |                            |           |
|            |        | DELDV   |                            |           |
|            | STORE  | DELDV   |                            |           |
|            | BDSU   |         |                            |           |
|            |        | DVPREV  |                            |           |
|            | STCALL | DELVCSI |                            |           |
|            |        | CSI/B1  |                            |           |
| K10RK2     | SQRT   | PUSH    | # TEMP3 = TEMP2**.5        | B29 PL10D |
|            | DCOMP  | DSU     |                            |           |
|            |        | 06D     | # -TEMP1-TEMP3 = K2 AT 10D |           |
|            | STODL  | 10D     | #                          | PL08D     |
|            | DSU    | STADR   | #                          | PL06D     |
|            | STORE  | 12D     | # -TEMP1+TEMP3 = K1 AT 12D |           |
|            | ABS    |         |                            |           |
|            | STODL  | 14D     |                            |           |
|            |        | 10D     |                            |           |
|            | ABS    | DSU     |                            |           |
|            |        | 14D     |                            |           |
|            | BMN    | DLOAD   |                            |           |
|            |        | K2.     |                            |           |
|            |        | 12D     |                            |           |
|            | STORE  | 10D     | # K = K1                   |           |
| K2.        | DLOAD  |         |                            |           |
|            |        | 10D     |                            |           |
|            | VXSC   | VSL1    |                            |           |
|            | VAD    | UNIT    | # V = RA3 + KU UNIT        | B1        |
|            |        | RACT3   |                            |           |
|            | PDVL   | UNIT    |                            |           |
|            |        | RPASS3  | #                          | PL06D     |
|            | PDVL   | UNIT    |                            |           |
|            |        | VPASS3  | #                          | PL12D     |
| # Page 639 |        |         |                            |           |
|            | VXV    | PDVL    | # UVP3 X URP3              | PL18D     |
|            |        | 06D     |                            |           |
|            |        | 06D     |                            |           |
|            | VXV    | DOT     |                            |           |
|            |        | 00D     |                            |           |
|            | STADR  |         | #                          | PL12D     |

|            |        |          |                                       |        |
|------------|--------|----------|---------------------------------------|--------|
|            | STOVL  | 12D      | # (URP3 X V).(UVP3 X URP3)=TEMP       | PI     |
|            | DOT    | SL1      | #                                     | PI     |
|            | ARCCOS | SIGN     |                                       |        |
|            |        | 12D      | #                                     | B0     |
|            | SR1    | PUSH     | # GAMMA = SIGN(TEMP)ARCOS(UNITV.URP3) | PI     |
|            | BON    | DLOAD    |                                       |        |
|            |        | S32.1F2  |                                       |        |
|            |        | FRSTPAS  |                                       |        |
|            |        | OOD      | # NOT THE FIRST PASS OF A CYCLE       |        |
|            | DSU    | PDDL     | # GAMMA-GAMPREV                       | B1 PI  |
|            |        | GAMPREV  |                                       |        |
|            |        | DELVCSI  |                                       |        |
|            | DSU    | NORM     | #                                     | B7     |
|            |        | DVPREV   |                                       |        |
|            |        | X1       |                                       |        |
|            | BDDV   | PDDL     | # (GAM-GAMPREV)/(DV-DVPREV)           | B-6+N1 |
|            |        | O2D      | # = SLOPE                             |        |
|            |        | DELVCSI  |                                       |        |
|            | STORE  | DVPREV   |                                       |        |
|            | BOFF   | BOFF     |                                       |        |
|            |        | S32.1F3A |                                       |        |
|            |        | THRDCHK  |                                       |        |
|            |        | S32.1F3B |                                       |        |
|            |        | THRDCHK  |                                       |        |
|            | DLOAD  | DMP      |                                       |        |
|            |        | O2D      |                                       |        |
|            |        | GAMPREV  |                                       |        |
|            | BPL    | DLOAD    |                                       |        |
|            |        | FIFTYFPS |                                       |        |
|            |        | INITST   |                                       |        |
|            | SIGN   |          |                                       |        |
|            |        | DELDV    |                                       |        |
|            | STORE  | DELDV    |                                       |        |
|            | SET    | CLEAR    |                                       |        |
|            |        | S32.1F3A |                                       |        |
|            |        | S32.1F3B |                                       |        |
| FRSTPAS    | DLOAD  |          |                                       |        |
|            |        | OOD      |                                       |        |
|            | STODL  | GAMPREV  |                                       |        |
|            |        | DELVCSI  |                                       |        |
|            | STORE  | DVPREV   |                                       |        |
|            | DSU    | CLEAR    |                                       |        |
|            |        | DELDV    |                                       |        |
|            |        | S32.1F2  |                                       |        |
| # Page 640 | STCALL | DELVCSI  |                                       |        |

July 12, 2016

1155

|          |        |                                                         |
|----------|--------|---------------------------------------------------------|
| THRDCHK  | BON    | CSI/B1<br>BON<br>S32.1F3A<br>NEWTN<br>S32.1F3B<br>NEWTN |
| FIFTYFPS | DLOAD  | SIGN<br>FIFPSDP<br>04D                                  |
|          | SIGN   | GAMPREV                                                 |
|          | STORE  | DELDV                                                   |
|          | DCOMP  | DAD                                                     |
|          |        | DELVCSI                                                 |
|          | STODL  | DELVCSI                                                 |
|          |        | OOD                                                     |
|          | SET    | SET                                                     |
|          |        | S32.1F3B<br>S32.1F3A                                    |
|          | STCALL | GAMPREV<br>CSI/B2                                       |
| NEWTN    | DLOAD  | NORM<br>04D<br>X2                                       |
|          | BDDV   | XSU,1<br>OOD<br>X2                                      |
|          | SR*    | 0,1                                                     |
|          | STODL  | DELDV<br>OOD                                            |
|          | STORE  | GAMPREV                                                 |
|          | DLOAD  | ABS                                                     |
|          |        | DELDV                                                   |
|          | PUSH   | DSU<br>EPSILN1                                          |
|          | BMN    | DLOAD<br>CSI/SOL                                        |
|          | DSU    | BMN<br>DELMAX1<br>CSISTEP                               |
|          | DLOAD  | SIGN<br>DELMAX1<br>DELDV                                |
|          | STORE  | DELDV                                                   |
| CSISTEP  | DLOAD  | DSU                                                     |

#

PL08D

```

 DELVCSI
 DELDV
 STCALL DELVCSI
Page 641
 CSI/B1
CSI/SOL DLOAD AXT,2
 POSTCSI
 2
 LXA,1
 RTX1
 DSU* BMN
 PMINE -2,1
 SCNDSOL
 AXT,2 DLOAD
 3
 POSTCDH
 DSU* BMN
 PMINE -2,1
 SCNDSOL
 DLOAD DSU
 TCDH
 TCSI
 STORE T1TOT2
 AXT,2 DSU
 4
 TMIN
 BMN AXT,2
 SCNDSOL
 5
 DLOAD DSU
 TTPI
 TCDH
 STORE T2TOT3
 DSU BPL
 TMIN
 P32/P72C
SCNDSOL BON BOFF
 S32.1F3A
 ALMXIT
 S32.1F3B
 ALMXIT
 SXA,2 DLOAD
 CSIALRM
 P30ZERO
 CLEAR SET
 S32.1F1

```

```

 S32.1F2
 CLEAR CLEAR
 S32.1F3A
 S32.1F3B
 STCALL LOOPCT
 CSI/B

```

```

Page 642
***** ADVANCE *****
#
SUBROUTINES USED
PRECSET
ROTATE

```

```

ADVANCE STQ DLOAD
 SUBEXIT
 TIG
 STCALL TDEC1
 PRECSET
 SET VLOAD
 XDELVFLG
 VPASS3
 STORE VPASS2
 STOVL VPASS1
 RPASS3
 STORE RPASS2
 STORE RPASS1
 UNIT VXV
 VPASS1
 UNIT
 STOVL UP1
 RACT3
 STCALL RTIG
 ROTATE
 STORE RACT2
 STOVL RACT1
 VACT3
 STCALL VTIG
 ROTATE
 STORE VACT2
 STCALL VACT1
 SUBEXIT

```

```

Page 643
***** ROTATE *****

```

|        |       |      |
|--------|-------|------|
| ROTATE | PUSH  | PUSH |
|        | DOT   | VXSC |
|        |       | UP1  |
|        |       | UP1  |
|        | VSL2  | BVSU |
|        | UNIT  | PDVL |
|        | ABVAL | VXSC |
|        | VSL1  | RVQ  |

# Page 644

# \*\*\*\*\* INTINTNA \*\*\*\*\*

|          |      |         |
|----------|------|---------|
| INTINT2C | PDDL | PDDL    |
|          |      | TCSI    |
|          |      | TCDH    |
|          | PDDL | PUSH    |
|          |      | TWOPI   |
|          | GOTO |         |
|          |      | INTINT  |
| INTINT3P | PDDL | PDDL    |
|          |      | TCDH    |
|          |      | TTPI    |
|          | PDDL | PUSH    |
|          |      | P30ZERO |
|          | GOTO |         |
|          |      | INTINT  |

# Page 645

# \*\*\*\*\* S32/33.1 \*\*\*\*\*

#

# SUBROUTINES USED

# S32/33.X

|          |       |          |
|----------|-------|----------|
| S32/33.1 | STQ   | AXT,1    |
|          |       | SUBEXIT  |
|          | VN    | 0681     |
|          | CALL  |          |
|          |       | DISDVLVC |
|          | CALL  |          |
|          |       | S32/33.X |
|          | VLOAD | VXM      |
|          |       | DELVLVC  |
|          |       | OD       |
|          | VSL1  |          |
|          | STORE | DELVSIN  |
|          | PUSH  | ABVAL    |

July 12, 2016

1159

STOVL DELVSAB  
GOTO  
SUBEXIT

# Page 646

# \*\*\*\*\* S32/33.X \*\*\*\*\*

S32/33.X        SETPD    VLOAD  
                      6D  
                      UP1  
                  VCOMP   PDVL  
                          RACT1  
                  UNIT    VCOMP  
                  PUSH    VXV  
                          UP1  
                  VSL1  
                  STORE   OD  
                  RVQ

# Page 647

# \*\*\*\*\* CDHMVR \*\*\*\*\*

#

# SUBROUTINES USED

#        VECSHIFT

#        TIMETHET

#        SHIFTR1

CDHMVR        STQ        VLOAD  
                          SUBEXIT  
                          RACT2  
                  PUSH    UNIT  
                  STOVL   UNVEC  
                          RPASS2  
                  UNIT    DOT  
                          UNVEC  
                  PUSH    SL1  
                  STODL   CSTD  
                  DSQ     PDDL  
                          DP1/4TH  
                  SR2     DSU  
                  SQRT    SL1  
                  PDVL    VCOMP  
                  VXV  
                          RPASS2  
                  DOT     PDDL  
                          UP1

# UR SUB A

# Page 648

|        |          |                          |      |     |
|--------|----------|--------------------------|------|-----|
| SIGN   | STADR    |                          |      |     |
| STOVL  | SNTH     |                          |      |     |
|        | RPASS2   |                          |      |     |
| PDVL   | CALL     |                          |      |     |
|        | VPASS2   |                          |      |     |
|        | VECSHIFT |                          |      |     |
| STOVL  | VVEC     |                          |      |     |
| CLEAR  |          |                          |      |     |
|        | RVSU     |                          |      |     |
| STCALL | RVEC     |                          |      |     |
|        | TIMETHET |                          |      |     |
| LXA,2  | VSL*     |                          |      |     |
|        | RTX2     |                          |      |     |
|        | 0,2      |                          |      |     |
| STORE  | 18D      |                          |      |     |
| DOT    | SL1R     |                          |      |     |
|        | UNVEC    |                          |      |     |
| PDVL   | ABVAL    | # OD = V SUB PV          |      |     |
| SL*    | PDVL     |                          |      |     |
|        | 0,2      |                          |      |     |
|        | RACT2    |                          |      |     |
| ABVAL  | PDDL     | # 2D = LENGTH OF R SUB A |      |     |
| DSU    |          |                          |      |     |
|        | 02D      |                          |      |     |
| STODL  | DIFFALT  | # DELTA H IN METERS      | B+29 |     |
|        | R1A      |                          |      |     |
| NORM   | PDDL     | # 2 - R V**/MU           |      | 04D |
|        | X1       |                          |      |     |
|        | R1       |                          |      |     |
| CALL   |          |                          |      |     |
|        | SHIFTR1  |                          |      |     |
| SR1R   | DDV      |                          |      |     |
| SL*    | PUSH     |                          |      |     |
|        | 0        | -5,1                     |      |     |
| DSU    | PDDL     | # A SUB A                | B+29 | 04D |
|        | DIFFALT  |                          |      |     |
| SR2    | DDV      | # A SUB P                | B+31 |     |
|        | 04D      | #                        | B+2  |     |
| PUSH   | SQRT     | # A SUB P/A SUB A        |      | 06D |
| DMPR   | DMP      |                          |      |     |
|        | 06D      |                          |      |     |
|        | 00D      |                          |      |     |
| SL3R   | PDDL     | # V SUB AV METERS/CS     | B+7  | 08D |
|        | 02D      | # R SUB A MAGNITUDE      | B+29 |     |
| NORM   | PDDL     |                          |      |     |



July 12, 2016

1161

```

X1
RTMU
SR1 DDV # 2MU B+38
SL* PDDL # 2 MU/R SUBAA B+14 10D
0 -5,1
04D # ASUBA B+29
NORM PDDL
X2
RTMU
SR1 DDV
SL* BDSU
0 -6,2 # 2U/R - U/A B+14 (METERS/CS)SQ
PDDL DSQ # 10D
08D
BDSU SQRT
PDVL VXV # SQRT(MU(2/R SUB A-1/A SUB A)-VSUBA2) 10D
UP1
UNVEC
UNIT VXSC
10D
PDVL VXSC
UNVEC
08D
VAD VSL1
STADR
STORE VACT3
VSU
VACT2
Page 649
STCALL DELVEET2 # DELTA VCDH -- REFERENCE COORDINATES
SUBEXIT

Page 650
***** COMPTGO *****
#
SUBROUTINES USED
CLOKTASK
2PHSCHNG

BANK 35
SETLOC CSI/CDH
BANK

EBANK= RTRN

COUNT* $$/P3575
```

```
COMPTGO EXTEND
 QXCH RTRN
 CAF ZERO
 TS DISPDEX
 CAF BIT2
 INHINT
 TC WAITLIST
 EBANK= WHICH
 2CADR CLOKTASK

 TC 2PHSCHNG
 OCT 40036
 OCT 05024
 OCT 13000
 TC RTRN
```

This code is written to file `src/P32-P35-P72-P75.s`.

**B.72 P34-35 P74-75**1163 *<src/P34-35-P74-75.s 1163>≡*

```

Copyright: Public domain.
Filename: P34-35_P74-75.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 460-504
Contact: Onno Hommes <ohommes@cmu.edu>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-10 OH Batch 2 Assignment Comanche Transcription
2009-05-23 RSB In DISPLAYE, corrected a CADR GOFLASHR
to CADR GOFLASH.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 460
TRANSFER PHASE INITITIATION (TPI) PROGRAMS (P34 AND P74)
MOD NO -1 LOG SECTION -- P32-P35, P72-P75
MOD BY WHITE, P. DATE: 1 JUNE 67
#
PURPOSE
(1) TO CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL CONDITIONS
REQUIRED BY THE ACTIVE VEHICLE FOR EXECUTION OF THE TRANSFER
PHASE INITITIATION (TPI) MANEUVER, GIVEN --
(A) TIME OF IGNITION TIG (TPI) OR THE ELEVATION ANGLE (E) OF
THE ACTIVE/PASSIVE VEHICLE LOS AT TIG (TPI).

```

```

(B) CENTRAL ANGLE OF TRANSFER (CENTANG) FROM TIG (TPI) TO
INTERCEPT TIME (TIG (TPF))).
#
(2) TO CALCULATE TIG (TPI) GIVEN E OR E GIVEN TIG (TPI).
(3) TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA
APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.
(4) TO DISPLAY TO THE ASTRONAUT AND THE GROUND CERTAIN DEPENDENT
VARIABLES ASSOCIATED WITH THE MANEUVER FOR APPROVAL BY THE
ASTRONAUT/GROUND.
(5) TO STORE THE TPI TARGET PARAMETERS FOR USE BY THE DESIRED
THRUSTING PROGRAM.
#
ASSUMPTIONS
(1) LM ONLY -- THIS PROGRAM IS BASED UPON PREVIOUS COMPLETION OF
THE CONSTANT DELTA ALTITUDE (CDH) PROGRAM (P33/P73).
THEREFORE --
(A) AT A SELECTED TPI TIME (NOW IN STORAGE) THE LINE OF SIGHT
BETWEEN THE ACTIVE AND PASSIVE VEHICLES WAS SELECTED TO BE
A PRESCRIBED ANGLE (E) (NOW IN STORAGE) FROM THE
HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION.
(B) THE TIME BETWEEN CDH IGNITION AND TPI IGNITION WAS
COMPUTED TO BE GREATER THAN 10 MINUTES.
(C) THE VARIATION OF TEH ALTITUDE DIFFERENCE BETWEEN THE
ORBITS WAS MINIMIZED.
(D) THE PERICENTER ALTITUDES OF ORBITS FOLLOWING CSI AND
CDH WERE COMPUTED TO BE GREATER THAN 35,000 FT FOR LUNAR
ORBIT OR 85 NM FOR EARTH ORBIT.
(E) THE CSI AND CDH MANEUVERS WERE ASSUMED TO BE PARALLEL TO
THE PLANE OF THE PASSIVE VEHICLE ORBIT. HOWEVER, CREW
MODIFICATION OF DELTA V (LV) COMPONENTS MAY HAVE RESULTED
IN AN OUT-OF-PLANE MANEUVER.
(2) STATE VECTOR UPDATED BY P27 ARE DISALLOWED DURING AUTOMATIC
STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION (4)).
(3) THIS PROGRAM MUST BE DONE OVER A TRACKING STATION FOR REAL
TIME GROUND PARTICIPATION IN DATA INPUT AND OUTPUT. COMPUTED
VARIABLES MAY BE STORED FOR LATER VERIFICATION BY THE GROUND.
THESE STORAGE CAPABILITIES ARE LIMITED ONLY TO THE PARAMETERS
FOR ONE THRUSTING MANEUVER AT A TIME EXCEPT FOR CONCENTRIC
FLIGHT PLAN MANEUVER SEQUENCES.
(4) THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM
OR CSM STATE VECTORS FOR THIS PROGRAM. IF RADAR USE IS
DESIRED THE RADAR WAS TURNED ON AND LOCKED ON THE CSM BY
PREVIOUS SELECTION OF P20. RADAR SIGHTING MARKS WILL BE MADE
AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE
TRACK AND UPDATE FLAGS (SEE P20). THE RENDEZVOUS TRACKING
MARK COUNTER IS ZEROED BY TEH SELECTION OF P20 AND AFTER EACH

```

```

THRUSTING MANEUVER.
(5) THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.
(6) THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
CALCULATES THE MANEUVER PARAMETERS. SET AT THE START OF
EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
SELECTED THE FINAL MANEUVER COMPUTATION CYCLE.
#
EXTERNAL DELTA V FLAG -- DESIGNATES THE TYPE OF STEERING
REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE THRUSTING
PROGRAM SELECTED AFTER COMPLETION OF THIS PROGRAM.
#
(7) ONCE THE PARAMETERS REQUIRED FOR COMPUTATION OF THE MANEUVER
HAVE BEEN COMPLETELY SPECIFIED, THE VALUE OF THE ACTIVE
VEHICLE CENTRAL ANGLE OF TRANSFER IS COMPUTED AND STORED.
THIS NUMBER WILL BE AVAILABLE FOR DISPLAY TO THE ASTRONAUT
THROUGH THE USE OF V06N52.
#
THE ASTRONAUT WILL CALL THIS DISPLAY TO VERIFY THAT THE
CENTRAL ANGLE OF TRANSFER OF THE ACTIVE VEHICLE IS NOT WITHIN
Page 462
170 TO 190 DEGREES. IF THE ANGLE IS WITHIN THIS ZONE THE
ASTRONAUT SHOULD REASSES THE INPUT TARGETING PARAMETERS BASED
UPON DELTA V AND EXPECTED MANEUVER TIME.
(8) THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY --
#
P34 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
P74 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
INPUT
(1) TTPI TIME OF THE TPI MANEUVER.
(2) ELEV DESIRED LOS ANGLE AT TPI
(3) CENTANG ORBITAL CENTRAL ANGLE OF THE PASSIVE VEHICLE DURING
TRANSFER FROM TPI TO TIME OF INTERCEPT
#
OUTPUT
(1) TRKMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO
(3) +MGA MIDDLE GIMBAL ANGLE
(4) TTPI COMPUTED TIME OF TPI MANEUVER
OR

```

```

ELEV COMPUTED LOS ANGLE AT TPI
(5) POSTTPI PERIGEE ALTITUDE AFTER THE TPI MANEUVER
(6) DELVTPI MAGNITUDE OF DELTA V AT TPI
(7) DELVTPF MAGNITUDE OF DELTA V AT INTERCEPT
(8) DVLOS DELTA VELOCITY AT TPI -- LINE OF SIGHT
(9) DELVLVC DELTA VELOCITY AT TPI -- LOCAL VERTICAL COORDINATES
#
DOWNLINK
(1) TTPI TIME OF TPI MANEUVER
(2) TIG TIME OF TPI MANEUVER
(3) ELEV DESIRED LOS ANGLE AT TPI
(4) CENTANG ORBITAL CENTRAL ANGLE OF THE PASSIVE VEHICLE DURING
TRANSFER FROM TPI TO TIME OF INTERCEPT
(5) DELVEET3 DELTA VELOCITY AT TPI -- REFERENCE COORDINATES
(6) TPASS4 TIME OF INTERCEPT
#
COMMUNICATION TO THRUSTING PROGRAMS
(1) TIG TIME OF THE TPI MANEUVER
(2) RTARG OFFSET TARGET POSITION
(3) TPASS4 TIME OF INTERCEPT
(4) XDELVFLG RESET TO INDICATE LAMBERT (AIMPOINT) VG COMPUTATION
#
SUBROUTINES USED
AVFLAGA
Page 463
AVFLAGP
VNPOOH
DISPLAYE
SELECTMU
PRECSET
S33/34.1
ALARM
BANKCALL
GOFLASH
GOTPOOH
TIMETHET
S34/35.2
PERIAP01
SHIFTR1
S34/35.5
VN1645

```

```

SETLOC CSI/CDH
BANK
EBANK= SUBEXIT
COUNT 35/P3474

```

July 12, 2016

1167

|            |        |          |                                        |
|------------|--------|----------|----------------------------------------|
| P34        | TC     | AVFLAGA  |                                        |
|            | TC     | P34/P74A |                                        |
| P74        | TC     | AVFLAGP  |                                        |
| P34/P74A   | TC     | P20FLGON | # SET UPDATFLG, TRACKFLG               |
|            | CAF    | V06N37   | # TTPI                                 |
|            | TC     | VNP00H   | # Onno: The scans look like 0 not zero |
|            | TC     | INTPRET  |                                        |
|            | SSP    | EXIT     |                                        |
|            |        | NN       |                                        |
|            |        | 0        |                                        |
|            | TC     | DISPLA   | # ELEV AND CENTANG                     |
|            | TC     | INTPRET  |                                        |
|            | CLEAR  | DLOAD    |                                        |
|            |        | ETPIFLAG |                                        |
|            |        | TTPI     |                                        |
|            | STODL  | TIG      |                                        |
|            |        | ELEV     |                                        |
|            | BZE    | SET      |                                        |
|            |        | P34/P74B |                                        |
|            |        | ETPIFLAG |                                        |
| P34/P74B   | CALL   |          |                                        |
|            |        | SELECTMU |                                        |
| DELELO     | EQUALS | 26D      |                                        |
| P34/P74C   | DLOAD  | SET      |                                        |
|            |        | ZEROVECS |                                        |
|            |        | ITSWICH  |                                        |
|            | BON    | CLEAR    |                                        |
|            |        | ETPIFLAG |                                        |
| # Page 464 |        |          |                                        |
|            |        | SWCHSET  |                                        |
|            |        | ITSWICH  |                                        |
| SWCHSET    | STORE  | NOMTPI   |                                        |
| INTLOOP    | DLOAD  | DAD      |                                        |
|            |        | TTPI     |                                        |
|            |        | NOMTPI   |                                        |
|            | STCALL | TDEC1    |                                        |
|            |        | PRECSET  |                                        |
|            | CALL   |          |                                        |
|            |        | S33/34.1 |                                        |
|            | BZE    | EXIT     |                                        |
|            |        | SWCHCLR  |                                        |
|            | TC     | ALARM    |                                        |
|            | OCT    | 611      |                                        |
|            | CAF    | V05N09   |                                        |
|            | TC     | BANKCALL |                                        |

|            |        |          |                            |
|------------|--------|----------|----------------------------|
|            | CADR   | GOFLASH  |                            |
|            | TC     | GOTOP00H |                            |
|            | TC     | P34/P74A | # PROCEED                  |
|            | TC     | -7       | # V32                      |
| SWCHCLR    | BONCLR | BON      |                            |
|            |        | ITSWICH  |                            |
|            |        | INTLOOP  |                            |
|            |        | ETPIFLAG |                            |
|            |        | P34/P74D | # DISPLAY TTPI             |
|            | EXIT   |          |                            |
|            | TC     | DISPLA   | # DISPLAY ELEV AND CENTANG |
|            | TC     | P34/P74E |                            |
| P34/P74D   | EXIT   |          |                            |
|            | CAF    | V06N37   | # TTPI                     |
|            | TC     | VNP00H   |                            |
| P34/P74E   | TC     | INTPRET  |                            |
|            | SETPD  | DLOAD    |                            |
|            |        | OD       |                            |
|            |        | RTX1     |                            |
|            | STODL  | X1       |                            |
|            |        | CENTANG  |                            |
|            | PUSH   | COS      |                            |
|            | STODL  | CSTH     |                            |
|            | SIN    |          |                            |
|            | STOVL  | SNTH     |                            |
|            |        | RPASS3   |                            |
|            | VSR*   |          |                            |
|            |        | 0,2      |                            |
|            | STOVL  | RVEC     |                            |
|            |        | VPASS3   |                            |
|            | VSR*   | SET      |                            |
|            |        | 0,2      |                            |
|            |        | RVSW     |                            |
| # Page 465 | STCALL | VVEC     |                            |
|            |        | TIMETHET |                            |
|            | DLOAD  |          |                            |
|            |        | TTPI     |                            |
|            | STORE  | INTIME   | # FOR INITVEL              |
|            | DAD    |          |                            |
|            |        | T        | # RENDEZVOUS TIME          |
|            | STCALL | TPASS4   | # FOR INITVEL              |
|            |        | S34/35.2 |                            |
|            | VLOAD  | ABVAL    |                            |
|            |        | DELVEET3 |                            |



July 12, 2016

1169

```
STOVL DELVTPI
 VPASS4
VSU ABVAL
 VTPRIME
STOVL DELVTPF
 RACT3
PDVL CALL
 VIPRIME
 PERIAP01
CALL
 SHIFTR1
STODL POSTTPI
 TTPI
STORE TIG
EXIT
CAF V06N58
TC VNP00H
TC INTPRET
CALL
 S34/35.5
CALL
 VN1645
GOTO
 P34/P74C
```

# Page 466

# RENDEZVOUS MID-COURSE MANEUVER PROGRAMS (P35 AND P75)

# MOD NO -1 LOG SECTION -- P32-P35, P72-P75

# MOD BY WHITE, P. DATE: 1 JUNE 67

#

# PURPOSE

# (1) TO CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL CONDITIONS  
# REQUIRED BY THE ACTIVE VEHICLE FOR EXECUTION OF THE NEXT  
# MID-COURSE CORRECTION OF THE TRANSFER PHASE OF AN ACTIVE  
# VEHICLE RENDEZVOUS.

# (2) TO DISPLAY TO THE ASTRONAUT AND THE GROUND CERTAIN DEPENDENT  
# VARIABLES ASSOCIATED WITH THE MANEUVER FOR APPROVAL BY THE  
# ASTRONAUT/GROUND.

# (3) TO STORE THE TPM TARGET PARAMETERS FOR USE BY THE DESIRED  
# THRUSTING PROGRAM.

#

# ASSUMPTIONS

# (1) THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.

# (2) STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC  
# STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION (3)).

# (3) THE RENDEZVOUS RADAR IS ON AND IS LOCKED ON THE CSM. THIS WAS

```

DONE DURING PREVIOUS SELECTION OF P20. RADAR SIGHTING MARKS
WILL BE MADE AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN
ENABLED BY THE TRACK AND UPDATE FLAGS (SEE P20). THE
RENDEZVOUS TRACKING MARK COUNTER IS ZEROED BY THE SELECTION OF
P20 AND AFTER EACH THRUSTING MANEUVER.
(4) THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
CALCULATES THE MANEUVER PARAMETERS. SET AT THE START OF
EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
SELECTED THE FINAL MANEUVER COMPUTATION CYCLE.
#
EXTERNAL DELTA V FLAG -- DESIGNATES THE TYPE OF STEERING
REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE THRUSTING
PROGRAM SELECTED AFTER COMPLETION OF THIS PROGRAM.
#
(5) THE TIME OF INTERCEPT (T(INT)) WAS DEFINED BY PREVIOUS
COMPLETION OF THE TRANSFER PHASE INITIATION (TPI) PROGRAM
(P34/P74) AND IS PRESENTLY AVAILABLE IN STORAGE.
Page 467
(6) ONCE THE PARAMETERS REQUIRED FOR COMPUTATION OF THE MANEUVER
HAVE BEEN COMPLETELY SPECIFIED, THE VALUE OF THE ACTIVE
VEHICLE CENTRAL ANGLE OF TRANSFER IS COMPUTED AND STORED.
THIS NUMBER WILL BE AVAILABLE FOR DISPLAY TO THE ASTRONAUT
THROUGH THE USE OF V06N52
#
THE ASTRONAUT WILL CALL THIS DISPLAY TO VERIFY THAT THE
CENTRAL ANGLE OF TRANSFER OF THE ACTIVE VEHICLE IS NOT WITHIN
170 TO 190 DEGREES. IF THE ANGLE IS WITHIN THIS ZONE THE
ASTRONAUT SHOULD REASSESS THE INPUT TARGETING PARAMETERS BASED
UPON DELTA V AND EXPECTED MANEUVER TIME.
(7) THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY --
#
P35 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
P75 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
INPUT
(1) TPASS4 TIME OF INTERCEPT -- SAVED FROM P34/P74
#
OUTPUT
(1) TRKMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO

```

July 12, 2016

1171

```
(3) +MGA MIDDLE GIMBAL ANGLE
(4) DVLOS DELTA VELOCITY AT MID -- LINE OF SIGHT
(5) DELVLC DELTA VELOCITY AT MID -- LOCAL VERTICAL COORDINATES
#
DOWNLINK
(1) TIG TIME OF THE TPM MANEUVER
(2) DELVEET3 DELTA VELOCITY AT TPM -- REFERENCE COORDINATES
(3) TPASS4 TIME OF INTERCEPT
#
COMMUNICATION TO THRUSTING PROGRAMS
(1) TIG TIME OF THE TPM MANEUVER
(2) RTARG OFFSET TARGET POSITION
(3) TPASS4 TIME OF INTERCEPT
(4) XDELVFLG RESET TO INDICATE LAMBERT (AIMPOINT) VG COMPUTATION.
#
SUBROUTINES USED
AVFLAGA
AVFLAGP
LOADTIME
SELECTMU
PRECSET
S34/35.1
S34/35.2
Page 468
S34/35.5
VN1645
```

```
 COUNT 35/P3575
 EBANK= KT

P35 TC AVFLAGA
 EXTEND
 DCA ATIGINC
 TC P35/P75A
P75 TC AVFLAGP
 EXTEND
 DCA PTIGINC
P35/P75A DXCH KT
 TC P20FLGON # SET UPDATFLG, TRACKFLG
 TC INTPRET
 CALL
 SELECTMU
P35/P75B RTB
 LOADTIME
 STORE TSTRT
 DAD
```

```

KT
STORE TIG
STORE INTIME # FOR INITVEL
STCALL TDEC1
PRECSET # ADVANCE BOTH VEHICLES
CALL
S34/35.1 # GET NORM AND LOS FOR TRANSFORM
CALL
S34/35.2 # GET DELTA V(LV)
CALL
S34/35.5
CALL
VN1645
GOTO
P35/P75B

Page 469
***** S33/34.1 *****

S33/34.1 STQ SSP
NORMEX
TITER
OCT 40000
DLOAD SETPD
MAX250
OD
STOVL SECMAX
RACT3
STOVL RAPREC
VACT3
STOVL VAPREC
RPASS3
STOVL RPPREC
VPASS3
STORE VPPREC
ELCALC CALL
S34/35.1 # NORMAL AND LOS
VXV PDVL
RACT3 # (RA*VA)*RA OD
PDVL UNIT # ULOS AT 6D
RACT3
PDVL VPROJ # XCHNJ AND UP
VSL2 BVSU
ULOS
UNIT PDVL # UP AT OD
DOT PDVL # UP.UN*RA AT OD
OD # UP IN MPAC

```

July 12, 2016

1173

|            |       |          |                          |
|------------|-------|----------|--------------------------|
|            | DOT   | SIGN     |                          |
|            |       | ULOS     |                          |
|            | SL1   | ACOS     |                          |
|            | PDVL  | DOT      | # EA AT OD               |
|            |       | ULOS     |                          |
|            |       | RACT3    |                          |
|            | BPL   | DLOAD    |                          |
|            |       | TESTY    |                          |
|            |       | DPPOSMAX |                          |
|            | DSU   | PUSH     |                          |
| TESTY      | BOFF  | DLOAD    |                          |
|            |       | ITSWICH  |                          |
|            |       | ELEX     |                          |
|            |       | DELEL    |                          |
|            | STODL | DELELO   |                          |
|            | DSU   |          |                          |
|            |       | ELEV     |                          |
|            | STORE | DELEL    |                          |
|            | ABS   | DSU      |                          |
|            |       | ELEPS    |                          |
| # Page 470 |       |          |                          |
|            | BMN   |          |                          |
|            |       | TIMEX    | # COMMERCIALS EVERYWHERE |
| FIGTIME    | SLOAD | SR1      |                          |
|            |       | TITER    |                          |
|            | BHIZ  | LXA,1    |                          |
|            |       | NORMEX   | # TOO MANY ITERATIONS    |
|            |       | MPAC     |                          |
|            | SXA,1 | VLOAD    |                          |
|            |       | TITER    |                          |
|            |       | RPASS3   |                          |
|            | UNIT  | PDDL     |                          |
|            |       | 36D      |                          |
|            | PDVL  | UNIT     |                          |
|            |       | RACT3    |                          |
|            | PDDL  |          |                          |
|            | PDDL  | PUSH     |                          |
|            |       | 36D      |                          |
|            | BDSU  |          |                          |
|            |       | 12D      |                          |
|            | STODL | 30D      | # RP-RA MAGNITUDES       |
|            |       | DPHALF   |                          |
|            | DSU   | PUSH     |                          |
|            |       | ELEV     |                          |
|            | SIGN  | BMN      |                          |
|            |       | 30D      |                          |

|            |        |                           |
|------------|--------|---------------------------|
|            | NORMEX |                           |
| DLOAD      | COS    |                           |
| DMP        | DDV    |                           |
|            | 14D    |                           |
|            | 12D    |                           |
| DCOMP      |        | # SINCE COS(180-A)=-COS A |
| STORE      | 28D    |                           |
| ABS        | BDSU   |                           |
|            | DPHALF |                           |
| BMN        | VLOAD  |                           |
|            | NORMEX |                           |
|            | UNRM   |                           |
| VXV        | UNIT   |                           |
|            | 6D     | # UN*RA                   |
| DOT        | DMP    |                           |
|            | VACT3  |                           |
|            | 12D    |                           |
| PDVL       | VXV    |                           |
|            | OD     |                           |
|            | VPASS3 |                           |
| VXV        | UNIT   |                           |
|            | OD     | # (RP*VP)*RP              |
| DOT        | DMP    |                           |
|            | VPASS3 |                           |
|            | 14D    |                           |
| # Page 471 |        |                           |
| BDSU       |        |                           |
| NORM       | PDVL   | # NORMALIZED WA-WP 12D    |
|            | X1     |                           |
|            | 6D     |                           |
| VXV        | DOT    |                           |
|            | OD     |                           |
|            | UNRM   | # RA*RP.UN 14D            |
| PDVL       | DOT    |                           |
|            | OD     |                           |
|            | 6D     |                           |
| SL1        | ACOS   |                           |
| SIGN       |        |                           |
| DSU        | DAD    | # ALPHA PI                |
|            | DPHALF |                           |
|            | ELEV   |                           |
| PDDL       | ACOS   |                           |
|            | 28D    |                           |
| BDSU       | SIGN   |                           |
|            | DPHALF |                           |
|            | 30D    | # CONTAINS RP-RA          |

July 12, 2016

1175

```
DAD
DMP DDV
 TWOPI

DMP
SL* DMP
 0 -3,1
PUSH ABS
DSU BMN
 SECMAX
 OKMAX
DLOAD SIGN # REPLACE TIME WITH MAX TIME SIGNED
 SECMAX

OKMAX SLOAD BPL # TEST FIRST ITERATION
 TITER
 REPETE
 SSP DLOAD
 TITER
 OCT 37777
 GOTO

REPETE DLOAD STORDEL
 DMP
 DELEL
 DELELO
 BPL DLOAD
 NEXTES
 SECMAX
 DMP
 THIRD
 STODL SECMAX

Page 472
 ABS SR1 # CROSSED OVER SOLUTION
 DCOMP GOTO # DT=(-SIGN(DTO)//DT//)/2
 RESIGN
NEXTES DLOAD ABS
 DELEL
 PDDL ABS
 DELELO
 DSU
 BMN DLOAD
 REVERS # WRONG DIRECTION
 ABS
RESIGN SIGN GOTO
 DELTEEO
 STORDEL
REVERS DLOAD DCOMP
```

```

 DELTEEO
 SR1
 STORE DELTEEO
 DAD
 GOTO
 ADTIME
STORDEL T STORE DELTEEO
ADTIME DAD
 NOMTPI # SUM OF DELTA T:S
 STORE NOMTPI
 VLOAD PDVL
 VAPREC
 RAPREC
 CALL
 GOINT
 CALL
 ACTIVE # STORE NEW RACT3 VACT3
 VLOAD PDVL
 VPPREC
 RPPREC
 CALL
 GOINT
 CALL
 PASSIVE # STORE NEW RPASS3 VPASS3
 GOTO
 ELCALC
ELEX DLOAD DAD
 TTPI
 NOMTPI
 STODL TTPI
 BON
 ETPIFLAG
 TIMEX
 STORE ELEV
TIMEX DLOAD GOTO
Page 473
 ZEROVECS
 NORMEX

Page 474
***** S34/35.1 *****

COMPUTE UNIT NORMAL AND LINE OF SIGHT VECTORS GIVEN THE ACTIVE AND
PASSIVE POS AND VEL AT TIME T3
 SETLOC S3435LOC
 BANK

```



July 12, 2016

1177

```
S34/35.1 VLOAD VSU
 RPASS3
 RACT3
 UNIT PUSH
 STOVL ULOS
 RACT3
 VXV UNIT
 VACT3
 STORE UNRM
 RVQ
```

# Page 475

# \*\*\*\*\* S34/35.2 \*\*\*\*\*

```
ADVANCE PASSIVE VEH TO RENDEZVOUS TIME AND GET REQ VEL FROM LAMBERT
 SETLOC CSI/CDH
 BANK
```

```
S34/35.2 STQ VLOAD
 SUBEXIT
 VPASS3
 PDVL PDDL
 RPASS3
 INTIME
 PDDL PDDL
 TPASS4
 TWOPI # CONIC
 PDDL BHIZ
 NN
 S3435.23
 DLOAD PDDL
 ZEROVECS # PRECISION
```

```
S3435.23 CALL INTINT # GET TARGET VECTOR
```

```
S3435.25 STOVL RTARG
 VATT
 STOVL VPASS4
 RTARG
```

```
COMPUTE PHI = PI + (ACOS(UNIT RA.UNIT RP) - PI) SIGN(RA*RP.U)
```

```
UNIT PDVL # UNIT RP
 RACT3
UNIT PUSH # UNIT RA
VXV DOT
 OD
 UNRM # RA*RP.U
```

|                        |        |          |                           |
|------------------------|--------|----------|---------------------------|
|                        | PDVL   |          |                           |
|                        | DOT    | SL1      | # UNIT RA.UNIT RP         |
|                        |        | OD       |                           |
|                        | ACOS   | SIGN     |                           |
|                        | BPL    | DAD      |                           |
|                        |        | NOPIE    |                           |
|                        |        | DPPOSMAX | # REASONABLE TWO PI       |
| NOPIE                  | STODL  | ACTCENT  |                           |
|                        |        | TPASS4   |                           |
|                        | DSU    |          |                           |
|                        |        | INTIME   |                           |
|                        | STORE  | DELLT4   |                           |
|                        | SLOAD  | SETPD    |                           |
|                        |        | NN       | # NUMBER OF OFFSETS       |
|                        |        | OD       |                           |
|                        | PDDL   | PDVL     |                           |
|                        |        | EPSFOUR  |                           |
| # Page 476             |        |          |                           |
|                        |        | RACT3    |                           |
|                        | STOVL  | RINIT    |                           |
|                        |        | VACT3    |                           |
|                        | STCALL | VINIT    |                           |
|                        |        | INITVEL  |                           |
|                        | CALL   |          |                           |
|                        |        | LOMAT    |                           |
|                        | VLOAD  | MXV      |                           |
|                        |        | DELVEET3 |                           |
|                        |        | OD       |                           |
|                        | VSL1   |          |                           |
|                        | STCALL | DELVLVC  |                           |
|                        |        | SUBEXIT  |                           |
| # Page 477             |        |          |                           |
| # ***** S34/35.3 ***** |        |          |                           |
| S34/35.3               | STQ    | CALL     |                           |
|                        |        | NORMEX   |                           |
|                        |        | LOMAT    | # GET MATRIX IN PUSH LIST |
|                        | VLOAD  | VXM      |                           |
|                        |        | DELVLVC  | # NEW DEL V TPI           |
|                        |        | OD       |                           |
|                        | VSL1   |          |                           |
|                        | STORE  | DELVEET3 | # SAVE FOR TRANSFORM      |
|                        | VAD    | PDVL     |                           |
|                        |        | VACT3    | # NEW V REQ               |
|                        |        | RACT3    |                           |

July 12, 2016

1179

|         |        |          |                             |
|---------|--------|----------|-----------------------------|
|         | PDDL   | PDDL     |                             |
|         |        | TIG      |                             |
|         |        | TPASS4   |                             |
|         | PDDL   | PUSH     |                             |
|         |        | DPPOSMAX |                             |
|         | CALL   |          | # INTEG. FOR NEW TARGET VEC |
|         |        | INTINT   |                             |
|         | VLOAD  |          |                             |
|         |        | RATT     |                             |
|         | STORE  | RTARG    |                             |
| NOVRWRT | VLOAD  | PUSH     |                             |
|         |        | ULOS     |                             |
|         | VXV    | VCOMP    |                             |
|         |        | UNRM     |                             |
|         | UNIT   | PUSH     |                             |
|         | VXV    | VSL1     |                             |
|         |        | ULOS     |                             |
|         | PDVL   |          |                             |
|         | PDVL   | MXV      |                             |
|         |        | DELVEET3 |                             |
|         |        | OD       |                             |
|         | VSL1   |          |                             |
|         | STCALL | DVLOS    |                             |
|         |        | NORMEX   |                             |

# Page 478

# \*\*\*\*\* S34/35.4 \*\*\*\*\*

|          |      |         |                        |
|----------|------|---------|------------------------|
| S34/35.4 | STQ  | SETPD   | NO ASTRONAUT OVERWRITE |
|          |      | NORMEX  |                        |
|          |      | OD      |                        |
|          | GOTO |         |                        |
|          |      | NOVRWRT |                        |

# Page 479

# \*\*\*\*\* LOMAT \*\*\*\*\*

|       |       |       |        |
|-------|-------|-------|--------|
| LOMAT | VLOAD | VCOMP |        |
|       |       | UNRM  |        |
|       | STOVL | 6D    | # Y    |
|       |       | RACT3 |        |
|       | UNIT  | VCOMP |        |
|       | STORE | 12D   |        |
|       | VXV   | VSL1  |        |
|       |       | UNRM  | # Z*-Y |
|       | STORE | OD    |        |

1180

July 12, 2016

|        |        |          |      |                               |
|--------|--------|----------|------|-------------------------------|
|        | SETPD  | RVQ      |      |                               |
|        |        | 18D      |      |                               |
| GOINT  | PDDL   | PDDL     | # DO |                               |
|        |        | ZEROVECS | #    | NOT                           |
|        |        | NOMTPI   | #    |                               |
|        | PUSH   | PUSH     | #    | ORDER OR INSERT BEFORE INTINT |
| INTINT | STQ    | CALL     |      |                               |
|        |        | RTRN     |      |                               |
|        |        | INTSTALL |      |                               |
|        | CLEAR  | DLOAD    |      |                               |
|        |        | INTYPFLG |      |                               |
|        | BZE    | SET      |      |                               |
|        |        | +2       |      |                               |
|        |        | INTYPFLG |      |                               |
|        | DLOAD  | STADR    |      |                               |
|        | STODL  | TDEC1    |      |                               |
|        | SET    | LXA,2    |      |                               |
|        |        | MOONFLAG |      |                               |
|        |        | RTX2     |      |                               |
|        | BON    | CLEAR    |      |                               |
|        |        | CMOONFLG |      |                               |
|        |        | ALLSET   |      |                               |
| ALLSET | STOVL  | TET      |      |                               |
|        | VSR*   |          |      |                               |
|        |        | 0,2      |      |                               |
|        | STOVL  | RCV      |      |                               |
|        | VSR*   |          |      |                               |
|        |        | 0,2      |      |                               |
|        | STCALL | VCV      |      |                               |
|        |        | INTEGRVS |      |                               |
|        | VLOAD  | GOTO     |      |                               |
|        |        | RATT     |      |                               |
|        |        | RTRN     |      |                               |

```

Page 480
***** S34/35.5 *****
#
SUBROUTINES USED
BANKCALL
GOFLASH
GOTOP00H
S34/35.3
S34.35.4
VNPO0H

```

July 12, 2016

1181

```
S34/35.5 STQ BON
 SUBEXIT
 FINALFLG
 FLAGON
 SET GOTO
 UPDATFLG
 FLAGOFF

FLAGON VLOAD
 DELVLVC
 STORE DVLOS # SAVE DELTA V BEFORE DISPLAY
 EXIT
 CAF V06N81
 TC VNPOOH
 TC INTPRET
 VLOAD VSU # TEST FOR OVERWRITE OF COMPUTED
 DELVLVC # DELTA V
 DVLOS
 ABVAL BZE
 NOCHG # NO OVERWRITE
 CALL
 S34/35.3
NOCHG CLEAR VLOAD
 XDELVFLG
 DELVEET3
 STORE DELVSIN
FLAGOFF CALL
 S34/35.4
 EXIT
 CAF V06N59
 TC VNPOOH
 TC INTPRET
 GOTO
 SUBEXIT
```

```
Page 481
***** VN1645 *****
#
SUBROUTINES USED
P3XORP7X
GET+MGA
BANKCALL
DELAYJOB
COMPTGO
GOFLASHR
GOTOP00H
FLAGUP
```

|            |        |          |                                               |
|------------|--------|----------|-----------------------------------------------|
| VN1645     | STQ    | DLOAD    |                                               |
|            |        | SUBEXIT  |                                               |
|            |        | DP-.01   |                                               |
|            | STORE  | +MGA     | # MGA = -.01                                  |
|            | BOFF   | DLOAD    |                                               |
|            |        | FINALFLG |                                               |
|            |        | GET45    |                                               |
|            |        | DP-.01   |                                               |
|            | DAD    |          |                                               |
|            |        | DP-.01   |                                               |
|            | STORE  | +MGA     | # MGA = -.02                                  |
|            | BOFF   | EXIT     |                                               |
|            |        | REFSMFLG |                                               |
|            |        | GET45    |                                               |
|            | TC     | P3XORP7X |                                               |
|            | TC     | +2       | # P3X                                         |
|            | TC     | GET45 +1 | # P7X                                         |
|            | TC     | INTPRET  |                                               |
|            | VLOAD  | PUSH     |                                               |
|            |        | DELVSIN  |                                               |
|            | CALL   |          | # COMPUTE MGA                                 |
|            |        | GET+MGA  |                                               |
| GET45      | EXIT   |          |                                               |
|            | TC     | COMPTGO  | # INITIATE TASK TO UPDATE TTOGO               |
|            | CA     | SUBEXIT  |                                               |
|            | TS     | QSAVED   |                                               |
|            | CAF    | 1SEC     |                                               |
|            | TC     | BANKCALL |                                               |
|            | CADR   | DELAYJOB |                                               |
|            | CAF    | V16N45   | # TRKMKCNT, TTOGO, +MGA                       |
|            | TC     | BANKCALL |                                               |
|            | CADR   | GOFLASH  |                                               |
|            | TC     | KILCLOCK | # TERMINATE                                   |
|            | TC     | N45PROC  | # PROCEED                                     |
|            | TC     | CLUPDATE | # RECYCLE -- RETURN FOR INITIAL COMPUTATION   |
| KILCLOCK   | CA     | Z        |                                               |
|            | TS     | DISPDEX  |                                               |
| # Page 482 |        |          |                                               |
|            | TC     | GOTOP00H |                                               |
| N45PROC    | CS     | FLAGWRD2 |                                               |
|            | MASK   | BIT6     |                                               |
|            | EXTEND |          |                                               |
|            | BZF    | KILCLOCK | # FINALFLG IS SET -- FLASH V37 -- AWAIT NEW P |
|            | TC     | PHASCHNG |                                               |
|            | OCT    | 04024    |                                               |

|          |       |          |            |
|----------|-------|----------|------------|
|          | TC    | UPFLAG   | # SET      |
|          | ADRES | FINALFLG | # FINALFLG |
| CLUPDATE | CA    | Z        |            |
|          | TS    | DISPDEX  |            |
|          | TC    | PHASCHNG |            |
|          | OCT   | 04024    |            |
|          | TC    | INTPRET  |            |
|          | CLEAR | GOTO     |            |
|          |       | UPDATFLG |            |
|          |       | QSAVED   |            |

```
Page 483
***** DISPLAYE *****
#
SUBROUTINES USED
BANKCALL
GOFLASHR
GOTOP00H
BLANKET
ENDOFJOB
```

|          |        |          |
|----------|--------|----------|
| DISPLAYE | EXTEND |          |
|          | QXCH   | NORMEX   |
|          | CAF    | V06N55   |
|          | TCR    | BANKCALL |
|          | CADR   | GOFLASH  |
|          | TCF    | GOTOP00H |
|          | TC     | NORMEX   |
|          | TCF    | -5       |

```
Page 484
***** P3XORP7X *****
```

|          |        |        |
|----------|--------|--------|
| P3XORP7X | CAF    | HIGH9  |
|          | MASK   | MODREG |
|          | EXTEND |        |
|          | BZF    | +2     |
|          | INCR   | Q      |
|          | RETURN |        |

```
***** VNPO0H *****
#
SUBROUTINES USED
BANKCALL
GOFLASH
GOTOP00H
```

```

SETLOC P30SUBS
BANK

VNPOOH EXTEND
 QXCH RTRN
 TS VERBNOUN
 CAF VNBANK # ***** THIS ROUTINE MUST REMAIN IN
 XCH FBANK # FIXED-FIXED *****
 TS TBASE5 # * WATCH OUT *

 CA VERBNOUN
 TCR BANKCALL
 CADR GOFLASH
 TCF GOTOP00H
 TCF +2
VNBANK TC -5

 CA TBASE5
 TS FBANK
 TC RTRN

Page 485
***** CONSTANTS *****

V06N37 VN 0637
V06N55 VN 0655
V06N58 VN 0658
V06N59 VN 0659
V06N81 VN 0681
V16N45 VN 1645
 SETLOC CSI/CDH
 BANK

TWOPI 2DEC 6.283185307 B-4
MAX250 2DEC 25 E3 B-28 # RSB 2004 added the B-28. OH 2009 leave?
THIRD 2DEC .333333333
ELEPS 2DEC .277777777 E-3
DECTWO OCT 2
DP- .01 OCT 77777 # CONSTANTS
 OCT 61337 # ADJACENT -.01 FOR MGA DSP
EPSFOUR 2DEC .04166666666

Page 486
***** INITVEL *****
MOD NO -1
LOG SECTION -- P34-P35, P74-P75

```



```

MOD BY WHITE, P. DATE: 21 NOV 67
#
FUNCTIONAL DESCRIPTION
THIS SUBROUTINE COMPUTES THE REQUIRED INITIAL VELOCITY VECTOR FOR
A TRAJECTORY OF SPECIFIC TRANSFER TIME BETWEEN SPECIFIED INITIAL
AND TARGET POSITIONS. THE TRAJECTORY MAY BE EITHER CONIC OR
PRECISION DEPENDING ON AN INPUT PARAMETER (NAMELY, NUMBER OF
OFFSETS). IN ADDITION, IN THE PRECISION TRAJECTORY CASE, THE
SUBROUTINE ALSO COMPUTES AN OFFSET TARGET VECTOR, TO BE USED
DURING PURE-CONIC CROSS-PRODUCT STEERING. THE OFFSET TARGET
VECTOR IS THE TERMINAL POSITION VECTOR OF A CONIC TRAJECTORY WHICH
HAS THE SAME INITIAL STATE AS A PRECISION TRAJECTORY WHOSE
TERMINAL POSITION VECTOR IS THE SPECIFIED TARGET VECTOR.
#
IN ORDER TO AVOID THE INHERENT SINGULARITIES IN THE 180 DEGREE
TRANSFER CASE WHEN THE (TRUE OR OFFSET) TARGET VECTOR MAY BE
SLIGHTLY OUT OF THE ORBITAL PLANE, THIS SUBROUTINE ROTATES THIS
VECTOR INTO A PLANE DEFINED BY THE INPUT INITIAL POSITION VECTOR
AND ANOTHER INPUT VECTOR (USUALLY THE INITIAL VELOCITY VECTOR),
WHENEVER THE INPUT TARGET VECTOR LIES INSIDE A CONE WHOSE VERTEX
IS THE ORIGIN OF COORDINATES, WHOSE AXIS IS THE 180 DEGREE
TRANSFER DIRECTION, AND WHOSE CONE ANGLE IS SPECIFIED BY THE USER.
#
THE LAMBERT SUBROUTINE IS UTILIZED FOR THE CONIC COMPUTATIONS AND
THE COASTING INTEGRATION SUBROUTINE IS UTILIZED FOR THE PRECISION
TRAJECTORY COMPUTATIONS.
#
CALLING SEQUENCE
L CALL
L+1 INITVEL
L+2 (RETURN -- ALWAYS)
#
INPUT
(1) RINIT INITIAL POSITION RADIUS VECTOR
(2) VINIT INITIAL POSITION VELOCITY VECTOR
(3) RTARG TARGET POSITION RADIUS VECTOR
(4) DELLT4 DESIRED TIME OF FLIGHT FROM RINIT TO RTARG
(5) INTIME TIME OF RINIT
(6) OD NUMBER OF ITERATIONS OF LAMBERT/INTEGRVS
(7) 2D ANGLE TO 180 DEGREES WHEN ROTATION STARTS
(8) RTX1 -2 FOR EARTH, -10D FOR LUNAR
(9) RTX2 COORDINATE SYSTEM ORIGIN -- 0 FOR EARTH, 2 FOR LUNAR
PUSHLOC SET AT 4D
#
Page 487
OUTPUT

```

```

(1) RTARG OFFSET TARGET POSITION VECTOR
(2) VIPRIME MANEUVER VELOCITY REQUIRED
(3) VTPRIME VELOCITY AT TARGET AFTER MANEUVER
(4) DELVEET3 DELTA VELOCITY REQUIRED FOR MANEUVER
#
SUBROUTINES USED
LAMBERT
INTSTALL
INTEGRVS

 SETLOC INTVEL
 BANK

INITVEL COUNT 11/INITV
SET # COGA GUESS NOT AVAILABLE

HAVEGUES VLOAD GUESSW
 STQ
 RTARG
 NORMEX
STORE RTARG1
SLOAD BHIZ
 RTX2
 INITVEL1
VLOAD VSL2
 RINIT # B29
STOVL RINIT # B27
 VINIT # B7
VSL2
STOVL VINIT # B5
 RTARG1
VSL2
STORE RTARG1

INITIALIZATION

INITVEL1 SSP DLOAD # SET ITCTR TO -1,LOAD MPAC WITH E4 (PL 2D)
 ITCTR
 0 -1
COSINE SR1 # CALCULATE COSINE (E4) (+2)
STODL COZY4 # SET COZY4 TO COSINE (E4) (PL 0D)
LXA,2 SXA,2
 MPAC
 VTARGETAG # SET VTARGETAG TO 0D (SP)
VLOAD
 RINIT
STOVL R1VEC # R1VEC EQ RINIT

```

```

Page 488
RTARG1
STODL R2VEC # R2VEC EQ RTARG
 DELLT4
STORE TDESIRE # TDESIRE EQ DELLT4
SETPD VLOAD
 OD # INITIALIZE PL TO OD
 RINIT # MPAC EQ RINIT (+29)
UNIT PUSH # UNIT(RI) (+1) (PL 6D)
VXV UNIT
 VINIT # MPAC EQ UNIT(RI) X VI (+8)
STOVL UN
 RTARG1
UNIT DOT # TEMP=URT.URI (+2) (PL 0D)
DAD CLEAR
 COZY4
 NORMSW
STORE COZY4
INITVEL2 BPL SET
 INITVEL3 # UN CALCULATED IN LAMBERT
 NORMSW

ROTATE RC INTO YC PLANE -- SET UNIT NORMAL TO YC

VLOAD PUSH # (PL 6D)
 R2VEC # RC TO 6D (+29)
ABVAL PDVL # RC TO MPAC, ABVAL(RC) (+29) TO OD(PL 2D)
PUSH VPROJ # (PL 8D)
 UN
VSL2 BVSU
UNIT VXSC # (PL 0D)
VSL1
STORE R2VEC
TLOAD SLOAD
 ZEROVEC
 ITCTR
BPL VLOAD
 INITVEL3
 R2VEC
STORE RTARG1
INITVEL3 DLOAD PDVL # (PL 2D)
 MUEARTH # POSITIVE VALUE
 R2VEC
UNIT PDVL # 2D = UNIT(R2VEC) (PL 8D)
 R1VEC
UNIT PUSH # 8D = UNIT(R1VEC) (PL 14D)

```

```

 VXV VCOMP # -N = UNIT(R2VEC) X UNIT(R1VEC)
 2D
 PUSH
 LXA,1 DLOAD #
 RTX1 (PL 20D)
 18D
 BMN INCR,1
Page 489
 +2
 DEC -8
 INCR,1 SLOAD
 10D
 X1
 BHIZ VLOAD #
 +2 (PL 14D)
 VCOMP PUSH #
 VLOAD (PL 20 D)
 VXV DOT #
 BPL DLOAD #
 INITVEL4 (PL 14D)
 DCOMP PUSH #
 LXA,2 SXA,2 (PL 2D)
INITVEL4 OD
 GEOMSGN

SET INPUTS UP FOR LAMBERT

 LXA,1 CALL
 RTX1
OPERATE THE LAMBERT CONIC ROUTINE (COASTFLT SUBROUTINE)

 LAMBERT

ARRIVED AT SOLUTION IS GOOD ENOUGH ACCORDING TO SLIGHTLY WIDER BOUNDS.

 CLEAR VLOAD
 GUESSW
 VVEC

STORE CALCULATED INITIAL VELOCITY REQUIRED IN VIPRIME

 STODL VIPRIME # INITIAL VELOCITY REQUIRED (+7)

IF NUMIT IS ZERO, CONTINUE AT INITVELB, OTHERWISE
SET UP INPUTS FOR ENCKE INTEGRATION (INTEGRVS).

```

```

 VTARGETAG
 BHIZ CALL
 INITVEL7
 INTSTALL
 SLOAD CLEAR
 RTX2
 MOONFLAG
 BHIZ SET
 INITVEL5
 MOONFLAG
INITVEL5 VLOAD
 RINIT
 STORE R1VEC
Page 490
 STOVL RCV
 VIPRIME
 STODL VCV
 INTIME
 STORE TET
 DAD CLEAR
 DELLT4
 INTYPFLG
 STCALL TDEC1
 INTEGRVS
 VLOAD
 VATT1
 STORE VTARGET

```

```

IF ITERATION COUNTER (ITCTR) EQ NO. ITERATIONS (NUMIT), CONTINUE AT
INITVELC, OTHERWISE REITERATE LAMBERT AND ENCKE

```

```

 LXA,2 INCR,2
 ITCTR
 1D # INCREMENT ITCTR
 SXA,2 XSU,2
 ITCTR
 VTARGETAG
 SLOAD BHIZ # IF SP(MPAC) EQ 0, CONTINUE AT INITVELC
 X2
 INITVEL6

```

```

OFFSET CONIC TARGET VECTGOR

```

```

 VLOAD VSU
 RTARG1
 RATT1

```

```

 VAD
 R2VEC
 STODL R2VEC
 COZY4
 GOTO INITVEL2 # CONTINUE ITERATING AT INITVEL2

COMPUTE THE DELTA VELOCITY

INITVEL6 VLOAD
 R2VEC
 STORE RTARG1
INITVEL7 VLOAD VSU
 VIPRIME
 VINIT
 STOVL DELVEET3 # DELVEET3 = VIPRIME-VINIT (+7)
 VTARGET
 STORE VTPRIME
 SLOAD BHIZ
 RTX2

Page 491

 INITVELX
 VLOAD VSR2
 VTPRIME
 STOVL VTPRIME
 VIPRIME
 VSR2
 STOVL VIPRIME
 RTARG1
 VSR2
 STOVL RTARG1
 DELVEET3
 VSR2
 STORE DELVEET3
INITVELX SETPD VLOAD
 OD
 RTARG1
 STCALL RTARG
 NORMEX

***** END OF INITVEL ROUTINE *****

Page 492
***** MIDGIM *****
MOD NO. 0, BY WILLMAN, SUBROUTINE RENDGUID, LOG P34-P35, P74-P75
REVISION 03, 17 FEB 67

```

```

#
IF THE ACTIVE VEHICLE IS DOING THE COMPUTATION, MIDGIM COMPUTES
THE POSITIVE MIDDLE GIMBAL ANGLE OF THE ACTIVE VEHICLE TO THE INPUT
DELTA VELOCITY VECTOR (OD IN PUSY LIST), OTHERWISE
MIDGIM CONVERTS THE INPUT DELTA VELOCITY VECTOR FROM INERTIAL COORDIN-
ATES TO LOCAL VERTICAL COORDINATES OF THE ACTIVE VEHICLE.
#
** INPUTS **
NAME MEANING UNITS/SCALING/MODE
AVFLAG INT FLAG -- 0 IS CSM ACTIVE, 1 IS LEM ACTIVE BIT
COMPUTER INT FLAG -- 0 IS LEM COMPUTER, 1 IS CSM COMPUTER BIT
RINIT ACTIVE VEHICLE RADIUS VECTOR METERS/CSEC (+7) VT
VINIT ACTIVE VEHICLE VELOCITY VECTOR METERS/CSEC (+7) VT
OD(PL) ACTIVE VEHICLE DELTA VELOCITY VECTOR METERS/CSEC (+7) VT
#
** OUTPUTS **
NAME MEANING UNITS/SCALING/MODE
+MGA + MIDDLE GIMBAL ANGLE REVOLUTIONS (+0) DP
DELVLVC DELTA VELOCITY VECTOR IN LV COORD. METERS/CSEC (+7) VT
MGLVFLAG INT FLAG: 0 IS +MGA COMUTED, 1 IS DELVLVC COMP. BIT
#
** CALLING SEQUENCE **
L CALL
L+1 MIDGIM
L+2 (RETURN -- ALWAYS)
#
** NO SUBROUTINES CALLED **
#
** DEBRIS -- ERASABLE TEMPORARY USAGE **
A,Q,L, PUSH LIST, MPAC.
#
** ALARMS -- NONE **

Page 493
MIDDLE GIMBAL ANGLE COMPUTATION
SETLOC MIDDGIM
BANK
#
COUNT* $$/MIDG
#
HALFREV 2DEC 1 B-1
#
MIDGIM BON BOFF
AVFLAG
MIDGIM1
COMPUTER

```

## GET.LVC

# COMPUTE +MGA IF AVFLAG AND COMPUTER HAVE OPPOSITE VALUES.

```

GET+MGA VLOAD UNIT # (PL OD) V (+7) TO MPAC UNITIZE UV (+1)
 DOT SL1 # DOT UV WITH Y(STABLE MEMBER) AND RESCALE
 REFMMAT +6 # FROM +2 TO +1 FOR ASIN ROUTINE
 ARCSIN BPL
 SETMGA
 DAD DAD # CONVERT -MGA TO +MGA BY
 HALFREV # ADDING ONE REVOLUTION
 HALFREV
SETMGA STORE +MGA
 CLR RVQ # CLEAR MGLVFLAG TO INDICATE +MGA CALC
 MGLVFLAG # AND EXIT

MIDGIM1 BOFF
 COMPUTER
 GET+MGA

```

# COMPUTE DELVLVC IF AVFLAG AND COMPUTER HAVE SAME VALUES

```

GET.LVC VLOAD UNIT # (PL 6D) R (+29) IN MPAC UNITIZE UR
 RINIT
 VCOMP # U(-R)
 STORE 18D # U(-R) TO 18D
 VXV UNIT # U(-R)*V EQ V*U(R), U(V*R)
 VINIT
 STORE 12D # U(V*R) TO 12D
 VXV UNIT # U(V*R)*U(-R), U((V*R)*(-R))
 18D
 STOVL 6D # TRANSFORMATION MATRIX IS IN 6D (+1)
 OD # DELTA V (+7) IN OD
 MXV VSL1 # CONVERT FROM INER COOR TO LV COOR (+8)
 6D # AND SCALE +7 IN MPAC
 STORE DELVLVC # STORE IN DELVLVC (+7)
 SET RVQ # SET MGLVFLAG TO INDICATE LVC CALC
 MGLVFLAG # AND EXIT

```

# \*\*\*\*\* END OF MIDGIM ROUTINE \*\*\*\*\*

# Page 494

```

SELECTMU AXC,1 AXT,2
 2D
 OD
 BOFF

```



```

 CMOONFLG
 SETMUER
 AXC,1 AXT,2
 10D
 2D
SETMUER DLOAD* SXA,1
 MUTABLE +4,1
 RTX1
 STODL* RTSR1/MU
 MUTABLE -2,1
 BOFF SR
 CMOONFLG
 RTRNMU
 6D
RTRNMU STORE RTMU
 SXA,2 CLEAR
 RTX2
 FINALFLG
 GOTO
 VN1645

```

# Page 495

# \*\*\*\*\* PERIAPO \*\*\*\*\*

# MOD NO -1

LOG SECTION -- P34-P35, P74-P75

# MOD BY WHITE, P.

DATE 18 JAN 68

#

# FUNCTIONAL DESCRIPTION

# THIS SUBROUTINE COMPUTES THE TWO BODY APOCENTER AND PERICENTER  
# ALTITUDES GIVEN THE POSITION AND VELOCITY VECTORS FOR A POINT ON  
# THE TRAJECTORY AND THE PRIMARY BODY.

#

# SETRAD IS CALLED TO DETERMINE THE RADIUS OF THE PRIMARY BODY.

#

# APSIDES IS CALLED TO SOVE FOR THE TWO BODY RADII OF APOCENTER AND  
# PERICENTER AND THE ECCENTRICITY OF THE TRAJECTORY.

#

# CALLING SEQUENCE

# L CALL

# L+1 PERIAPO

# L+2 (RETURN -- ALWAYS)

#

# INPUT

# (1) RVEC POSITION VECTOR IN METERS

# SCALE FACTOR -- EARTH +29, MOON +27

# (2) VVEC VELOCITY VECTORY IN METERS/CENTISECOND

# SCALE FACTOR -- EARTH +7, MOON +5

```

(3) X1 PRIMARY BODY INDICATOR
EARTH -2, MOON -10
#
OUTPUT
(1) 2D APOCENTER RADIUS IN METERS
SCALE FACTOR -- EARTH +29, MOON +27
(2) 4D APOCENTER ALTITUDE IN METERS
SCALE FACTOR -- EARTH +29, MOON +27
(3) 6D PERICENTER RADIUS IN METERS
SCALE FACTOR -- EARTH +29, MOON +27
(4) 8D PERICENTER ALTITUDE IN METERS
SCALE FACTOR -- EARTH +29, MOON +27
(5) ECC ECCENTRICITY OF CONIC TRAJECTORY
SCALE FACTOR -- +3
(6) XXXALT RADIUS OF THE PRIMARY BODY IN METERS
SCALE FACTOR -- EARTH +29, MOON +27
(7) PUSHLOC EQUALS 10D
#
SUBROUTINES USED
SETRAD
Page 496
APSIDES

```

```

SETLOC APOPERI
BANK

```

```

COUNT* $$/PERAP

```

```

RPAD 2DEC 6373338 B-29 # STANDARD RADIUS OF PAD 37-B.
 # = 20 909 901.57 FT

```

```

PERIAPO1 LXA,2 VSR*
 RTX2
 0,2
 STOVL VVEC
 LXA,1 VSR*
 RTX1
 0,2
 STORE RVEC
PERIAPO STQ CALL
 NORMEX
 SETRAD
 STCALL XXXALT
 APSIDES
 SETPD PUSH
 # 2D = APOCENTER RADIUS
 2D

```

B29 OR B27

July 12, 2016

1195

|            |        |                                   |                          |            |
|------------|--------|-----------------------------------|--------------------------|------------|
|            | DSU    | PDDL<br>XXXALT<br>OD              | # 4D = APOGEE ALTITUDE   | B29 OR B27 |
|            | PUSH   | DSU<br>XXXALT                     | # 6D = PERICENTER RADIUS | B29 OR B27 |
|            | PUSH   | GOTO<br>NORMEX                    | # 8D = PERIGEE ALTITUDE  | B29 OR B27 |
| # Page 497 |        |                                   |                          |            |
| # SETRAD   |        |                                   |                          |            |
| SETRAD     | DLOAD  | PUSH<br>RPAD                      |                          |            |
|            | SXA,1  | INCR,2<br>X2<br>2D                |                          |            |
|            | SLOAD  | BHIZ<br>X2<br>SETRADX             |                          |            |
|            | VLOAD  | ABVAL<br>RLS                      |                          |            |
|            | PDDL   |                                   |                          |            |
| SETRADX    | DLOAD  | RVQ                               |                          |            |
| # Page 498 |        |                                   |                          |            |
| # PRECSET  |        |                                   |                          |            |
| PRECSET    | STQ    | NORMEX                            |                          |            |
|            | STCALL | TDEC2<br>LEMPREC                  |                          |            |
|            | CALL   | LEMSTORE                          |                          |            |
|            | DLOAD  | TDEC2                             |                          |            |
|            | STCALL | TDEC1<br>CSMPREC                  |                          |            |
|            | CALL   | CSMSTORE                          |                          |            |
|            | GOTO   | NORMEX                            |                          |            |
| LEMSTORE   | VLOAD  | BOFF<br>RATT<br>AVFLAG<br>PASSIVE |                          |            |
| ACTIVE     | STOVL  | RACT3<br>VATT                     |                          |            |
|            | STORE  | VACT3                             |                          |            |
|            | RVQ    |                                   |                          |            |

|          |       |        |
|----------|-------|--------|
| CSMSTORE | VLOAD | BOFF   |
|          |       | RATT   |
|          |       | AVFLAG |
|          |       | ACTIVE |
| PASSIVE  | STOVL | RPASS3 |
|          |       | VATT   |
|          | STORE | VPASS3 |
|          | RVQ   |        |

# Page 499

# VECSHIFT

|          |       |      |
|----------|-------|------|
| VECSHIFT | LXA,2 | VSR* |
|          |       | RTX2 |
|          |       | 0,2  |
|          | LXA,1 | PDVL |
|          |       | RTX1 |
|          | VSR*  | PDVL |
|          |       | 0,2  |
|          | RVQ   |      |

# Page 500

# SHIFTR1

|         |       |      |
|---------|-------|------|
| SHIFTR1 | LXA,2 | SL*  |
|         |       | RTX2 |
|         |       | 0,2  |
|         | RVQ   |      |

# Page 501

# PROGRAM DESCRIPTION

#

# SUBROUTINE NAME R36 OUT-OF-PLANE RENDEZVOUS ROUTINE

# MOD NO. 2 DATE 2 JANUARY 1969

# MOD BY A.W.BANCROFT LOG SECTION EXTENDED VERBS

#

# FUNCTIONAL DESCRIPTION

#

# TO DISPLAY AT ASTRONAUT REQUEST LGC CALCULATED RENDEZVOUS

# OUT-OF-PLANE PARAMETERS (Y, YDOT, PSI). (REQUESTED BY DSKY).

#

# CALLING SEQUENCE

# ASTRONAUT REQUEST THROUGH DSKY V 90 E

#

# SUBROUTINES CALLED

# EXDSPRET

# GOMARKF

# CSMPREC

```

LEMPREC
SGNAGREE
LOADTIME
#
NORMAL EXIT MODES
ASTRONAUT REQUEST THROUGH DSKY TO TERMINATE PROGRAM V 34 E
#
ALARM OR ABORT EXIT MODES
NONE
#
OUTPUT
DECIMAL DISPLAY OF TIME, Y, YDOT AND PSI
#
DISPLAYED VALUES Y, YDOT, AND PSI, ARE STORED IN ERASABLE
REGISTERS RANGE, RRATE, AND RTHETA RESPECTIVELY.
#
ERASABLE INITIALIZATION REQUIRED
CSM AND LEM STATE VECTORS
#
DEBRIS
CENTRALS A,Q,L
OTHER: THOSE USED BY THE ABOVE LISTED SUBROUTINES

 BANK 20
 SETLOC R36CM
 BANK

Page 502

 EBANK= RPASS36

 SBANK= R36A
 COUNT* $$/R36

R36 CAF TWO
 TS OPTIONX
 CAF ONE
 TS OPTIONX +1
 CAF OPTION36 # V 04 N 12
 TC BANKCALL
 CADR GOXDSPF
 TC ENDEXT # TERMINATE
 TC +2 # PROCEED
 TC -5 # R2 LOADED VIA DSKY
 TC POSTJUMP
 CADR R36A

OPTION36 VN 0412

```

```

 SETLOC R36LM
 BANK

R36A ZL
 CAF ZERO # SET TIME OF EVENT TO ZERO FOR FIRST
 DXCH DSPTMX # DISPLAY
 LXCH OPTIONY # SAVE VEH. OPTION

R36P3 CAF V06N16N
 TC BANKCALL
 CADR GOMARKF
 TCF ENDEXT # TERMINATE
 TCF +2 # PROCEED
 TCF -5 # RECYCLE FOR ASTRONAUT INPUT TIME
 DXCH DSPTMX
 EXTEND
 BZF LREGCHK # A-REG ZERO GOTO CHECK L-REG FOR ZERO
 DXCH MPAC # A-REG NON-ZERO, TIME = ASTRO INPUT TIME
 TC INTPRET
 RTB GOTO
 DPMODE
 R36INT

 SETLOC R36LM1
 BANK

R36INT STORE TDEC1
 SLOAD SR1
 OPTIONY
 BHIZ CALL
 R36PROG2 # FOR CSM DISPLAY

Page 503
 GOTO THISPREC # FOR LEM DISPLAY
 R36PROG3
R36PROG2 CALL
 OTHPREC
R36PROG3 VLOAD PDVL
 VATT
 RATT # -
 STORE RPASS36 # R
 UNIT PDVL # P
 VXV UNIT # -
 STADR
 STODL UNP36 # U
 TAT

```

July 12, 2016

1199

|            |        |          |                                                         |                          |
|------------|--------|----------|---------------------------------------------------------|--------------------------|
|            | STORE  | TDEC1    |                                                         |                          |
|            | SLOAD  | SR1      |                                                         |                          |
|            |        | OPTIONY  |                                                         |                          |
|            | BHIZ   | CALL     |                                                         |                          |
|            |        | R36PROG4 | # FOR CSM DISPLAY                                       |                          |
|            |        | OTHPREC  | # FOR LEM DISPLAY                                       |                          |
|            | GOTO   |          |                                                         |                          |
|            |        | R36PROG5 |                                                         |                          |
| R36PROG4   | CALL   |          |                                                         |                          |
|            |        | THISPREC |                                                         |                          |
| R36PROG5   | VLOAD  | PDVL     | #                                                       |                          |
|            |        | VATT     | # VELOCITY VECTOR                                       | $\bar{V}$ 00D            |
|            |        | RATT     | #                                                       | A                        |
|            | PDDL   |          |                                                         |                          |
|            |        | TAT      | # SAVE TIME IN LOCATION 30D FOR REDISPLAY               |                          |
|            | STOVL  | 30D      | #                                                       |                          |
|            | PUSH   | PUSH     | # POSITION VECTOR                                       | $\bar{R}$ IN 06D AND 12D |
|            | BVSU   | PDVL     | #                                                       | $\bar{A}$                |
|            |        | RPASS36  | # LINE OF SIGHT VECTOR                                  | $\bar{R} - \bar{R}$ 12D  |
|            | DOT    | SL1      | #                                                       | P A                      |
|            |        | UNP36    | #                                                       |                          |
|            | STOVL  | RANGE    | # $\bar{Y} = \bar{U} \cdot \bar{R}$                     |                          |
|            |        | 00D      | #                                                       | A                        |
|            | DOT    | SL1      |                                                         |                          |
|            |        | UNP36    | # $\cdot$                                               | $\bar{A}$                |
|            | STOVL  | RRATE    | # $\bar{Y} = \bar{U} \cdot \bar{V}$                     |                          |
|            |        | 06D      | #                                                       | A                        |
|            | UNIT   | PUSH     | # $\bar{U} = \text{UNIT} ( \bar{R} )$                   | 18D                      |
|            | VXV    | VXV      | # RA                                                    | A                        |
|            |        | 00D      | #                                                       |                          |
|            |        | 18D      | # $( \bar{U} \times \bar{V} ) \times \bar{U} = \bar{U}$ |                          |
|            | VSL2   | UNIT     | # RA                                                    | A RA A                   |
|            | UNIT   | GOTO     |                                                         |                          |
|            |        | R36B     |                                                         |                          |
|            | SETLOC | R36CM1   |                                                         |                          |
| # Page 504 |        |          |                                                         |                          |
|            | BANK   |          |                                                         |                          |
| R36B       | STOVL  | 00D      | # UNIT HORIZONTAL IN FORWARD DIR. 00D                   |                          |
|            |        | 18D      |                                                         |                          |
|            | DOT    | VXSC     | #                                                       | $\bar{U}$                |
|            |        | 12D      | #                                                       | L                        |
|            | VSL2   |          |                                                         |                          |
|            | BVSU   | UNIT     |                                                         |                          |
|            | UNIT   |          |                                                         |                          |

```

PUSH DOT # LOS PROJECTED INTO HORIZONTAL 12D
 OOD # PLANE
SL1 ARCCOS #
STOVL RTHETA # PSI = ARCCOS($\frac{\bar{U}}{A}$. $\frac{\bar{U}}{L}$)
VXV DOT #
 OOD
BPL DLOAD
 R36TAG2
 DPPOSMAX
DSU
 RTHETA
STCALL RTHETA
 R36TAG2

SETLOC R36LM
BANK

R36TAG2 DLOAD RTB
 30D
 SGNAGREE
STORE DSPTEMX
EXIT
CAF V06N90N # DISPLAY Y, YDOT, AND PSI.
TC BANKCALL
CADR GOMARKF
TCF ENDEXT # TERMINATE
TCF ENDEXT # PROCEED, END OF PROGRAM
TCF R36P3 # REDISPLAY OUTPUT
LREGCHK XCH L
 EXTEND
 BZF ENTTIM2 # L-REG ZERO, SET TIME = PRESENT TIME
 XCH L # L-REG NON ZERO, TIME = ASTRO INPUT TIME
 TCF ASTROTIM
ENTTIM2 TC INTPRET
 RTB GOTO
 LOADTIME
 R36INT
V06N16N VN 00616
V06N90N VN 00690
 SBANK= LOWSUPER

```

This code is written to file `src/P34-35-P74-75.s`.



**B.73 P37 P70**

1201

*<src/P37-P70.s 1201>*≡

```

Copyright: Public domain.
Filename: P37_P70.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Jim Lawton <jim.lawton@gmail.com>.
Website: www.ibiblio.org/apollo.
Pages: 890-933
Mod history: 2009-05-11 JVL Adapted from the Colossus249/ file
of the same name, using Comanche055 page
images.
2009-05-20 RSB Added missing label V2T179. Fixed POOD00 -> P00D00.
2009-05-23 RSB In RTD18, corrected a STOVL DELVLVC to
STODL DELVLVC and a STODL 02D to STORE 02D.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 890

 BANK 31
 SETLOC RTE1
 BANK

 EBANK= RTEDVD
 COUNT 31/P37

PROGRAM DESCRIPTION: P37, RETURN TO EARTH
#
```

```

DESCRIPTION
A RETURN TO EARTH TRAJECTORY IS COMPUTED PROVIDED THE CSM IS OUTSIDE THE LUNA
TIME OF IGNITION. INITIALLY A CONIC TRAJECTORY IS DETERMINED AND RESULTING I
DISPLAYED TO THE ASTRONAUT. THEN IF THE ASTRONAUT SO DESIRES, A PRECISION T
RESULTING IGNITION AND REENTRY PARAMETERS DISPLAYED. UPON FINAL ACCEPTANCE
COMPUTES AND STORES THE TARGET PARAMETERS FOR RETURN TO EARTH FOR USE BY SPS
#
CALLING SEQUENCE
L TC P37
#
SUBROUTINES CALLED
PREC100
V2T100
RTENCK2
RTENCK3
TIMERAD
PARAM
V2T100
GAMDV10
XT1LIM
DVCALC
RTENCK1
INTSTALL
INTEGRVS
RTEVN
RETDISP
TMRAD100
AUGKUGL
LAT-LONG
TMRAD100
TIMERAD
INVC100
CSMPREC
GETERAD
TIMETHET
P37OALRM
VN1645
POLY
#
ERASABLE INITIALIZATION REQUIRED
CSM STATE VECTOR
Page 891
NJETSFLG NUMBER OF JETS IF THE RCS PROPULSION SYSTEM SELECTED STAT
#
ASTRONAUT INPUT
SPRTEIG TIME OF IGNITION (OVERLAYS TIG) DP

```

July 12, 2016

1203

|        |                       |                                                         |                          |                                |
|--------|-----------------------|---------------------------------------------------------|--------------------------|--------------------------------|
| #      | VPRED                 | DESIRED CHANGE IN VELOCITY AT TIG(PROGRM COMPUTED IF 0) | DP                       | B7                             |
| #      | GAMMAEI               | DESIRED FLIGHT PATH ANGLE AT REENTRY (COMPUTED IF 0)    | DP                       | B0                             |
| #      | OPTION2               | PROPULSION SYSTEM OPTION                                | SP                       | B14                            |
| #      |                       |                                                         |                          |                                |
| #      | OUTPUT                |                                                         |                          |                                |
| #      | CONIC OR PRECISION    | TRAJECTORY DISPLAY                                      |                          |                                |
| #      | VPRED                 | VELOCITY MAGNITUDE AT 400,000 FT. ENTRY ALTITUDE        | DP                       | B7                             |
| #      | T3TOT4                | TRANSIT TIME TO 400,000 FT. ENTRY ALTITUDE              | DP                       | B28                            |
| #      | GAMMAEI               | FLIGHT PATH ANGLE AT 400,00 FT. ENTRY ALTITUDE          | DP                       | B0                             |
| #      | DELVLVC               | INITIAL VELOCITY CHANGE VECTOR IN LOCAL VERTICAL COORD. | VECTOR                   | B7                             |
| #      | LAT(SPL)              | LATITUDE OF THE LANDING SITE                            | DP                       | B0                             |
| #      | LNG(SPL)              | LONGITUDE OF THE LANDING SITE                           | DP                       | B0                             |
| #      | TARGETING COMPUTATION | DISPLAY                                                 |                          |                                |
| #      | TIG                   | RECOMPUTED TIG BASED ON THRUST OPTION                   | DP                       | B28                            |
| #      | TTOGO                 | TIME FROM TIG                                           | DP                       | B28                            |
| #      | +MGA                  | POSITIVE MIDDLE GIMBAL ANGLE                            | DP                       | B0                             |
| #      | THRUST PROGRAM        | COMMUNICATION                                           |                          |                                |
| #      | XDELVFLG              | EXTERNAL DELTA V FLAG                                   | STATE                    | FLAG                           |
| #      | NORMSW                | LAMBERT AIMPT ROTATION SWITCH                           | STATE                    | FLAG                           |
| #      | ECSTEER               | CROSS PRODUCT STEERING CONSTANT                         | SP                       | B2                             |
| #      | RTARG                 | CONICALLY INTEGRATED REENTRY POSITION VECTOR            | VECTOR                   | B29                            |
| #      | TPASS4                | REENTRY TIME                                            | DP                       | B28                            |
| P37    | TC                    | PHASCHNG                                                | # P37 IS NOT RESTARTABLE |                                |
|        | OCT                   | 4                                                       |                          |                                |
|        | TC                    | INTPRET                                                 |                          |                                |
|        | AXT,1                 | SXA,1                                                   |                          |                                |
|        | OCT                   | 04000                                                   |                          |                                |
|        |                       | ECSTEER                                                 |                          |                                |
|        | DLOAD                 |                                                         |                          |                                |
|        |                       | ZEROVECS                                                |                          |                                |
|        | STORE                 | VPRED                                                   |                          |                                |
|        | STORE                 | GAMMAEI                                                 |                          |                                |
|        | EXIT                  |                                                         |                          |                                |
|        | CAF                   | V6N33RTE                                                | # INPUT TIG              | STORED IN SPRTETIG             |
|        | TCR                   | P37OGOF                                                 | #                        | OVERLAYED WITH TIG             |
|        | TCF                   | -2                                                      | #                        | DISPLAY NEW DATA               |
|        | CAF                   | V6N60RTE                                                | #                        | INPUT REENTRY ANGLE IN GAMMAEI |
|        | TCR                   | P37GFRB1                                                | #                        | AND DESIRED DELTA V IN RETDVD  |
|        | TCF                   | -2                                                      | #                        | DISPLAY NEW DATA               |
| RTE299 | TC                    | INTPRET                                                 |                          |                                |
|        | SSP                   | DLOAD                                                   |                          |                                |
|        |                       | OVFIND                                                  |                          |                                |
|        |                       | 0                                                       |                          |                                |
|        |                       | VPRED                                                   |                          |                                |

# Page 892

|        |        |                      |                                         |      |
|--------|--------|----------------------|-----------------------------------------|------|
|        | STODL  | RTEDVD               |                                         |      |
|        |        | GAMMAEI              |                                         |      |
|        | STODL  | RTEGAM2D             |                                         |      |
|        |        | 1RTEB13              |                                         |      |
|        | STODL  | CONICX1              |                                         |      |
|        |        | C4RTE                |                                         |      |
|        | STCALL | MAMAX1               |                                         |      |
|        |        | INVC100              | # GET R(T1)/,V(T1)/,UR1/,UH/            |      |
|        | CLEAR  | DLOAD                |                                         |      |
|        |        | SLOWFLG              |                                         |      |
|        |        | RTEDVD               |                                         |      |
|        | BPL    | ABS                  |                                         |      |
|        |        | RTE317               |                                         |      |
|        | STORE  | RTEDVD               |                                         |      |
|        | DLOAD  | DSU                  |                                         |      |
|        |        | R(T1)                |                                         |      |
|        |        | K1RTE                |                                         |      |
|        | BMN    | SET                  |                                         |      |
|        |        | RTE317               |                                         |      |
|        |        | SLOWFLG              |                                         |      |
| RTE317 | DLOAD  | EXIT                 |                                         |      |
|        |        | R(T1)                |                                         |      |
|        | TC     | POLY                 |                                         |      |
|        | DEC    | 2                    |                                         |      |
|        | 2DEC   | 181000434. B-31      |                                         |      |
|        | 2DEC   | 1.50785145 B-2       |                                         |      |
|        | 2DEC*  | -6.49993057 E-9 B27* |                                         |      |
|        | 2DEC*  | 9.76938926 E-18 B56* |                                         |      |
|        | TC     | INTPRET              |                                         |      |
|        | SL1    |                      |                                         |      |
|        | STODL  | MAMAX2               | # C0+C1*R+C2*R**2+C3*R**3=MAMAX2 B30    |      |
|        |        | M9RTEB28             |                                         |      |
|        | STODL  | NN1A                 |                                         |      |
|        |        | K2RTE                |                                         |      |
| RTE320 | STODL  | RCON                 | # RCON=K2                               |      |
|        |        | RTEGAM2D             |                                         |      |
|        | BZE    | BDSU                 |                                         |      |
|        |        | RTE340               | # GOTORTE340 IF REENTRY ANGLE NOT INPUT |      |
|        |        | 1RTEB2               |                                         |      |
|        | PUSH   | COS                  | #                                       | PL02 |
|        | PDDL   | SIN                  |                                         |      |
|        | BDDV   | STADR                | #                                       | PL00 |
|        | STCALL | X(T2)                | # X(T2)=COT(GAM2D)                      | B0   |
|        |        | RTE360               |                                         |      |
| RTE340 | DLOAD  | DSU                  |                                         |      |

```

Page 893
R(T1)
K1RTE
BMN DLOAD
RTE350
K4RTE
STCALL X(T2) # X(T2)=K4
RTE360
DLOAD
K3RTE
STORE X(T2) # X(T2)=K3
RTE360
CALL
V2T100
BZE GOTO
RTE367
RTEALRM
RTE367
VLOAD
R(T1)/
STODL RVEC
RCON
STOVL RDESIRED
V2(T1)/
STCALL VVEC
TMRAD100
DAD
T1
STODL T2
RTEGAM2D
BZE GOTO
RTE369
RTE372
RTE369
VLOAD ABVAL
V(T2)/
EXIT
TC POLY
DEC 2
2DEC 0
2DEC -4.8760771 E-2 B4
2DEC 4.5419476 E-4 B11
2DEC -1.4317675 E-6 B18
TC INTPRET
DAD
RTED1
SL3 GOTO # X(T2),=D1+D2V2+D3V2**2+D4V2**3
RTE373

```

|            |        |                   |                                 |       |
|------------|--------|-------------------|---------------------------------|-------|
| RTE372     | DLOAD  |                   | # X(T2),=X(T2)                  |       |
| RTE373     | DSU    | X(T2)<br>PUSH     | # X(T2)ERR                      | B0 PL |
| # Page 894 |        |                   |                                 |       |
|            | VLOAD  | X(T2)<br>UNIT     |                                 |       |
|            |        | R(T2)/            | #                               | B58   |
|            | STCALL | ALPHAV<br>GETERAD |                                 |       |
|            | DAD    |                   |                                 |       |
|            |        | E3RTE             |                                 |       |
|            | PUSH   | DSU<br>RCON       | # RCON,=(E1/1+E2BETA11)**.5)+E3 | B29 P |
|            | ABS    | DSU               |                                 |       |
|            |        | EPC2RTE           |                                 |       |
|            | BMN    | GOTO              |                                 |       |
|            |        | RTE374            |                                 |       |
|            |        | RTE375            |                                 |       |
| RTE374     | DLOAD  | ABS               |                                 |       |
|            |        | OOD               |                                 |       |
|            | DSU    | BMN               |                                 |       |
|            |        | EPC3RTE           |                                 |       |
|            |        | P37E              |                                 |       |
| RTE375     | DLOAD  | DAD               |                                 |       |
|            |        | NN1A              |                                 |       |
|            |        | 1RTEB28           |                                 |       |
|            | BMN    | SLOAD             |                                 |       |
|            |        | RTE380            |                                 |       |
|            |        | OCT605            |                                 |       |
|            | GOTO   |                   |                                 |       |
|            |        | RTEALRM           | # TOO MANY ITERATIONS           |       |
| RTE380     | STORE  | NN1A              |                                 |       |
|            | DSU    | BZE               |                                 |       |
|            |        | M8RTEB28          |                                 |       |
|            |        | RTE385            |                                 |       |
|            | DLOAD  | DSU               |                                 |       |
|            |        | OOD               |                                 |       |
|            |        | DRCON             |                                 |       |
|            | NORM   | PDDL              | # X(T2)ERR-X(T2)ERR,=Z1         | PL061 |
|            |        | X1                |                                 |       |
|            |        | RPRE'             |                                 |       |
|            | DSU    | DDV               | # X(T2)PRI-X(T2)=Z2             | PL041 |
|            |        | X(T2)             |                                 |       |
|            | DMP    | SL*               | # DX(T2)=X(T2)ERR(Z2/Z1)        |       |
|            |        | OOD               |                                 |       |
|            |        | 0,1               |                                 |       |

```

 GOTO RTE390
RTE385 DLOAD 00D # DX(T2)=X(T2)ERR
RTE390 STODL 16D # DX(T2) PL02D
 STADR
 STODL RCON # RCON=RCON,
 BOV
Page 895
 RTE360
 STODL DRCON # X(T2)ERR,=X(T2)ERR
 X(T2)
 STODL RPRE' # X(T2)PRI=X(T2)
 16D
 DAD
 X(T2)
 STCALL X(T2) # X(T2)=X(T2)+DX(T2)
 RTE360 # REITERATE
P37E CALL # DISPLAY CONIC SOLUTION
 RTEVN
RTE505 DLOAD DMP
 PCON
 BETA1
 BDSU BZE
 RCON
 RTE510
 BMN DLOAD
 RTE510
 1RTEB2
 GOTO # ENTRY NEAR APOGEE
 RTE515
RTE510 DLOAD DCOMP # ENTRY NEAR PERIGEE
 1RTEB2
RTE515 STCALL PHI2
 PREC100 # PRECISION TRAJECTORY COMPUTATION
RTE625 BZE
 P37G
RTEALRM CALL
 P370ALRM
 EXIT
 TCF P37 # RECYCLE AFTER ALARM DISPLAY

RETURN TO EARTH DISPLAY SUBROUTINE
RTEVN STQ CALL
 VNSTORE

```

```

 RTEDISP # DISPLAY PREPARATION
EXIT
CAF V6N61RTE # LATITUDE, LONGITUDE, BLANK
TCR P370GOFR # IN LAT(SPL), LNG(SPL), -
CAF FOUR
TCR 37BLANK +1
TCF +5
TCF P37 # RECYCLE
CAF V6N39RTE # T21 HRS, MIN, SEC IN T3TOT4
TCR P370GOF
TCF P37 # RECYCLE
CAF V6N60RTE # DISPLAY BLANK, V(T2), FPA2
TCR P37GFRB1 # IN -, VPRED, GAMMAEI

Page 896
TCF P37 # RECYCLE
CAF V6N81RTE # DISPLAY DELTA V (LV) IN DELVLVC
TCR P370GOF
TCF P37 # RECYCLE
TCR INTPRET
GOTO VNSTORE

PRECISION DISPLAY, TARGETING COMPUTATION AND RTE END PROCESSING

P37G CALL
 RTEVN
EXIT
P37N CAF SEVEN
TS OPTION1
CAF ONE
TS OPTION2
CAF V4N06RTE # DISPLAY RCS OR SPS OPTION SPS ASSUMED
TCR P370GOF
TCF -2 # RECYCLE
TC INTPRET # PROCEED
SETPD SLOAD
 OOD
 OPTION2
DSU BZE
 1RTEB13
 P37Q
SLOAD NORM # SPS
 EMDOT
 X1
PDDL GOTO
 VCSPS

```



July 12, 2016

1209

|            |       |                    |                                           |               |       |
|------------|-------|--------------------|-------------------------------------------|---------------|-------|
| P37Q       | DLOAD | P37T<br>BON        | # RCS                                     |               |       |
|            |       | MDOTRCS            |                                           |               |       |
|            |       | NJETSFLG           |                                           |               |       |
|            |       | P37R               |                                           |               |       |
| P37R       | SL1   |                    |                                           |               |       |
|            | SL1   |                    |                                           |               |       |
|            | NORM  | PDDL               |                                           |               |       |
|            |       | X1                 |                                           |               |       |
|            |       | VCRCS              |                                           |               |       |
| P37T       | PDDL  | DDV                | # DV/VC                                   | B7 -B5 = B2   | PL02D |
|            |       | DV                 |                                           |               |       |
|            | EXIT  |                    |                                           |               |       |
|            | TC    | POLY               |                                           |               |       |
|            | DEC   | 1                  |                                           |               |       |
|            | 2DEC  | 5.66240507 E-4 B-3 |                                           |               |       |
|            | 2DEC  | 9.79487897 E-1 B-1 |                                           |               |       |
| # Page 897 |       |                    |                                           |               |       |
|            | 2DEC  | -.388281955 B1     |                                           |               |       |
|            | TC    | INTPRET            |                                           |               |       |
|            | PUSH  | SLOAD              | # (1-E)**(-DV/VC)=A                       | B3            | PL04D |
|            |       | WEIGHT/G           |                                           |               |       |
|            | DMP   | DDV                | # DTB=(M0/MDOT)A                          | B16+B3-B3=B16 | PL00D |
|            | SL*   | DMP                |                                           |               |       |
|            |       | 0 -12D,1           |                                           |               |       |
|            |       | CSUBT              |                                           |               |       |
|            | BDSU  |                    |                                           |               |       |
|            |       | T1                 |                                           |               |       |
|            | STORE | TIG                | # TIG=T1-CT*DTB                           | B28           |       |
|            | EXIT  |                    |                                           |               |       |
|            | CAF   | V6N33RTE           | # DISPLAY BIASED TIG                      |               |       |
|            | TCR   | P370GOF            |                                           |               |       |
|            | TCF   | -2                 |                                           |               |       |
|            | CAF   | ZERO               |                                           |               |       |
|            | TS    | VHFCNT             |                                           |               |       |
|            | TS    | TRKMKCNT           |                                           |               |       |
|            | TC    | INTPRET            |                                           |               |       |
|            | CALL  |                    | # CONICALLY INTEGRATE FROM R1,V1 OVER T12 |               |       |
|            |       | RTENCK1            |                                           |               |       |
|            | VLOAD | UNIT               | #                                         |               | PL00D |
|            |       | R(T2)/             |                                           |               |       |
|            | PDVL  | VXSC               | # UR2                                     | B1            | PL06D |
|            |       | UR1/               |                                           |               |       |
|            |       | MCOS7.5            |                                           |               |       |
|            | PDVL  | VXSC               | # -UR1(COS7.5)                            | B1            | PL12D |
|            |       | UH/                |                                           |               |       |

```

MSIN7.5
VAD DOT # K/=-UR1(COS7.5)-UH(SIN7.5) B2 PLOO
DAD BMN
MCOS22.5
P37W
VLOAD DOT # K/ . UR2 GR COS22.5
UH/
R(T2)/
BMN DLOAD
P37U
THETA165
PUSH GOTO
P37V
P37U DLOAD PUSH
THETA210
P37V SIN
STODL SNTH
COS CLEAR
RVSW
STOVL CSTH
R(T1)/
Page 898
STOVL RVEC
V2(T1)/
STCALL VVEC
TIMETHET
P37W CLEAR CLEAR
XDELVFLG
NORMSW
SET VLOAD
FINALFLG
STADR
STODL RTARG
T
DAD
T1
STOVL TPASS4
V2(T1)/
VSU
V(T1)/
STCALL DELVSIN
VN1645
GOTO
P37W

```

# SUBROUTINE TO GO TO GOFLASHR AND BLANK R1

```

P37GFRB1 EXTEND
 QXCH SPRTEX
 TCR P370GOFR
37BLANK CAF ONE
 TCR BLANKET
 TCF ENDOFJOB
 TC SPRTEX # RECYCLE
 TCF P37PROC # PROCEED

```

```

SUBROUTINE TO GO TO GOFLASHR

```

```

P370GOFR EXTEND
 QXCH RTENCKEX
 TCR BANKCALL
 CADR GOFLASHR
 TCF GOTOP00H # TERMINATE
 TCF +3
 TCF +4
 TC RTENCKEX # IMMEDIATE RETURN
 INDEX RTENCKEX # PROCEED
 TCF 0 +4
 INDEX RTENCKEX # RECYCLE
 TCF 0 +3

```

```

SUBROUTINE TO GO TO GOFLASH

```

```

Page 899

```

```

P370GOF EXTEND
 QXCH SPRTEX
 TCR BANKCALL
 CADR GOFLASH
 TCF GOTOP00H
 TCF +2
 TC SPRTEX
P37PROC INDEX SPRTEX
 TCF 0 +1
V6N33RTE VN 0633
V4N06RTE VN 0406
V6N61RTE VN 0661
V6N39RTE VN 0639
V6N60RTE VN 0660
V6N81RTE VN 0681
 BANK 32
 SETLOC RTE
 BANK

```

COUNT 32/RTE

# Page 900

# ALARM DISPLAY SUBROUTINE

```

P370ALRM STQ EXIT
 SPRTEX
 CA MPAC
 TC VARALARM
 CAF V5N09RTE
 TC BANKCALL
 CADR GOFLASH
 TCF GOTOPOOH
 TCF -4
 TC INTPRET
 GOTO
 SPRTEX
V5N09RTE VN 0509

```

# Page 901

# TIME RADIUS CALLING SUBROUTINE

#

# INPUT

```

RVEC INITIAL POSITION VECTOR VECTOR
VVEC INITIAL VELOCITY VECTOR VECTOR
RDESIRED FINAL RADIUS FOR WHICH TRANSFER TIME IS TO BE COMPUTED DP
CONICX1 X1 SETTING FOR CONIC SUBROUTINES -2=EARTH SP
#

```

# OUTPUT

```

R(T2)/ FINAL POSITION VECTOR VECTOR
V(T2)/ FINAL VELOCITY VECTOR VECTOR
T12 TRANSFER TIME TO FINAL RADIUS DP

```

```

TMRAD100 STQ CLEAR
 RTENCKEX
 RVSW
 AXC,2 SXA,2
 OCT 20000
 SGNRDOT
 LXC,1 CALL
 CONICX1
 TIMERAD
 STOVL V(T2)/
 STADR
 STODL R(T2)/
 T

```

PLOOR

July 12, 2016

1213

STCALL T12  
RTENCKEX

```
Page 902
DISPLAY CALCULATION SUBROUTINE
#
DESCRIPTION
OUTPUT FOR DISPLAY IS CONVERTED TO PROPER UNITS AND PLACED IN OUTPUT STORAGE REGISTERS.
COMPUTATION FOR DETERMINING LANDING SITE LATITUDE AND LONGITUDE IS INCLUDED IN THE ROUTINE
#
CALLING SEQUENCE
L CALL
L+1 RTEDISP
#
SUBROUTINES CALLED
TMRAD100
AUGKUGL
LAT-LONG
#
ERASABLE INITIALIZATION REQUIRED
PUSHLIST
NONE
MPAC
NONE
OTHER
R(T2)/ FINAL POSITION VECTOR VECTOR B29
V(T2)/ FINAL VELOCITY VECTOR VECTOR B7
T2 FINAL TIME DP B28
V2(T1)/ POST IMPULSE INITIAL VELOCITY VECTOR VECTOR B7
V(T1)/ INITIAL VELOCITY VECTOR VECTOR B7
UR1/ UNIT INITIAL VECTOR VECTOR B1
UH/ UNIT HORIZONTAL VECTOR VECTOR B1
#
OUTPUT
VPREL VELOCITY MAGNITUDE AT 400,000 FT. ENTRY ALTITUDE DP B7
T3TOT4 TRANSIT TIME TO 400,000 FT. ENTRY ALTITUDE DP B28
GAMMAEI FLIGHT PATH ANGLE AT 400,000 FT. ENTRY ALTITUDE DP B0
DELVLVC INITIAL VELOCITY CHANGE VECTOR IN LOCAL VERTICAL COORD. VECTOR B7
LAT(SPL) LATITUDE OF THE LANDING SITE DP B0
LNG(SPL) LONGITUDE OF THE LANDING SITE DP B0

RTEDISP STQ VLOAD # DISPLAY
 SPRTEX
 V(T2)/
 UNIT PDDL
 36D
```

```

STODL VPRED # V(T2)
 T2
DSU
 SPRTETIG
STOVL T3TOT4 # T21
 R(T2)/
UNIT DOT
SL1

Page 903
ARCCOS BDSU
 1RTEB2
STOVL GAMMAEI # FLIGHT PATH ANGLE T2
 V2(T1)/
VSU PUSH
 V(T1)/
DOT DCOMP
 UR1/
PDVL PUSH
DLOAD PDVL
 ZERORTE
DOT VDEF
 UH/
VSL1
STODL DELVLVC
 DELVLVC
BOFF DCOMP
 RETROFLG
 RTD18
STORE DELVLVC # NEGATE X COMPONENT, RETROGRADE
VLOAD ABVAL
 DELVLVC
STOVL VGDISP
 R(T2)/
STORE RVEC # ***** LANDING SITE COMPUTATION *****
ABVAL DSU
 3048ORTE
STOVL RDESIRED
 V(T2)/
STCALL VVEC
 TMRAD100 # R3,V3,T23 FROM TIMERAD
VLOAD UNIT
 R(T2)/
PDVL UNIT # UR3
 V(T2)/
DOT SL1 # GAMMAE=ARCSIN(UR3 . UV3)

```

July 12, 2016

1215

|       |        |          |                            |       |
|-------|--------|----------|----------------------------|-------|
|       | ARCSIN | PDDL     | # V(T3)                    | PL02D |
|       |        | 36D      |                            |       |
|       | PDDL   | ABS      |                            |       |
|       | PUSH   | CALL     | # /GAMMAE/                 | PL04D |
|       |        | AUGEKUGL | # PHIE                     | PL06D |
|       | DAD    | DAD      |                            |       |
|       |        | T12      | # T23                      |       |
|       |        | T2       |                            |       |
|       | STORE  | 02D      | # T(LS)=T2&T23&TE          |       |
|       | SLOAD  | BZE      |                            |       |
|       |        | P37RANGE |                            |       |
|       |        | RTD22    |                            |       |
|       | STORE  | 04D      | # OVERRIDE RANGE (PCR 261) |       |
| RTD22 | DLOAD  | SIN      |                            |       |

# Page 904

|  |        |          |                                    |       |
|--|--------|----------|------------------------------------|-------|
|  |        | 04D      |                                    |       |
|  | STODL  | LNG(SPL) | # LNG(SPL)=SIN(PHIE)               | PL04D |
|  | COS    |          |                                    |       |
|  | STORE  | LAT(SPL) | # LAT(SPL)=COS(PHIE)               |       |
|  | VLOAD  | UNIT     |                                    |       |
|  |        | R(T2)/   |                                    |       |
|  | PUSH   | PUSH     |                                    |       |
|  | PDVL   | UNIT     | #                                  | PL22D |
|  |        | V(T2)/   |                                    |       |
|  | PDVL   | VXV      |                                    |       |
|  | VXV    | UNIT     | # UH3=UNIT(UR3 X UV3 X UR3)        | PL10D |
|  | VXSC   | PDVL     |                                    |       |
|  |        | LNG(SPL) |                                    |       |
|  | VXSC   | VAD      | #                                  | PL04D |
|  |        | LAT(SPL) |                                    |       |
|  | CLEAR  | CLEAR    | # T(LS) IN MPAC                    |       |
|  |        | ERADFLAG |                                    |       |
|  |        | LUNAFLAG |                                    |       |
|  | STODL  | ALPHAV   | # ALPHAV=UR3(COSPHIE)+UH3(SINPHIE) | PL02D |
|  | CALL   |          |                                    |       |
|  |        | LAT-LONG |                                    |       |
|  | DLOAD  |          |                                    |       |
|  |        | LAT      |                                    |       |
|  | STODL  | LAT(SPL) | # LATITUDE LANDING SITE *****      |       |
|  |        | LONG     |                                    |       |
|  | STCALL | LNG(SPL) | # LONGITUDE LANDING SITE *****     |       |
|  |        | SPRTEX   |                                    |       |
|  | COUNT* | \$\$/RTE |                                    |       |

# Page 905

```

INITIAL VECTOR SUBROUTINE
#
DESCRIPTION
A PRECISION INTEGRATION OF THE STATE VECTOR TO THE TIME OF IGNITION IS PERFORMED
#
CALLING SEQUENCE
L CALL
L+1 INVC100
#
NORMAL EXIT MODE
AT L+2 OF CALLING SEQUENCE WITH MPAC = 0
#
ALARM EXIT MODE
AT L+2 OF CALLING SEQUENCE WITH MPAC = OCTAL 612 FOR STATE VECTOR IN MOONS STATE
#
SUBROUTINES CALLED
CSMPREC
#
ERASABLE INITIALIZATION REQUIRED
PUSHLIST
NONE
MPAC
NONE
OTHER
SPRTETIG TIME OF IGNITION
CSM STATE VECTOR
#
OUTPUT
R(T1)/ INITIAL POSITION VECTOR AT TIG
V(T1)/ INITIAL VELOCITY VECTOR AT TIG
T1 INITIAL VECTOR TIME (TIG)
UR1/ UNIT INITIAL VECTOR
UH/ UNIT HORIZONTAL VECTOR
CFPA COSINE OF INITIAL FLIGHT PATH ANGLE
#
INVC100 STQ DLOAD
 SPRTEX
 SPRTETIG
 STCALL TDEC1
 CSMPREC # PRECISION INTEGRATION R0,V0 TO R1,V1
 VLOAD SXA,2
 RATT
 P(T1)
 STOVL R(T1)/
 VATT
 STODL V(T1)/

```



July 12, 2016

1217

|            |       |          |                                           |     |
|------------|-------|----------|-------------------------------------------|-----|
|            |       | TAT      |                                           |     |
|            | STORE | T1       |                                           |     |
|            | SLOAD | BZE      |                                           |     |
|            |       | P(T1)    |                                           |     |
| # Page 906 |       |          |                                           |     |
|            |       | INVC109  |                                           |     |
| INVC107    | SLOAD | GOTO     |                                           |     |
|            |       | OCT612   |                                           |     |
|            |       | RTEALRM  | # R1,V1 NOT IN PROPER SPHERE OF INFLUENCE |     |
| INVC109    | VLOAD | UNIT     |                                           |     |
|            |       | R(T1)/   |                                           |     |
|            | STODL | UR1/     | # UR1/                                    | B1  |
|            |       | 36D      |                                           |     |
|            | STOVL | R(T1)    | # R(T1)                                   | B29 |
|            |       | V(T1)/   |                                           |     |
|            | UNIT  |          |                                           |     |
|            | STORE | UV1/     |                                           |     |
|            | DOT   | SL1      |                                           |     |
|            |       | UR1/     |                                           |     |
|            | STORE | CFPA     | # CFPA                                    | B1  |
|            | ABS   | DSU      |                                           |     |
|            |       | EPC1RTE  |                                           |     |
|            | BMN   | DLOAD    |                                           |     |
|            |       | INVC115  | # NOT NEAR RECTILINEAR                    |     |
|            |       | 1RTEB2   |                                           |     |
|            | PDDL  | PUSH     |                                           |     |
|            |       | ZERORTE  |                                           |     |
|            | VDEF  | PUSH     | # N/ = (0,0,1)                            |     |
|            | GOTO  |          |                                           |     |
|            |       | INVC120  |                                           |     |
| INVC115    | VLOAD | VXV      |                                           |     |
|            |       | UR1/     |                                           |     |
|            |       | UV1/     |                                           |     |
|            | PUSH  |          | # N/ = UR X UV                            | B2  |
| INVC120    | CLEAR | DLOAD    |                                           |     |
|            |       | RETROFLG |                                           |     |
|            | PUSH  | BPL      |                                           |     |
|            |       | INVC125  |                                           |     |
|            | VLOAD | VCOMP    | # RETROGRADE ORBIT                        |     |
|            | PUSH  | SET      |                                           |     |
|            |       | RETROFLG |                                           |     |
| INVC125    | VLOAD |          |                                           |     |
|            | VXV   | UNIT     |                                           |     |
|            |       | UR1/     |                                           |     |
|            | STORE | UH/      | # UH/                                     | B1  |
|            | GOTO  |          |                                           |     |

## SPRTEX

```

Page 907
PRECISION TRAJECTORY COMPUTATION SUBROUTINE
#
DESCRIPTION
A NUMERICALLY INTEGRATED TRAJECTORY IS GENERATED WHICH FOR THE RETURN TO EARTH
CONSTRAINTS (RCON AND X(T2)) ACHIEVED BY THE INITIAL CONIC TRAJECTORY AND MEET
AS POSSIBLE.
#
CALLING SEQUENCE
L CALL
L+1 PREC100
#
NORMAL EXIT MODE
AT L+2 OF CALLING SEQUENCE WITH MPAC = 0
#
ALARM EXIT MODE
AT L+2 OF CALLING SEQUENCE WITH MPAC =
OCTAL 605 FOR EXCESS ITERATIONS
OCTAL 613 FOR REENTRY ANGLE OUT OF LIMITS
#
SUBROUTINES CALLED
INTSTALL
RTENCK2
RTENCK3
TIMERAD
PARAM
V2T100
#
ERASABLE INITIALIZATION REQUIRED
PUSHLIST
NONE
MPAC
NONE
OTHER
R(T1)/ INITIAL POSITION VECTOR VECTOR
V2(T1)/ POST IMPULSE INITIAL VELOCITY VECTOR VECTOR
V(T1)/ INITIAL VELOCITY VECTOR VECTOR
T1 INITIAL VECTOR TIME DP
T12 INITIAL TO FINAL POSITION TIME DP
RCON CONIC FINAL RADIUS DP
R(T1) MAGNITUDE OF INITIAL POSITION VECTOR DP
X(T2) COTANGENT OF FINAL FLIGHT PATH ANGLE DP
X(T1) COTANGENT OF INITIAL FLIGHT PATH ANGLE DP
RTEDVD DELTA VELOCITY DESIRED DP

```

July 12, 2016

1219

|         |          |                                                    |                   |         |
|---------|----------|----------------------------------------------------|-------------------|---------|
| #       | MAMAX1   | MAJOR AXIS LIMIT FOR LOWER BOUND ON GAMDV ITERATOR | DP                | B30/B28 |
| #       | MAMAX2   | MAJOR AXIS LIMIT FOR UPPER BOUND ON GAMDV ITERATOR | DP                | B30/B28 |
| #       | UR1/     | UNIT INITIAL VECTOR                                | VECTOR            | B1      |
| #       | UH/      | UNIT HORIZONTAL VECTOR                             | VECTOR            | B1      |
| #       | BETA1    | $1+X(T2)**2$                                       | DP                | B1      |
| #       | PHI2     | PERIGEE OR APOGEE INDICATOR                        | DP                | B2      |
| #       |          |                                                    |                   |         |
| #       | Page 908 |                                                    |                   |         |
| #       |          |                                                    |                   |         |
| #       | OUTPUT   |                                                    |                   |         |
| #       | V2(T1)/  | POST IMPULSE INITIAL VELOCITY VECTOR               | VECTOR            | B7      |
| #       | R(T2)/   | FINAL POSITION VECTOR                              | VECTOR            | B29     |
| #       | V(T2)/   | FINAL VELOCITY VECTOR                              | VECTOR            | B7      |
| #       | T2       | FINAL TIME                                         | DP                | B28     |
| #       |          |                                                    |                   |         |
| #       | DEBRIS   |                                                    |                   |         |
| #       | RD       | FINAL R DESIRED                                    | DP                | B29/B27 |
| #       | R/APRE   | R/A                                                | DP                | B6      |
| #       | P/RPRE   | P/R                                                | DP                | B4      |
| #       | RPRE     | MAGNITUDE OF R(T2)/                                | DP                | B29/B27 |
| #       | X(T2)PRE | COTANGENT OF GAMMA2                                | DP                | B0      |
| #       | DT12     | CORRECTION TO FINAL TIME T2                        | DP                | B28     |
| #       | RCON     | FINAL RADIUS                                       | DP                | B29/B27 |
| #       | DRCON    | DELTA RCON                                         | DP                | B29/B27 |
|         |          |                                                    |                   |         |
| PREC100 | STQ      | DLOAD                                              |                   |         |
|         |          | SPRTEX                                             |                   |         |
|         |          | 1ORTE                                              |                   |         |
|         | STODL    | NN1A                                               |                   |         |
|         |          | RCON                                               |                   |         |
|         | STORE    | RD                                                 |                   |         |
| PREC120 | DLOAD    |                                                    |                   |         |
|         |          | 2RTEB1                                             |                   |         |
|         | STODL    | DT21PR                                             | # DT21PR = POSMAX |         |
|         |          | M15RTE                                             |                   |         |
|         | STCALL   | NN2                                                |                   |         |
|         |          | RTENCK3                                            |                   |         |
| PREC125 | CALL     |                                                    |                   |         |
|         |          | PARAM                                              |                   |         |
|         | DLOAD    |                                                    |                   |         |
|         |          | P                                                  |                   |         |
|         | STODL    | P/RPRE                                             |                   |         |
|         |          | R1A                                                |                   |         |
|         | STODL    | R/APRE                                             |                   |         |
|         |          | R1                                                 |                   |         |
|         | STODL    | RPRE                                               |                   |         |

```

 COGA
SL 5
STORE X(T2)PRE
DCOMP DAD
 X(T2)
ABS DSU
 EPC4RTE
BOV BMN
 PREC130
 PREC175

DESIRED REENTRY ANGLE NOT ACHIEVED

Page 909
PREC130 DLOAD BMN
 NN2
 PREC140
PREC132 SLOAD GOTO # TOO MANY ITERATIONS
 OCT605 # EXIT WITH ALARM
 PRECX

DETERMINE RADIUS AT WHICH THE DESIRED REENTRY ANGLE WILL BE ACHIEVED

PREC140 DLOAD BZE
 NN1A
 PREC162
PREC150 DLOAD SL2 B2
 P/RPRE
 DMP SL1 # BETA2=BETA1*P/R B2 PL02
 BETA1
 PUSH DLOAD
 R/APRE
 SL4 DMP
 OOD
 BDSU BMN # BETA3=1-BETA2*R/A
 1RTEB4
 PREC160
PREC155 SL2 SQRT
 DMP BDSU
 PHI2
 1RTEB3
 NORM PDDL
 X1
 SR1 DDV # BETA4=BETA2/(1-PHI2*SQRT(BETA3))
 SL* GOTO B1

```

July 12, 2016

1221

|            |       |          |                   |    |
|------------|-------|----------|-------------------|----|
|            |       | 0        | -1,1              |    |
|            |       | PREC165  |                   |    |
| PREC160    | DLOAD | NORM     |                   |    |
|            |       | R/APRE   |                   |    |
|            |       | X1       |                   |    |
|            | BDDV  | SL*      |                   | B1 |
|            |       | 1RTEB1   |                   |    |
|            |       | 0        | -6,1              |    |
|            | GOTO  |          |                   |    |
|            |       | PREC165  |                   |    |
| PREC162    | DLOAD | NORM     |                   |    |
|            |       | RPRE     |                   |    |
|            |       | X1       |                   |    |
|            | BDDV  | SL*      | # BETA4=RD/RPRE   | B1 |
|            |       | RD       |                   |    |
|            |       | 0        | -1,1              |    |
| PREC165    | SETPD | PUSH     |                   |    |
|            |       | 0        |                   |    |
|            | DSU   | DCOMP    |                   |    |
| # Page 910 |       |          |                   |    |
|            |       | 1RTEB1   |                   |    |
|            | STORE | BETA12   |                   |    |
|            | BMN   | DLOAD    |                   |    |
|            |       | PREC168  |                   |    |
|            |       | X(T2)PRE |                   |    |
|            | BMN   | DLOAD    |                   |    |
|            |       | PREC167  |                   |    |
|            |       | BETA12   |                   |    |
|            | DCOMP |          |                   |    |
|            | STORE | BETA12   |                   |    |
| PREC167    | DLOAD |          |                   |    |
|            |       | BETA12   |                   |    |
| PREC168    | ABS   | DSU      |                   |    |
|            |       | EPC6RTE  |                   |    |
|            | BMN   | DLOAD    |                   |    |
|            |       | PREC175  |                   |    |
|            | DMP   | SL1      |                   |    |
|            |       | RPRE     |                   |    |
|            | PUSH  |          | # RF = NEW RADIUS |    |
| PREC170    | DLOAD | DAD      |                   |    |
|            |       | NN2      |                   |    |
|            |       | 1RTEB28  |                   |    |
|            | STORE | NN2      |                   |    |
|            | VLOAD | SET      |                   |    |
|            |       | R(T2)/   |                   |    |
|            |       | RVSW     |                   |    |

|            |        |          |                                        |     |      |
|------------|--------|----------|----------------------------------------|-----|------|
|            | STOVL  | RVEC     |                                        |     |      |
|            |        | V(T2)/   |                                        |     |      |
|            | SIGN   |          |                                        |     |      |
|            |        | BETA12   |                                        |     |      |
|            | STODL  | VVEC     |                                        |     |      |
|            |        | 1RTEB1   |                                        |     |      |
|            | SIGN   | DCOMP    |                                        |     |      |
|            |        | BETA12   |                                        |     |      |
|            | LXA,2  | DLOAD    |                                        |     |      |
|            |        | MPAC     |                                        |     |      |
|            | LXC,1  | SXA,2    |                                        |     |      |
|            |        | CONICX1  |                                        |     |      |
|            |        | SGNRDOT  |                                        |     |      |
|            | STCALL | RDESIRED | # COMPUTED DT12 (CORRECTION TO TIME OF |     |      |
|            |        | TIMERAD  | # NEW RADIUS)                          |     |      |
|            | DLOAD  | SIGN     |                                        |     |      |
|            |        | T        |                                        |     |      |
|            |        | BETA12   |                                        |     |      |
|            | PDDL   | NORM     | # DT21=(PHI4)DT21                      |     | PL02 |
|            |        | DT21PR   |                                        |     |      |
|            |        | X1       |                                        |     |      |
|            | BDDV   | SL*      |                                        |     |      |
|            |        | OOD      |                                        |     |      |
|            |        | 0 -3,1   |                                        |     |      |
| # Page 911 |        |          |                                        |     |      |
|            | PUSH   | BMN      | # BETA13=(DT21)/(DT21PR)               | R3  | PL04 |
|            |        | PREC172  |                                        |     |      |
|            | DLOAD  | PDDL     | # BETA14=1                             | B0  | PL04 |
|            |        | 2RTEB1   |                                        |     |      |
|            | GOTO   |          |                                        |     |      |
|            |        | PREC173  |                                        |     |      |
| PREC172    | DLOAD  | PDDL     | # BETA14=.6                            | B0  | PL04 |
|            |        | M.6RTE   |                                        |     |      |
| PREC173    | DDV    | DSU      |                                        |     |      |
|            |        | O2D      |                                        |     |      |
|            |        | 1RTEB3   |                                        |     |      |
|            | BMN    | DLOAD    |                                        |     |      |
|            |        | PREC174  |                                        |     |      |
|            | DMP    |          |                                        |     |      |
|            |        | DT21PR   |                                        |     |      |
|            | STORE  | OOD      | # DT21=(BETA14)DT21PR                  | B28 |      |
| PREC174    | DLOAD  | PUSH     |                                        |     |      |
|            |        | OOD      |                                        |     |      |
|            | STCALL | DT21PR   |                                        |     |      |
|            |        | RTENCK2  |                                        |     |      |
|            | GOTO   |          |                                        |     |      |

July 12, 2016

1223

|                                        |       |                              |                               |         |
|----------------------------------------|-------|------------------------------|-------------------------------|---------|
| PREC175                                | DLOAD | PREC125<br>DSU<br>RPRE<br>RD |                               |         |
|                                        | PUSH  | ABS                          | # RPRE-RD = RERR              |         |
|                                        | DSU   | BMN<br>EPC7RTE<br>PREC220    |                               |         |
| # DESIRED RADIUS HAS NOT BEEN ACHIEVED |       |                              |                               |         |
|                                        | DLOAD | BZE<br>NN1A<br>PREC132       | # TOO MANY ITERATIONS         |         |
|                                        | DSU   | BZE<br>1ORTE<br>PREC207      |                               |         |
| PREC205                                | DLOAD | DSU                          | # NOT FIRST PASS OF ITERATION |         |
|                                        |       | RPRE'                        |                               |         |
|                                        |       | RPRE                         | # RPRE'-RPRE                  | B29/B27 |
|                                        | NORM  | BDDV<br>X2<br>DRCON          |                               |         |
|                                        | SL*   | PUSH                         | # DRCON/(RPRE'-RPRE)=S        | B2      |
|                                        |       | 0 -2,2                       |                               |         |
|                                        | DAD   | BOV                          | # S GR +4 OR LS -4            |         |
|                                        |       | 1RTEB1<br>PREC205M           |                               |         |
|                                        | ABS   | DSU                          |                               |         |
| # Page 912                             |       |                              |                               |         |
|                                        |       | 1RTEB1                       |                               |         |
|                                        | BMN   |                              |                               |         |
|                                        |       | PREC206                      |                               |         |
| PREC205M                               | DLOAD | DCOMP                        | # S GR 0 OR LS -4             |         |
|                                        |       | 2RTEB1                       |                               |         |
|                                        | PDDL  |                              | # S=-4                        | B2      |
| PREC206                                | DLOAD | DMP                          |                               |         |
|                                        | SL2   |                              |                               |         |
|                                        | STORE | DRCON                        | # DRCON=S(RERR)               | B29     |
|                                        | DAD   |                              |                               |         |
|                                        |       | RCON                         |                               |         |
|                                        | STORE | RCON                         | # RCON+DRCON=RCON             |         |
|                                        | GOTO  |                              |                               |         |
|                                        |       | PREC210                      |                               |         |
| PREC207                                | DLOAD | DSQ                          | # FIRST PASS OF ITERATION     |         |
|                                        |       | RD                           |                               |         |

```

 NORM SR1
 X1
 PDDL NORM
 RPRE
 X2
 XSU,1 BDDV
 X2
 SR*
 0 -1,1
 STORE RCON # RD**2/RPRE=RCON
 DSU
 RD
 STORE DRCON # RCON-RD=DRCON
PREC210 DLOAD # PREPARE FOR NEXT ITERATION
 RPRE
 STODL RPRE'
 NN1A
 DSU
 1RTEB28
 STCALL NN1A
 V2T100
 BHIZ GOTO
 PREC120
 PRECX

DESIRED RADIUS ACHIEVED

 SETLOC RTE2
 BANK
PREC220 DLOAD DSU
 X(T2)
 X(T2)PRE
 ABS DSU
 EPC8RTE

Page 913
 BMN SLOAD
 PREC225
 OCT613
 GOTO
 PRECX # IF REENTRY ANGLE OUT OF LIMITS

EPC8RTE 2DEC .002

OCT613 OCT 613

DESIRED FINAL ANGLE HAS BEEN REACHED.

```



```

 SETLOC RTE
 BANK
PREC225 DLOAD
 ZERORTE
PRECX GOTO
 SPRTEX

```

```
Page 914
```

```
INTEGRATION CALLING SUBROUTINE
```

```
#
```

```
DESCRIPTION
```

```
PERFORMS CONIC AND PRECISION INTEGRATIONS USING SUBROUTINE INTEGRVS. THERE ARE THREE E
RTENCK2, AND RTENCK3) FOR DIFFERENT SOURCES OF INPUT AND DIFFERENT OPTIONS. THERE IS A
WHICH INCLUDES SET UP OF INPUT FOR THE PARAM SUBROUTINE.
```

```
#
```

```
RTENCK1 (CONIC INTEGRATION)
```

```
#
```

```
CALLING SEQUENCE
```

```
L CALL
L+1 RTENCK1
```

```
#
```

```
ERASABLE INITIALIZATION REQUIRED
```

```
SAME AS FOR THE RTENCK3 ENTRANCE
```

```
#
```

```
RTENCK2 (PRECISION INTEGRATION)
```

```
#
```

```
CALLING SEQUENCE
```

```
L CALL
L+1 RTENCK2
```

```
#
```

```
ERASABLE INITIALIZATION REQUIRED
```

```
PUSHLIST
```

```
PUSHLOC-2 INTEGRATION TIME DT12 (CORRECTION TO T2) DP B28
```

```
OTHER
```

```
R(T2)/ FINAL POSITION VECTOR VECTOR B29
```

```
V(T2)/ FINAL VELOCITY VECTOR VECTOR B7
```

```
T2 FINAL TIME DP B28
```

```
#
```

```
RTENCK3 (PRECISION INTEGRATION)
```

```
#
```

```
CALLING SEQUENCE
```

```
L CALL
L+1 RTENCK3
```

```
#
```

```
ERASABLE INITIALIZATION REQUIRED
```

```

R(T1)/ INITIAL POSITION VECTOR VECTOR
V2(T1)/ POST IMPULSE INITIAL VELOCITY VECTOR VECTOR
T1 INITIAL VECTOR TIME DP
T2 FINAL TIME DP
#
EXIT MODE
AT L+2 OF CALLING SEQUENCE
#
SUBROUTINES CALLED
INTSTALL
INTEGRVS
#
OUTPUT
PUSHLIST
Page 915
PUSHLOC-6 FINAL POSITION VECTOR R(T2)/ VECTOR
X1 CONICS MUTABLE ENTRY FOR EARTH (-2) SP
MPAC
#
FINAL VELOCITY VECTOR V(T2)/ VECTOR
OTHER
R(T2)/ AS IN PUSHLIST
V(T2)/ AS IN MPAC
T2 FINAL TIME DP
#
SETLOC RTE3
BANK
RTENCK1 STQ CALL
RTENCKEX
INTSTALL
VLOAD SET
R(T1)/
INTYPFLG
GOTO RTENCK3B
#
RTENCK2 STQ CALL
RTENCKEX
INTSTALL
CLEAR VLOAD
INTYPFLG
R(T2)/
STOVL RCV
V(T2)/
STODL VCV
T2
STORE TET

```

July 12, 2016

1227

```

DAD
GOTO RTENCK3D

RTENCK3 STQ CALL
 RTENCKEX
 INTSTALL

RTENCK3A VLOAD CLEAR
 R(T1)/
 INTYPFLG

RTENCK3B STOVL RCV
 V2(T1)/
 STODL VCV
 T1
 STODL TET
 T2

Page 916
RTENCK3D STORE TDEC1
 CLEAR CALL
 MOONFLAG
 INTEGRVS
 VLOAD
 RATT
 STORE R(T2)/
 PDDL LXC,1
 TAT
 CONICX1
 STOVL T2
 VATT
 STORE V(T2)/
 GOTO
 RTENCKEX
 SETLOC RTE
 BANK
```

# Page 917

# V2(T1) COMPUTATION SUBROUTINE

#

# DESCRIPTION

# A POST IMPULSE VELOCITY VECTOR (V2(T1)) IS COMPUTED WHICH EITHER

# (1) MEETS THE INPUT VELOCITY CHANGE DESIRED (RTEDVD) IN A MINIMUM TIME OR

# (2) IF A VELOCITY CHANGE ISN'T SPECIFIED (RTEDVD = 0), A V2(T1) IS COMPUTED WHICH M  
# AND CONSEQUENTLY FUEL.

#

# CALLING SEQUENCE

# L CALL

```

L+1 V2T100
#
NORMAL EXIT MODE
AT L+2 OF CALLING SEQUENCE WITH MPAC = 0
#
ALARM EXIT MODE
AT L+2 OF CALLING SEQUENCE WITH MPAC = OCTAL 605 FOR EXCESS ITERATIONS.
#
SUBROUTINES CALLED
GAMDV10
XT1LIM
DVCALC
#
ERASABLE INITIALIZATION REQUIRED
PUSHLIST
NONE
MPAC
NONE
OTHER
R(T1) MAGNITUDE OF INITIAL POSITION VECTOR DP
RCON MAGNITUDE OF FINAL POSITION VECTOR DP
V(T1)/ INITIAL VELOCITY VECTOR VECTOR
RTEDVD DELTA VELOCITY DESIRED DP
UR1/ UNIT INITIAL VECTOR VECTOR
UH/ UNIT HORIZONTAL VECTOR VECTOR
X(T2) COTANGENT OF FINAL FLIGHT PATH ANGLE DP
X(T1) COTANGENT OF INITIAL FLIGHT PATH ANGLE (INPUT FOR PREC) DP
CFPA COSINE OF INITIAL FLIGHT PATH ANGLE DP
MAMAX1 MAJOR AXIS LIMIT FOR LOWER BOUND ON GAMDV ITERATOR DP
MAMAX2 MAJOR AXIS LIMIT FOR UPPER BOUND ON GAMDV ITERATOR DP
PHI2 REENTRY NEAR PERIGEE OR APOGEE INDICATE (RTE ONLY) DP
N1 CONIC OR PRECISION ITERATION OPERATOR DP
#
OUTPUT
V2(T1)/ POST IMPULSE INITIAL VELOCITY VECTOR VECTOR
DV INITIAL VELOCITY CHANGE DP
X(T1) COTANGENT OF INITIAL FLIGHT PATH ANGLE (POST IMPULSE) DP
PCON SEMI-LATUS RECTUM DP
BETA1 1+X(T2)**2 DP
#
Page 918
#
DEBRIS
PUSHLIST
OOD X(T1),,=PREVIOUS PRECISION X(T1) DP
O2D THETA1=BETA5*LAMBDA-1 TP

```

July 12, 2016

1229

|          |       |                                                 |                                     |    |         |
|----------|-------|-------------------------------------------------|-------------------------------------|----|---------|
| #        | 05D   | THETA2=2*R(T1)*(LAMBDA-1)                       |                                     | TP | B38/B36 |
| #        | 08D   | THETA3=MU**0.5/R(T1)                            |                                     | DP | B-4/B-5 |
| #        | 10D   | X(T1)MIN=LOWER BOUND ON X(T1) IN GAMDV ITERATOR |                                     | DP | B5      |
| #        | 12D   | DX(T1)MAX=MAXIMUM DELTA X(T1)                   |                                     | DP | B5      |
| #        | 14D   | X(T1)MAX=UPPER BOUND ON X(T1) IN GAMDV ITERATOR |                                     | DP | B5      |
| #        | 16D   | DX(T1)=ITERATOR INCREMENT                       |                                     | DP | B5      |
| #        | 31D   | GAMDV10 SUBROUTINE RETURN ADDRESS               |                                     |    |         |
| #        | 32D   | DVCALC SUBROUTINE RETURN ADDRESS                |                                     |    |         |
| #        | 33D   | V2T100 SUBROUTINE RETURN ADDRESS                |                                     |    |         |
| V2T100   | STQ   | DLOAD                                           |                                     |    |         |
|          |       | 33D                                             |                                     |    |         |
|          |       | RCON                                            |                                     |    |         |
|          | BMN   | DSU                                             | # ABORT IF RCON NEGATIVE            |    |         |
|          |       | V2TERROR                                        |                                     |    |         |
|          |       | R(T1)                                           |                                     |    |         |
|          | BMN   |                                                 |                                     |    |         |
|          |       | V2T101                                          |                                     |    |         |
| V2TERROR | EXIT  |                                                 | # OR IF LAMBDA LESS THAN ONE        |    |         |
|          | TC    | P00D00                                          | # NO SOLUTION IF LAMBDA LESS THAN 1 |    |         |
|          | OCT   | 00610                                           |                                     |    |         |
| V2T101   | SETPD | CLEAR                                           |                                     |    |         |
|          |       | 0                                               | #                                   |    | PLOOD   |
|          |       | F2RTE                                           |                                     |    |         |
|          | DLOAD | NORM                                            |                                     |    |         |
|          |       | RCON                                            |                                     |    |         |
|          |       | X1                                              |                                     |    |         |
|          | PDDL  | NORM                                            |                                     |    |         |
|          |       | R(T1)                                           |                                     |    |         |
|          |       | S1                                              |                                     |    |         |
|          | STORE | 10D                                             |                                     |    |         |
|          | SR1   | DDV                                             | # R1/RCON = LAMBDA                  | B1 |         |
|          | XSU,1 | PDDL                                            | #                                   |    | PL02D   |
|          |       | S1                                              |                                     |    |         |
|          |       | X(T2)                                           |                                     |    |         |
|          | DSQ   |                                                 |                                     |    |         |
|          | SR1   | DAD                                             |                                     |    |         |
|          |       | 1RTEB1                                          |                                     |    |         |
|          | STORE | BETA1                                           | # 1+X(T2)**2 = BETA1                | B1 |         |
|          | DMP   |                                                 |                                     |    |         |
|          |       | 00D                                             |                                     |    |         |
|          | STORE | 28D                                             | # BETAI*LAMBDA = BETA5              |    |         |
|          | DMP   | SL*                                             |                                     |    |         |
|          |       | 00D                                             |                                     |    |         |
|          |       | 0 -7,1                                          |                                     |    |         |
|          | SL*   | DSU                                             |                                     |    |         |

# Page 919

```

 0 -7,1
 1RTEB17
RTB PDDL # BETA5*LAMBDA-1 = THETA1 B17 PL051
 TPMODE
 1RTEB1
SR* DCOMP
 0,1
DAD DMP
 OOD
 R(T1)
SL* RTB
 0 -7D,1
 TPMODE
PDDL # 2*R(T1)*(LAMBDA-1)=THETA2 B38/B36 PL081
 RTMURTE
NORM SR1
 X2
XSU,2 DDV
 S1
 10D
SR* PDDL # MU**.5/R(T1)=THETA3 B-4/B-5 PL101
 6,2
 MAMAX1
PUSH PUSH # MAMAX1=MA
CALL
 XT1LIM
DCOMP PUSH # X(T1)MIN B5 PL121
DCOMP SR4
PDDL PUSH # DX(T1)MAX B5 PL141
 MAMAX2
PUSH CALL
 XT1LIM
PDDL BMN # X(T1)MAX B5 PL161
 NN1A
 V2T102
GOTO V2T110

```

# PROCEED HERE IF NOT PRECISION COMPUTATION

```

V2T102 DLOAD
 RTEDVD
 GOTO
 V2T105
 V2T140

```

July 12, 2016

1231

V2T105           DLOAD   BMN  
                          CFPA  
                          V2T140  
                  GOTO  
                          V2T145

# Page 920

# DURING A PRECISION TRAJECTORY ITERATION CONSTRAIN THE INDEPENDENT

# VARIABLE TO INSURE THAT ALL CONICS PASS THROUGH RCON ON THE SAME PASS

# THROUGH X(T2)

V2T110           DLOAD   RTB  
                          1RTEB17  
                          TPMODE  
                  DCOMP   PDDL           # -1           B17       PL19D  
                          2RTEB1  
                  SR\*     DSU  
                          0,1  
                          OOD  
                  DMP     SL\*  
                          28D  
                          0 -7,1  
                  SL\*     TAD  
                          0 -7,1  
                  RTB     PDDL           # BETA5(2-LAMBDA)-1=BETA6       B17       PL19D  
                          TPMODE  
                          X(T1)  
                  STORE   OOD           # X(T1),,           B5  
                  TLOAD               #                       PL16D  
                  BMN     BZE  
                          V2T115  
                          V2T115  
                  SL     GOTO  
                          7  
                          V2T120  
V2T115           DLOAD   BMN  
                          PHI2  
                          V2T125  
                  DCOMP  
                  STODL   PHI2  
                          1ORTE  
                  STORE   NN1A  
                  GOTO  
                          V2T125  
V2T120           SQRT    RTB  
                          DPMODE  
                  PDDL    BMN           # BETA6\*\*0.5=X(T1)LIM       B5       PL18D

```

 PHI2
 V2T130
 DLOAD STADR
 STORE 14D # X(T1)LIM = X(T1)MAX
 DCOMP
 STORE 10D # -X(T1)LIM = X(T1)MIN
V2T125 DLOAD BZE
 X(T1)
 V2T140
 BMN GOTO
Page 921
 V2T140
 V2T145
V2T130 DLOAD BZE
 X(T1)
 V2T135
 BMN DLOAD #
 V2T135
 STADR
 STORE 10D # X(T1)LIM = X(T1)MIN
 GOTO
 V2T145
V2T135 DLOAD DCOMP #
 STADR
 STORE 14D # -X(T1)LIM = X(T1)MAX
V2T140 DLOAD
 10D
 STODL X(T1) # X(T1)MIN = X(T1)
 12D
 PUSH GOTO # DX(T1)MAX = DX(T1)
 V2T150
V2T145 DLOAD
 14D
 STODL X(T1) # X(T1)MAX = X(T1)
 12D
 DCOMP PUSH # -DX(T1)MAX = DX(T1)
V2T150 CALL # GOTO X(T1)-DV ITERATOR
 GAMDV10
 DLOAD BZE # EXIT IF MINIMUM FUEL MODE
 RTEDVD
 V2T1X

CONTINUE IF TIME CRITICAL MODE

 DSU BMN
 DV

```

PL161

PL161

PL181

PL181



```

 V2T155
 GOTO
V2T155 DLOAD BMN
 NN1A
 V2T160
 GOTO
 V2T185

CONIC TRAJECTORY COMPUTATION

V2T160 DLOAD BZE
 X(T1)
 V2T165
 BMN GOTO

Page 922

 V2T165
 V2T300
V2T165 DLOAD BZE
 CFPD
 V2T300
 BMN DLOAD
 V2T300
 14D
 STODL X(T1) # X(T1)MAX=X(T1)
 12D
 DCOMP
 STCALL 16D # -DX(T1)MAX=DX(T1)
 GAMDV10
 DLOAD DSU
 RTEDVD
 DV
 BMN
 V2T300
V2T175 SET DLOAD
 F2RTE
 X(T1)
 BOFF
 SLOWFLG
 V2T177
 STODL 10D # X(T1)MIN
 12D # DX(T1)MAX
 GOTO
 V2T179
V2T177 STODL 14D
 12D

```

```

DCOMP
V2T179 STCALL 16D # DX(T1)
 GAMDV10
 DLOAD BMN
 NN1A
 V2T300

PREVENT A LARGE CHANGE IN INDEPENDENT VARIABLE DURING AN ITERATION FOR A
PRECISION TRAJECTORY

V2T185 DLOAD DSU
 X(T1)
 OOD
 ABS PDDL # /X(T1)-X(T1),,/ = BETA7
 12D
 SL1 BDSU
 BMN DLOAD
 V2T300
 OOD # CONTINUE IF BETA7 LARGER THAN 2DX(T1)MAX
 STORE X(T1) # X(T1),, = X(T1)

Page 923
 DSU BMN
 14D
 V2T195
 DLOAD
 14D
 STORE X(T1) # X(T1)MAX = X(T1)
 GOTO
 V2T205
V2T195 DLOAD DSU
 X(T1)
 10D
 BMN GOTO
 V2T200
 V2T205
V2T200 DLOAD
 10D
 STORE X(T1) # X(T1)MIN = X(T1)
V2T205 CALL
 DVCALC
V2T300 DLOAD
 ZERORTE
V2T1X GOTO
 33D

Page 924

```

```

X(T1)-DV ITERATOR SUBROUTINE
#
DESCRIPTION
COMPUTES A POST IMPULSE VELOCITY VECTOR (V2(T1)) WHICH REQUIRES A MINIMUM DV.
#
CALLING SEQUENCE
L CALL
L+1 GAMDV10
#
NORMAL EXIT MODE
AT L+2 OF CALLING SEQUENCE
#
ALARM EXIT MODE
AT V2T1X WITH MPAC = OCTAL 605 FOR EXCESS ITERATIONS
#
SUBROUTINES CALLED
DVCALC
#
ERASABLE INITIALIZATION REQUIRED
PUSHLIST
02D THETA1=BETA5*LAMBDA-1 TP B17
05D THETA2=2*R(T1)*(LAMBDA-1) TP B38/B36
08D THETA3=MU**.5/R(T1) DP B-4/B-5
10D X(T1)MIN=LOWER BOUND ON INDEPENDENT VARIABLE X(T1) DP B5
12D DX(T1)MAX=MAXIMUM DX(T1) DP B5
14D X(T1)MAX=UPPER BOUND ON INDEPENDENT VARIABLE X(T1) DP B5
16D DX(T1)=ITERATOR INCREMENT DP B5
MPAC
NONE
OTHER
V(T1)/ INITIAL VELOCITY VECTOR VECTOR B7/B5
RTEDVD DELTA VELOCITY DESIRED DP B7/B5
UR1/ UNIT INITIAL VECTOR VECTOR B1
UH/ UNIT HORIZONTAL VECTOR VECTOR B1
X(T1) COTANGENT OF INITIAL FLIGHT PATH ANGLE (FROM VERTICAL) DP B5
F2RTE TIME CRITICAL OR MINIMUM FUEL MODE INDICATOR STATE AREA
#
OUTPUT
V2(T1)/ POST IMPULSE INITIAL VELOCITY VECTOR VECTOR B7/B5
DV INITIAL VELOCITY CHANGE DP B7/B5
X(T1) COTANGENT OF INITIAL FPA MEASURED FROM VERTICAL DP B5
PCON SEMI-LATUS RECTUM DP B28/B26
#
DEBRIS
PUSHLIST
00D X(T1),,

```

|         |          |                                   |                              |    |       |
|---------|----------|-----------------------------------|------------------------------|----|-------|
| #       | 02D      | THETA1                            |                              |    |       |
| #       | 05D      | THETA2                            |                              |    |       |
| #       | 08D      | THETA3                            |                              |    |       |
| #       | 10D      | X(T1)MIN                          |                              |    |       |
| #       | 12D      | DX(T1)MAX                         |                              |    |       |
| #       | Page 925 |                                   |                              |    |       |
| #       | 14D      | X(T1)MAX                          |                              |    |       |
| #       | 16D      | DX(T1)                            |                              |    |       |
| #       | 22D      | DV,=PREVIOUS DV                   |                              |    | DP    |
| #       | 24D      | BETA9=X(T1)+1.1DX(T1)             |                              |    | DP    |
| #       | 31D      | GAMDV10 SUBROUTINE RETURN ADDRESS |                              |    |       |
| #       | 32D      | DVCALC SUBROUTINE RETURN ADDRESS  |                              |    |       |
| #       | 33D      | V2T100 SUBROUTINE RETURN ADDRESS  |                              |    |       |
|         |          |                                   |                              |    |       |
| GAMDV10 | STQ      | 31D                               |                              |    |       |
|         | SETPD    | CALL                              |                              |    |       |
|         |          | 18D                               | #                            |    | PL18D |
|         |          | DVCALC                            |                              |    |       |
|         | DLOAD    | DSU                               |                              |    |       |
|         |          | 14D                               |                              |    |       |
|         |          | 10D                               |                              |    |       |
|         | BOV      |                                   |                              |    |       |
|         |          | GAMDV20                           |                              |    |       |
|         | PUSH     | DSU                               | # X(T1)MAX-X(T1)MIN=BETA8    | B5 | PL20D |
|         |          | EPC9RTE                           |                              |    |       |
|         | BMN      | DLOAD                             |                              |    |       |
|         |          | GAMDVX                            | # BOUNDS CLOSE TOGETHER      |    |       |
|         |          | 18D                               |                              |    |       |
|         | DSU      | BMN                               | # BETA8-DX(T1)MAX            |    |       |
|         |          | 12D                               |                              |    |       |
|         |          | GAMDV15                           |                              |    |       |
|         | SETPD    | GOTO                              | #                            |    | PL18D |
|         |          | 18D                               |                              |    |       |
|         |          | GAMDV20                           |                              |    |       |
| GAMDV15 | DLOAD    |                                   | #                            |    | PL18D |
|         | SIGN     | SR1                               |                              |    |       |
|         |          | 16D                               |                              |    |       |
|         | STORE    | 16D                               | # BETA8(SIGNDX(T1))/2=DX(T1) |    |       |
| GAMDV20 | DLOAD    |                                   |                              |    |       |
|         |          | M144RTE                           |                              |    |       |
|         | STORE    | NN2                               |                              |    |       |
| GAMDV25 | DLOAD    | DAD                               |                              |    |       |
|         |          | NN2                               |                              |    |       |
|         |          | 1RTEB28                           |                              |    |       |
|         | BMN      | SLOAD                             |                              |    |       |

July 12, 2016

1237

|                      |        |          |                                  |          |       |
|----------------------|--------|----------|----------------------------------|----------|-------|
|                      |        | GAMDV30  |                                  |          |       |
|                      |        | OCT605   |                                  |          |       |
|                      | GOTO   | V2T1X    |                                  |          |       |
| GAMDV30              | STORE  | NN2      | # NN2=NN2+1                      |          |       |
|                      | DLOAD  | PDDL     | # X(T1)=X(T1),                   | B5       | PL20D |
|                      |        | X(T1)    |                                  |          |       |
|                      |        | DV       |                                  |          |       |
|                      | PDDL   | DAD      | # DV=DV,                         | B7/B5    | PL22D |
|                      |        | X(T1)    |                                  |          |       |
|                      |        | 16D      |                                  |          |       |
| # Page 926           |        |          |                                  |          |       |
|                      | STCALL | X(T1)    | # X(T1)+DX(T1)=X(T1)             | B5       |       |
|                      |        | DVCALC   |                                  |          |       |
|                      | BON    | DLOAD    |                                  |          |       |
|                      |        | F2RTE    |                                  |          |       |
|                      |        | GAMDV35  |                                  |          |       |
|                      |        | DV       |                                  |          |       |
|                      | DSU    | BMN      | # CONTINUE IF FUEL CRITICAL MODE |          |       |
|                      |        | 20D      |                                  |          |       |
|                      |        | GAMDV33  |                                  |          |       |
| GAMDV32              | DLOAD  | DCOMP    |                                  |          |       |
|                      |        | 16D      |                                  |          |       |
|                      | SR1    |          |                                  |          |       |
|                      | STORE  | 16D      |                                  |          |       |
| GAMDV33              | SETPD  | GOTO     |                                  |          |       |
|                      |        | 18D      | #                                |          | PL18D |
|                      |        | GAMDV50  |                                  |          |       |
| # TIME CRITICAL MODE |        |          |                                  |          |       |
| GAMDV35              | DLOAD  | DSU      |                                  |          |       |
|                      |        | RTEDVD   |                                  |          |       |
|                      |        | DV       |                                  |          |       |
|                      | PDDL   | PUSH     | # DVD-DV=DVERR                   | B7/B5    | PL22D |
| GAMDV40              | DLOAD  | ABS      | # DV,                            |          | PL24D |
|                      |        | 20D      |                                  |          |       |
|                      | DSU    | BMN      |                                  |          |       |
|                      |        | EPC10RTE |                                  |          |       |
|                      |        | GAMDVX   |                                  |          |       |
| GAMDV45              | BOVB   | DLOAD    |                                  |          |       |
|                      |        | TCDANZIG | # ASSURE OVFind IS 0             |          |       |
|                      | BDSU   | NORM     |                                  |          |       |
|                      |        | DV       |                                  |          |       |
|                      |        | X2       |                                  |          |       |
|                      | PDDL   |          | # DV-DV,                         | B7/B5-N2 | PL22D |

July 12, 2016

PL18D

```

 10D
 X(T1)
 SR1
 STORE 16D # (X(T1)MIN-X(T1))/2=DX(T1) B5
GAMDV65 DLOAD ABS
 16D
 DSU BMN
 EPC9RTE
 GAMDVX
 GOTO
 GAMDV25
GAMDVX GOTO
 31D

```

# Page 928

# DV CALCULATION SUBROUTINE

#

# INPUT

# PUSHLIST

|   |     |                           |    |         |
|---|-----|---------------------------|----|---------|
| # | 02D | THETA1=BETA5*LAMBDA-1     | TP | B17     |
| # | 05D | THETA2=2*R(T1)*(LAMBDA-1) | TP | B38/B36 |
| # | 08D | THETA3=MU** .5/R(T1)      | DP | B-4/B-5 |

# OTHER

|   |        |                                                     |        |       |
|---|--------|-----------------------------------------------------|--------|-------|
| # | X(T1)  | COTANGENT OF POST IMPULSE INITIAL FLIGHT PATH ANGLE | DP     | B5    |
| # | V(T1)/ | INITIAL VELOCITY VECTOR (PRE IMPULSE)               | VECTOR | B7/B5 |
| # | UR1/   | UNIT INITIAL VECTOR                                 | VECTOR | B1    |
| # | UH/    | UNIT HORIZONTAL VECTOR                              | VECTOR | B1    |

#

# OUTPUT

|   |         |                                      |        |         |
|---|---------|--------------------------------------|--------|---------|
| # | V2(T1)/ | POST IMPULSE INITIAL VELOCITY VECTOR | VECTOR | B7/B5   |
| # | DV      | INITIAL VELOCITY CHANGE              | DP     | B7/B5   |
| # | PCON    | SEMI-LATUS RECTUM                    | DP     | B28/B26 |

#

# DEBRIS

|   |            |                                       |        |         |
|---|------------|---------------------------------------|--------|---------|
| # | 28D        | THETA3*PCON** .5                      | DP     | B10/B8- |
| # | C(PUSHLOC) | THETA3(PCON** .5)*X(T1)*UR1/          | VECTOR | B7/B5   |
| # | 32D        | DVCALC SUBROUTINE RETURN ADDRESS      |        |         |
| # | X1         | NORMALIZATION FACTOR FOR VALUE IN 28D |        |         |

#

# PUSHLOC IS RESTORED TO ITS ENTRANCE VALUE UPON EXITING DVCALC

```

DVCALC STQ DLOAD
 32D
 X(T1)
 DSQ SR
 7

```

```

DCOMP TAD
 02D
NORM PUSH
 X1
TLOAD NORM
 05D
 X2
RTB SR1
 DPMODE
XSU,2 DDV
 X1
SR*
 6,2
STORE PCON # THETA2/(THETA1-X(T1)**2)=PCON B28/26
SQRT DMP
 08D
NORM
 X1
STODL 28D # THETA3*PCON**.5 B10/B8 -N1
Page 929
 X(T1)
NORM VXSC
 X2
 UR1/ # X(T1)*UR1/ B5+B1 -N2
XAD,2 VXSC
 X1
 28D
VSR* PDVL # THETA3(PCON**.5)X(T1)*UR1/ B7/B5
 0 -9D,2 #
 UH/
VXSC VSR* # THETA3(PCON**.5)UH/ B7/B5
 28D
 0 -4,1 #
VAD STADR
STORE V2(T1)/ # V2(T1)/ B7/B5
VSU ABVAL
 V(T1)/
STORE DV # ABVAL(V2(T1)/-V1(T)/)=DV B7/B5
GOTO 32D

```

# Page 930

# SUBROUTINE TO COMPUTE BOUNDS ON INDEPENDENT VARIABLE X(T1)

#

# INPUT

#     PUSHLIST



July 12, 2016

1241

```

PUSHLOC -4 MAJOR AXIS (MA) DP B30/B28
PUSHLOC -2 MAJOR AXIS (MA) AGAIN DP B30/B28
28D BETA5=LAMBDA*BETA1 DP B9
OTHER
RCON DP B29/B27
R(T1) DP B29/B27
#
OUTPUT
MPAC
X(T1)LIM LIMIT ON INDEPENDENT VARIABLE X(T1) DP B5
#
DEBRIS
PUSHLIST
C(PUSHLOC) MA-RCON DP (B30/28
C(PUSHLOC) +2 MA DP B30/B28
X1 NORMALIZATION FACTOR FOR MA-RCON
20D XT1LIM SUBROUTINE RETURN ADDRESS
#
PUSHLOC IS RESTORED TO ITS ENTRANCE VALUE UPON EXITING XT1LIM

XT1LIM STQ DLOAD
 20D
 RCON
 SR1 BDSU
 NORM PDDL # MA-RCON B30-N1
 X2
 PDDL SR1
 R(T1)
 BDSU DDV
 SL* DMP
 0 -3,2
 28D
 SL* DSU # BETA10=BETA5(MA-RT)/(MA-RC)-1 B11
 0 -6,1
 1RTEB25 +1 # 1.0 B-11
 SL1 BOV
 XT1LIM2
 BMN GOTO
 XT1LIM5
 XT1LIM3
XT1LIM2 DLOAD # BETA10=POSMAX IF OVERFLOW
 2RTEB1
XT1LIM3 SQRT # X(T1)=SQRT(BETA10)
 XT1LIMX
XT1LIM5 DLOAD
 ZERORTE

```

XT1LIMX           GOTO  
                    20D

# Page 931

# CONSTANTS FOR THE P37 AND P70 PROGRAMS AND SUBROUTINES

|            |        |                |                                         |
|------------|--------|----------------|-----------------------------------------|
|            | BANK   | 36             |                                         |
|            | SETLOC | RTECON1        |                                         |
|            | BANK   |                |                                         |
| 1RTEB1     | 2DEC   | 1. B-1         |                                         |
| 1RTEB2     | 2DEC   | 1. B-2         |                                         |
| 1RTEB3     | 2DEC   | 1. B-3         |                                         |
| 1RTEB4     | 2DEC   | 1. B-4         |                                         |
| 1RTEB10    | 2DEC   | 1. B-10        |                                         |
| 1RTEB12    | 2DEC   | 1. B-12        |                                         |
| 1RTEB13    | 2DEC   | 1. B-13        |                                         |
| 1RTEB17    | 2DEC   | 1. B-17        |                                         |
| 1RTEB25    | 2DEC   | 1. B-25        |                                         |
| #          |        |                | * * B25 AND B28 MUST BE CONSECUTIVE * * |
| 1RTEB28    | 2DEC   | 1. B-28        |                                         |
| ZERORTE    | 2DEC   | 0              |                                         |
| M144RTE    | 2DEC   | -144. B-28     |                                         |
| M15RTE     | 2DEC   | -15            |                                         |
| 10RTE      | 2DEC   | 10             |                                         |
| M.6RTE     | 2DEC   | -.6            |                                         |
| 1.1RTEB1   | 2DEC   | 1.1 B-1        |                                         |
| M6RTEB28   | 2DEC   | -6             |                                         |
| 2RTEB1     | 2OCT   | 3777737777     |                                         |
| M9RTEB28   | 2DEC   | -9             |                                         |
| M8RTEB28   | 2DEC   | -8             |                                         |
| 3048ORTE   | 2DEC   | 30480. B-29    |                                         |
| VCSPS      | 2DEC   | 31.510396 B-5  | # (SEE 2VEXHUST)                        |
| # Page 932 |        |                |                                         |
| VCRCS      | 2DEC   | 27.0664 B-5    |                                         |
| MDOTRCS    | 2DEC   | .0016375 B-3   |                                         |
| CSUBT      | 2DEC   | .5             |                                         |
| OCT605     | OCT    | 00605          |                                         |
| OCT612     | OCT    | 00612          |                                         |
| MCOS7.5    | 2DEC   | -.99144486     |                                         |
| MSIN7.5    | 2DEC   | -.13052619     |                                         |
| MCOS22.5   | 2DEC   | -.92387953 B-2 |                                         |
| THETA165   | 2DEC   | .4583333333    |                                         |
| THETA210   | 2DEC   | .5833333333    |                                         |
| EPC1RTE    | 2DEC   | .99966 B-1     |                                         |
| EPC2RTE    | 2DEC   | 100. B-29      |                                         |

July 12, 2016

1243

|            |         |                   |
|------------|---------|-------------------|
| EPC3RTE    | 2DEC    | .001              |
| EPC4RTE    | 2DEC    | .00001            |
| EPC5RTE    | 2DEC    | .01 B-6           |
| EPC6RTE    | 2DEC    | .000007 B-1       |
| EPC7RTE    | 2DEC    | 1000. B-29        |
| EPC9RTE    | 2DEC    | 1. B-25           |
| EPC10RTE   | 2DEC    | .0001 B-7         |
|            |         |                   |
| BANK       | 35      |                   |
| SETLOC     | RTECON1 |                   |
| BANK       |         |                   |
|            |         |                   |
| C4RTE      | 2DEC    | -6.986643 E7 B-30 |
| K1RTE      | 2DEC    | 7. E6 B-29        |
| K2RTE      | 2DEC    | 6495000. B-29     |
| K3RTE      | 2DEC    | -.06105           |
| K4RTE      | 2DEC    | -.10453           |
| RTMURTE    | 2DEC    | 199650.501 B-18   |
| # Page 933 |         |                   |
| E3RTE      | 2DEC    | 121920. B-29      |

This code is written to file `src/P37-P70.s`.

## B.74 P40-P47

```

1244 <src/P40-P47.s 1244>≡
 # Copyright: Public domain.
 # Filename: P40-P47.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 684-736
 # Mod history: 2009-05-11 RSB Adapted from the Colossus249/ file
 # of the same name, using Comanche055 page
 # images.
 # 2009-05-20 RSB In S20.1, a DMP DDV was corrected to DMPR DDV.
 # 2009-05-22 RSB In BESTTRIM, TC PACTOFF corrected to
 # TS PACTOFF.
 # 2009-05-23 RSB Prior to the 2CADR at T5IDLDAP, added an
 # SBANK.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 684
 # PROGRAM DESCRIPTION ** P40CSM **

 EBANK= DAPDATR1
 BANK 31
 SETLOC P40S
 BANK

```

July 12, 2016

1245

|            |        |             |                                           |
|------------|--------|-------------|-------------------------------------------|
|            | COUNT  | 24/P40      |                                           |
| P40CSM     | TC     | DOWNFLAG    |                                           |
|            | ADRES  | ENG2FLAG    |                                           |
|            | TC     | INTPRET     |                                           |
|            | SLOAD  | BOFF        |                                           |
|            |        | ECSTEER     | # IS THIS AN EXTERNAL DELTA V BURN        |
|            |        | XDELVFLG    |                                           |
|            |        | P40S/C      | # NO CSTEER = ECSTEER                     |
|            | DLOAD  |             | # YES CSTEER = ZERO                       |
|            |        | HI6ZEROS    |                                           |
| P40S/C     | STODL  | CSTEER      |                                           |
|            |        | FENG        | # SET UP THRUST FOR P40 20,000 LBS        |
| P40S/F     | STODL  | F           | # P41 ENTERS HERE                         |
|            |        | TIG         | # ORIGINAL TIG MAY BE SLIPPED BY P40S/SV  |
|            | STORE  | NOMTIG      | # SET ORIGINAL TIME OF IGNITION FOR S40.9 |
|            | EXIT   |             |                                           |
|            | TC     | BANKCALL    |                                           |
|            | CADR   | R02BOTH     | # IMU STATUS CHECK                        |
| P40PVA     | TC     | INTPRET     |                                           |
|            | CALL   |             |                                           |
|            |        | S40.1       | # COMPUTE VGTIG,UT                        |
|            | CALL   |             |                                           |
|            |        | S40.2,3     | # COMPUTE PREFERRED ATTITUDE              |
|            | SET    | EXIT        |                                           |
|            |        | PFRATFLG    |                                           |
| P40SXTY    | TCR    | SETMINDB -1 | # NARROW DEADBAND FOR MANEUVER (EBANK6)   |
|            | RELINT |             |                                           |
|            | TC     | BANKCALL    |                                           |
|            | CADR   | R60CSM      | # ATTITUDE MANEUVER                       |
|            | CS     | ONE         | # FOR UPDATEVG                            |
|            | TS     | NBRCYCLS    |                                           |
|            | TC     | UPFLAG      |                                           |
|            | ADRES  | TIMRFLAG    | # ALLOW CLOCTASK                          |
|            | TC     | P41/P40     |                                           |
|            | TC     | P41/DSP     | # P41                                     |
| P40TTOG    | CAF    | V06N40      | # INITIALIZE FOR CLOCTASK WHICH IS CALLED |
| # Page 685 | TS     | NVWORD1     | # BELOW                                   |
|            | TC     | INTPRET     |                                           |

|          |        |          |                                             |
|----------|--------|----------|---------------------------------------------|
|          | VLOAD  | ABVAL    | # FOR R2                                    |
|          |        | VGTIG    |                                             |
|          | STODL  | VGDISP   |                                             |
|          |        | HI6ZEROS |                                             |
|          | STORE  | DVTOTAL  |                                             |
|          | EXIT   |          |                                             |
|          | EXTEND |          |                                             |
|          | DCA    | STEERADS | # SET FOR UPDATEVG AND TEST FOR STEERING    |
|          | DXCH   | AVEGEXIT | # AFTER AVERAGE G                           |
| P40GMB   | CAF    | P40CKLS2 | # (4.1 PROTECTION)                          |
|          | TC     | BANKCALL |                                             |
|          | CADR   | GOPERF1  |                                             |
|          | TCF    | POST41   | # V34                                       |
|          | TCF    | TST,TRIM | # V33                                       |
| TRIMONLY | CS     | BIT1     | # SET MRKRTEMP FOR GIMBAL TRIM (-1)         |
| +1       | TS     | MRKRTEMP | # ENTRY FROM TST,TRIM                       |
|          | CAF    | ZERO     | # SET CNTR +0 FOR RESTART LOGIC IN S40.6    |
|          | TS     | CNTR     | # +0 SAYS NORMAL ENTRY                      |
|          |        |          | # +1 (PRE40.6) SAYS RESTART ENTRY           |
|          | CAF    | ONE      |                                             |
|          | TC     | WAITLIST |                                             |
|          | EBANK= | DAPDATR1 |                                             |
|          | 2CADR  | S40.6    |                                             |
|          | CCS    | MRKRTEMP | # TEST TO FIND TIME TO WAIT FOR GIMBAL TEST |
|          | CAF    | 18SEC    | # PLUS, DELAY FOR 18 SECONDS                |
|          | TCF    | +2       | # HOLE                                      |
|          | CAF    | 5SEC     | # DELAY FOR TRIM ONLY TASK                  |
|          | TC     | BANKCALL |                                             |
|          | CADR   | DELAYJOB |                                             |
|          | TC     | 2PHSCHNG |                                             |
|          | OCT    | 40026    | # 6.2 = PRE40.6(-OCS), CLOKTASK(100CS)      |
|          | OCT    | 00234    | # 4.23 = P40S/SV (PRIO12)                   |
| P40S/RS  | CAF    | ONE      |                                             |
|          | TC     | WAITLIST | # P41/SDP                                   |
|          | EBANK= | TIG      |                                             |
|          | 2CADR  | CLOKTASK |                                             |
|          | RELINT |          |                                             |
| P40S/SV  | TCR    | E7SETTER | # JOB, 4.23 PRETECTS, PRE012                |
|          | EBANK= | TIG      |                                             |

July 12, 2016

1247

# Page 686

|          |        |          |                                         |
|----------|--------|----------|-----------------------------------------|
|          | TC     | INTPRET  |                                         |
|          | DLOAD  | DSU      |                                         |
|          |        | TIG      |                                         |
|          |        | SEC29.96 |                                         |
|          | STORE  | TDEC1    |                                         |
|          | CALRB  |          | # RETURN IN BASIC                       |
|          |        | MIDTOAV1 |                                         |
|          | TCF    | +2       |                                         |
|          | TC     | P40SNEWM | # INTEGRATION TIME GREATER THAN ALLOWED |
| P40SET   | EXTEND |          |                                         |
|          | DCA    | MPAC     | # DELTA TIME TO PREREAD (INT.INIT.)     |
|          | DXCH   | P40TMP   |                                         |
|          | EXTEND |          |                                         |
|          | DCS    | 5SECDP   | # FOR TIGBLNK                           |
|          | DAS    | P40TMP   |                                         |
|          | EXTEND |          |                                         |
|          | DCA    | P40TMP   |                                         |
|          | TC     | LONGCALL |                                         |
|          | EBANK= | TIG      |                                         |
|          | 2CADR  | TIGBLNK  |                                         |
|          | TC     | PHASCHNG |                                         |
|          | OCT    | 20214    | # 4.21 = TIGBLNK (P40TMP CS)            |
|          | TCF    | ENDOFJOB |                                         |
| P40BLNKR | TC     | BANKCALL |                                         |
|          | CADR   | CLEANDSP | # REMOVE RESIDUE                        |
|          | TCF    | ENDOFJOB |                                         |
|          | EBANK= | TIG      |                                         |
| P40SNEWM | EXTEND |          |                                         |
|          | DCA    | PIPTIME1 |                                         |
|          | DXCH   | TIG      | # SET NEW TIG FOR 06 40                 |
|          | EXTEND |          |                                         |
|          | DCA    | SEC29.96 |                                         |
|          | DAS    | TIG      |                                         |
|          | TCF    | P40SET   | # FOR LONGCALL OF TIG-30 (OR -35)       |
|          | EBANK= | DAPDATR1 |                                         |
| POSTBURN | CAF    | V16N40   |                                         |
|          | TC     | BANKCALL |                                         |
|          | CADR   | REFLASH  |                                         |
|          | TCF    | POST41   | # V34 GO FINISH                         |
|          | TCF    | P40RCS   | # PROCEED                               |
|          | TCF    | POSTBURN | # RECYCLE                               |

1248

July 12, 2016

```

P4ORCS EXTEND # V99N40 ENTERS HERE ON A P40 BYPASS SPS
 DCA ACADN85
 DXCH AVEGEXIT
 CAF 2SECS # WAIT FOR CALCN85 VIA AVEGEXIT
 TC BANKCALL

Page 687

 CADR DELAYJOB

P40MINDB TCR SETMINDB -1
 RELINT

TIGNOW TC PHASCHNG
 OCT 05024 # TYPE C GROUP 4 BELOW FOR NOUN 85
 OCT 20000 # PRIO 20
 CAF V16N85B
 TC BANKCALL
 CADR REFLASH
 TCF POST41 # FINISH P40/P41
 TCF POST41 # V03 PROCEED WITH REST OF THE CLEAN-UP
 TCF TIGNOW # V32 NOT GSOP RESPONSE BUT REDISPLAY N85

POST41 EXTEND
 DCA SERVCADR
 DXCH AVEGEXIT
 TCF GOTOPOOH

MINDB DEC 46
MAXDB DEC 455
 EBANK= DAPDATR1
 -1 INHINT
SETMINDB CA CDUX # ROUTINE FOR SETTING
 TS THETADX # THE MINIMUM DEADBAND
 EXTEND # IN AUTOPILOT
 DCA CDUY
 DXCH THETADY
 CA MINDB # SHOULD BE CALLED UNDER
 TS ADB # INTERRUPT INHIBITED
 CS BIT4 # EBANK = E6
 MASK DAPDATR1
 TS DAPDATR1
 TC Q

 EBANK= DAPDATR1
 -1 INHINT
SETMAXDB CA MAXDB # ROUTINE FOR SETTING
 TS ADB # THE MAXIMUM DEADBAND IN AUTOPILOT
 CS DAPDATR1

```



July 12, 2016

1249

|      |          |                          |
|------|----------|--------------------------|
| MASK | BIT4     | # SHOULD BE CALLED UNDER |
| ADS  | DAPDATR1 | # INTERRUPT INHIBITED    |
| TC   | Q        | # EBANK = E6             |

# Page 688

# PROGRAM DESCRIPTION \*\* P41CSM \*\*

|         |        |          |                              |
|---------|--------|----------|------------------------------|
|         | SETLOC | P40S2    |                              |
|         | BANK   |          |                              |
|         | EBANK= | DAPDATR1 |                              |
|         | COUNT  | 24/P41   |                              |
| P41CSM  | TC     | UPFLAG   |                              |
|         | ADRES  | ENG2FLAG | # SET FOR RCS                |
|         | TC     | INTPRET  |                              |
|         | DLOAD  |          |                              |
|         |        | HI6ZEROS | # FOR P41 CSTEER =0          |
|         | STORE  | CSTEER   |                              |
|         | DLOAD  | BON      |                              |
|         |        | FRCS2    | # 2JET THRUST FOR S40.1      |
|         |        | NJETSFLG |                              |
|         |        | P40S/F   | # NJETS = 1 2-JET            |
|         | DAD    | GOTO     | # NJETS = 0 4-JET            |
|         |        | FRCS2    |                              |
|         |        | P40S/F   |                              |
|         | SETLOC | P40S     |                              |
|         | BANK   |          |                              |
| P41/P40 | CS     | MODREG   |                              |
|         | MASK   | ONE      | # P41EXITS AT CALL LOC +1    |
|         | EXTEND |          |                              |
|         | BZF    | +2       | # P41                        |
|         | INCR   | Q        | # P40 EXITS AT CALL LOC +2   |
|         | TC     | Q        |                              |
| TTG/0   | CAF    | PRI020   | # TASK (4.4 PROTECTS IN P41) |
|         | TC     | NOVAC    |                              |
|         | EBANK= | DAPDATR1 |                              |
|         | 2CADR  | TIGNOW   |                              |
| P40CLK  | TC     | DOWNFLAG |                              |
|         | ADRES  | TIMRFLAG |                              |

|            |           |                    |                                            |
|------------|-----------|--------------------|--------------------------------------------|
|            | TCF       | TASKOVER           |                                            |
| P41/DSP    | CAF<br>TS | V06N85B<br>NVWORD1 | # SET UP FOR NONFLASH V 06 N85 BY CLOCKJOB |
| # Page 689 | TC        | INTPRET            |                                            |
|            | CALL      |                    | # COMPUTE                                  |
|            |           | P40CNV85           | # VGTIG IN CTRL COORDS                     |
|            | EXIT      |                    |                                            |
|            | EXTEND    |                    | # DO CONTROL COORD CALCULATION AFTER AVEG  |
|            | DCA       | ACADN85            |                                            |
|            | DXCH      | AVEGEXIT           |                                            |
|            | TC        | 2PHSCHNG           |                                            |
|            | OCT       | 40036              | # 6.3=CLOKTASK(100CS)                      |
|            | OCT       | 234                | # 4.23=P40S/SV(PRI012)                     |
| P41REDSP   | TCF       | P40S/RS            |                                            |
|            | CAF       | V16N85B            | # ENTER FROM P41 SIDE OF TIGAVEG           |
|            | TS        | NVWORD1            | # REDISPLAY NONFLASHING                    |
|            | CAF       | SEC29.96 +1        |                                            |
|            | TC        | WAITLIST           |                                            |
|            | EBANK=    | DAPDATR1           |                                            |
|            | 2CADR     | TTG/0              |                                            |
| P40CNV85   | CS        | BIT3               |                                            |
|            | TCF       | TTGPHS             |                                            |
|            | STQ       | SETPD              |                                            |
|            |           | QTEMP1             |                                            |
|            |           | 0                  |                                            |
|            | VLOAD     | PUSH               |                                            |
|            |           | VGPREV             | # EQUALS VGTIG (TARGETTING INPUT)          |
|            | CALL      |                    |                                            |
|            |           | S41.1              |                                            |
|            | STCALL    | VGBODY             |                                            |
|            |           | QTEMP1             |                                            |
| CALCN85    | EBANK=    | DAPDATR1           |                                            |
|            | TC        | INTPRET            |                                            |
|            | CALL      |                    |                                            |
|            |           | UPDATEVG           | # NEW VG, S40.8 (+MAYBE S40.9)             |
|            | CALL      |                    |                                            |
|            |           | P40CNV85           | # COMPUTE VGBODY                           |
|            | EXIT      |                    |                                            |
|            | TC        | SERVXT             |                                            |

July 12, 2016

1251

|            |        |                 |                                            |
|------------|--------|-----------------|--------------------------------------------|
| FENG       | 2DEC   | 9.1188544 B-7   | # SPS THRUST (20500LBS), SC.AT B+7 NEWT/E4 |
| FRCS2      | 2DEC   | .087437837 B-7  | # RCS ULLAGE (199.6COS10 LBS), SC.AT       |
|            |        |                 | # B+7 NEWTONS/E+4                          |
| SEC24.96   | DEC    | 2496            |                                            |
| SEC29.96   | 2DEC   | 2996            |                                            |
| 18SEC      | DEC    | 1800            |                                            |
| P40CKLS2   | OCT    | 204             |                                            |
| 40CST5     | OCT    | 37730           | # 40 CS FOR THE T5 CLOCK                   |
| OCT12      | =      | TEN             |                                            |
| # Page 690 |        |                 |                                            |
| V1683      | VN     | 1683            |                                            |
| V06N85B    | VN     | 0685            |                                            |
| V16N85B    | VN     | 1685            |                                            |
| V06N40     | VN     | 0640            |                                            |
| V16N40     | VN     | 1640            |                                            |
| OCT27/24   | OCT    | 27              |                                            |
| OCT53      | OCT    | 53              |                                            |
| OCT35      | OCT    | 35              |                                            |
|            | EBANK= | DAPDATR1        |                                            |
| T5IDL24    | 2CADR  | T5IDLOC         |                                            |
| 3MDOT      | DEC    | 86.6175796 B-16 | # 3SEC MASS LOSS (63.8 LBS/SEC), SC.AT     |
|            |        |                 | # B+16 KB/SEC (NOT, EMDOT IS PAD-LOADED,   |
|            |        |                 | # BUT 3MDOT IS NOT A CRITICAL QUANTITY, SO |
|            |        |                 | # IT CAN REMAIN IN FIXED MEMORY)           |
| TST,TRIM   | CAF    | BIT1            | # SET UP FOR GIMB DRIVE TEST AND TRIM (+1) |
|            | TCF    | TRIMONLY +1     |                                            |
| TIGBLNK    | CAF    | 5SEC            | # CALL TIGAVEG IN FIVE SEC AT TIG-30       |
|            | TC     | WAITLIST        |                                            |
|            | EBANK= | TIG             |                                            |
|            | 2CADR  | TIGAVEG         |                                            |
|            | CAF    | ZERO            | # DISABLE HERE, NOT IN P40BLNKR            |
|            | TS     | NVWORD1         |                                            |
|            | CAF    | PRI014          |                                            |
|            | TC     | NOVAC           |                                            |
|            | EBANK= | TIG             |                                            |
|            | 2CADR  | P40BLNKR        | # DON'T PROTECT -- RESTARTS BLANK DSKY     |
|            | CS     | OCT37           | # 4.37 = TIGAVEG (500CS)                   |
| P40TSK     | TC     | NEWPHASE        |                                            |
|            | OCT    | 4               |                                            |

|                                    |        |          |                                           |
|------------------------------------|--------|----------|-------------------------------------------|
|                                    | TC     | TASKOVER |                                           |
| ACADN83                            | EBANK= | TIG      |                                           |
|                                    | 2CADR  | CALCN83  |                                           |
| SERVCADR                           | EBANK= | TIG      |                                           |
|                                    | 2CADR  | SERVEXIT |                                           |
| ACADN85                            | EBANK= | DAPDATR1 |                                           |
|                                    | 2CADR  | CALCN85  |                                           |
| # Page 691                         |        |          |                                           |
| # PROGRAM DESCRIPTION ** P47CSM ** |        |          |                                           |
|                                    | COUNT  | 24/P47   |                                           |
| P47CSM                             | EBANK= | TIG      |                                           |
|                                    | TC     | BANKCALL | # IMU STATUS CHECK                        |
|                                    | CADR   | R02BOTH  |                                           |
|                                    | TC     | INTPRET  |                                           |
|                                    | CALRB  |          |                                           |
|                                    |        | MIDTOAV2 |                                           |
|                                    | CA     | MPAC +1  | # DELTA TIME TO RPEREAD (LESS THAN 100    |
|                                    | TS     | P40TMP   | # CS, WITH A TPAGREE, INT.INIT.)          |
|                                    | TC     | WAITLIST |                                           |
|                                    | EBANK= | TIG      |                                           |
|                                    | 2CADR  | TIGON    | # TIGON IS REQUIRED TO MATHCHTAT AND AVEG |
|                                    | TC     | PHASCHNG |                                           |
|                                    | OCT    | 40574    | # A, 4.57 = TIGON (P40TMP CS)             |
|                                    | TCF    | ENDOFJOB |                                           |
| TIGON                              | EBANK= | P40TMP   |                                           |
|                                    | EXTEND |          |                                           |
|                                    | DCA    | ACADN83  |                                           |
|                                    | DXCH   | AVEGEXIT |                                           |
|                                    | CAF    | PRI030   | # FORCE ZEROING OF N83 BEFORE SERVICER    |
|                                    | TC     | NOVAC    |                                           |
|                                    | EBANK= | TIG      |                                           |
|                                    | 2CADR  | P47BODY  |                                           |
|                                    | CS     | BIT2     | # 4.2 = PRECHECK (-OCS), P47BODY (PRI030) |
|                                    | TCF    | TTGPHS   |                                           |
| CALCN83                            | EBANK= | TIG      |                                           |
|                                    | TC     | INTPRET  |                                           |

July 12, 2016

1253

```

 SETPD # SET UP PUSHLIST FOR S41.1
 0
 VLOAD VAD
 DELVCTL
 DELVREF
 STORE DV47TEMP # FOR COPYCYCLE BELOW
 PUSH CALL
 S41.1
 STCALL DELVIMU
 S11.1 # CALC. VI, H, HDOT FOR NOUN 62
 EXIT
 TC PHASCHNG
 OCT 10035
Page 692
 CAF FIVE
 TC GENTRAN
 ADRES DV47TEMP
 ADRES DELVCTL
 TC SERVXT
P47BODY TC INTPRET
 VLOAD
 HI6ZEROS
 STORE DELVIMU # CLEAR DISPLAY AND ACCUMULATOR STORAGE
 STORE DELVCTL # UPON INITIATION OR ENTER RESPONSE
 EXIT
P47BOD CAF PRI015 # LOWER PRIO THAN CALCN83 (20)
 TC PRI0CHNG # TO PREVENT INTERRUPTION OF CALCN83
 TC PHASCHNG
 OCT 05024 # TYPE C GROUP 4 BELOW FOR NOUN 83
 OCT 15000 # PRIO 15
P47/DSP CAF V1683
 TC BANKCALL
 CADR GOFLASH
 TC GOTOP00H
 TC GOTOP00H
 TCF P47BODY # RECYCLE -- CLEAR ACCUMULATED VELOCITY

Page 693
ROUTINE ** TIG-30 ** DESCRIPTION

 EBANK= TIG
 COUNT 24/P40

TIGAVEG TC P41/P40 # TASK (4.37 PROTECTS)
 TCF P41REDSP
```

|          |        |          |                                           |
|----------|--------|----------|-------------------------------------------|
|          | CAF    | V06N40   | # UNBLANK DISPLAY                         |
|          | TS     | NVWORD1  |                                           |
|          | CAF    | SEC24.96 |                                           |
|          | TC     | WAITLIST |                                           |
|          | EBANK= | TIG      |                                           |
|          | 2CADR  | TIG-5    |                                           |
| TTGPHS   | CS     | SIX      | # 4.6 = TIG-5 (2496CS), PRECHECK (-OCS)   |
|          | TC     | NEWPHASE | # ENTRY FROM P41REDSP (P41) WITH A=-4, OR |
|          | OCT    | 4        | # FROM TIGON (P47) WITH A=-1              |
| PRECHECK | CCS    | PHASE5   | # HAS SERVICER BEEN RESTARTED             |
|          | TCF    | TASKOVER | # YES, DON'T START ANOTHER ONE            |
|          | TC     | POSTJUMP |                                           |
|          | CADR   | PREREAD  |                                           |

# Page 694

# ROUTINE \*\* TIG-5 \*\* DESCRIPTION

|       |        |          |                            |
|-------|--------|----------|----------------------------|
| TIG-5 | EBANK= | TIG      |                            |
|       | CAF    | 5SEC     |                            |
|       | TC     | WAITLIST |                            |
|       | EBANK= | DAPDATR1 |                            |
|       | 2CADR  | TIG-0    |                            |
|       | CS     | BIT9     | # WILL CAUSE V99 FLASH     |
|       | TS     | NVWORD1  |                            |
|       | TC     | 2PHSCHNG |                            |
|       | OCT    | 40074    | # A, 4.7 = TIG-0 (500CS)   |
|       | OCT    | 00033    | # A, 3.3 = S40.13 (PRI020) |
|       | CAF    | PRI020   |                            |
|       | TC     | FINDVAC  |                            |
|       | EBANK= | TGO      |                            |
|       | 2CADR  | S40.13   |                            |
|       | TCF    | TASKOVER |                            |

# Page 695

# ROUTINES \*\* TIG-0 \*\* AND \*\* IGNITION \*\* DESCRIPTION

|       |        |          |                                          |
|-------|--------|----------|------------------------------------------|
| TIG-0 | EBANK= | DAPDATR1 | # TASK, 4.7 PHASE, OR 4.77 (-OCS) IN R40 |
|       | CS     | FLAGWRD7 | # SET IGN FLAG                           |

July 12, 2016

1255

|          |        |          |                                            |
|----------|--------|----------|--------------------------------------------|
|          | MASK   | BIT13    |                                            |
|          | ADS    | FLAGWRD7 |                                            |
|          | CAE    | FLAGWRD7 | # CHECK ASTN FLAG FOR V99 RESPONSE         |
|          | MASK   | BIT12    |                                            |
|          | EXTEND |          |                                            |
|          | BZF    | TASKOVER | # WAIT FOR V99P                            |
|          | CAF    | V06N40   | # CLEAR THE V99 (IN CASE OF A RESTART      |
|          | TS     | NVWORD1  | # DURING THE V99 SEQUENCE)                 |
|          | TC     | PHASCHNG | # V99P HAS COME ALREADY, DO IGNITION NOW   |
|          | OCT    | 00614    | # A, 4.61 = IGNITION (-OCS) TBASE OLD      |
| IGNITION | CAE    | CDUX     | # SAVE FOR ROLL DAP REFERENCE OGAD         |
|          | TS     | OGAD     | # V99PJOB (CLOCKJOB) SETS UP IGNITION      |
|          | EXTEND |          | # TASK (4.61 PROTECTION)                   |
|          | DCA    | TIME2    | # FOR RESTARTS                             |
|          | DXCH   | TEVENT   |                                            |
|          | CS     | FLAGWRD5 | # SET ENGONFLG                             |
|          | MASK   | BIT7     |                                            |
|          | ADS    | FLAGWRD5 |                                            |
| SPSON    | CAF    | BIT13    | # TURN ON SPS ENGINE                       |
|          | EXTEND |          |                                            |
|          | WOR    | DSALMOUT |                                            |
| IMPULCHK | CAF    | BIT9     | # CHECK FOR IMPULSIVE BURN                 |
|          | MASK   | FLAGWRD2 |                                            |
|          | CCS    | A        |                                            |
|          | TCF    | IMPLBURN | # IMPULSIVE                                |
|          | CS     | FLAGWRD6 | # NON-IMPULSIVE, SET STRULLSW FOR STEERULL |
|          | MASK   | BIT13    |                                            |
|          | ADS    | FLAGWRD6 |                                            |
| PREPTVC  | CS     | OCT60000 | # RESET T5 BITS                            |
|          | MASK   | FLAGWRD6 |                                            |
|          | TS     | FLAGWRD6 |                                            |
|          | EXTEND |          | # KILL RCS                                 |
|          | DCA    | T5IDL24  |                                            |
|          | DXCH   | T5LOC    |                                            |
|          | CS     | THREE    | # 4.3 = DOTVCON (40CS)                     |
|          | TC     | NEUPHASE |                                            |
|          | OCT    | 4        |                                            |

# Page 696

|          |        |          |                                             |
|----------|--------|----------|---------------------------------------------|
|          | TC     | FIXDELAY |                                             |
|          | DEC    | 40       | # 0.4 SECOND DELAY FOR THRUST BUILDUP       |
| DOTVCON  | CS     | BIT1     | # SET TVCPHASE = TVCDAPON CALL (FRESHDAP)   |
|          | TS     | TVCPHASE |                                             |
|          | CAF    | ZERO     | # SET TVCEXECUTIVE PHASE                    |
|          | TS     | TVCEXPHS |                                             |
|          | CS     | OCT60000 | # SET T5 BITS TO INDICATE TVC TAKEOVER .... |
|          | MASK   | FLAGWRD6 | # BITS 15,14 = 10                           |
|          | AD     | BIT15    |                                             |
|          | TS     | FLAGWRD6 |                                             |
|          | CAF    | THREE    | # 6.3 = CLOKTASK (100CS), DROPPING PRE40.6  |
|          | TS     | L        | # WHICH IS HANDLED NOW BY REDOTVC           |
|          | COM    |          |                                             |
|          | DXCH   | -PHASE6  |                                             |
|          | EXTEND |          | # STORE RCS ATTITUDE ERRORS FOR USE IN      |
|          | DCS    | ERRORY   | # INITIALIZING TVC ATTITUDE ERRORS          |
|          | DXCH   | ERRBTMP  |                                             |
|          | CS     | FIVE     | # 4.5 = DOSTRULL (160 CS)                   |
|          | TC     | NEWPHASE |                                             |
|          | OCT    | 4        |                                             |
|          | CAF    | POSMAX   | # SET TIME5 FOR STARTING RIGHT AWAY         |
|          | TS     | TIME5    |                                             |
|          | EXTEND |          |                                             |
|          | DCA    | TVCON2C  | # (TVCDAPON)                                |
|          | DXCH   | T5LOC    | # (KILLS RCS DAP)                           |
|          | TC     | FIXDELAY | # 0.4 + 1.6 = 2.0 SEC FOR ULLAGE-OFF AND    |
|          | DEC    | 160      | # STEERING (IF NON-IMPULSIVE)               |
| DOSTRULL | CAF    | BIT13    | # CHECK STRULLSW FOR IMPULSIVE BURN         |
|          | MASK   | FLAGWRD6 |                                             |
|          | CCS    | A        |                                             |
|          | TCR    | STEERULL | # NON-IMPULSIVE, STEERING AND ULLAGE OFF    |
|          | TCR    | ULAGEOFF | # ULLAGE OFF (ONLY, OR AGAIN)               |
|          | EXTEND |          |                                             |
|          | DCA    | NEGO     | # KILL GROUP 4 (DP NEG0 = -0,+0)            |
|          | DXCH   | -PHASE4  |                                             |
| ENDIGN   | TCF    | TASKOVER |                                             |



July 12, 2016

1257

```
STEERULL CS FLAGWRD2 # SET STEERSW
 MASK BIT11
 ADS FLAGWRD2

Page 697

ULAGEOFF CAF ZERO
 EXTEND
 WRITE CHAN5 # ZERO CHANNEL 5
 TC Q

IMPLBURN CS BIT13 # RESET STRULLSW (COULD BE AN IMPULSIVE
 MASK FLAGWRD6 # ENGINE FAIL)
 TS FLAGWRD6

 TCR E7SETTER

 EBANK= TIG
 EXTEND
 DCA TGO
 DXCH TIG
 EXTEND
 DCA TIME2
 DAS TIG

 TC 2PHSCHNG
 OCT 40153 # A, 3.15 = ENGINOFF (TGO+1) NOT GROUP
 OCT 07014 # C, DELTAT NEXT, TASK BELOW, IN
 DEC -0 # -0 CS
 EBANK= DAPDATR1
 2CADR IMPLCONT

 CAE TGO +1 # (TPAGREE IN S40.13, LESS THAN 600CS)
 TC WAITLIST
 EBANK= TGO
 2CADR ENGINOFF

IMPLCONT CS BIT9 # RESET IMPULSW, ENGINOFF IS NOW SET UP
 MASK FLAGWRD2
 TS FLAGWRD2

 TCR E6SETTER
 EBANK= DAPDATR1

 CAF ZERO # SET UP V97VCNTR IN CASE ENGINOFF (MASS=
```

|            |        |          |   |                                          |
|------------|--------|----------|---|------------------------------------------|
|            | TS     | V97VCNTR | # | BACK) ARRIVES BEFORE TVCDAPON            |
|            | TCF    | PREPTVC  |   |                                          |
| ENGINEOFF  | EBANK= | TGO      | # | E7 FORCED BY 3.15SPOT VARIABLE DELTA-T   |
|            | TCR    | E6SETTER | # | TASK, 3.15 PHASE (TGO+1 CS) GET E6       |
|            | EBANK= | DAPDATR1 |   |                                          |
|            | CAE    | CSMMASS  |   |                                          |
| # Page 698 | TS     | MASSTMP  | # | COPYCYCLE FOR MASSBACK                   |
|            | TC     | 2PHSCHNG |   |                                          |
|            | OCT    | 00003    | # | KILL GROUP 3 PROTECTION OF ENGINEOFF, DO |
|            | OCT    | 40634    | # | A, 4.63 = DOSPSOFF (-OCS)                |
| DOSPSOFF   | TCR    | SPSOFF   | # | SHUTDOWN SPS, MASS UPDATES, ETC.         |
|            | CS     | OCT27/24 | # | (OCTAL 27)                               |
|            | TC     | NEWPHASE |   |                                          |
|            | OCT    | 4        | # | 4.27 = DOTVCRCS (250 CS)                 |
|            | TC     | FIXDELAY | # | 2.5 SECOND DELAY FOR SPS TAILOFF         |
|            | DEC    | 250      |   |                                          |
| DOTVCRCS   | TCR    | SETMAXDB | # | WIDE DEADBAND FOR CUTOFF TRANSIENT       |
|            | TC     | IBNKCALL | # | SET UP RCS DAP (KILLS TVCDAPS, SETS T5   |
|            | CADR   | RCSDAPON | # | BITS, WAITS 0.6SEC FOR TVCEXEC DIE)      |
|            | TC     | IBNKCALL | # | UPDATE WEIGHT/G AND MASS-PROPERTIES FOR  |
|            | CADR   | MASSPROP | # | RCS DAP STARTUP IN 0.6 SECONDS           |
|            | TCR    | TVCZAP   | # | WIPE OUT TVC, TURN OFF CLOKTASK          |
|            | TC     | PHASCHNG |   |                                          |
|            | OCT    | 00354    | # | A, 4.35 = POSTBURN (NOVAC, PRI012)       |
|            | CAF    | PRI012   | # | SET UP POSTBURN V16N40 JOB               |
|            | TC     | NOVAC    |   |                                          |
|            | EBANK= | DAPDATR1 | # | (SET MAXDB IN POST41)                    |
|            | 2CADR  | POSTBURN |   |                                          |
|            | TCF    | TASKOVER |   |                                          |
| SPSOFF     | EBANK= | DAPDATR1 | # | ESTABLISH SPSOFF TEVENT                  |
|            | EXTEND |          |   |                                          |
|            | DCA    | TIME2    |   |                                          |
|            | DXCH   | TEVENT   |   |                                          |
|            | CS     | BIT7     | # | RESET ENGONFLG                           |
|            | MASK   | FLAGWRD5 |   |                                          |

July 12, 2016

1259

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | TS     | FLAGWRD5 | # (RESTARTS WILL SHUT DOWN SPS NOW)        |
|            | CS     | BIT13    | # SHUT DOWN SPS ENGINE                     |
|            | EXTEND |          |                                            |
|            | WAND   | DSALMOUT |                                            |
|            | CAF    | BIT14    | # ISSUE SIV CUTOFF COMMAND                 |
|            | EXTEND |          | # FOR POSSIBLE BACK-UP USE                 |
| MASSBACK   | WOR    | CHAN12   |                                            |
|            | CAE    | V97VCNTR | # RESTORE PART OF PRE-DECREMENTED MASS     |
|            |        |          | # V97CNTR = VCNTR UNLESS V97 IS            |
|            |        |          | # ACTIVE. ONLY V97CNTR IS THEN RIGHT.      |
|            | EXTEND |          | # VCNTR COUNTS 1/2-SECONDS IN TVC EXEC     |
|            | MP     | EMDOT    | # MDOT, SC.AT B+3 KG/CS                    |
|            | LXCH   | A        |                                            |
| # Page 699 |        |          |                                            |
|            | EXTEND |          |                                            |
|            | MP     | 1SEC     | # DEC 100                                  |
|            | AD     | MASSTMP  | # CORRECTION IS ACCURATE TO 5 CS OF FLOW   |
|            | TS     | CSMMASS  | # (1.44 KG OR 0.4 BITS)                    |
|            | CA     | TVCPHASE | # CHECK IF OK FOR TRIM UPDATE              |
|            | AD     | ONE      | # THESE CHECKS ARE ONLY NEEDED             |
|            | EXTEND |          | # FOR A LESS THAN 0.4 SEC BURN             |
|            | BZF    | BTRIMR   | # NO. INITIALIZATION NOT COMPLETE          |
|            | CS     | FLAGWRD6 | # YES, CHECK IF TVC                        |
|            | MASK   | OCT60000 |                                            |
|            | EXTEND |          |                                            |
|            | BZMF   | BTRIMR   | # NO, NOT TVC YET                          |
| BESTTRIM   | CAE    | DELPBAR  | # UPDATE TRIMS WITH DELFILTER VALUES       |
|            | TS     | PACTOFF  |                                            |
|            | CAE    | DELYBAR  |                                            |
|            | TS     | YACTOFF  |                                            |
| BTRIMR     | TC     | Q        |                                            |
|            | EBANK= | DAPDATR1 |                                            |
| STEERADS   | 2CADR  | STEERING |                                            |
| .6SECT5    | OCT    | 37703    |                                            |
| 5SECDP     | DEC    | 0        | # MAKE DP 5SEC                             |
| 5SEC       | DEC    | 500      |                                            |
| OCT02202   | OCT    | 02202    | # BITS 2, 8, 11 FOR CHANNEL 12 TVC/OPTICS  |
|            | EBANK= | DAPDATR1 |                                            |
| TVCON2C    | 2CADR  | TVCDAPON |                                            |
|            |        |          |                                            |
|            | INHINT |          |                                            |
| -1         | CS     | OCT02202 | # DISABLE TVC AND OPT ERR CNTRLS, REENGAGE |
| TVCZAP     | EXTEND |          | # OPTICS DAC                               |

```

WAND CHAN12
CS BIT1 # ENABLE T4RUPT OPTICS MONITOR PERMIT
TS OPTIND # OPTICS-ZERO BUT NOT OPTICS-DRIVE
CAF ZERO # CLEAR NVWORD1 IN CASE CLOCKJOB WAITING
TS NVWORD1
CS BIT11 # CLEAR TIMRFLAG TO STOP CLOKTASK
MASK FLAGWRD7
TS FLAGWRD7
TC Q
EBANK= DAPDATR1
UPDATEVG STQ BON
 QTEMP1
 XDELVFLG
 CALL40.8

SLOAD BMN
 NBRCYCLS
 SETUP.9

Page 700

VLOAD VAD
 DELVSUM
 DELVREF
STORE DELVSUMP
EXIT
CA ONE
AD NBRCYCLS
TS NBRCYCLP

TC PHASCHNG # TYPE B RESTART RESTART BELOW AND 5.3 REREAD
OCT 10035

CA NBRCYCLP
TS NBRCYCLS
TC INTPRET
VLOAD
 DELVSUMP
STORE DELVSUM

CALL40.8 CALL
 S40.8
 GOTO
 QTEMP1

SETUP.9 BON SLOAD
 FIRSTFLG

```

July 12, 2016

1261

```
SURELY.9
NBRCYCLP
NORM VXSC # (NORM HANDLES ZERO PROPERLY)
 X1
 BDT
VSR* VAD
 0 -14D,1
 VGTEMP
VSU
 DELVSUM
STORE VGPREV
EXIT
CAF PRI010
TC FINDVAC
EBANK= DAPDATR1
2CADR S40.9

TC 2PHSCHNG
OCT 00051 # A, 1.5 = REDO40.9, PRI0 10
OCT 10035
TC INTPRET
VLOAD

RN # ACTIVE VEHICLE RADIUS VECTOR AT T1
STOVL RINIT

VN # ACTIVE VEHICLE VELOCITY VECTORY AT T1
STODL VINIT
 PIPTIME
STORE TNIT
BDSU

TPASS4
STOVL DELLT4
 HI6ZEROS
STODL DELVSUM
 HI6ZEROS
STORE NBRCYCLS
GOTO

CALL40.8
EBANK= DAPDATR1
TC INTPRET
CALL

UPDATEVG
EXIT
CAF BIT9 # CHECK IMPULSW
MASK FLAGWRD2
CCS A
```

SURELY.9

# Page 701

STEERING

|        |        |          |                                            |
|--------|--------|----------|--------------------------------------------|
| SERVXT | TCF    | +3       | # PRE-IGNITE, REQUEST ENG-OFF, OR POST-OFF |
|        | TC     | POSTJUMP |                                            |
|        | CADR   | SERVEXIT |                                            |
|        | CAF    | BIT13    | # CHECK ENGINE-ON/-OFF                     |
|        | EXTEND |          |                                            |
|        | RAND   | DSALMOUT |                                            |
|        | EXTEND |          |                                            |
|        | BZF    | SERVXT   | # ENGINE-OFF, SO PRE-IGNITE OR POST-OFF    |
|        | TCR    | E7SETTER |                                            |
|        | EBANK= | TIG      |                                            |
|        | INHINT |          |                                            |
|        | EXTEND |          |                                            |
|        | DCA    | TIG      |                                            |
|        | DXCH   | MPAC     |                                            |
|        | EXTEND |          |                                            |
|        | DCS    | TIME2    |                                            |
|        | DAS    | MPAC     |                                            |
|        | TCR    | DPAGREE  |                                            |
|        | CAE    | MPAC +1  | # (LESS THAN 6 (OR 4) SECONDS TO GO)       |
|        | CCS    | A        | # PROTECT AGAINST NEG/ZRO W.L. CALL        |
|        | TCF    | +3       |                                            |
|        | TCF    | +2       |                                            |
|        | CAF    | ZERO     |                                            |
|        | AD     | ONE      |                                            |
|        | XCH    | L        |                                            |
|        | CA     | ZERO     |                                            |
|        | DXCH   | TGO      |                                            |
|        | CA     | TGO +1   |                                            |
|        | TC     | WAITLIST |                                            |

# Page 702

|        |           |
|--------|-----------|
| EBANK= | TGO       |
| 2CADR  | ENGINEOFF |

|       |          |
|-------|----------|
| TC    | 2PHSCHNG |
| OCT   | 40153    |
| OCT   | 10035    |
| TC    | DOWNFLAG |
| ADRES | IMPULSW  |
| TCF   | SERVXT   |

|                                               |
|-----------------------------------------------|
| # A, 3.15 = ENGINEOFF (TGO+1) .... NOTE GROUP |
| # B, 5.3 = REREADAC, AND START BELOW          |
| # CLEAR IMPULSW, ENGINEOFF IS NOW SET UP      |
| # RESTARTS OK                                 |

# Page 703

# ROUTINE \*\* CLOKTASK \*\* DESCRIPTION

|          |        |          |                   |
|----------|--------|----------|-------------------|
| CLOKTASK | EBANK= | TIG      |                   |
|          | CAF    | BIT11    | # IS TIMRFLAG SET |
|          | MASK   | FLAGWRD7 |                   |

July 12, 2016

1263

|                       |        |          |                                            |
|-----------------------|--------|----------|--------------------------------------------|
|                       | CCS    | A        |                                            |
|                       | TCF    | CLOCKON  |                                            |
|                       | TC     | PHASCHNG |                                            |
|                       | OCT    | 00006    | # KILL RESTART                             |
|                       | TC     | TASKOVER |                                            |
| CLOCKON               | EXTEND |          |                                            |
|                       | DCA    | TIME2    |                                            |
|                       | DXCH   | TTOGO    |                                            |
|                       | EXTEND |          |                                            |
|                       | DCS    | TIG      |                                            |
|                       | DAS    | TTOGO    |                                            |
| SETCLOCK              | CAF    | 1SEC     |                                            |
|                       | TC     | WAITLIST |                                            |
|                       | EBANK= | TIG      |                                            |
|                       | 2CADR  | CLOKTASK |                                            |
|                       | CCS    | NVWORD1  |                                            |
|                       | TCF    | +3       |                                            |
|                       | TCF    | SETTB6   |                                            |
|                       | TCF    | +1       |                                            |
|                       | CS     | V06N85B  | # CHECK FOR V06N85B (P41)                  |
|                       | AD     | NVWORD1  |                                            |
|                       | EXTEND |          |                                            |
|                       | BZF    | SETUPDYN | # V06N85, SO UPDATE N85 FOR DYNAMIC DISP   |
|                       | CAF    | PRI027   |                                            |
|                       | TC     | NOVAC    |                                            |
|                       | EBANK= | DAPDATR1 |                                            |
|                       | 2CADR  | CLOCKJOB |                                            |
| SETTB6                | CS     | TIME1    | # SET GROUP6 TIMEBASE                      |
|                       | TS     | TBASE6   |                                            |
|                       | TCF    | TASKOVER |                                            |
| SETUPDYN              | CAF    | PRI027   | # SET UP A JOB TO UPDATE N85 (FOR P41=V06) |
|                       | TC     | FINDVAC  |                                            |
|                       | EBANK= | DAPDATR1 |                                            |
|                       | 2CADR  | DYNDISP  |                                            |
|                       | TCF    | SETTB6   | # CLOSE OUT CLOKTASK                       |
| # Page 704<br>DYNDISP | TC     | INTPRET  | # UPDATE N85 FOR A DYNAMIC V06N85 IN P41.  |
|                       | CALL   |          | # PRIOR TO BLANKING AND AVEG (V16N85)      |

P40CNV85

EXIT

TCF CKNVWRD1

# Page 705

# ROUTINE \*\* CLOCKJOB \*\* DESCRIPTION

|          |        |          |                                            |
|----------|--------|----------|--------------------------------------------|
|          | EBANK= | DAPDATR1 |                                            |
| CLOCKJOB | CA     | CDUX     |                                            |
|          | TS     | CDUSPOTX |                                            |
|          | CA     | CDUY     |                                            |
|          | TS     | CDUSPOTY |                                            |
|          | CA     | CDUZ     |                                            |
|          | TS     | CDUSPOTZ |                                            |
|          | TC     | BANKCALL |                                            |
|          | CADR   | QUICTRIG |                                            |
| CKNVWRD1 | INHINT |          |                                            |
|          | CCS    | NVWORD1  | # DETERMINE FUNCTION, INDICATED BY NVWORD1 |
|          | TCF    | NOFLASH  |                                            |
|          | TCF    | ENDOFJOB |                                            |
|          | TCF    | ENGREQST | # SPS ENGINE-ON-ENABLE V99 FLASH           |
| FAILDSP  | CAF    | V06N40   | # SPS ENGINE-FAILED V97 FLASH              |
|          | TC     | BANKCALL |                                            |
|          | CADR   | CLOCPLAY |                                            |
|          | TCF    | V97T     | # TERMINATE                                |
|          | TCF    | V97P     | # PROCEED                                  |
|          | TCF    | V97E     | # ENTER                                    |
| ENGREQST | CAF    | V06N40   |                                            |
|          | TC     | BANKCALL |                                            |
|          | CADR   | CLOCPLAY | # LINUS MAKES IT A REDO, INHINT OK         |
|          | TCF    | V99T     | # TERMINATE                                |
|          | TCF    | V99P     | # PROCEED                                  |
|          | TCF    | V99E     | # ENTER                                    |
| NOFLASH  | CAE    | NVWORD1  | # DISPLAY NVWORD1 NORMALLY                 |
|          | TC     | BANKCALL |                                            |
|          | CADR   | REGODSP  |                                            |
| E7SETTER | CAF    | EBANK7   |                                            |
|          | TS     | EBANK    |                                            |
|          | EBANK= | TIG      |                                            |
|          | TC     | Q        |                                            |
| E6SETTER | CAF    | EBANK6   | # SET UP EBANK6                            |
|          | TS     | EBANK    |                                            |



July 12, 2016

1265

|            |        |           |                                          |
|------------|--------|-----------|------------------------------------------|
|            | EBANK= | DAPDATR1  |                                          |
|            | TC     | Q         |                                          |
| V99E       | EBANK= | DAPDATR1  |                                          |
|            | TC     | 2PHSCHNG  |                                          |
|            | OCT    | 00006     | # KILL PRE40.6/CLOKTASK PROTECTION       |
|            | OCT    | 05024     | # C, PRIORITY NEXT, JOB BELOW            |
| # Page 706 |        |           |                                          |
|            | OCT    | 27000     |                                          |
| V99EJOB    | TCR    | TVCZAP -1 | # WIPE OUT TVC, CLOKTASK                 |
|            | TCF    | P40RCS    | # V16N85 POST-BURN OPERATIONS            |
| V99T       | EBANK= | DAPDATR1  |                                          |
|            | TC     | 2PHSCHNG  | # (ENTRY FROM V97T FLOW TOO)             |
|            | OCT    | 00006     | # KILL PRE40.6/CLOKTASK PROTECTION       |
|            | OCT    | 05024     | # C, PRIORITY NEXT, JOB BELOW            |
|            | OCT    | 27000     |                                          |
| V99TJOB    | TCR    | TVCZAP -1 | # WIPE OUT TVC, CLOKTASK                 |
|            | TCF    | POST41    | # AVEGEXIT, SETMAXDB, GOTOPOOH           |
| V99P       | INHINT |           |                                          |
|            | CAE    | FLAGWRD7  | # CHECK ASTN FLAG FOR PRIOR V99P         |
|            | MASK   | BIT12     |                                          |
|            | CCS    | A         |                                          |
|            | TCF    | V99P/TIG  | # YES, THIS MUST BE A RESTART ENTRY      |
| ASTNV99P   | CAF    | BIT12     | # SET ASTN FLAG                          |
|            | ADS    | FLAGWRD7  |                                          |
|            | CAE    | FLAGWRD7  | # CHECK IGN FLAG FOR TIG-0 ARRIVAL       |
|            | MASK   | BIT13     |                                          |
|            | EXTEND |           |                                          |
|            | BZF    | V99P/TIG  | # NO, CLEAR THE V99 AND WAIT FOR TIG-0   |
| ENDV99PI   | CAF    | BIT1      | # TIG-0 HAS COME ALREADY                 |
|            | TC     | WAITLIST  | # SET UP IGNITION HERE                   |
|            | EBANK= | DAPDATR1  |                                          |
|            | 2CADR  | IGNITION  |                                          |
| V99P/TIG   | CAF    | V06N40    | # CLEAR THE V99 FLASH AND WAIT FOR TIG-0 |
|            | TS     | NVWORD1   |                                          |
| ENDV99P    | TCF    | ENDOFJOB  |                                          |
| V97T       | EBANK= | CSMMASS   |                                          |
|            | TC     | 2PHSCHNG  |                                          |
|            | OCT    | 00006     | # KILL GROUP 6 (CLOKTASK)                |
|            | OCT    | 40674     | # A, 4.67 = V97TTASK (-0 CS), TBASE NOW  |

```

CAF BIT1
TC TWIDDLE
ADRES V97TTASK
TCF ENDOFJOB # KEEP EBANK6 FOR MASSES, SPSOFF, ETC.

EBANK= CSMMASS
V97TTASK CAF ZERO # DISABLE CLOCKJOB
TS NVWORD1
CAF 3MDOT # 3 SECONDS OF MDOT (2-4 SEC ENGFAIL
AD CSMMASS # DETECTION) NOT LOST BECAUSE THRUST
TS MASSTMP # FAILED. COPYCYCLE FOR MASSBACK

Page 707
TC PHASCHNG
OCT 05014 # C, DELTAT NEXT, TASK BELOW, IN
DEC -0 # -0 CS

TCR SPSOFF # SHUTDOWN SPS ENGINE, MASS UPDATE, ETC.
TC PHASCHNG
OCT 00714 # A, 4.71 = V97TRCS (250 CS), TBASE OLD
TC FIXDELAY # DELAY 2.5 SECONDS FOR (POSSIBLE) TAIL-
DEC 250 # OFF (FALSE THRUST-LOSS)

EBANK= DAPDATR1
V97TRCS TC IBNKCALL # RCS DAP IN 0.6SEC, SETTING T5 BITS TO
CADR RCSDAPON # KILL TVCEXEC/TVCROLLDAP STARTS
CAF PRI027 # SET UP V99T FOR TVCZAP AND POST41 (SET-
TC NOVAC # MAXDB AND GOTOPOOH)
EBANK= DAPDATR1 # EBANK6 FOR SETMAXDB IN POST41
2CADR V99T

ENDV97T TCF TASKOVER

EBANK= V97VCNTR
V97P TC PHASCHNG
OCT 40734 # A, 4.73 = V97PTASK (-0 CS), TBASE NOW
CAF BIT1
TC TWIDDLE
ADRES V97PTASK
TCF ENDOFJOB

EBANK= V97VCNTR
V97PTASK CAE V97VCNTR # GET MASS UPDATES (TVCEXEC) GOING AGAIN
TS VCNTR # (ERRORS IF FLASE THRUST-LOSS AND/OR
POOR SYNC OF MANUAL ENGINE-ON AND
THE VERB 97 PROCEED)
CAF V06N40 # REDISPLAY V06N40

```

July 12, 2016

1267

```
TS NVWORD1
TC UPFLAG # SET IDLEFAIL TO ALLOW R41-BYPASS, IN
ADRES IDLEFAIL # CASE OF UNFAVORABLE S40.8 SYNCH
TC UPFLAG # SET STEERSW TO RE-ENABLE STEERING
ADRES STEERSW
TC PHASCHNG
OCT 00134 # A, 4.13 = R40ENABL (200 CS), TBASE OLD
TC FIXDELAY # WAIT 2 SECONDS, THEN
DEC 200

R40ENABL EBANK= WHOCARES
TC DOWNFLAG # RE-ENABLE R40 BY CLEARING IDLEFAIL
ADRES IDLEFAIL
TC PHASCHNG
OCT 00004 # KILL GROUP 4

Page 708
ENDV97P TCF TASKOVER

V97E EBANK= WHOCARES
TC PHASCHNG
OCT 40534 # A, 4.53 = V97ETASK (-0 CS), TBASE NOW
CAF BIT1
TC WAITLIST
EBANK= TIG
2CADR V97ETASK

TCF ENDOFJOB

V97ETASK EBANK= TIG
CS OCT24 # FORCE R1 OF V06N40 TO READ 59X59
TS TIG
CAF V06N40 # REDISPLAY V06N40
TS NVWORD1
TCR E6SETTER # RETURN TO EBANK6 FOR REST OF V97ETASK
EBANK= CSMMASS
CAF 3MDOT # 3 SECONDS OF MDOT (2-4 SEC ENGFAIL
AD CSMMASS # DETECTION) NOT LOST BECAUSE THRUST
TS MASSTMP # FAILED....COPYCYCLE FOR MASSBACK
TC PHASCHNG
OCT 00754 # A, 4.75 = SPSOFF97 (-0 CS), TBASE OLD
TCR SPSOFF
TC PHASCHNG
OCT 00114 # A, 4.11 = V97E40.6 (250 CS), TBASE OLD
TC FIXDELAY # DELAY 2.5 SECONDS FOR (POSSIBLE) TAIL-
DEC 250 # OFF (FALSE THRUST-LOSS)
```

```

V97E40.6 EBANK= DAPDATR1
 CAF BIT1
 TC WAITLIST
 EBANK= CNTR
 2CADR PRE40.6 # USE S40.6 RESTART ENTRY TO TRIM ENGINE

 TC IBNKCALL # RCS DAP IN 0.6SEC, SETTING T5 BITS TO
 CADR RCSDAPON # KILL TVCEXEC/TVCROLLDAP STARTS.
 # LEAVE NARROW DEADBAND FOR REIGNITE.

 TC 2PHSCHNG
 OCT 00026 # A, 6.2 = PRE40.6 (-0 CS), CLOKTASK (1 SEC)
 OCT 05014 # C, DELTAT NEXT, TASK BELOW, IN
 DEC -0 # -0 CS.

QUICKIGN CS PRI014 # CLEAR ASTNFLAG AND SET IGNFLAG FOR
 MASK FLAGWRD7 # IMMEDIATE V99 RESPONSE.
 AD BIT13
 TS FLAGWRD7
 TC FIXDELAY # DELAY TO ALLOW TIME FOR PRE40.6

Page 709
 DEC 30

V99FLASH CS BIT9 # CAUSE V99 TO FLASH
 TS NVWORD1
 TC 2PHSCHNG
 OCT 40774 # A, 4.77 = TIG-0 (-0 CS) TBASE FOR PREPTVC
 OCT 00033 # A, 3.3 = S40.13 (PRIO 20)
 CAF PRI020 # SET UP TIMEBURN
 TC FINDVAC
 EBANK= TGO
 2CADR S40.13

ENDV97E TCF TASKOVER # WAIT FOR CLOCKJOB (IMMEDIATE) REACTION
 # TO FLASHING V99 RESPONSE.

MOD NO2
MOD BY ZELDIN
#
FUNCTIONAL DESCRIPTION
COMPUTE INITIAL THRUST DIRECTION(UT) AND INITIAL VALUE OF VG
VECTOR(VGTIG).
#
CALLING SEQUENCE
L CALL
L+1 S40.1

```

LOG SECTION P40-P47

```

#
NORMAL EXIT MODE
AT L+2 OF CALLING SEQUENCE (GOTO L+2) NORMAL RETURN OR
ERROR RETURN IF NOSOFLAG =1
#
SUBROUTINES CALLED
CSMPREC
INITVEL
CALCGRAV
MIDGIM
#
ALARM OR ABORT EXIT MODES
L+2 OF CALLING SEQUENCE, UNSOLVABLE CONIC IF NOSOFLAG=1
#
ERASABLE INITIALIZATION REQUIRED
WEIGHT/G ANTICIPATED VEHICLE MASS SP B16 KGM
XDELVFLG 1=DELTA-V MANEUVER, 0=AIMPT STEER
IF DELTA-V MANEUVER:
DELVSIN SPECIFIED DELTA-V REQUIRED IN
INERTIAL COORDS. OF ACTIVE VEHICLE
AT TIME OF IGNITION VECTOR B7 M/CS
DELVSAB MAG. OF DELVSIN DP B7 M/CS
RTIG POSITION AT TIME OF IGNITION VECTOR B29 M
VTIG VELOCITY AT TIME OF IGNITION VECTOR B7 M/CS.
CSTEER = 0 DP
IF AIMPOINT STEERING:
IF AIMPT STEER
TIG TIME OF IGNITION DP B28 CS
RTARG POSITION TARGET TIME VECTOR B29 M
CSTEER = ECSTEER (GR 0) DP B1
Page 710
TPASS4 -- TIME OF ARRIVAL AT AIMPOINT
#
OUTPUT
UT 1/2 UNIT VECTOR ALIGNED WITH THRUST DIRECTION IN REF COOR
VGTIG INITIAL VALUE OF VELOCITY
TO BE GAINED (INERT. COORD.) VECTOR B7 M/CS
DELVLVC VGTIG IN LOC. VERT. COORDS. B7 M/CS
F NOMINAL THRUST FOR ENG USED FOR S40.13 DP B7 M-NEWT
BDT V REQUIRED AT TIG -V REQUIRED AT (TIG-2SEC)
-GDT FOR S40.13 VECT B7 M/CS
RTIG CALC IN S40.1B (AIMPT) FOR S40.2,3 VECTOR B29M
POSITION AT TIME OF IGNITION
#
DEBRIS QTEMP1
MPAC, QPRET

```

1270

July 12, 2016

```
PUSHLIST
RTX2,RTX1

 BANK 14
 SETLOC P40S1
 BANK

 COUNT 16/S40.1

S40.1 SET VLOAD
 FIRSTFLG
 LO6ZEROS
 STORE BDT
 STQ BOF
 QTEMP
 XDELVFLG
 S40.1B # LAMBERT
 VLOAD ABVAL # EXTERNAL DELTA-V
 DELVSIN
 STORE DELVSAB # COMPUTE FOR P30/P40 INTERFACE
 # THUS PERMITTING MODULE-ONLY CHANGE

 SETPD VLOAD
 0
 VTIG
 STORE VINIT
 VXV UNIT
 RTIG
 STOVL UT # UP IN UT
 RTIG
 STORE RINIT
 VSQ PDDL
 36D
 DMPR DDV
 THETACON
 DMP DMP
 DELVSAB
 WEIGHT/G
 DDV

Page 711

 F
 STOVL 14D
 DELVSIN

 DOT VXSC
 UT
 UT
```

July 12, 2016

1271

```

VSL2 PUSH # (DELTA V.UP) UP SCALED AT 2(+7) P.D.L. 0
BVSU PDDL # DELTA VP SCALED AT 2(+7) P.D.L. 6
 DELVSIN
 14D
SIN PDVL
 6D
VXV UNIT
 UT
VXSC STADR
STOVL VGTIG # UNIT(VP X UP) SIN(THETAT/2) IN VGTIG.
UNIT PDDL # UNIT(DELTA VP) IN P.D.L. 6
 14D
COS VXSC
VAD VXSC
 VGTIG
 36D
VSL2 VAD
STADR
STORE VGTIG # VG IGNITION SCALED AT 2(+7) M/CS

UNIT
STOVL UT # THRUST DIRECTION SCALED AT 2(+1)
 VGTIG
PUSH SET
 AVFLAG
CALL
 MIDGIM # VGTIG IN LV COOR AT 2(+7) M/CS IN DELVLVC
GOTO
 QTEMP
S40.1B DLOAD # LAMBERT
 DSU
 TIG
 TWODT
STODL TDEC1
 TPASS4
DSU
 TDEC1
STCALL DELLT4
 AGAIN
VLOAD
 VIPRIME
STODL UT
 TIG
STORE TDEC1

Page 712
BDSU
 TPASS4
```

|          |        |               |
|----------|--------|---------------|
|          | STCALL | DELLT4        |
|          |        | AGAIN         |
|          | VLOAD  | PUSH          |
|          |        | DELVEET3      |
|          | STORE  | VGITG         |
|          | SET    | CALL          |
|          |        | AVFLAG        |
|          |        | MIDGIM        |
|          | SETPD  | GOTO          |
|          |        | 0             |
|          |        | CALCUT        |
| THETACON | 2DEC   | .31830989 B-8 |
|          | SETLOC | P40S3         |
|          | BANK   |               |
|          | COUNT  | 24/S40.1      |
| EP4(45)H | 2DEC   | .125          |
| EP4(10)H | 2DEC   | .027777777    |
| AGAIN    | STQ    | CALL          |
|          |        | QTEMP1        |
|          |        | THISPREC      |
|          | SXA,2  | SXA,1         |
|          |        | RTX2          |
|          |        | RTX1          |
|          | VLOAD  |               |
|          |        | RATT          |
|          | STORE  | RTIG          |
|          | STOVL  | RINIT         |
|          |        | VATT          |
|          | STORE  | VTIG          |
|          | STORE  | VINIT         |
|          | SETPD  | SLOAD         |
|          |        | 0             |
|          |        | HI6ZEROS      |
|          | PDDL   | BON           |
|          |        | EP4(45)H      |
|          |        | NORMSW        |
|          |        | +3            |
|          | DLOAD  |               |
|          |        | EP4(10)H      |
|          | PUSH   | CALL          |



July 12, 2016

1273

|            |        |          |                                          |
|------------|--------|----------|------------------------------------------|
|            |        | INITVEL  |                                          |
|            | SETPD  | GOTO     |                                          |
| # Page 713 |        | 0        |                                          |
|            |        | QTEMP1   |                                          |
| CALCUT     | VLOAD  | CALL     |                                          |
|            |        | RTIG     |                                          |
|            |        | CALCGRAV | # GDELTAT IN MPAC AT 2(+7) M/CS          |
|            | VSL1   | V/SC     |                                          |
|            |        | 200CS    | # G AT 2(-5) M/CS. CS                    |
|            | PDVL   | VSU      |                                          |
|            |        | VIPRIME  |                                          |
|            |        | UT       |                                          |
|            | V/SC   | VSU      |                                          |
|            |        | 200CS    |                                          |
|            | VXSC   | VSL2     |                                          |
|            |        | CSTEER   |                                          |
|            | STOVL  | 12D      | # B.C SCALED AT 2(-15) PDL 12D           |
|            |        | VG TIG   |                                          |
|            | UNIT   | PUSH     | # UG PDL 0 SCALED AT 2(+1)               |
|            | DOT    | VXSC     |                                          |
|            |        | 12D      |                                          |
|            |        | 0        |                                          |
|            | VSL2   | BVSU     |                                          |
|            |        | 12D      |                                          |
|            | STODL  | 12D      | # Q PDL SCALED AT 2(-5)                  |
|            |        | F        |                                          |
|            | SRR    | DDV      |                                          |
|            |        | 4        |                                          |
|            |        | WEIGHT/G |                                          |
|            | DSQ    | PDVL     | # F/MASS SQUARED PDL 6 AT 2(-10M/(CS.CS) |
|            |        | 12D      |                                          |
|            | VSQ    |          |                                          |
|            | BDSU   | SQRT     |                                          |
|            | VXSC   | VSL1     |                                          |
|            | VAD    | UNIT     |                                          |
|            |        | 12D      |                                          |
|            | STCALL | UT       |                                          |
|            |        | QTEMP    |                                          |
| 200CS      | 2DEC   | 200 B-12 |                                          |

# Page 714

# PROGRAM DESCRIPTION S40.2,3

# MOD NO 2

# MOD BY ZELDIN

DATE 15 NOV 66

LOG SECTIONS P40-P47

```

#
FUNCTIONAL DESCRIPTION
#
COMPUTE GIMBAL ANGLES IF THRUSTING OCCURRED WITH PRESENT IMU
ORIENTATION, WINGS LEVEL SPACECRAFT, HEADS UP
COMPUTE X AXIS OF ENGINE BELL
COMPUTE PREFERRED IMU ORIENTATION (XSCREF)
FOR THIS CALCULATION, ASSUME X AXIS OF SC ALONG UT INITIALLY,
YSC=UNIT(XXR), ZSC=UNIT(XX(XXR)) AND ROTATE ENGINE BELL ALONG UT.
NEW SC AXES WILL BE APPROX. WINGS LEVEL AND NEW SC AXES IN REF.
COORDS. WILL BE PREFERRED IMU ORIENTATION.
COMPUTE DESIRED THRUST DIRECTION IN SM COORDS.
#
CALLING SEQUENCE
L CALL
L+1 S40.2,3
#
NORMAL EXIT MODE
AT L+2 OF CALLING SEQUENCE (GOTO L+2)
#
SUBROUTINES CALLED
CALCGA
#
ALARM OR ABORT MODES
NONE
#
ERASABLE INITIALIZATION REQUIRED
PACTOFF TOTAL PITCH TRIM ANGLE SP AT 1.0795111 REV.
YACTOFF TOTAL YAW TRIM ANGLE SP AT 1.0795111 REV.
UT DESIRED THRUST DIRECTION VECT. B2 M/(CS.CS)
RTIG POSITION AT TIME OF IGNITION VECT. B29 M
ENG2FLAG ON=RCS OFF=SPS
#
OUTPUT
SCAXIS UNIT VECT. ALIGNED WITH ENG BELL IN SC COOR. B1
XSCREF UNIT VECTORS ALIGNED WTH PREFERRED IMU B1
YSCREF
ZSCREF
GIMBAL ANGLES IN THETAD
POINTVSM UNIT VECT ALONG DESIRED THRUST DIRECTION IN SM B1
#
DEBRIS
PUSHLIST, QPRET, MPAC
QTEMP TEMP. ERASABLE

```

July 12, 2016

1275

```
S40.2,3
SETLOC P40S
BANK
COUNT* $$/S40.2
VLOAD MXV
 UT
 REFSMMAT
VSL1 STQ
 QTEMP
STORE POINTVSM # THRUST IN SM AT 2
SETPD BON
 0

Page 715
 ENG2FLAG
 S40.2,3B
DLOAD
 HI6ZEROS
PUSH SLOAD # ZERO PDL 0
 YACTOFF
DMP SL1
 TRIMSCAL
DAD PUSH
 YBIAS
COS PDDL # COS(Y +Y0) PDL 2
SIN PUSH # SIN(Y +Y0) PDL 4
SLOAD
 PACTOFF
DMP SL1
 TRIMSCAL
DAD PUSH
 PBIAS
COS PDDL # COS(P +P0) PDL 6
SIN PUSH # SIN(P +P0) PDL 8D
STODL ZSCREF # SIN(P +P0)
 6
DMP SL1
 4
DCOMP PDDL # -SIN(Y+Y0)COS(P+P0) PDL 10
 6
DMP SL1
 2
VDEF

STODL XSCREF # PD POINTER AT 6 NEW SC X AXIS SCALED AT
 ZSCREF
DMP SL1
 4
```

# Page 716

S40.2,3B

|       |         |                                             |
|-------|---------|---------------------------------------------|
| PDDL  | DMP     |                                             |
|       | ZSCREF  |                                             |
|       | 2       |                                             |
| SL1   | DCOMP   |                                             |
| VDEF  |         |                                             |
| STODL | ZSCREF  | # PD POINTER AT 4 NEW SC Z AXIS SCALED AT 2 |
| VDEF  |         |                                             |
| STODL | YSCREF  | # PD POINTER AT 0 NEW SC Y AXIS SCALED AT 2 |
|       | ZSCREF  |                                             |
| PDDL  | PDDL    |                                             |
|       | YSCREF  |                                             |
|       | XSCREF  |                                             |
| VDEF  |         |                                             |
| STOVL | SCAXIS  | # ENGINE BELL SCALED AT 2                   |
|       | UT      |                                             |
| PDVL  | UNIT    |                                             |
|       | RTIG    |                                             |
| VXV   | VCOMP   |                                             |
|       | 0       |                                             |
| UNIT  | PUSH    |                                             |
| CALL  |         |                                             |
|       | TSTRXUT |                                             |
| VXV   | VCOMP   |                                             |
|       | 0       |                                             |
| VSL1  | PDVL    | # 2 RF/SC IN PDL 12D                        |
|       | XSCREF  |                                             |
| VXM   | VSL1    |                                             |
|       | 0       |                                             |
| STOVL | XSCREF  | # X OF PREF. IMU,X OF SC IN REF COOR. AT 2  |
|       | YSCREF  |                                             |
| VXM   | VSL1    |                                             |
|       | 0       |                                             |
| STOVL | YSCREF  | # Y OF PREF. IMU,Y OF SC IN REF COOR. AT 2  |
|       | ZSCREF  |                                             |
| VXM   | VSL1    |                                             |
|       | 0       |                                             |
| STORE | ZSCREF  | # Z OF PREF. IMU,Z OF SC IN REF COOR. AT 2  |
| SETPD | GOTO    |                                             |
|       | 0       |                                             |
|       | QTEMP   |                                             |
| VLOAD |         |                                             |
|       | UNITX   |                                             |
| STOVL | SCAXIS  |                                             |

July 12, 2016

1277

```

 UT
 XSCREF
STORE VXV UNIT
 RTIG
STCALL 6D
 TSTRXUT
STORE YSCREF
VXV VCOMP
 XSCREF
VSL1
STCALL ZSCREF # ZNB AXIS IN REF COOR
 QTEMP
TSTRXUT DLOAD BHIZ
 36D
 BADVCTOR
VLOAD RVQ
 6D
BADVCTOR VLOAD UNIT
 RTIG
PDVL UNIT
Page 717
 VTIG
VSR3 VAD
VXV UNIT
 UT
VCOMP
STORE 6D
RVQ
TRIMSCAL 2DEC 1.07975111 B-1
YBIAS 2DEC +.00263888889 # YAW MECH BIAS (+0.95 DEG, THRUST ON)
PBIAS 2DEC -.00597222222 # PITCH MECH BIAS (-2.15 DEG, THRUST ON)
 # REFERENCE, TRW 68.6520.3.3-40 27 FEB, 1968
PROGRAM DESCRIPTION S41.1 DATE 8 DEC 66
MOD NO 1 LOG SECTION P40-P47
MOD BY ZELDIN
#
FUNCTIONAL DESCRIPTION
#
COMPUTE VELOCITY TO BE GAINED INITIALLY IN REF COORDS.
TO CONTROL COORDS.
#
CALLING SEQUENCE
```

```

#
L CALL
L+1 S41.1
#
NORMAL EXIT MODE
#
AT L +2 OF CALLING SEQUENCE
#
SUBROUTINES CALLED:
#
CALCSMSC
CDUTRIG
#
ALARM OR ABORT MODES
#
NONE
#
ERASABLE INITIALIZATION REQUIRED
#
VG IN REF. COORD. PDL L POINTER AT L+5. S41.1 WILL RETURN WITH
POINTER AT L (L MUST BE LESS THAN OR = TO 14D)
#
OUTPUT
#
MPAC CONTAINS VG IN CONTROL COORDS VECT. B7 M/CS
#
DEBRIS:
#
QTEMP TEMP ERASABLE
QPRET
#
COUNT 22/S41.1
#
SETLOC P40S5
BANK
#
S41.1 STQ CALL
QTEMP
CDUTRIG
#
VLOAD
MXV CALL
REFSMMAT
SMNB
#
Page 718
MXV VXSC
QUADROT

```

July 12, 2016

1279

```
TENBNK14 TENBNK14 # VG IN CONTROL COORD IN MPAC SCALED AT
VSL5 GOTO # VG IN CONTROL COORDS. IN MPAC AT 2(+7)
 QTEMP
TENBNK14 2DEC 10. B-4

Page 719
NAME S40.8 -- CROSS PRODUCT STEERING
FUNCTION (1) UPDATES THE VELOCITY-TO-BE-GAINED VECTOR.
(2) GENERATES ANGULAR RATE STEERING COMMANDS FOR AUTOPILOT.
(3) ESTABLISHES ENGINE CUT-OFF SIGNALS AT APPROPRIATE TIMES.
(4) INITIATES THRUST-FAIL ROUTINE, R40
CALLING SEQ CALL S40.6
INPUT VGPREV LAST VALUE OF THE VELOCITY-TO-BE-GAINED VECTOR
DELVREF CHANGE IN VEHICLE VELOCITY SINCE LAST MEASUREMENT
IN METERS/CS AT +7.
BDT EFFECT OF RATE OF CHANGE OF REQUIRED VELOCITY AND
GRAVITY DURING DT UPON VELOCITY-TO-BE-GAINED IN
METERS/CS AT +7.
CSTEER A SCALAR OF THE STEERING LAW, SC.AT B+1, USED FOR
SPS AIMPOINT STEERING MANEUVERS.
IDLEFAIL A FLAG TO INHIBIT (IDLE) THE THRUST-FAIL ROUTINE.
STEERSW A SWITCH TO PRECLUDE NEEDLESS CONDUCT OF STEERING.
REFMMAT, DAPDATR1, PIPTIME
EREFFRAC, ETDECAY, KPRIMEDT FOR TVC.
OUTPUT TTOGO TIME REMAINING FOR ENGINE BURN IN CS AT +28.
OMEGAC DP VECTOR RATE COMMAND, SC.AT 1/(2TVCDT) REVS/SEC.
VG, VGPREV, VGDISP, TGO, TIG, SCALED AS NOTED IN CODING
STEERSW, IMPULSW, NVWORD1
REPFRAC, CNTR, VCNTR, VCNTRTMP FOR TVC (R40 INTERFACING)
DEBRIS OMEGAXC, +1
SUBROUTINES USED: *SMNB*, ALARM

 SETLOC P40S1
 BANK
 EBANK= DAPDATR1
 COUNT 16/S40.8

S40.8 SETPD STQ
SPBIT1 OOD
 QTEMP
 VLOAD BVSU # CONSTRUCT DELVG, SC.AT B+7 M/CS
 DELVREF
 BDT
 VAD
 VGPREV
```

# Page 720

TGOALC

|       |           |                                               |
|-------|-----------|-----------------------------------------------|
| STORE | VG        | # VELOCITY-TO-BE-GAINED, SC.AT B+7 M/CS       |
| ABVAL |           |                                               |
| STORE | VGDISP    | # FOR DISPLAY PURPOSES                        |
| EXIT  |           |                                               |
| TC    | PHASCHNG  |                                               |
| OCT   | 10035     | # TYPE B RESTART RESTART BELOW AND 5.3 REREAD |
| TC    | INTPRET   |                                               |
| VLOAD |           |                                               |
|       | VG        |                                               |
| STORE | VGPREV    |                                               |
| BOFF  | VLOAD     |                                               |
|       | STEERSW   | # SKIP TGO AND CROSS-PRODUCT                  |
|       | QTEMP     |                                               |
|       | DELVREF   |                                               |
| ABVAL | PUSH      | # CHECK FOR LOTHURST                          |
| SLOAD | DMP       |                                               |
|       | DVTHRESH  | # SC.AT B-2 M/CS                              |
|       | DPB-9     |                                               |
| BDSU  |           |                                               |
| BMN   | EXIT      |                                               |
|       | LOTHRUST  |                                               |
| CAE   | DAPDATR1  | # ENABLE TVCDAP CG TRACKING                   |
| MASK  | BIT14     |                                               |
| CCS   | A         |                                               |
| CAF   | BIT1      |                                               |
| INDEX | A         | # LM-OFF, LM-ON VALUE                         |
| CAE   | EREPPFRAC |                                               |
| TS    | REPFRAC   |                                               |
| TC    | INTPRET   |                                               |
| VLOAD | BVSU      | # GET DELVG                                   |
|       | DELVREF   |                                               |
|       | BDT       |                                               |
| UNIT  |           |                                               |
| DOT   | PUSH      | # (OOD)                                       |
|       | VG        |                                               |
| BPL   | DDV       | # ANGLE SHOULD BE GREATER THAN PI/2           |
|       | INCRSVG   | # DISPLAY ALARM IF NOT                        |
|       | 2VEXHUST  |                                               |
| DAD   | DMP       | # (DOT PRODUCT UP FROM OOD)                   |
|       | LODPHALF  |                                               |
| NORM  | SR1       |                                               |
|       | X1        |                                               |



July 12, 2016

1281

```

PDDL NORM
 36D # (MAG DELVG)
 X2
BDDV
XSU,2 SL*
 X1
 0 -9D,2
DMP PUSH # (OOD)
 -FOURDT
SLOAD SR
 ETDECAY # ETDECAY SC.AT B+14 CS
 14D
BDSU STADR
STORE TGO # TIME TO GO IN CS. AT +28
DAD

Page 721

 PIPTIME
STODL TIG
 TGO
DSU BMN
 FOURSEC
 S40.81

XPRODUCT VLOAD VXSC
 BDT
 CSTEER
VSL2 VSU
 DELVREF
UNIT PDVL
 VG
UNIT VXV
MXV CALL
 REFSMMAT # (REFSMMAT/2)
 SMNB
VXSC
 KPRIMEDT # (KPRIMEDT SC.AT PI/8 RAD)
OMEGACLC STORE OMEGAC
 GOTO
 QTEMP

 SETLOC DAPS7
 BANK
COUNT 17/S40.8

TWODT 2DEC 200.0 B-28 # 2 SEC
```

|            |        |               |                                            |
|------------|--------|---------------|--------------------------------------------|
| -FOURDT    | 2DEC   | -800 B-18     | # -4(200CS), SC.AT B+18CS (-4 FOR SCALING) |
| 2VEXHUST   | 2DEC   | 63.020792 B-7 | # 2(10338.0564 FPS), SC.AT B+7 M/CS        |
| FOURSEC    | 2DEC   | 400.0 B-28    | # 4 SEC                                    |
| DPB-9      | 2DEC   | 1 B-9         |                                            |
|            | SETLOC | DAPS6         |                                            |
|            | BANK   |               |                                            |
|            | COUNT  | 20/S40.8      |                                            |
| S40.81     | SET    | VLOAD         | # TGO LESS THAN 4 SECONDS                  |
|            |        | IMPULSW       | # FOR ENGINE-OFF CALL                      |
|            |        | HI6ZEROS      |                                            |
| RATEZRO    | STORE  | OMEGAC        | # TVC TO ATTITUDE HOLD                     |
|            | EXIT   |               |                                            |
|            | CAF    | POSMAX        | # INHIBIT SWITCHOVER/TVC EG TRACKING       |
|            | TS     | CNTR          |                                            |
| # Page 722 | TC     | INTPRET       |                                            |
|            | CLEAR  | GOTO          |                                            |
|            |        | STEERSW       | # RESTARTS OK                              |
|            |        | QTEMP         |                                            |
| INCRSVG    | EXIT   |               | # ALARM INDICATING THAT THRUST IS POINTING |
|            | TC     | ALARM         | # IN WRONG DIRECTION.                      |
|            | OCT    | 01407         |                                            |
|            | TC     | INTPRET       |                                            |
|            | GOTO   |               |                                            |
|            |        | QTEMP         |                                            |
| LOTHRUST   | BON    | VLOAD         | # THRUST FAILURE (LO-OR-NO) INDICATED      |
|            |        | IDLEFAIL      | # SET BY V97P. ALLOWS 1 BYPASS IN CASE OF  |
|            |        | QTEMP         | # UNFAVORABLE S40.8 SYNCH.                 |
|            |        | HI6ZEROS      | # START OF ENGINE-FAIL (R40) OPERATIONS    |
|            | STORE  | OMEGAC        | # PUT TVC IN ATTITUDE HOLD                 |
|            | EXIT   |               |                                            |
|            | CS     | ZERO          |                                            |
|            | TS     | VCNTR         | # KILL CSMMASS UPDATING                    |
|            | TS     | VCNTRTMP      | # (TVCEXEC LOGIC REQUIRES THIS TOO)        |
|            | TS     | REPFRAC       | # KILL TVCDAP CG TRIM TRACKING             |
|            | TS     | NVWORD1       | # SET UP ENGINE-FAIL V97FLASH (CLOCKJOB)   |
|            | TC     | INTPRET       |                                            |

```

CLEAR GOTO # INHIBIT STEERING AND TGO CALC (MANUAL
 STEERSW # SHUTDOWN IF NOT SET UP AGAIN)
 QTEMP # RESTARTS OK

```

```
Page 723
```

```

NAME S40.9 -- VTOGAIN (AIMPOINT MANEUVERS ONLY)
FUNCTION (1) GENERATES REQUIRED VELOCITY AND VELOCITY-TO-BE-GAINED
VECTORS FOR USE DURING AIMPOINT MANEUVERS.
(2) UPDATES THE B VECTOR WHICH IS USED IN THE FINAL
CALCULATION OF EXTRAPOLATING THE VELOCITY-TO-BE-GAINED.
CALLING SEQ VIA FINDVEC AS NEW JOB.
INPUT RNIT ACTIVE VEHICLE RADIUS VECTOR IN METERS AT +29.
VNIT ACTIVE VEHICLE VELOCITY VECTOR IN METERS/CS AT +7.
VRPREV LAST COMPUTED VELOCITY REQUIRED VECTOR IN
METERS/CS AT +7.
NONTIG TIME OF IGN. USED IN TARGETTING ROUTINES B+28
DELLT4 TRANSFER TIME FROM PIPTIME TO TARGET B+28
TNIT TIME OF RNIT AND VNIT IN CS AT +28
GDT/2 HALF OF VELOCITY GAINED IN DELTA T TIME DUE TO
ACCELERATION OF GRAVITY IN METERS/CS AT +7.
DELVREF CHANGE IN VELOCITY DURING LAST 2 SEC IN
METERS/CS AT +7.
NORMSW SET=CENTRAL ANGLE BETWEEN RTARG AND RTIG IS BETWEEN
165 TO 195 DEGREES.
RESET=CENTRAL ANGLE OUTSIDE CONE DESCRIBED ABOVE.
OUTPUT VGTEMP VELOCITY TO BE GAINED VECTOR IN METERS/CS AT +7.
COGA INPUT OF INITIAL GUESS FOR LAMBERT FROM S40.1
OR PREVIOUS PASS THRU S40.9.
GOBL/2 OBLATENESS TERM IN AVG GRAV CALC: GOBL*RSQ/MU
VRPREV/ VELOCITY REQUIRED VECTOR IN METERS/CS AT +7.
BDT B VECTOR IN METERS/CS AT +7.
SUBROUTINES USED -- INITVEL

```

```

SETLOC P40S1
BANK

```

```

EBANK= NBRCYCLS
COUNT 16/S40.9

```

```

S40.9 TC INTPRET
 SETPD DLOAD
 OOD
 LO6ZEROS
 PDDL
 EP4(45)L
 BON DLOAD

```

|            |       |          |                                          |
|------------|-------|----------|------------------------------------------|
|            |       | NORMSW   |                                          |
|            |       | +2       |                                          |
|            |       | EP4(10)L |                                          |
|            | PUSH  |          |                                          |
|            | CLEAR | CALL     |                                          |
|            |       | GUESSW   |                                          |
|            |       | HAVEGUES |                                          |
|            | EXIT  |          |                                          |
|            | TC    | PHASCHNG | # SAVE TIME BY NOT REDOING LAMBERT CALCS |
|            | OCT   | 05021    | # C, PRIORITY NEXT, JOB BELOW            |
| # Page 724 |       |          |                                          |
|            | OCT   | 10000    |                                          |
|            | TC    | INTPRET  |                                          |
| ENDLAMB    | BON   |          |                                          |
|            |       | FIRSTFLG |                                          |
|            |       | FIRSTTME |                                          |
|            | VLOAD | VSU      |                                          |
|            |       | VIPRIME  |                                          |
|            |       | VRPREV   |                                          |
|            | PDDL  | DSU      |                                          |
|            |       | TNIT     |                                          |
|            |       | TNITPREV |                                          |
|            | SL    | BDDV     |                                          |
|            |       | 17D      |                                          |
|            |       | 200CSHI  |                                          |
|            | VXSC  |          |                                          |
|            | VSU   | VSL1     |                                          |
|            |       | GDT/2    |                                          |
|            | STORE | BDT      |                                          |
| FIRSTTME   | SLOAD | DCOMP    |                                          |
|            |       | RTX2     |                                          |
|            | BMN   |          |                                          |
|            |       | MOONCASE |                                          |
|            | VLOAD | UNIT     |                                          |
|            |       | RN       |                                          |
|            | DLOAD | DSU      |                                          |
|            |       | PIPTIME  |                                          |
|            |       | NOMTIG   |                                          |
|            | DMP   | DDV      |                                          |
|            |       | EARTHMU  |                                          |
|            |       | 34D      |                                          |
|            | VXSC  | VAD      |                                          |
|            |       | GOBL/2   |                                          |
|            |       | VGTEMP   |                                          |
|            | STORE | VGTEMP   | # NOTE: NO TEST IS MADE TO SUBTRACT GOBL |
| MOONCASE   | EXIT  |          | # INSIDE 165-195 DEGREE CONE AREA.       |

July 12, 2016

1285

```

TC PHASCHNG
OCT 04021 # C, JOB BELOW

COPY40.9 TC INTPRET
 DLOAD
 TNIT
 STOVL TNITPREV
 VIPRIME
 STORE VRPREV
 CLEAR EXIT
 FIRSTFLG
 -2 CS ONE # RED040.9 (RESTART) ENTRY TO END S40.9
 TS NBRCYCLS
ENDS40.9 TC PHASCHNG
 OCT 00001

Page 725 TCF ENDOFJOB

RED040.9 TC INTPRET # S40.9 RESTARTS COME HERE TO GRACEFULLY
 VLOAD # TERMINATE S40.9 SO THAT IT CAN BE
 LO6ZEROS # SET UP WITH LATEST R,V,T NEXT PASS
 STODL DELVSUM # (TYPE C PHASE POINTS '04021' WILL
 LO6ZEROS # FORCE NORMAL S40.9 TERMINATIONS,
 STOVL NBRCYCLS # RATHER THAN LOSE TIME OF BRAND NEW
 VGPREV # PASS -- QUICK OLD DATA BETTER THAN
 STORE VGTEMP # NONE) NOW CAN GO THRU SETUP.9
 EXIT # WITHOUT DISTURBING VGPREV.
 TCF ENDS40.9 -2 # STORE 0,0 COVERED NBRCYCLS,P -- FIX UP S

200CSHI 2DEC 200 B-12

EARTHMU 2DEC* -3.986032 E10 B-36*

EP4(45)L 2DEC .125

EP4(10)L 2DEC .027777777

Page 726
NAME: S40.13 -- TIMEBURN
#
FUNCTION (1) DETERMINE WHETHER A GIVEN COMBINATION OF VELOCITY-TO-
BE-GAINED AND ENGINE CHOICE RESULT IN A BURN TIME SUFFICIENT
TO ALLOW STEERING AT THE VEHICLE DURING THE BURN, AND
(2) THE MAGNITUDE OF THE RESULTING BURN TIME -- IF IT IS SHORT --
AND THE ASSOCIATED TIME OF THE ENGINE OFF SIGNAL.
#
```

```

CALLING SEQUENCE VIA FINDVAC AS A NEW JOB
#
INPUT VGTIG -- VELOCITY TO BE GAINED VECTOR (METERS/CS) AT +7
WEIGHT/G -- MASS OF VEHICLE IN KGM AT TIG
F -- ENGINE THRUST IN M.NEWTONS AT +7
MDOT -- RATE OF DECREASE OF VEHICLE MASS DURING ENGINE BURN
IN KILOGRAMS/CENTISECOND AT +3. THIS SCALING MAY
REQUIRE MODIFICATION FOR SATURN BURNS.
#
OUTPUT IMPULSW ZERO FOR STEERING
ONE FOR ATTITUDE HOLD
TGO TIME TO BURN IN CENTISECONDS AT +14
THE QUANTITY M.NEWTON SHALL BE USED TO EXPRESS WEIGHT IN TER
(KILOGRAM*METER)/(CENTISECOND*CENTISECOND)
(1) M.NEWTON = (10000) NEWTONS.

EBANK= TGO
COUNT 16/40.13

S40.13 TC INTERPRET
 SETPD SET
 OOD
 IMPULSW # ASSUME NO STEERING UNTIL FOUND OTHERWISE
 VLOAD ABVAL
 VGTIG # VELOCITY TO BE GAINED AT +7
 EXIT
 CAF BIT7 # TEST +X TRANSLATION
 EXTEND
 RXOR CHAN31
 MASK BIT7
 EXTEND
 BZF NOTADDUL
 TC INTERPRET
 PDDL DDV # OOD = MAG OF VGTIG AT +7
 S40.135 # COMPENSATION FOR 2 JET ULLAGE AT +24
 WEIGHT/G # MASS IN KGMS AT +16
 BON SL1 # DOUBLE CORRECTION IF FOUR JETS
 NJETSFLG
 S40.130

S40.130 BDSU
 PDDL DDV # OOD = MAG OF VGTIG CORRECTED FOR ULLAGE
 K1VAL # M.NEWTON-CS AT +24
 WEIGHT/G
 BDSU BMN
 OOD
 S40.131 # TGO LESS THAN 100 CS

```

July 12, 2016

1287

# Page 727

|          |        |            |                                            |
|----------|--------|------------|--------------------------------------------|
|          | PDDL   | DMP        | # 02D = TEMP1 AT +7                        |
|          |        | EMDOT      | # SPS FLOW RATE SC.AT B+3 KG/CS (SP, NOTE) |
|          |        | 3.5SEC     | # 350 CS AT +14                            |
|          | BDSU   | PDDL       |                                            |
|          |        | WEIGHT/G   |                                            |
|          |        | FANG       |                                            |
|          | DMP    | SR2        |                                            |
|          |        | 5SECOND    | # 500 CS AT +14                            |
|          | DDV    | PUSH       | # 04D = TEMP2                              |
|          | BDSU   | BPL        |                                            |
|          |        | 02D        |                                            |
|          |        | S40.133    | # TGO GREATER THAN 600 CS                  |
|          | DLOAD  | BDDV       |                                            |
|          | DMP    | DAD        |                                            |
|          |        | 5SECOND    | # 500 CS AT +14                            |
|          |        | 1SEC2D     | # 100 CS AT +14                            |
|          | GOTO   |            |                                            |
|          |        | S40.132    |                                            |
| S40.131  | DLOAD  | DMP        | # TGO LESS THAN 100 CS                     |
|          |        | WEIGHT/G   |                                            |
|          | DAD    | DDV        |                                            |
|          |        | K2VAL      | # M.NEWTON CS AT +24                       |
|          |        | K3VAL      | # M.NEWTON AT +10                          |
| S40.132  | EXIT   |            |                                            |
|          | EBANK= | TGO        |                                            |
|          | TC     | TPAGREE    |                                            |
|          | CA     | MPAC       |                                            |
|          | XCH    | L          |                                            |
|          | CA     | ZERO       |                                            |
|          | DXCH   | TGO        | # TGO IN CS AT +28                         |
|          | TC     | S40.134    |                                            |
| S40.133  | CLEAR  | EXIT       | # WILL STEER VEHICLE                       |
|          |        | IMPULSW    |                                            |
| S40.134  | TC     | PHASCHNG   | # KILL GROUP 3                             |
|          | OCT    | 3          |                                            |
|          | TCF    | ENDOFJOB   |                                            |
| NOTADDUL | TC     | INTPRET    |                                            |
|          | GOTO   |            |                                            |
|          |        | S40.130 +1 | # DO NOT COMPENSATE FOR 7 SEC OF ULLAGE    |
|          | SETLOC | DAPS7      |                                            |
|          | BANK   |            |                                            |
|          | COUNT  | 17/40.13   |                                            |

```

K1VAL = EK1VAL # DP PAD LOAD B+23 NEWTON-SEC/E+2
K2VAL = EK2VAL # DP PAD LOAD B+23 NEWTON-SEC/E+2
K3VAL = EK3VAL # DP PAD LOAD B+09 NEWTONS/E+4
1SEC2D 2DEC 100.0 B-14 # 100.0 CS AT +14
Page 728
3.5SEC 2DEC 350.0 B-13 # 350 CS AT +13

5SECOND 2DEC 500.0 B-14 # 500.0 CS AT +14

S40.135 2DEC 69.6005183 B-23 # IMPULSE FROM 7.96 SECS OF 2-JET FIRING
7.96 (199.6)COS(10) LB-SEC, SC.AT
B+23 NEWTON-SEC/E+2 (7 SEC ULLAGE
TO GO, PLUS 0.96 SEC FROM PIPTIME)

Page 729
NAME S40.6 GIMBAL DRIVE TEST AND/OR GIMBAL TRIM
MOD NO 5 DATE 9 MARCH, 1967
MOD BY ENGEL LOG SECTION P40-P47
#
FUNCTIONAL DESCRIPTION
GIMBAL DRIVE TEST....0,+2,-2,0 DEGREE ENGINE COMMANDS, AT 2 SECOND
INTERVALS, FIRST IN PITCH, THEN IN YAW. ASTRONAUT VERIFICATION
OF GIMBAL MOTION ON GPI
GIMBAL TRIM....AFTER A 4 SECOND DELAY, ENGINE COMMANDED TO
PRE-COMPUTED TRIM POSITION. ASTRONAUT VERIFICATION ON GPI.
PRE40.6....RESTART ENTRY TO RE-DO S40.6, ONLY IF RCS IS ON --- IF TVC
IS NOT ON --- PRIMARILY TO GET ACTUATORS TRIMMED FOR IGNITION.
BYPASS 4 SEC DELAY. SPEED IS CRITICAL NEAR IGNITION.
IF TVC IS ON (TVCDAPON OR LATER) THEN REDOTVC WILL TAKE CARE
OF RESTARTING ACTUATORS.
#
CALLING SEQUENCE....
WAITLIST, WITH 2CADR FOR S40.6 (OR PRE40.6), WITH EBANK= CNTR
#
NORMAL EXIT MODE -- FIXDELAY, TASKOVER
#
SUBROUTINES CALLED....
OUTPUT (INTERNAL)
FIXDELAY
#
ALARM OR ABORT EXIT MODES --- NONE
#
ERASABLE INITIALIZATION REQUIRED
CNTR = +0, NORMALLY SET BY THE P40 CALL AT TST,TRIM.
MRKRTMP....POSITIVE FOR GIMBAL DRIVE TEST AND GIMBAL TRIM (BOTH)

```



July 12, 2016

1289

```
NEGATIVE FOR GIMBAL TRIM ONLY
PACTOFF, YACTOFF SC.AT 85.41 ARCSEC/BIT (V48N48 P, YTRIM)
"SC CONT" SWITCH AT "CMC" (A/P CONTROL SWITCH AT "GNC")
ACTIVE SPS GIMBAL MOTOR POWER(S), PITCH, YAW
#
OUTPUT
TVCYAW, TVCPITCH (BITS RELEASED)
TVC ENABLE AND OPTICS ERROR COUNTER ENABLE
#
DEBRIS
TBMPR60, CNTR

 BANK 17
 SETLOC DAPS6
 BANK

 EBANK= CNTR
 COUNT 20/S40.6

PRE40.6 CS FLAGWRD6 # RESTART ENTRY TO S40.6 (DO NOT PERMIT
 MASK OCT60000 # IF TVC, BITS 15,14 = 1,0)
 EXTEND
 BZMF +2
 TCF TASKOVER # TVC, REDOTVC WILL REESTABLISH INTERFACE

 CS BIT1 # RCS, SO DO S40.6, GIMTRIM ONLY
Page 730
 TS MRKRTMP

 CAF BIT1 # FOR REVISED S40.6 TIMING FOR RESTARTS...
 TS CNTR # TO INDICATE A RESTART ENTRY (CNTR 1S
 # NORMALLY +0, BY S40.6)

 EBANK= CNTR
S40.6 CS ZERO # INHIBIT OPTICS ACTIVITY
 TS OPTIND

 CS BIT2 # DISENABLE OPTICS ERROR COUNTERS (ZERO,
 EXTEND # AND INHIBIT PULSE TRANSMISSION --
 WAND CHAN12 # NORMAL STATE)

 CAF OCT02200 # TVC ENABLE (SPS SERVO AMPS SEE DAC
 EXTEND # VOLTAGES) AND DISENGAGE OPTICS/DAC
 WOR CHAN12

 TC FIXDELAY # 60MS PROCEDURAL DELAY (40MS MINIMUM) FOR
```

|            |        |            |                                           |
|------------|--------|------------|-------------------------------------------|
|            | DEC    | 6          | # RELAY LATCHING                          |
|            | CAF    | BIT2       | # ENABLE OPTICS ERROR COUNTERS            |
|            | EXTEND |            |                                           |
|            | WOR    | CHAN12     |                                           |
|            | TC     | FIXDELAY   | # 20MS PROCEDURAL DELAY (4MS MINIMUM) FOR |
|            | DEC    | 2          | # RELAY LATCHING                          |
| RSTRTST    | CCS    | CNTR       | # CHECK FOR RESTART ENTRY (PRE40.6)       |
|            | TCF    | GIMTRIM +2 | # RESTART ENTRY...BYPASS 4 SECOND DELAY   |
|            |        |            | # TST,TRIM SETS +0 ON NORMAL ENTRY        |
|            | CAE    | MRKRTMP    | # CHECK FOR TEST/TRIM OR TRIM ONLY        |
|            | TS     | CNTR       | # MRKRTMP SAVES CNTR FOR RESTARTS         |
|            | EXTEND |            |                                           |
|            | BZMF   | GIMTRIM    | # (TRIM ONLY)                             |
| GDTSETUP   | CS     | ZERO       | # GIMBAL DRIVE TEST SETUP, FOR PITCH      |
|            | TS     | CNTR       |                                           |
| GIMDTEST   | CAF    | +2ACTDEG   | # GIMBAL DRIVE TEST, 1ST INCREMENT        |
|            | TC     | OUTPUT     | # (LEAVES GIMBAL AT +2 DEG)               |
|            | CAF    | -4ACTDEG   | # 2ND INCREMENT (LEAVES GIMBAL AT -2)     |
|            | TC     | OUTPUT     |                                           |
|            | CAF    | +2ACTDEG   | # 3RD INCREMENT (LEAVES GIMBAL AT -0)     |
|            | TC     | OUTPUT     |                                           |
|            | CS     | CNTR       | # CHECK FOR COMPLETION OF YAW TEST.       |
| # Page 731 | CCS    | A          |                                           |
|            | TCF    | GIMTRIM    | # COMPLETED, GO TO GIMBAL TRIM ROUTINE    |
|            | CS     | BIT1       | # SET UP YAW TEST                         |
|            | TS     | CNTR       |                                           |
|            | TCF    | GIMDTEST   | # FOR YAW TEST                            |
| OUTPUT     | EXTEND |            | # OUTPUT THE INCREMENT....SAVE Q          |
|            | QXCH   | TEMPR60    |                                           |
|            | INDEX  | CNTR       |                                           |
|            | TS     | TVCPITCH   |                                           |
|            | INDEX  | CNTR       |                                           |
|            | CAF    | BIT11      |                                           |
|            | EXTEND |            |                                           |
|            | WOR    | CHAN14     |                                           |

July 12, 2016

1291

```

TC FIXDELAY # WAIT 2SEC, WHILE ASTRONAUT VERIFIES
DEC 200 # GIMBAL MOTION ON GPI
TC TEMPR60

GIMTRIM TC FIXDELAY # WAIT 4 SECONDS BEFORE GIMBAL TRIM
DEC 400

 +2 CS ZERO # PICK UP TRIM VALUES AND OUTPUT THEM
 AD PACTOFF # (AVOID +0) ENTRY POINT FROM RSTRTST
 TS TVCPITCH # ON A RESTART, TO AVOID 4SEC DELAY
 CS ZERO
 AD YACTOFF
 TS TVCYAW

CAF PRI06 # RELEASE THE COUNTERS, BITS 11,12
EXTEND
WOR CHAN14

ENDS40.6 TCF TASKOVER

OCT02200 OCT 02200 # BITS 8,11 FOR CHANNEL 12 TVC/OPTICS
-4ACTDEG DEC -168 # -2(+2ACTDEG), WHOLE BITS, NO ROUNDUP
+2ACTDEG DEC +84 # +2 DEG, SC.AT 85.41 ARCSEC/BIT (+84D)

CALLED BY "DONOUN46" (VERB 48), OR DIRECTLY BY "FRESHDAP" (RCS DAP) VIA IBNKCALL

COUNT 20/S41.2

S41.2 CA DAPDATR1
Page 732

MASK THREE
AD A
TS RATEINDX

INHINT
CAE DAPDATR1 # IS LEM ATTACHED (BITS 14,13 OF DAPDATR1
MASK PRI030 # =10)
AD -BIT14 # (OCT57777)
EXTEND
BZF TOGETHER # YES

CS BIT2 # NO, UNSET FLAG
MASK FLAGWRD7
TS FLAGWRD7
```

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | TCF    | +4       |                                            |
| TOGETHER   | CS     | FLAGWRD7 | # ATTACHED, SET FLAG FOR INTEGRATION       |
|            | MASK   | BIT2     |                                            |
|            | ADS    | FLAGWRD7 |                                            |
|            | RELINT |          |                                            |
|            | CA     | DAPDATR1 |                                            |
|            | MASK   | BIT4     |                                            |
|            | EXTEND |          |                                            |
|            | BZMF   | +2       | # DEC 46 MEANS NARROW DB                   |
|            | CA     | DEC409   |                                            |
|            | AD     | DEC46    | # DEC 455 MEANS WIDE DB                    |
|            | TS     | ADB      |                                            |
|            | CA     | DAPDATR1 |                                            |
|            | MASK   | BIT7     | # QUAD BD                                  |
|            | EXTEND |          |                                            |
|            | BZMF   | +2       |                                            |
|            | CA     | ONE      |                                            |
|            | TS     | XTRANS   |                                            |
|            | CA     | DAPDATR1 |                                            |
|            | MASK   | BIT10    | # QUAD AC                                  |
|            | EXTEND |          |                                            |
|            | BZMF   | +2       |                                            |
|            | CS     | ONE      |                                            |
|            | ADS    | XTRANS   |                                            |
|            | INHINT |          |                                            |
|            | EXTEND |          |                                            |
|            | BZF    | +5       | # CLEAR NJETSFLG (4 JETS, OR NO JETS)      |
|            | CS     | FLAGWRD1 | # SET NJETSFLG (2 JETS, AC OR BD QUADS)    |
|            | MASK   | BIT15    | # NJETSFLG = 1 FOR 2 JET ULLAGE (AC OR BD) |
|            | ADS    | FLAGWRD1 |                                            |
| # Page 733 | TCF    | +4       |                                            |
|            | CS     | BIT15    | # KJETSFLG = 0 FOR 4 JET (OR 0 JET) ULLAGE |
|            | MASK   | FLAGWRD1 |                                            |
|            | TS     | FLAGWRD1 |                                            |
|            | RELINT |          |                                            |
|            | CA     | DAPDATR2 |                                            |
|            | MASK   | BIT13    |                                            |
|            | EXTEND |          |                                            |
|            | BZMF   | +2       |                                            |
|            | TCF    | +2       |                                            |

July 12, 2016

1293

|        |      |          |                               |
|--------|------|----------|-------------------------------|
|        | CS   | ONE      |                               |
|        | COM  |          |                               |
|        | TS   | ACORBD   | # MINUS FOR A-C, PLUS FOR B-D |
|        | CA   | DAPDATR2 |                               |
|        | MASK | BIT10    |                               |
|        | CCS  | A        |                               |
|        | TCF  | +4       |                               |
|        | CA   | ONE      |                               |
|        | TS   | RACFAIL  |                               |
|        | TCF  | BDFAIL   |                               |
|        | CA   | ZERO     |                               |
|        | TS   | RACFAIL  |                               |
|        | CA   | DAPDATR2 |                               |
|        | MASK | BIT4     |                               |
|        | CCS  | A        |                               |
|        | TCF  | BDFAIL   |                               |
|        | CS   | ONE      |                               |
|        | TS   | RACFAIL  |                               |
| BDFAIL | CA   | DAPDATR2 |                               |
|        | MASK | BIT7     |                               |
|        | CCS  | A        |                               |
|        | TCF  | +4       |                               |
|        | CA   | ONE      |                               |
|        | TS   | RBDFAIL  |                               |
|        | TC   | Q        |                               |
|        | CA   | ZERO     |                               |
|        | TS   | RBDFAIL  |                               |
|        | CA   | DAPDATR2 |                               |
|        | MASK | BIT1     |                               |
|        | CCS  | A        |                               |
|        | TC   | Q        |                               |
|        | CS   | ONE      |                               |
|        | TS   | RBDFAIL  |                               |
|        | TC   | Q        |                               |

# DAPFIG ENTRY VIA TC POSTJUMP AS JOB FROM "STABLISH" (VERB 46)

|            |        |        |
|------------|--------|--------|
|            | BANK   | 42     |
|            | SETLOC | EXTVBS |
| # Page 734 | BANK   |        |

|        |        |        |                          |
|--------|--------|--------|--------------------------|
| DAPFIG | CS     | BIT9   | # TURN OFF SIVB TAKEOVER |
|        | EXTEND |        |                          |
|        | WAND   | CHAN12 |                          |

```

CAE DAPDATR1 # DETERMINE VEHICLE CONFIGURATION
EXTEND
MP BIT3 # RIGHT SHIFT 4 OCTAL DIGITS
MASK THREE # (IN CASE BIT 15 IS USED)
INDEX A
TCF +1 # BRANCH BASED ON CONFIG....

TCF NODAPUP # CM.....ACTIVATE NODAP
TCF RCSDAPUP # CSM.....ACTIVATE RCSDAP
TCF RCSDAPUP # CSM/LEM..ACTIVATE RCSDAP
TC POSTJUMP
CADR SATSTKON
INHINT
RCSDAPUP INHINT # CALL TO ACTIVATE RCSDAP, AND RETURN
TCR IBNKCALL
CADR RCSDAPON
RELINT
TCF ENDFIG # CAME IN VIA V46, GO OUT VIA GOPIN
EXTEND
NODAPUP DCA T5IDLDAP
DXCH T5LOC
TC DOWNFLAG # RESET T5-USAGE FLAGS FOR NODAP
ADRES DAPBIT1 # BIT 15 FLAG 6 = 0
TC DOWNFLAG
ADRES DAPBIT2 # BIT 14 FLAG 6 = 0
INHINT
TC IBNKCALL # ZERO JET CHANNELS IN 14 MS AND THEN
CADR ZEROJET # LEAVE THE T6 CLOCK DISABLED.
RELINT
CAF BIT1 # KILL KALCMANU JOB
TS HOLDFLAG
TC POSTJUMP # CAME IN VIA V46, GO OUT VIA GOPIN
CADR GOPIN
SBANK= PINSUPER # Added by RSB 2009
EBANK= PACTOFF
T5IDLDAP 2CADR T5IDLOC

SBANK= LOWSUPER
BANK 17
SETLOC DAPS6
BANK

DEC409 DEC 409
DEC46 DEC 46

```

July 12, 2016

1295

# CALLED BY "DONOUN47" (VERB 48), OR DIRECTLY BY "FRESHDAP" (RCS DAP)

S40.14 CAE IXX # RCS ENTRY

EXTEND

MP CONTONE

TS J/M

CA IAVG

EXTEND

MP CONTONE

TS J/M1

TS J/M2

EXTEND

DCA CONTTWO

EXTEND

DV IXX

TS KMJ

EXTEND

DCA CONTTWO

EXTEND

DV IAVG

TS KMJ1

TS KMJ2

TC Q

CONTONE DEC .662034 # 2PI/M

CONTTWO 2DEC .00118

COUNT 24/TVNG

BANK 31

SETLOC P40S

BANK

POS-2.5 OCT 37405

EBANK= DAPDATR1

RCSCADR 2CADR RCSUP

6SECT5 OCT 37704

COUNT 21/RCSUP

BANK 20

```

 SETLOC DAPS3
 BANK

Page 736

RCSUP LXCH BANKRUPT
 EXTEND
 QXCH QRUPT

 TCR RCSDAPON # ACTIVATE RCS DAP
 TCF RESUME

 EBANK= DAPDATR1
RCSADDR 2CADR RCSATT

0.6SECT5 OCT 37704

RCSDAPON CAF 0.6SECT5 # RCSDAPON ENTRY MUST BE UNDER INT-INHIBIT
+1 TS TIME5 # 0.6 SEC ALLOWS TVCEXEC/ROLLDAP TO DIE
 TS T5PHASE # ENTRY FROM ROOTOP00
 # WILL CAUSE FRESHDAP (+1)

 CS RCSFLAGS # SET BIT3 TO REINITIALIZE FDAI ERROR
 MASK BIT3 # DISPLAY, IN CASE SC CONT SWITCH
 ADS RCSFLAGS # IN SCS NOT GNC (GUIDEMODE PRIMARY)

 EXTEND
 DCA RCSADDR # (RCSATT)
 DXCH T5LOC

 CS OCT60000 # SEE BITS 15,14 TO 01 TO INDICATE
 MASK FLAGWRD6 # T5 TAKEOVER BY RCSDAP
 AD BIT14
 TS FLAGWRD6 # KILLS TVCEXEC AND ROLLDAP STARTS

 TC Q # RETURN TO CALLER (TVCDAPOF OR RCSDAPUP)

```

This code is written to file `src/P40-P47.s`.



**B.75 P51-P53**

1297

*<src/P51-P53.s 1297>≡*

```

Copyright: Public domain.
Filename: P51-P53.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 737-784
Contact: Ron Burkey <info@sandroid.org>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-12 RSB Adapted from Colossus249 file of the same
name, and Comanche 055 page images.
2009-05-20 RSB Corrections: SETI/PDT -> SET1/PDT,
GOTOP00H -> GOTOP00H, R33EXIT -> R53EXIT,
V853 -> VB53, R56A -> R56A1 (some places
only), added missing R56A1 label, added a
missing CAF in COARSTYP, corrected a SETLOC
from P50S to P50S3.
2009-05-21 RSB In COARFINE, a TC BANKCALL was corrected to
TC PHASCHNG. In R53C, a CADR GOFLASHR was
corrected to CADR GOFLASH.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 737
PROGRAM NAME -- PROG52
MOD NO -- 2
MODIFICATION BY -- LONSKE
DATE -- NOV 30, 1966
LOG SECTION -- P51-P53
ASSEMBLY -- SUNDISK REV 30

```

```

#
FUNCTIONAL DESCRIPTION --
#
ALIGNS THE IMU TO ONE OF THREE ORIENTATIONS SELECTED BY THE ASTRONAUT. THE IMU ORIENTATION
AND IS STORED IN REFSMMAT. THE THREE POSSIBLE ORIENTATIONS MAY BE:
#
(A) PREFERRED ORIENTATION
#
AN OPTIMUM ORIENTATION FOR A PREVIOUSLY CALCULATED MANEUVER. THIS ORIENTATION IS
STORED BY A PREVIOUSLY SELECTED PROGRAM.
#
(B) NOMINAL ORIENTATION
#

$$\begin{matrix} X & = & \text{UNIT} & (& Y & & x & Z & &) \\ -\text{SM} & & & & -\text{SM} & & & -\text{SM} \end{matrix}$$

#

$$\begin{matrix} Y & = & \text{UNIT} & (& V & X & R &) \\ -\text{SM} & & & & - & & - \end{matrix}$$

#

$$\begin{matrix} Z & = & \text{UNIT} & (& -R &) \\ -\text{SM} & & & & - \end{matrix}$$

#
WHERE:
#
R = THE GEOMETRIC RADIUS VECTOR AT TIME T(ALIGN) SELECTED BY THE ASTRONAUT.
-
#
V = THE INERTIAL VELOCITY VECTOR AT TIME T(ALIGN) SELECTED BY THE ASTRONAUT.
-
#
(C) RERSMMAT ORIENTATION
#
THIS SELECTION CORRECTS THE PRESENT IMU ORIENTATION. THE PRESENT ORIENTATION
WAS LAST ALIGNED ONLY DUE TO GYRO DRIVE (I.E., NEITHER GIMBAL LOCK NOR GYRO
SINCE THE LAST ALIGNMENT).
#
AFTER A IMU ORIENTATION HAS BEEN SELECTED ROUTINE S52.2 IS OPERATED TO COMPUTE THE
NEW ORIENTATION AND THE PRESENT VEHICLE ATTITUDE. CAL52A THEN USES THESE ANGLES TO
COARSE ALIGN THE IMU. THE STARS SELECTION ROUTINE, R56, IS THEN OPERATED. A MESSAGE
IS FLASHED TO NOTIFY THE ASTRONAUT. AT THIS POINT THE ASTRONAUT WILL MANEUVER THE
EITHER MANUALLY OR AUTOMATICALLY. AFTER 2 STARS HAVE BEEN SELECTED THE IMU IS
THE RENDEZVOUS NAVIGATION PROCESS IS OPERATING (INDICATED BY RNDVZFLG) P20 IS
REQUESTED.
#
CALLING SEQUENCE --
#

```

```

THE PROGRAM IS CALLED BY THE ASTRONAUT BY DSKY ENTRY.
Page 738
#
SUBROUTINES CALLED --
#
1. FLAGDOWN 7. S52.2 13. NEWMODEX
2. R02BOTH 8. CAL53A 14. PRIOLARM
3. GOPERF4 9. FLAGUP
4. MATMOVE 10. R56
5. GOFLASH 11. R51
6. S52.3 12. GOPERF3
#
NORMAL EXIT MODES --
#
EXITS TO ENDOFJOB
#
ALARM OR ABORT EXIT MODES --
#
NONE
#
OUTPUT --
#
THE FOLLOWING MAY BE FLASHED ON THE DSKY
1. IMU ORIENTATION CODE
2. ALARM CODE 215 -- PREFERRED IMU ORIENTATION NOT SPECIFIED
3. TIME OF NEXT IGNITION
4. GIMBAL ANGLES
5. ALARM CODE 405 -- TWO STARS NOT AVAILABLE
6. PLEASE PERFORM P00
THE MODE DISPLAY MAY BE CHANGED TO 20
#
ERASABLE INITIALIZATION REQUIRED --
#
PFRATFLG SHOULD BE SET IF A PREFERRED ORIENTATION HAS BEEN COMPUTED. IF IT HAS BEEN CO
XSMD, YSMD, ZSMD.
#
RNDVZFLG INDICATES WHETHER THE RENDEZVOUS NAVIGATION PROCESS IS OPERATING.
#
DEBRIS --
#
WORK AREA
#
P54 = PROG52
 BANK 33
 SETLOC P50S
 BANK

```

|            |        |          |                                          |
|------------|--------|----------|------------------------------------------|
|            | SBANK= | LOWSUPER |                                          |
|            | EBANK= | SAC      |                                          |
|            | COUNT  | 15/P52   |                                          |
| PROG52     | TC     | PHASCHNG |                                          |
|            | OCT    | 00254    |                                          |
|            | TC     | DOWNFLAG |                                          |
|            | ADRES  | UPDATFLG | # BIT 7 FLAG 1                           |
| # Page 739 |        |          |                                          |
|            | TC     | DOWNFLAG |                                          |
|            | ADRES  | TRACKFLG | # BIT 5 FLAG 1                           |
|            | TC     | BANKCALL |                                          |
|            | CADR   | R02BOTH  | # IMU STATUS CHECK                       |
|            | CAF    | BIT4     |                                          |
|            | MASK   | STATE +2 | # IS PFRATFLG SET?                       |
|            | CCS    | A        |                                          |
|            | TC     | P52A     | # YES                                    |
|            | CAF    | BIT2     | # NO                                     |
|            | TC     | P52A +1  |                                          |
| P52A       | CAF    | BIT1     |                                          |
|            | TS     | OPTION2  |                                          |
| P52B       | CAF    | BIT1     |                                          |
|            | TC     | BANKCALL | # FLASH OPTION CODE AND ORIENTATION CODE |
|            | CADR   | GOPERF4R |                                          |
|            | TC     | GOTOP00H |                                          |
|            | TC     | +5       |                                          |
|            | TC     | P52B     | # NEW CODE -- NEW ORIENTATION CODE INPUT |
|            | TC     | PHASCHNG |                                          |
|            | OCT    | 00014    |                                          |
|            | TC     | ENDOFJOB |                                          |
|            | CA     | OPTION2  |                                          |
|            | MASK   | THREE    |                                          |
|            | INDEX  | A        |                                          |
|            | TC     | +1       |                                          |
|            | TC     | P52T     | # L.S.                                   |
|            | TC     | P52J     | # PREF                                   |
|            | TC     | P52T     | # NORM                                   |
|            | TCF    | P52C     | # REF                                    |
| P52T       | EXTEND |          |                                          |
|            | DCA    | NEGO     |                                          |
|            | DXCH   | DSPTM1   |                                          |
|            | CAF    | V06N34   |                                          |
|            | TC     | BANKCALL |                                          |
|            | CADR   | GOFLASH  |                                          |
|            | TC     | GOTOP00H |                                          |

July 12, 2016

1301

```
TC +2
TC -5
EXTEND
DCA DSPTEM1
EXTEND
BZF +2
TCF +4

EXTEND
DCA TIME2
DXCH DSPTEM1
CA OPTION2
MASK BIT2
CCS A

Page 740

TCF +6 # NOM
TC INTERPRET # LS
CALL
P52LS
GOTO
P52D
TC INTERPRET
DLOAD
DSPTEM1
CALL # COMPUTE NOMINAL IMU
S52.3 # ORIENTATION
P52D CALL # READ VEHICLE ATTITUDE AND
S52.2 # COMPUTE GIMBAL ANGLES
EXIT
CAF VB06N22
TC BANKCALL # DISPLAY GIMBAL ANGLES
CADR GOFLASH
TC GOTOP00H
TC COARSTYP
P52J TC INTERPRET # RECYCLE: VEHICLE HAS BEEN MANEUVERED
GOTO
P52D
TC INTERPRET
CALL
CAL53A # DO COARSE ALIGN
CAL53RET SET EXIT # ROUTINE
REFSMFLG
P52C TC PHASCHNG
OCT 04024
CAF ALRM15
TC BANKCALL
```

|            |       |          |                                        |
|------------|-------|----------|----------------------------------------|
|            | CADR  | GOPERF1  |                                        |
|            | TC    | GOTPOOH  |                                        |
|            | TC    | +2       | # V33                                  |
|            | TC    | P52F     | # E                                    |
|            | TC    | INTPRET  |                                        |
|            | RTB   | DAD      |                                        |
|            |       | LOADTIME |                                        |
|            |       | TSIGHT1  |                                        |
|            | CALL  |          |                                        |
|            |       | LOCSAM   |                                        |
|            | EXIT  |          |                                        |
| P52E       | TC    | BANKCALL | # DO STAR SELECTION                    |
|            | CADR  | PICAPAR  |                                        |
|            | TC    | P52I     | # 2 STARS NOT AVAILABLE                |
| P52F       | TC    | INTPRET  | # 2 STARS AVAILABLE                    |
|            | CALL  |          |                                        |
|            |       | R51      |                                        |
| ENDP50S    | EXIT  |          |                                        |
|            | TC    | GOTPOOH  |                                        |
| # Page 741 |       |          |                                        |
| P52I       | TC    | ALARM    |                                        |
|            | OCT   | 405      |                                        |
|            | CAF   | V05N09   |                                        |
|            | TC    | BANKCALL |                                        |
|            | CADR  | GOFLASH  |                                        |
|            | TC    | GOTPOOH  |                                        |
|            | TC    | P52F     | # PROCEED: DO FINE ALIGN-R51           |
|            | TC    | P52C     | # RECYCLE: VEHICLE HAS BEEN MANEUVERED |
| V06N34     | VN    | 0634     |                                        |
| VB06N22    | VN    | 00622    |                                        |
| COARSTYP   | CAF   | OCT13    |                                        |
|            | TC    | BANKCALL |                                        |
|            | CADR  | GOPERF1  |                                        |
|            | TCF   | GOTPOOH  | # V34                                  |
|            | TCF   | P52J +3  | # NORMAL                               |
|            | TC    | INTPRET  | # GYRO COARSE                          |
| GYCRS      | VLOAD | MXV      |                                        |
|            |       | XSMD     |                                        |
|            |       | REFSMMAT |                                        |
|            | UNIT  |          |                                        |
|            | STOVL | XDC      |                                        |
|            |       | YSMD     |                                        |
|            | MXV   | UNIT     |                                        |
|            |       | REFSMMAT |                                        |
|            | STOVL | YDC      |                                        |
|            |       | ZSMD     |                                        |

July 12, 2016

1303

MXV UNIT  
REFSMMAT  
STCALL ZDC  
CALCGTA  
CLEAR CLEAR  
DRIFTFLG  
REFSMFLG  
EXIT  
CAF V16N20  
TC BANKCALL  
CADR GODSPR  
CA R55CDR  
TC BANKCALL  
CADR IMUPULSE  
TC BANKCALL  
CADR IMUSTALL  
TC CURTAINS  
TC PHASCHNG  
OCT 04024  
TC INTPRET  
AXC,1 AXC,2  
XSMD  
REFSMMAT  
CALL

# Page 742

MATMOVE  
CLEAR SET  
PFRATFLG  
REFSMFLG  
RTB VLOAD  
SET1/PDT  
ZEROVEC  
STORE GCOMP  
SET GOTO  
DRIFTFLG  
R51K  
V16N20 VN 1620  
ALRM15 EQUALS OCT15  
SETLOC P50S2  
BANK  
V06N89\* VN 0689

# NAME -- P52LS

#

# FUNCTION -- TO DISPLAY THE LANDING SITE LATITUDE,  
# LONGITUDE AND ALTITUDE. TO ACCEPT NEW DATA VIA

```

THE KEYBOARD. TO COMPUTE THE LANDING SITE
ORIENTATIION FOR P52 OR P54.
#
LET:
RLS = LANDING SITE VECTOR IN REF COORDINATES
R = CSM POSITION VECTOR IN REF COORDINATES
V = CSM VELOCITY VECTOR IN REF COORDINATES
THEN THE LANDING SITE ORIENTATION IS:
XSMD = UNIT(RLS)
YSMD = UNIT(ZSMD*XSMD)
ZSMD = UNIT((R*V)*RLS)
#
CALL: CALL
P52LS
#
INPUTS: DSPTM1 = TIME OF ALIGNMENT
RLS = LANDING SITE VECTOR IN MOON FIXED COORDINATES
#
OUTPUTS: XSMD, YSMD, ZSMD
#
SUBROUTINES: RP-TO-R, LAT-LONG, LLASRD, LLASRDA, CSMPREC
#
DEBRIS: VAC, SEE SUBROUTINES

P52LS STQ SET
 QMAJ
 LUNAFLAG
 DLOAD
 DSPTM1
 STORE TSIGHT
 VLOAD SET
 RLS
 ERADFLAG
 STODL OD
 TSIGHT
 STCALL 6D

Page 743

 RP-TO-R
 VSR2
 STODL ALPHAV
 TSIGHT
 CALL
 LAT-LONG
 CALL
 LLASRD
 EXIT

```



July 12, 2016

1305

```
LSDISP CAF V06N89*
 TC BANKCALL
 CADR GOFLASH
 TC GOTOPOOH
 TC +2
 TC LSDISP
 TC INTPRET
 CALL
 LLASRDA
DLOAD CALL
 TSIGHT
 LALOTORV
VLOAD UNIT
 ALPHAV
STODL XSMD
 TSIGHT
STCALL TDEC1
 CSMPREC
VLOAD VXV
 RATT
 VATT
VXV UNIT
 XSMD
STORE ZSMD
VXV UNIT
 XSMD
STCALL YSMD
 QMAJ
SETLOC P50S1
BANK
```

```
NAME: AUTOMATIC OPTICS POSITIONING ROUTINE
#
FUNCTION: (1) TO POINT THE STAR LOS OF THE OPTICS AT A STAR OR LANDMARK DEFINED BY THE PR
(2) TO POINT THE STAR LOS OF THE OPTICS AT THE LEM DURING RENDEZVOUS TRACKING C
#
CALLING: CALL R52
#
INPUT: 1. TARG1FLG AND TARG2FLG: PRESET BY CALLER
2. RNDVZFLG AND TRACKFLG: PRESET BY CALLER
3. STAR CODE: PRESET BY CALLER. ALSO INPUT THROUGH DSKY
4. LAT, LONG, AND ALT OF LANDMARK: INPUT THROUGH DSKY
Page 744
5. NO. OF MARKS (MARKINDX): PRESET BY CALLER
#
OUTPUT: DRIVE SHAFT AND TRUNNION CDUS.
```

```

#
SUBROUTINES: 1. FIXDELAY 7. CLEANDSP
2. GOPERF1 8. GODSPR
3. GOFLASH 9. REFLASHR
4. R53 10. R52.2
5. ALARM 11. R52.3
6. SR52.1

COUNT 15/R52

R52 STQ CLEAR
 SAVQR52
 ADVTRK

R52VRB EXIT
 EXTEND
 DCA CDUT
 DXCH DESOPTT
 TC INTPRET
 SSP CLEAR
 OPTIND
 0
 R53FLAG

R52A EXIT
 TC INTPRET
 SET BON
 TRUNFLAG
 TARG1FLG
 R52H
 CLEAR EXIT
 TERMIFLG

R52C CA SWSAMPLE # IS OPTICS MODE IN AGC
 EXTEND
 BZMF R52M # MANUAL
R52D TC BANKCALL # AGC
 CADR SR52.1
 TCF R52L # GR 90 DEGREES
 TCF R52J # GR 50 DEGREES
 TC UPFLAG # LS 50 DEGREES
 ADRES TRUNFLAG # SET TRUNFLAG BIT 4 FLAG 0
R52JA CAF BIT10 # IS THIS A LEM
 MASK STATE +1
 CCS A
 TC R52E # YES
 CAF BIT6 # NO, IS R53FLAG SET
 MASK STATE
 CCS A

```

July 12, 2016

1307

|            |        |          |                        |
|------------|--------|----------|------------------------|
| # Page 745 | TCF    | R52E     | # YES                  |
|            | CAF    | V06N92   | # NO                   |
|            | TC     | BANKCALL |                        |
|            | CADR   | GODSPR   |                        |
| R52E       | CA     | SWSAMPLE | # IS OSS IN CMC MODE   |
|            | EXTEND |          |                        |
|            | BZMF   | R52F     | # NO                   |
|            | CS     | STATE    | # YES: IS TRUNFLAG SET |
|            | MASK   | BIT4     |                        |
|            | CCS    | A        |                        |
|            | TC     | +3       | # NO                   |
|            | CA     | PAC      | # YES                  |
|            | TS     | DESOPTT  |                        |
|            | CA     | SAC      |                        |
|            | TS     | DESOPTS  |                        |
| R52F       | CAF    | .5SEC    | # WAIT 1/2 SEC         |
|            | TC     | BANKCALL |                        |
|            | CADR   | DELAYJOB |                        |
|            | CAF    | BIT10    |                        |
|            | MASK   | STATE +1 |                        |
|            | CCS    | A        |                        |
|            | TCF    | R52HA    | # YES, LEM             |
|            | CAF    | BIT15    | # NO                   |
|            | MASK   | STATE +7 | # IS TERMIFLG SET      |
|            | EXTEND |          |                        |
|            | BZF    | R52C     | # NO                   |
| R52Q       | TC     | INTPRET  | # YES                  |
|            | GOTO   |          |                        |
|            |        | SAVQR52  |                        |
| R52H       | EXIT   |          | # LEM                  |
| R52HA      | TC     | BANKCALL |                        |
|            | CADR   | R61CSM   |                        |
|            | CA     | STATE +1 |                        |
|            | MASK   | BIT5     |                        |
|            | EXTEND |          | # TRACKFLG             |
|            | BZF    | R52Q     |                        |
|            | CS     | STATE +1 |                        |
|            | MASK   | BIT7     | # UPDATFLG             |
|            | CCS    | A        |                        |
|            | TCF    | R52SYNC  |                        |
| R52I       | CA     | STATE +5 |                        |
|            | MASK   | BIT10    |                        |
|            | CCS    | A        |                        |

|            |        |                 |                                             |
|------------|--------|-----------------|---------------------------------------------|
| R52SYNC    | TC     | R52D            | # PRFTRKAT = 1                              |
|            | CAF    | 1.8SEC          | # MAKE UP FOR LOST TIME                     |
|            | TCF    | R52F +1         |                                             |
| R52J       | TC     | DOWNFLAG        | # CLEAR TRUNFLAG                            |
|            | ADRES  | TRUNFLAG        | # BIT 4 FLAG 0                              |
| # Page 746 |        |                 |                                             |
|            | TC     | ALARM           | # SET 407 ALARM                             |
|            | OCT    | 407             |                                             |
|            | TC     | R52JA           |                                             |
| R52M       | CAF    | BIT6            | # IS R53FLAG SET                            |
|            | MASK   | STATE           |                                             |
|            | CCS    | A               |                                             |
|            | TC     | R52F            | # YES                                       |
|            | INHINT |                 | # NO                                        |
|            | CAF    | PRI024          |                                             |
|            | TC     | FINDVAC         |                                             |
|            | EBANK= | SAC             |                                             |
|            | 2CADR  | R53JOB          |                                             |
|            | RELINT |                 |                                             |
|            | TCF    | R52F            |                                             |
| R53JOB     | TC     | INTPRET         |                                             |
|            | CALL   |                 |                                             |
|            |        | R53             |                                             |
| ENDPLAC    | EXIT   |                 | # INTERPRETER RETURN TO ENDOFJOB (R22 USES) |
|            | TC     | ENDOFJOB        |                                             |
| V06N92     | VN     | 00692           |                                             |
| V06N89A    | VN     | 0689            |                                             |
| SHAXIS     | 2DEC   | .5376381241 B-1 |                                             |
|            | 2DEC   | 0               |                                             |
|            | 2DEC   | .8431766920 B-1 |                                             |
| R52L       | CAF    | BIT10           | # IS THIS A LEM                             |
|            | MASK   | STATE +1        |                                             |
|            | CCS    | A               |                                             |
|            | TC     | R52J            | # YES                                       |
|            | CAF    | OCT404          |                                             |
|            | TC     | BANKCALL        |                                             |
|            | CADR   | PRIOLARM        |                                             |
|            | TCF    | TERM52          | # TERMINATE                                 |
|            | TCF    | R52F            | # PROCEED                                   |
|            | TCF    | R52F            | # NO PROVISION FOR NEW DATA                 |
|            | TCF    | ENDOFJOB        |                                             |

July 12, 2016

1309

|        |      |          |                    |
|--------|------|----------|--------------------|
| OCT404 | OCT  | 404      |                    |
| 1.8SEC | DEC  | 180      |                    |
| TERM52 | TC   | CLEARMRK |                    |
|        | TC   | BANKCALL | # KILL MARK SYSTEM |
|        | CADR | MKRELEAS |                    |

# Page 747

|        |      |          |                                   |
|--------|------|----------|-----------------------------------|
|        | CAF  | ZERO     |                                   |
|        | TS   | OPTCADR  |                                   |
|        | TC   | BANKCALL | # CLEAR OUT EXTENDED VERBS        |
|        | CADR | KLEENEX  |                                   |
|        | TC   | GOTOPOOH | # NO GO TO POO                    |
| ADVORB | STQ  | SET      | # SETS UP ADVANCED ORBIT TRACKING |
|        |      | SAVQR52  |                                   |
|        |      | ADVTRK   |                                   |
|        | SET  | SET      |                                   |
|        |      | LUNAFLAG |                                   |
|        |      | ERADFLAG |                                   |
|        | GOTO |          |                                   |
|        |      | R52VRB   |                                   |

# Page 748

# NAME -- S50 ALIAS LOCSAM

# NAME: LOCSAM

#

# FUNCTION -- TO COMPUTE QUATITIGS LISTED BELOW, USED IN THE  
# IMU ALIGNMENT PROGRAMS.

#

# DEFINE:

#

# RATT = POSITION VECTOR OF CM WRT PRIMARY BODY

#

# VATT = VELOCITY VECTOR OF CM WRT PRIMARY BODY

#

# RE = RADIUS OF EARTH

#

# RM = RADIUS OF MOON

#

# ECLIPOL = POLE OF ECLIPTIC SCALED BY TANGENTIAL VELOCITY OF EARTH

```

WRT TO SUN OVER THE VELOCITY OF LIGHT
#
REM = POSITION OF MOON WRT EARTH
#
RES = POSITION OF SUN WRT EARTH
#
C = VELOCITY OF LIGHT
#
EARTH IS PRIMARY MOON IS PRIMARY
#
$VEARTH = -1(RATT)$ $VEARTH = -1(REM + RATT)$
#
$VMOON = 1(REM - RATT)$ $VMOON = -1(RATT)$
#
$VSUN = 1(RES)$ $VSUN = 1(RES - REM)$
#
$CEARTH = \cos(\sin^{-1}(RE/RATT) + 5)$ $CEARTH = \cos 5$
#
$CMOON = \cos 5$ $CMOON = \cos(\sin^{-1} CRM/RATT) + 5)$
#
$CSUN = \cos 15$ $CSUN = \cos 15$
#
$VEL/C = VSUN \times ECLIPOL + VATT/C$
#
CALL: DLOAD CALL
DESIRED TIME
LOCSAM
#
INPUTS: MPAC = TIME
#
OUTPUTS: VEARTH, VMOON, VSUN, CEARTH, CMOON, CSUN, VEL/C
#
SUBROUTINES: LSPOS, CSMCONIC
#
DEBRIS: VAC AREA, SEE SUBROUTINES.
#
Page 749
#
SETLOC P50S1
BANK
#
COUNT* $$/S50
#
LOCSAM = S50
S50 STQ
#
QMAJ
STCALL TSIGHT

```

|            |        |          |
|------------|--------|----------|
|            |        | LSPOS    |
|            | STOVL  | VMOON    |
|            |        | 2D       |
|            | STODL  | VSUN     |
|            |        | TSIGHT   |
|            | STCALL | TDEC1    |
|            |        | CSMCONIC |
|            | SSP    | TIX,2    |
|            |        | S2       |
|            |        | 0        |
|            |        | MOONCNTR |
| EARTCNTR   | VLOAD  | VSU      |
|            |        | VMOON    |
|            |        | RATT     |
|            | UNIT   |          |
|            | STOVL  | VMOON    |
|            |        | RATT     |
|            | UNIT   | VCOMP    |
|            | STODL  | VEARTH   |
|            |        | RSUBE    |
|            | CALL   |          |
|            |        | OCCOS    |
|            | STODL  | CEARTH   |
|            |        | CSS5     |
|            | STOVL  | CMOON    |
|            |        | VSUN     |
|            | UNIT   |          |
|            | STCALL | VSUN     |
|            |        | ENDSAM   |
| MOONCNTR   | VLOAD  | VSR8     |
|            |        | VMOON    |
|            | VSR1   | BVSU     |
|            |        | VSUN     |
|            | UNIT   |          |
|            | STOVL  | VSUN     |
|            |        | VMOON    |
|            | VAD    | UNIT     |
|            |        | RATT     |
|            | VCOMP  |          |
|            | STOVL  | VEARTH   |
| # Page 750 |        |          |
|            |        | RATT     |
|            | UNIT   | VCOMP    |
|            | STODL  | VMOON    |
|            |        | RSUBM    |

|          |        |                |   |                       |
|----------|--------|----------------|---|-----------------------|
|          | CALL   |                |   |                       |
|          |        | OCCOS          |   |                       |
|          | STODL  | CMOON          |   |                       |
|          |        | CSS5           |   |                       |
|          | STOVL  | CEARTH         |   |                       |
|          |        | VSUN           |   |                       |
| ENDSAM   | VXV    |                |   |                       |
|          |        | ECLIPOL        |   |                       |
|          | STOVL  | VEL/C          |   |                       |
|          |        | VATT           |   |                       |
|          | VXSC   | VAD            |   |                       |
|          |        | 1/C            |   |                       |
|          |        | VEL/C          |   |                       |
|          | STODL  | VEL/C          |   |                       |
|          |        | CSSUN          |   |                       |
|          | STCALL | CSUN           |   |                       |
|          |        | QMAJ           |   |                       |
| OCCOS    | DDV    | SR1            |   |                       |
|          |        | 36D            |   |                       |
|          | ASIN   | DAD            |   |                       |
|          |        | 5DEGREES       |   |                       |
|          | COS    | SR1            |   |                       |
|          | RVQ    |                |   |                       |
|          | SETLOC | P50S           |   |                       |
|          | BANK   |                |   |                       |
| RSUBM    | 2DEC   | 1738090 B-29   | # | MOON RADIUS IN METERS |
| RSUBE    | 2DEC   | 6378166 B-29   |   |                       |
| 5DEGREES | 2DEC   | .013888889     | # | SCALED IN REVS        |
| 1/C      | 2DEC   | .000042699 B-1 | # | *                     |
| ECLIPOL  | 2DEC   | 0              | # | *                     |
|          | 2DEC   | -.00007896 B-1 | # | *                     |
|          | 2DEC   | .00018209 B-1  | # | *                     |
|          |        |                |   | * FOR USE BY CSM ONLY |
| TSIGHT1  | 2DEC   | 24000          |   |                       |
| CEARTH   | =      | 14D            |   |                       |
| CSUN     | =      | 16D            |   |                       |
| CMOON    | =      | 18D            |   |                       |
| CSS5     | 2DEC   | .2490475       | # | (COS 5)/4             |



July 12, 2016

1313

```
CSSUN 2DEC .24148 # (COS 15)/4

Page 752
PROGRAM NAME -- PICAPAR DATE: DEC 20 66
MOD 1 LOG SECTION: P51-P53
ASSEMBLY: SUNDISK REV40
BY KEN VINCENT
#
FUNCTION
THIS PROGRAM READS THE IMU-CDUS AND COMPUTES THE VEHICLE ORIENTATION
WITH RESPECT TO INERTIAL SPACE. IT THEN COMPUTES THE SHAFT AXIS (SAX)
WITH RESPECT TO REFERENCE INERTIAL. EACH STAR IN THE CATALOG IS TESTED
TO DETERMINE IF IT IS OCCULTED BY EITHER EARTH, SUN OR MOON. IF A
STAR IS NOT OCCULTED THEN IT IS PAIRED WITH ALL STARS OF LOWER INDEX.
THE PAIRED STAR IS TESTED FOR OCCULTATION. PAIRS OF STARS THAT PASS
THE OCCULTATION TESTS ARE TESTED FOR GOOD SEPARATION. A PAIR OF STARS
HAVE GOOD SEPARATION IF THE ANGLE BETWEEN THEM IS LESS THAN 66 DEGREES
AND MORE THAN 40 DEGREES. THOSE PAIRS WITH GOOD SEPARATION
ARE THEN TESTED TO SEE IF THEY LIE IN CURRENT FIELD OF VIEW. (WITHIN
33 DEGREES OF SAX). THE PAIR WITH MAX SEPARATION IS CHOSEN FROM
THOSE WITH GOOD SEPARATION, AND IN FIELD OF VIEW.
#
CALLING SEQUENCE
L TC BANKCALL
L+1 CADR PICAPAR
L+2 ERROR RETURN -- NO STARS IN FIELD OF VIEW
L+3 NORMAL RETURN
#
OUTPUT
BESTI, BESTJ -- SINGLE PREC, INTEGERS, STAR NUMBERS TIMES 6
VFLAG -- FLAG BIT SET IMPLIES NO STARS IN FIELD OF VIEW
#
INITIALIZATION
1) A CALL TO LOCSAM MUST BE MADE
2) VEARTH = -UNIT(R) WHERE R HAS BEEN UPDATED TO APPROXIMATE TIME OF
SIGHTINGS.
#
DEBRIS
WORK AREA
X,Y,ZNB
SINCDU, COSCDU
STARAD -- STAR +5

COUNT 14/PICAP

SETLOC P50S1
```

```

PICAPAR BANK
 TC MAKECADR
 TS QMIN
 TC INTPRET
 CALL
 CDUTRIG
 CALL
 CALCSMSC

Page 753

SETPD
0
SET DLOAD # VFLAG = 1
 VFLAG
 DPZERO
STOVL BESTI
 XNB
VXSC PDVL
 SIN33
 ZNB
AXT,1 VXSC
 228D # X1 = 37 X 6 + 6
 COS33
VAD
VXM UNIT
 REFSMMAT
STORE SAX # SAX = SHAFT AXIS
SSP SSP # S1 = S2 = 6
 S1
 6
 S2
 6
PIC1 TIX,1 GOTO # MAJOR STAR
 PIC2
 PICEND
PIC2 VLOAD* CALL
 CATLOG,1
 OCCULT
BON LXA,2
 CULTFLAG
 PIC1
 X1
PIC3 TIX,2 GOTO
 PIC4
 PIC1
PIC4 VLOAD* CALL
 CATLOG,2

```

July 12, 2016

1315

```

 OCCULT
 BON VLOAD*
 CULTFLAG
 PIC3
 CATLOG,1
 DOT* DSU
 CATLOG,2
 CSS66 # SEPARATION LESS THAN 66 DEG.
 BMN DAD
 PIC3
 CSS6640 # SEPARATION MORE THAN 40 DEG.
 BPL
 PIC3
Page 754
 VLOAD* DOT
 CATLOG,1
 SAX
 DSU BMN # MAJOR STAR IN CONE
 CSS33
 PIC1
 VLOAD* DOT
 CATLOG,2
 SAX
 DSU BPL
 CSS33
 STRATGY
 GOTO
 PIC3
STRATGY BONCLR
 VFLAG
 NEWPAR
 XCHX,1 XCHX,2
 BESTI
 BESTJ
STRAT VLOAD* DOT*
 CATLOG,1
 CATLOG,2
 PUSH BOFINV
 VFLAG
 STRAT -3
 DLOAD DSU
 BPL
 PIC3
NEWPAR SXA,1 SXA,2
 BESTI
 BESTJ
```

|            |       |             |                     |
|------------|-------|-------------|---------------------|
|            | GOTO  | PIC3        |                     |
| OCCULT     | MXV   | BVSU        |                     |
|            |       | CULTRIX     |                     |
|            |       | CSS         |                     |
|            | BZE   |             |                     |
|            |       | CULTED      |                     |
|            | BMN   | SIGN        |                     |
|            |       | CULTED      |                     |
|            |       | MPAC +3     |                     |
|            | BMN   | SIGN        |                     |
|            |       | CULTED      |                     |
|            |       | MPAC +5     |                     |
|            | BMN   | CLRG0       |                     |
|            |       | CULTED      |                     |
|            |       | CULTFLAG    |                     |
|            |       | QPRET       |                     |
| CULTED     | SETGO |             |                     |
| # Page 755 |       |             |                     |
|            |       | CULTFLAG    |                     |
|            |       | QPRET       |                     |
| CSS        | =     | CEARTH      |                     |
| SIN33      | 2DEC  | .5376381241 |                     |
| COS33      | 2DEC  | .8431756920 |                     |
| CSS66      | 2DEC  | .060480472  | # (COS76)/4         |
| CSS6640    | 2DEC  | -.15602587  | # (COS76 - COS30)/4 |
| CSS33      | 2DEC  | .197002688  | # (COS(1/2(76)))/4  |
| PICEND     | BOFF  | EXIT        |                     |
|            |       | VFLAG       |                     |
|            |       | PICGXT      |                     |
|            | TC    | PICBXT      |                     |
| PICGXT     | EXIT  |             |                     |
|            | INCR  | QMIN        |                     |
| PICBXT     | CA    | QMIN        |                     |
|            | TC    | SWCALL      |                     |
| #V1        | =     | 12D         |                     |

# Page 756

# NAME -- R51 FINE ALIGN

# FUNCTION -- TO ALIGN THE STABLE MEMBER TO REFSSMAT

# CALLING SEQ -- CALL R51

```
INPUT -- BESTI, BESTJ (PAIR OF STAR NO)
OUTPUT -- GYRO TORQUE PULSES
SUBROUTINES -- R52, R54, R55 (SXTNB, NBSM, AXISGEN)
```

```

COUNT 14/R51

R51 EXIT
 CAF BIT1
 TS STARIND
 TS MARKINDX
R51.2 TC INTPRET
R51.3 CLEAR CLEAR
 TARG2FLG
 TARG1FLG

 EXIT
 TC PHASCHNG
 OCT 05024 # RESTART GR 4 FOR R52-R53
 OCT 13000
 INDEX STARIND
 CA BESTI
 EXTEND
 MP 1/6TH
 TS STARCODE
R51DSP CAF V01N70
 TC BANKCALL
 CADR GOFLASHR
 TC GOTOP00H
 TC +5
 TC -5
 CAF SIX
 TC BLANKET
 TCF ENDOFJOB
 TC CHKSCODE
 TC FALTON
 TC R51DSP
 TC INTPRET
 RTB CALL
 LOADTIME
 PLANET
 SSP LXA,1
 S1
 0
 STARIND
 TIX,1
 R51ST
 STCALL STARS2V2 # 2ND STAR
```

|            |        |                      |                                    |
|------------|--------|----------------------|------------------------------------|
| R51ST      | STORE  | R51ST +1<br>STARSAV1 | # 1ST STAR                         |
| # Page 757 | EXIT   |                      |                                    |
|            | CS     | MODREG               | # IS THIS P54                      |
|            | AD     | OCT66                |                                    |
|            | EXTEND |                      |                                    |
|            | BZF    | R51B                 | # YES                              |
|            | TC     | INTPRET              |                                    |
|            | CALL   |                      |                                    |
|            |        | R52                  | # AOP WILL MAKE CALLS TO SIGHTING  |
| R51A       | CALL   |                      | # COMPUTE LOS IN SM FROM MARK DATA |
|            |        | SXTSM                |                                    |
|            | STORE  | STARSAV2             |                                    |
|            | EXIT   |                      |                                    |
|            | TC     | BANKCALL             |                                    |
|            | CADR   | MKRELEAS             |                                    |
|            | TC     | INTPRET              |                                    |
|            | DLOAD  | CALL                 |                                    |
|            |        | TSIGHT               |                                    |
|            |        | PLANET               |                                    |
|            | EXIT   |                      |                                    |
|            | CCS    | STARIND              |                                    |
|            | TC     | R51.4                |                                    |
|            | TC     | INTPRET              |                                    |
|            | MXV    | UNIT                 |                                    |
|            |        | REFSMMAT             |                                    |
|            | STORE  | STARAD               |                                    |
|            | VLOAD  |                      |                                    |
|            |        | STARSAV2             |                                    |
|            | STOVL  | 6D                   |                                    |
|            |        | STARSAV1             |                                    |
|            | STOVL  | 12D                  |                                    |
|            |        | PLANVEC              |                                    |
|            | STCALL | STARAD +6            |                                    |
|            |        | R54                  | # STAR DATA TEST                   |
|            | BOFF   | CALL                 |                                    |
|            |        | FREEFLAG             |                                    |
|            |        | R51K                 |                                    |
|            |        | AXISGEN              |                                    |
|            | CALL   |                      |                                    |
|            |        | R55                  | # GYRO TORQUE                      |
|            | CLEAR  |                      |                                    |
|            |        | PFRATFLG             |                                    |
| R51K       | EXIT   |                      |                                    |
|            | CAF    | OCT14                |                                    |

July 12, 2016

1319

```

TC BANKCALL
CADR GOPERF1
TC GOTOPOOH
TC +2 # V33
TC +3
TC BANKCALL
CADR P52C

Page 758
TC INTERPRET
GOTO

R51.4 TC ENDP50S
 MXV INTERPRET
 REFMMAT
STOVL PLANVEC
 STARS2
STORE STARS1
SSP
 STARIND
 0
GOTO

R51B TC R51.3
 CALL INTERPRET
 R56
GOTO

OCT66 OCT R51A
V01N70 VN 00066
1/6TH DEC 0170
 .166667

Page 759
NAME: R55 GYRO TORQUE
FUNCTION -- COMPUTE AND SEND GYRO PULSES
CALLING SEQ -- CALL R55
INPUT -- X,Y,ZDC -- REFMMAT WRT PRESENT STABLE MEMBER
OUTPUT -- GYRO PULSES
SUBROUTINES -- CALCGTA, GOFLASH, GODSPR, IMUFINE, IMUPULSE, GOPERF1

R55 SETLOC P50S
 BANK
COUNT* $$/R55
STQ
 QMIN
CALL
 CALCGTA
```

```

PULSEM EXIT
R55.1 CAF V06N93
 TC BANKCALL
 CADR GOFLASH
 TC GOTOPOOH
 TC R55.2
 TC R55RET
R55.2 TC PHASCHNG
 OCT 00314
 CA R55CDR
 TC BANKCALL
 CADR IMUPULSE
 TC BANKCALL
 CADR IMUSTALL
 TC CURTAINS
 TC PHASCHNG
 OCT 05024
 OCT 13000
R55RET TC INTPRET
 GOTO
 QMIN
V06N93 VN 0693
R55CDR ECADR OGC
R54 = CHKSDATA

```

```

ROUTINE NAME -- CHKSDATA DATE -- JAN 9, 1967
MOD NO -- 0 LOG SECTION -- P51-P53
MODIFICATION BY -- LONSKE ASSEMBLY --
#
FUNCTIONAL DESCRIPTION -- CHECKS THE VALIDITY OF A PAIR OF STAR SIGHTINGS. WHEN A
BY THE ASTRONAUT THIS ROUTINE OPERATES AND CHECKS THE OBSERVED SIGHTINGS AGAINST S
COMPUTER TO INSURE A PROPER SIGHTING WAS MADE. THE FOLLOWING COMPUTATIONS ARE PER
#
OS1 = OBSERVED STAR 1 VECTOR
OS2 = OBSERVED STAR 2 VECTOR
SS1 = STORED STAR 1 VECTOR
SS2 = STORED STAR 2 VECTOR
A1 = ARCCOS(OS1 - OS2)
Page 760
A2 = ARCCOS(SS1 - SS2)
A = ABS(2(A1 - A2))
#
THE ANGULAR DIFFERENCE IS DISPLAYED FOR ASTRONAUT ACCEPTANCE.
#
EXIT MODE -- 1. FREEFLAG SET IMPLIES ASTRONAUT WANTS TO PROCEED
2. FREEFLAG RESET IMPLIES ASTRONAUT WANTS TO RECYCLE
#

```



July 12, 2016

1321

```
OUTPUT -- 1. VERB 6,NOUN 3 -- DISPLAYS ANGULAR DIFFERENCE BETWEEN 2 SETS OF STARS.
2. STAR VECTORS FROM STAR CATALOG ARE LEFT IN 6D AND 12D.
#
ERASABLE INITIALIZATION REQUIRED --
1. MARK VECTORS ARE STORED IN STARAD AND STARAD +6.
2. CATALOG VECTORS ARE STORED IN 6D AND 12D.
#
DEBRIS --
```

```
 SETLOC P50S1
 BANK
 COUNT* $$/R50
CHKSDATA STQ SET
 QMIN
 FREEFLAG
CHKXSAB AXC,1 # SET X1 TO STORE EPHEMERIS DATA
 STARAD
CHKSAB VLOAD* DOT* # CAL. ANGLE THETA
 0,1
 6,1
 SL1 ACOS
 STORE THETA
 BOFF INVERT # BRANCH TO CHKSD IF THIS IS 2ND PASS
 FREEFLAG
 CHKSD
 FREEFLAG # CLEAR FREEFLAG
 AXC,1 DLOAD # SET X1 TO MARK ANGLES
 6D
 THETA
 STORE 18D
 GOTO
 CHKSAB # RETURN TO CAL. 2ND ANGLE
CHKSD DLOAD DSU
 THETA # COMPUTE POS DIFF
 18D
 ABS RTB
 SGNAGREE
 STORE NORMTEM1
 SET EXIT
 FREEFLAG
 CAF ZERO
 TC BANKCALL
 CADR CLEANDSP
 CAF VB6N5
```

# Page 761

```

TC BANKCALL
CADR GOFLASH
TCF GOTOPOOH
TC CHKSDA # PROCEED
TC INTPRET
CLEAR GOTO
 FREEFLAG
 QMIN
CHKSDA TC INTPRET
 GOTO
 QMIN
VB6N5 VN 605

NAME -- CAL53A
FUNCTION -- COARSE ALIGN THE IMU, IF NECESSARY
CALLING SEQUENCE -- CALL CAL53A
INPUT -- PRESENT GIMBAL ANGLES -- CDUX, CDUY, CDUZ
DESIRED GIMBAL ANGLES -- THETAD,+1,+2
OUTPUT -- THE IMU COORDINATES AT STORED IN REFSMMAT
SUBROUTINES -- 1.IMUCOARS, 2.IMUSTALL, 3CURTAINS

COUNT 14/R50

CAL53A CALL
 S52.2 # MAKE FINAL COMP OF GIMBAL ANGLES
RTB SSP
 RDCDUS # READ CDUS
 S1
 1
AXT,1 SETPD
 3
 4
CALOOP DLOAD* SR1
 THETAD +3D,1
PDDL* SR1
 4,1
DSU ABS
PUSH DSU
 DEGREE1
BMN DLOAD
 CALOOP1
DSU BPL
 DEG359
 CALOOP1
COARFINE EXIT

```

July 12, 2016

1323

```
Page 762
TC PHASCHNG
OCT 04024
TC BANKCALL
CADR IMUCOARS # PERFORM COARSE ALIGNMENT
TC BANKCALL
CADR IMUSTALL # REQUEST MODE SWITCH

TC CURTAINS
TC BANKCALL
CADR IMUFIN20
TC BANKCALL
CADR IMUSTALL
TC CURTAINS # TEST FOR MALFUNCTION
TC INTPRET
RTB VLOAD
 SET1/PDT
 ZEROVEC
STORE GCOMP
SET GOTO
 DRIFTFLG
 FINEONLY
CALOOP1 TIX,1
 CALOOP
FINEONLY AXC,1
 AXC,2
 XSM
 REFSMMAT
CALL
 MATMOVE
GOTO
 CAL53RET
MATMOVE VLOAD* # TRANSFER MATRIX
 0,1
STORE 0,2
VLOAD*
 6D,1
STORE 6D,2
VLOAD*
 12D,1
STORE 12D,2
RVQ
DEGREE1 DEC 46
DEG359 DEC 16338
 SETLOC P50S
 BANK
RDCDUS INHINT # READ CDUS
 CA CDUX
```

```

INDEX FIXLOC
TS 1
CA CDUY
INDEX FIXLOC
TS 2
CA CDUZ
INDEX FIXLOC
TS 3
RELINT
TC DANZIG

```

# Page 763

# NAME: GIMB

#

# FUNCTION: DETERMINE AND COMPUTE THE DESIRED GIMBAL ANGLES TO BE USED FOR COARSE

#

# CALLING SEQUENCE: CALL GIMB

#

# INPUT: DESIRED IMU INERTIAL ORIENTATION VECTORS: XSMD, YSMD, ZSMD

#

# OUTPUT: GIMBAL ANGLES LEFT IN THETAD, +1, +2

#

# SUBROUTINES USED: 1.CDUTRIG 2.CALCSMSC 3.CALCGA

```

SETLOC P50S2
BANK
COUNT 14/INFLT

```

```

CALCSMSC DLOAD DMP
 SINCDUY
 COSCDUZ
 DCOMP
 PDDL SR1
 SINCDUZ
 PDDL DMP
 COSCDUY
 COSCDUZ
 VDEF VSL1
 STORE XNB
 DLOAD DMP
 SINCDUX
 SINCDUZ
 SL1
 STORE 26D
 DMP
 SINCDUY

```

```

PDDL DMP
 COSCDUX
 COSCDUY

DSU
PDDL DMP
 SINCDUX
 COSCDUZ

DCOMP
PDDL DMP
 COSCDUX
 SINCDUY

PDDL DMP
 COSCDUY
 26D

DAD VDEF
VSL1
STORE ZNB
VXV VSL1
 XNB
STORE YNB
RVQ

```

# NAME -- P51 -- IMU ORIENTATION DETERMINATION

# MOD. NO. 2 21 DEC 66

LOG SECTION -- P51-P53

# Page 764

# MOD BY STURLAUGSON

ASSEMBLY SUNDISK REV15

#

# FUNCTIONAL DESCRIPTION

# DETERMINES THE INERTIAL ORIENTATION OF THE IMU. THE PROGRAM IS SELECTED BY DSKY ENTRY.  
# ROUTINE IS CALLED TO COLLECT THE CDU COUNTERS AND SHAFT AND TRUNNION ANGLES FOR A SIGHT  
# THEN PROCESSED AS FOLLOWS.

#

# 1. SEXTANT ANGLES ARE COMPUTED IN TERMS OF NAVIGATIONAL BASE COORDINATES. LET SA AND  
# TRUNNION ANGLES, RESPECTIVELY. THEN,

#

#  $V = (\sin(TA) \cdot \cos(SA), \sin(TA) \cdot \sin(SA), \cos(TA))$  (A COLUMN VECTOR)  
# NB

# THE OUTPUT IS A HALF-UNIT VECTOR STORED IN STARM.

#

# 2. THIS VECTOR IN NAV. BASE COORDS. IS THEN TRANSFORMED TO ONE IN STABLE MEMBER COORDI

#

#  $V = Q \begin{matrix} T & T & T \\ 1 & 2 & 3 \end{matrix} \begin{matrix} T \\ T \\ NB \end{matrix} V$  , WHERE

#

#  $\begin{pmatrix} \cos(IG) & 0 & -\sin(IG) \end{pmatrix}$   
#  $\begin{pmatrix} \end{pmatrix}$

THE GIMBAL ANGL

```

Q = (0 1 0), IG= INNER GIMBAL ANGLE
1 (
(SIN(IG) 0 COS(IG))
#
(COS(MG) SIN(MG) 0)
(
Q = (-SIN(MG) COS(MG) 0), MG= MIDDLE GIMBAL ANGLE
2 (
(0 0 1)
#
(1 0 0)
(
Q = (0 COS(OG) SIN(OG)), OG= OUTER GIMBAL ANGLE
3 (
(0 -SIN(OG) COS(OG))
#

```

```

3. THE STAR NUMBER IS SAVED AND THE SECOND STAR IS THEN SIMILARLY PROCESSED
#

```

```

4. THE ANGLE BETWEEN THE TWO STARS IS THEN CHECKED AT CKSDATA.
#

```

```

5. REFSMMAT IS THEN COMPUTED AT AXISGEN AS FOLLOWS.
#

```

```

LET \bar{S}_1 AND \bar{S}_2 BE TWO STAR VECTORS EXPRESSED IN TWO COORDINATE SYSTEMS
#
Page 765
DEFINE,
#
$\bar{U}_A = \bar{S}_{A1}$
#
$\bar{V}_A = \text{UNIT}(\bar{S}_{A1} \times \bar{S}_{A2})$
#
$\bar{W}_A = \bar{U}_A \times \bar{V}_A$
#
AND,
#
$\bar{U}_B = \bar{S}_{B1}$
#
$\bar{V}_B = \text{UNIT}(\bar{S}_{B1} \times \bar{S}_{B2})$
#
$\bar{W}_B = \bar{U}_B \times \bar{V}_B$
#

```

```

#
THEN
#
$\bar{X} = U \begin{matrix} \bar{U} \\ B1 \end{matrix} + V \begin{matrix} \bar{V} \\ B1 \end{matrix} + W \begin{matrix} \bar{W} \\ B1 \end{matrix}$
#
$\bar{Y} = U \begin{matrix} \bar{U} \\ B2 \end{matrix} + V \begin{matrix} \bar{V} \\ B2 \end{matrix} + W \begin{matrix} \bar{W} \\ B2 \end{matrix}$ (REFSMAT)
#
$\bar{Z} = U \begin{matrix} \bar{U} \\ B3 \end{matrix} + V \begin{matrix} \bar{V} \\ B3 \end{matrix} + W \begin{matrix} \bar{W} \\ B3 \end{matrix}$
#
THE INPUTS CONSIST OF THE FOUR HALF-UNIT VECTORS STORED AS FOLLOWS
#
\bar{S} IN 6-11 OF THE VAC AREA
A1
#
\bar{S} IN 12-17 OF THE VAC AREA
A2
#
\bar{S} IN STARAD
B1
Page 766
\bar{S} IN STARAD +6
B2
#
CALLING SEQUENCE:
#
THE PROGRAM IS CALLED BY THE ASTRONAUT BY DSKY ENTRY.
#
SUBROUTINES CALLED:
#
GOPERF3
GOPERF1R
GODSPR
IMUCOARS
IMUFIN20
R53
SXTNB
NBSM
MKRELEAS
CHKSDATA
MATMOVE
#
ALARMS
#

```

```

NONE
#
ERASABLE INITIALIZATION:
#
IMU ZERO FLAG SHOULD BE SET.
#
OUTPUT
#
REFSMMAT
REFSMFLG
#
DEBRIS
#
WORK AREA
STARAD
STARIND
BESTI
BESTJ

 SETLOC P50S1
 BANK
 COUNT 14/P5153

P53 EQUALS P51
P51 CS IMODES30
 MASK BIT9
 CCS A

Page 767

 TC P51A
 TC ALARM
 OCT 210
 TC GOTOP00H
P51A TC BANKCALL
 CADR R02ZERO

P51AA CAF PRFMSTAQ
 TC BANKCALL
 CADR GOPERF1
 TC GOTOP00H # TERM.
 TC P51B # V33
 TC PHASCHNG
 OCT 05024
 OCT 13000
 CAF P51ZERO
 TS THETAD # ZERO THE GIMBALS
 TS THETAD +1

```



July 12, 2016

1329

```
TS THETAD +2
CAF V6N22
TC BANKCALL
CADR GODSPRET
CAF V41K # NOW DISPLAY COARSE ALIGN VERB 41
TC BANKCALL
CADR GODSPRET
TC BANKCALL
CADR IMUCOARS
TC BANKCALL
CADR IMUSTALL
TC CURTAINS # CAGING OR BAD END
TC BANKCALL # SCHEDULE IFAILOK AND IMUFINED TASKS, IN 5
CADR IMUFIN20 # AND 20 SECS. DIRECT RETURN AND NO STALL,
TC BANKCALL # IF CAGING, BUT T4 WILL ZERO C/A ENABLE.
CADR IMUSTALL # IF PUT TO SLEEP, IMUFINED WILL WAKE US
TC CURTAINS # UP.
TC PHASCHNG
OCT 05024
OCT 13000
TCF P51AA # COARSE ALIGN DONE: RECYCLE FOR FINE
```

# Page 768

# DO STAR SIGHTING AND COMPUTE NEW REFSMMAT

```
P51B TC PHASCHNG
 OCT 00014
 TC INTPRET
 SSP SETPD
 STARIND # INDEX -- STAR 1 OR 2
 0
 0
 RTB VLOAD
 SET1/PDT
 ZEROVEC
 STORE GCOMP
 SET CLEAR
 DRIFTFLG # ENABLE T4 COMPENSATION
 TARG2FLG # SHOW MARK IS STAR --- NOT LANDMARK
 EXIT
 CAF BIT1
 TS MARKINDX # INITIALIZE FOR ONE MARK

P51C TC PHASCHNG
 OCT 05024
 OCT 13000
 TC CHECKMM
```

1330

July 12, 2016

|            |       |          |                                    |
|------------|-------|----------|------------------------------------|
|            | MM    | 53       | # BACKUP PROGRAM                   |
|            | TCF   | P51C.1   | # NOT P53                          |
|            | TC    | INTPRET  |                                    |
|            | CALL  |          |                                    |
|            |       | R56      |                                    |
|            | GOTO  |          |                                    |
|            |       | P51C.2   |                                    |
| P51C.1     | TC    | INTPRET  |                                    |
|            | CALL  |          |                                    |
|            |       | R53      | # SIGHTING ROUTINE                 |
| P51C.2     | CALL  |          | # COMPUTE LOS IN SM FROM MARK DATA |
|            |       | SXTSM    |                                    |
|            | PUSH  |          |                                    |
|            | SLOAD | BZE      |                                    |
|            |       | STARIND  |                                    |
|            |       | P51D     |                                    |
|            | VLOAD | STADR    |                                    |
|            | STORE | STARSAV2 | # DOWNLINK                         |
|            | GOTO  |          |                                    |
|            |       | P51E     |                                    |
| P51D       | VLOAD | STADR    |                                    |
|            | STODL | STARSAV1 |                                    |
|            |       | TSIGHT   |                                    |
|            | CALL  |          |                                    |
|            |       | PLANET   |                                    |
|            | STORE | PLANVEC  |                                    |
| # Page 769 |       |          |                                    |
| P51E       | EXIT  |          |                                    |
|            | TC    | PHASCHNG |                                    |
|            | OCT   | 05024    |                                    |
|            | OCT   | 13000    |                                    |
|            | TC    | BANKCALL |                                    |
|            | CADR  | MKRELEAS | # ZERO MARKSTAT                    |
|            | CCS   | STARIND  |                                    |
|            | TCF   | P51F     | # STAR 2                           |
|            | TC    | PHASCHNG |                                    |
|            | OCT   | 05024    |                                    |
|            | OCT   | 13000    |                                    |
|            | CAF   | BIT1     |                                    |
|            | TS    | STARIND  |                                    |
|            | TCF   | P51C     | # GO DO SECOND STAR                |
| P51F       | TC    | PHASCHNG |                                    |
|            | OCT   | 05024    |                                    |
|            | OCT   | 13000    |                                    |
|            | TC    | INTPRET  |                                    |
|            | DLOAD | CALL     |                                    |

July 12, 2016

1331

```

 TSIGHT
 PLANET
 STOVL 12D
 PLANVEC
 STOVL 6D
 STARS AV1
 STOVL STARAD
 STARS AV2
 STCALL STARAD +6
 CHKS DATA # CHECK STAR ANGLES IN STARAD AND
 BON EXIT
 FREEFLAG
 P51G
 TC P51AA
P51G CALL
 AXISGEN # COME BACK WITH REFSMMAT IN XDC
 AXC,1 AXC,2
 XDC
 REFSMMAT
 CLEAR CALL
 REFSMFLG
 MATMOVE
 SET GOTO
 REFSMFLG
 ENDP50S
PRFMSTAQ = OCT15
P51ZERO = ZERO
P51FIVE = FIVE
V6N22 VN 0622
V41K VN 4100
SET1/PDT CA TIME1
Page 770
 TS 1/PIPADT
 TCF DANZIG
```

# Page 771

# SXTSM COMPUTES AN LOS VECTOR IN SM COORD FROM OCDU AND ICDU MARK DATA

```

 SETLOC P50S3
 BANK
SXTSM STQ
 QMAJ
 LXC,1 DLOAD*
 MARKSTAT
 OD,1
 STORE TSIGHT
```

```

 LXC,2 SLOAD*
 STARIND
 MKDNCDR,2
 LXC,2 VLOAD*
 MPAC
 0,1
 STORE 0,2
 DLOAD*
 5,1
 STORE 5,2
 CALL
 SXTNB # COMPUTE LOS VECTOR FROM OCDU IN MKVAC
 LXA,1 INCR,1
 MARKSTAT
 2 # INCREMENT TO BASE ADR OF ICPU
 SXA,1 CALL
 S1
 NBSM # TRANSFORM LOS TO SM
 GOTO
 QMAJ
MKDNCDR ECADR MARKDOWN
 ECADR MARK2DWN

```

# Page 772

# PROGRAM DESCRIPTION: R53 -- SIGHTING MARK ROUTINE

# MOD. NO. 2

21 DEC 66

# MOD. BY STURLAUGSON

#

# FUNCTIONAL DESCRIPTION:

#

# TO PERFORM A SATISFACTORY NUMBER OF SIGHTING MARKS FOR THE REQUESTING PROGRAM  
 # CAN BE MADE ON A STAR OR LANDMARK. WHEN THE CMC ACCEPTS A MARK IT RECORDS AN  
 # (OCDUS) AND THE TIME OF THE MARK.

#

# CALLING SEQUENCE:

#

# R53 IS CALLED AND RETURNS IN INTERPRETIVE CODE. RETURN IS VIA QPRET.  
 # THERE IS NO ERROR EXIT IN THIS ROUTINE ITSELF.

#

# SUBROUTINES CALLED

#

# SXTMARK  
 # OPTSTALL  
 # GOFLASH

#

# ERASABLE INITIALIZATION:

```

#
TARGET FLAG -- STAR OR LANDMARK
MARKINDX -- NUMBER OF MARKS WANTED
STARIND -- INDEX TO BESTI OR BESTJ (STAR NUMBER)
#
OUTPUT
#
MARKSTAT CONTAINS INDEX TO VACANT AREA WEHRE MARK DATA IS STORED
BESTI (INDEXED BY STARIND) CONTAINS STAR NUMBER SIGHTED.
#
DEBRIS
#
MARKINDX CONTAINS NUMBER OF MARKS DESIRED

 SETLOC RT53
 BANK

 COUNT 14/R53

R53 STQ SET # SET SIGHTING MARK FLAG
 R53EXIT
 R53FLAG

 EXIT
R53A CA MARKINDX # NUMBER OF MARKS
 MASK LOW3
 TC BANKCALL
 CADR SXTMARK
 TC BANKCALL
 CADR OPTSTALL
 TC CURTAINS
 INDEX MARKSTAT
 CCS QPRET # NUMNBER OF MARKS ACTUALLY DONE
 TCF R53B
 TCF +2 # ZERO
 TCF +1 # CCS HOLE
 CAF ZERO # HOUSEKEEP VAC AREA SAVE
 XCH MARKSTAT # AND MARKSTAT

Page 773

 CCS A
 INDEX A
 TS 0
 TCF R53A
R53B TC CHECKMM
 MM 22
 TCF +2
 TCF R53D

```

|          |        |          |                         |
|----------|--------|----------|-------------------------|
|          | TC     | CHECKMM  |                         |
|          | MM     | 23       |                         |
|          | TCF    | R53C1    |                         |
|          | TCF    | R53D     |                         |
| R53C1    | CAF    | ZERO     |                         |
|          | TC     | BANKCALL |                         |
|          | CADR   | CLEANDSP |                         |
| R53C     | CAF    | V01N71   |                         |
|          | TC     | BANKCALL |                         |
|          | CADR   | GOFLASH  |                         |
|          | TC     | GOTOPOOH | # TERM.                 |
|          | TCF    | R53Z     |                         |
|          | TC     | R53C     | # RECYCLE               |
| R53Z     | TC     | CHKSCODE |                         |
|          | TC     | FALTON   |                         |
|          | TC     | R53C     |                         |
|          | CS     | HIGH9    |                         |
|          | MASK   | STARCODE |                         |
|          | EXTEND |          |                         |
|          | MP     | SIGHTSIX |                         |
|          | XCH    | L        |                         |
|          | INDEX  | STARIND  |                         |
|          | TS     | BESTI    |                         |
| R53D     | TC     | INTPRET  |                         |
| R53OUT   | SETGO  |          |                         |
|          |        | TERMIFLG | # SET TERMINATE FOR R52 |
|          |        | R53EXIT  |                         |
| SIGHTSIX | =      | SIX      |                         |
| V01N71   | VN     | 0171     |                         |

# \*\*\*\*\* KEEP IN SAME BANK AS R51 AND R53 \*\*\*\*\*

|          |       |          |                     |
|----------|-------|----------|---------------------|
| CHKSCODE | CCS   | STARCODE |                     |
|          | AD    | NEG47    |                     |
|          | CCS   | A        |                     |
|          | TC    | Q        | # SC < 0 OR SC > 50 |
|          | TCF   | +2       | # SC = + OR - 0     |
|          | TCF   | +1       | # 0 <= SC< 50       |
|          | INDEX | Q        | # SC = 50           |
|          | TC    | 00002    |                     |
| NEG47    | OCT   | 77730    |                     |

# Page 774

# NAME -- S52.2

# FUNCTION -- COMPUTE GIMBAL ANGLES FOR DESIRED SM AND PRESENT VEHICLE

# CALL -- CALL S52.2

# INPUT -- X,Y,ZSMD

July 12, 2016

1335

# OUTPUT -- OGC,IGC,MGC,THETAD,+1,+2  
# SUBROUTINES -- CDUTRIG, CALCSMSC, MATMOVE, CALCGA

```

 SETLOC S52/2
 BANK
S52.2 COUNT 13/S52.2
 STQ
 QMAJ
 CALL
 CDUTRIG
 CALL
 CALCSMSC
 AXT,1 SSP
 18D
 S1
 6D
S52.2A VLOAD* VXM
 XNB +18D,1
 REFSMMAT
 UNIT
 STORE XNB +18D,1
 TIX,1
S52.2.1 AXC,1 S52.2A
 AXC,2
 XSMD
 XSM
 CALL
 MATMOVE
 CALL
 CALCGA
 GOTO
 QMAJ
```

# Page 775

# PROGRAM NAME: SR52.1

DATE: DEC 20 1968

# MOD 1

LOG SEC: P51-P53

# BY KEN VINCENT

ASSEMBLY: SUNDISK REV 40

#

# FUNCTION

#

# TARG1 AND TARG2 FLAGS ARE LOOKED AT TO DETERMINE IF THE TARGET IS THE  
# LEM, STAR, OR LANDMARK. IN CASE OF LEM OR LMK, THE PRESENT TIME PLUS  
# 2 SECONDS IS SAVED IN AOPTIME (ALIAS STARAD, +1). IF THE LEM IS  
# THE TARGET THEN CONIC UPDATES OF TEH CSM AND LEM ARE MADE TO  
# THE TIME IN AOPTIME. THE UNIT OF THE DIFFERENCE OF LEM AND CSM

```

POSITION VECTORS BECOMES THE REFERENCE SIGHTING VECTOR USED IN THE
COMMON PART OF THE THIS PROGRAM.
#
IN THE CASE OF LANDMARK, THE CSM IS UPDATED CONICALLY. THE RADIUS
VECTOR FOR THE LANDMARK IS OBTAINED FROM LALOTORV. BOTH OF THESE ARE
FOUND FOR THE TIME IN AOPTIME. THE UNIT OF THE DIFFERENCE BETWEEN
THE LANDMARK AND CSM RADIUS VECTORS BECOMES THE REFERENCE SIGHTING
VECTOR FOR THE COMMON PART OF THIS ROUTINE.
#
IF A STAR IS THE TARGET, THE PROPER STAR IS OBTAINED FROM THE CATALOG
AND THIS VECTOR BECOMES THE REFERENCE SIGHTING VECTOR.
#
THE COMMON PART OF THIS PROGRAM TRANSFORMS THE REFERENCE SIGHTING
VECTOR INTO STABLE MEMBER COORDINATES. IT READS THE IMU-CDUS AND USES
THIS DATA IN A CALL TO CALCSXA. ON RETURN FROM CALCSXA A TEST IS
MADE TO SEE IF THE TRUNNION ANGLE IS GREATER THAN 90DEG OR 38DEG.
MADE TO SEE IF THE TRUNNION ANGLE IS GREATER THAN 90DEG. OR 50DEG.
#
CALLING SEQUENCE
#
L+4 RETURN WHEN SHAFT OR TRUNION NOT WITHIN 5 DEG OF DESIRED
L TC BANKCALL
L+1 CADR SR52.1
L+2 ERROR RETURN TRUNNION GREATER THAN 90 DEG.
L+3 ERROR RETURN TRUNNION GREATER THAN 50 DEG
L+4 NORMAL RETURN
#
OUTPUT
#
SAC: SINGLE PREC, 2'S COMP, SCALED AT HALF REVS -- SHAFT ANGLE DESIRED.
PAC: SINGLE PREC, 2'S COMP, SCALED AT EIGHTH REVS -- TRUNNION ANGLE DESIRED.
#
INITIALIZATION
#
IF TARG1FLG =1 THEN TARGET IS LEM -- NO OTHER INPUT REQUIRED.
#
IF TARG1FLG =0 AND TARG2FLG =0 THE TARGET IS STAR, STARIND SHOULD
0 OR 1 DENOTING BESTI OR BESTJ RESPECTIVELY AS STAR CODE. STAR CODES
ARE 6 TIMES STAR NUMBER.
#
IF TARG1FLG =0 AND TARG2FLG =1 THEN TARGET IS LANDMARK. SETT ROUTINE
LALOTORV FOR INPUT REQUIREMENTS. HERE FIXERAD=1 FOR CONSTANT EARTH
RADIUS
#
DEBRIS
#

```



July 12, 2016

1337

# WORK AREA  
# STARAD -- STAR+5 (STAR IS DESIRED LOS IN STABLE MEMBER COORDINATES)

COUNT\* \$\$/SR521  
# Page 776  
SETLOC SR52/1  
BANK

SR52.1 TC MAKECADR  
TS QMIN  
TC INTPRET  
RTB DAD  
LOADTIME  
1.3SECDP  
STORE AOPTIME  
BON BON  
TARG1FLG  
LEM52  
TARG2FLG  
LMK52

GOTO  
STAR52  
LEM52 DLOAD  
AOPTIME  
STCALL TDEC1  
LEMCONIC  
VLOAD  
RATT  
GOTO

LMK52 LMKLMCOM  
BON DLOAD  
ADVTRK  
ADVTRACK  
AOPTIME  
CALL  
LALOTORV  
VLOAD

LMKLMCOM ALPHAV  
STODL STAR  
AOPTIME  
STCALL TDEC1  
CSMCONIC  
VLOAD VSU  
STAR  
RATT  
UNIT GOTO

|            |       |                                       |                                          |
|------------|-------|---------------------------------------|------------------------------------------|
| STAR52     | SSP   | COM52<br>LXA,1<br>S1<br>0<br>STARIND  |                                          |
|            | TIX,1 | ST52ST                                |                                          |
|            | VLOAD | GOTO<br>STARSAV2                      |                                          |
| # Page 777 |       |                                       |                                          |
| ST52ST     | VLOAD | COM52                                 |                                          |
| COM52      | MXV   | STARSAV1<br>UNIT<br>REFSMMAT          |                                          |
|            | STORE | STAR                                  |                                          |
|            | SETPD | CALL<br>0<br>CDUTRIG                  | # COMPUTES SINES AND COSINES FOR CALCSXA |
|            | CALL  |                                       | # NOW EXPECT TO SEE THE CDU ANGLES.      |
|            | BOFF  | CALCSXA<br>EXIT<br>CULTFLAG<br>TRUN38 |                                          |
| TRUN38     | TC    | SR52E1                                |                                          |
|            | DLOAD | DSU<br>PAC<br>38TRDEG                 |                                          |
|            | BPL   | DLOAD<br>SR52E22<br>PAC               |                                          |
|            | DSU   | BPL<br>20DEGSMN<br>SR52E3             |                                          |
| SR52E22    | EXIT  |                                       |                                          |
|            | TC    | SR52E2                                |                                          |
| SR52E3     | EXIT  |                                       |                                          |
|            | INCR  | QMIN                                  |                                          |
| SR52E2     | INCR  | QMIN                                  |                                          |
| SR52E1     | CA    | QMIN                                  |                                          |
|            | TC    | SWCALL                                |                                          |
| 38TRDEG    | 2DEC  | .66666667                             | # CORRESPONDS TO 50 DEGS IN TRUNION      |
| 1.3SECDP   | 2DEC  | 130                                   |                                          |
| 20DEGSMN   | DEC   | -07199                                |                                          |

DEC -0

# Page 778

# THE ADVTRACK ROUTINE IS USED TO COMPUTE AN OPTICS LOS VECTOR TO  
 # A POINT ON THE GROUND TRACK 60 DEGREES FORWARD OF THE LOCAL VERTICAL  
 # OF AN ADVANCED ORBIT A SPECIFIED NUMBER OF REVOLUTIONS FROM NOW.

```

ADVTRACK SETLOC 26P50S
 BANK
 SETPD
 0
 VLOAD PUSH # INITIALIZE FOR RP-TO-R
 UNITZ # UZ VEC IN PD 0-5
 RTB PUSH # TIME IN PD 6-7
 LOADTIME
 STCALL AOPTIME # TIME ALSO IN AOPTIME FOR CSMCONIC
 RP-TO-R # GET MOON ROTATION VEC IN REF
 STODL STAR
 AOPTIME # PICK UP TIME
 STCALL TDEC1 # UPDATE STATE TO TIME
 CSMCONIC
 VLOAD VXV
 VATT
 RATT
 UNIT
 STOVL 24D # SAVE -UNIT(VxR) FOR 2ND ROTATION
 RATT
 UNIT VCOMP
 SETPD PUSH # PUSH LOS=-UNIT(RVEC) PD 0-5
 0
 EXIT
 CA LANDMARK
 MASK SEVEN # GET NUMBER OF ADVANCE PERIODS
 EXTEND
 MP BIT11 # GET N/16
 XCH L
 INDEX FIXLOC
 TS 30D # TEMP STORE N/16
 TC INTPRET
 SLOAD DMP
 30D
 MPERIOD
 STCALL AOPTIME # ROTATE ANG ABOUT UR
 ROTA
 VLOAD
 24D # PICK UP 2ND ROTATION AXIS

```

```

STODL STAR
 DP1/6
DSU
 AOPTIME # 2ND RAT ANGLE = 60 - A
STCALL AOPTIME
 ROTA # GO ROTATE 2ND TIME
VLOAD
Page 779
 0
STCALL STAR # STORE FINAL LOS IN STAR
 COM52 # RETURN TO SR52.1
ROTA
DLOAD SIN
 AOPTIME
PDVL VXV # PUSH 1/2SIN(A) PD 6-7
 STAR # UR VEC
 0 # LOS
VXSC VSL2 # 1/2SIN(A)(URXLOS) PD 6-11
PDVL DOT
 STAR
 0
VXSC VSL2
 STAR
PDDL COS # 1/2(UR . LOS)UR 12-17
 AOPTIME
PDVL BVSU # PUSH 1/2COS(A) 18-19
 12D
 0
VXSC VSL1 # UP 18-19
VAD VAD # UP 12-17 UP 6011
UNIT SETPD
 0
PUSH RVQ
DP1/6 2DEC .16666666
MPERIOD 2DEC .047619 # APPROX LUNAR ROT ANG IN 2HRS x 16
Page 780
NAME -- S52.3
FUNCTION -- XSMD= UNIT(YSMD x ZSMD)
YSMD= UNIT(V X R)
ZSMD= UNIT(-R)
CALL -- DLOAD CALL
TALIGN
S52.3

```

July 12, 2016

1341

```
INPUT -- TIME OF ALIGNMENT IN MPAC
OUTPUT -- X,Y,ZSMD
SUBROUTINES -- CSMCONIC
```

```

 SETLOC P50S2
 BANK
S52.3 COUNT 15/S52.3
 STQ
 QMAJ
 STCALL TDEC1
 CSMPREC
 SETPD
 0
 VLOAD VCOMP
 RATT
 UNIT
 STOVL ZSMD
 VATT
 VXV UNIT
 RATT
 STORE YSMD
 VXV UNIT
 ZSMD
 STCALL XSMD
 QMAJ
```

# Page 781

# PROGRAM DESCRIPTION: R56 -- ALTERNATE LOS SIGHTING MARK ROUTINE

#

# FUNCTIONAL DESCRIPTION

#

# TO PERFORM SIGHTING MARKS FOR THE BACK-UP ALIGNMENT PROGRAMS (P53,P54). THE ASTRONAUT  
# COORDINATES (OPTICS) OF THE ALTERNATE LINE OF SIGHT HE MUST USE FOR THIS ROUTINE. WHEN  
# ENTER IN RESPONSE TO THE FLASHING V50 N25 R1-XXXXX THE CMC STORES THE THREE ICDU ANGLES  
# IN N92.

#

# CALLING SEQUENCE

#

# CALL

# R56

#

# SUBROUTINES CALLED

#

# A PORTION OF SXTMARK (VAC.AREA SEARCH)

# GOFLASH

```

GOPERF1
#
ERASABLE INITIALIZATION
#
STARIND: INDEX TO STAR NUMBER
#
OUTPUT
#
MARKSTAT: INDEX TO VAC.AREA WHERE OUTPUT IS STORED.
BESTI (INDEXED BY STARIND) CONTAINS STAR NUMBER.
ICDU AND OCDU ANGLES IN VAC. AREA AS FOLLOWS:
#
VAC +2 CDUY
VAC +3 CDUS
VAC +4 CDUZ
VAC +5 CDUT
VAC +6 CDUX
#
COUNT* $$/R56
SETLOC P50S3
BANK
R56 STQ EXIT
R53EXIT
CAF V06N94B
TC BANKCALL
CADR GOFLASH
TC GOTOP00H # TERM.
TC R56A # PROCEED: ANGLES OK
TC -5 # ENTER: NEW ANGLES
R56A TC BANKCALL
CADR SXTMARK +2 # INHIBIR EXT VB ACT AND FIND VAC AREA
#
CAF ZERO
TC BANKCALL
CADR CLEANDSP
#
R56A1 CAF VB53 # DISPLAY V53 REQUESTING ALTERNATE MARK
TC BANKCALL
#
Page 782
CADR GOMARK2
TCF GOTOP00H # V34: TERMINATE
TCF R56A1 # V33: DON'T PROCEED -- JUST ENTER TO MARK
TC INTPRET
DLOAD
MRKBUF1 +3
STODL SAC
MRKBUF1 +5

```

July 12, 2016

1343

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | STORE  | PAC      |                                            |
|            | EXIT   |          |                                            |
|            | INHINT |          |                                            |
|            | EXTEND |          |                                            |
|            | DCA    | TIME2    |                                            |
|            | INDEX  | MARKSTAT |                                            |
|            | DXCH   | 0        |                                            |
|            | CA     | CDUY     | # ENTER: THIS IS A BACKUP SYSTEM MARK      |
|            | INDEX  | MARKSTAT |                                            |
|            | TS     | 2        |                                            |
|            | CA     | SAC      |                                            |
|            | INDEX  | MARKSTAT |                                            |
|            | TS     | 3        |                                            |
|            | CA     | CDUZ     |                                            |
|            | INDEX  | MARKSTAT |                                            |
|            | TS     | 4        |                                            |
|            | CA     | PAC      |                                            |
|            | INDEX  | MARKSTAT |                                            |
|            | TS     | 5        |                                            |
|            | CA     | CDUX     |                                            |
|            | INDEX  | MARKSTAT |                                            |
|            | TS     | 6        |                                            |
|            | RELINT |          |                                            |
|            | TC     | CLEARMRK | # ENABLE EXTENDED VERBS                    |
|            | CAF    | OCT16    |                                            |
|            | TC     | BANKCALL |                                            |
|            | CADR   | GOPERF1  |                                            |
|            | TC     | GOTOP00H | # TERM.                                    |
|            | TCF    | R56B     | # PROCEED: MARK COMPLETED                  |
|            | TCF    | R56A +2  | # RECYCLE: DO ANOTHER MARK -- LIKE REJECT. |
| R56B       | TC     | BANKCALL |                                            |
|            | CADR   | R53C1    |                                            |
| VB53       | VN     | 05300    | # ALTERNATE MARK VERB                      |
| V06N94B    | VN     | 00694    |                                            |
|            | SETLOC | P50S     |                                            |
|            | BANK   |          |                                            |
| PLANET     | STORE  | TSIGHT   |                                            |
|            | STQ    | CALL     |                                            |
|            |        | QMIN     |                                            |
|            |        | LOCSAM   |                                            |
|            | VLOAD  |          |                                            |
| # Page 783 |        | VEARTH   |                                            |
|            | STOVL  | OD       |                                            |
|            |        | VSUN     |                                            |

|         |        |                |  |
|---------|--------|----------------|--|
|         | STOVL  | VEARTH         |  |
|         |        | OD             |  |
|         | STORE  | VSUN           |  |
| NOSAM   | EXIT   |                |  |
|         | CS     | HIGH9          |  |
|         | MASK   | STARCODE       |  |
|         | EXTEND |                |  |
|         | MP     | SIGHTSIX       |  |
|         | XCH    | L              |  |
|         | INDEX  | STARIND        |  |
|         | TS     | BESTI          |  |
|         | CCS    | A              |  |
|         | TCF    | NOTPLAN        |  |
|         | CAF    | VNPLANV        |  |
|         | TC     | BANKCALL       |  |
|         | CADR   | GOFLASH        |  |
|         | TC     | GOTOPOOH       |  |
|         | TC     | +2             |  |
|         | TC     | -5             |  |
|         | TC     | INTPRET        |  |
|         | VLOAD  | VXSC           |  |
|         |        | STARSAV3       |  |
|         |        | 1/SQR3         |  |
|         | UNIT   | GOTO           |  |
|         |        | CORPLAN        |  |
| NOTPLAN | CS     | A              |  |
|         | AD     | DEC227         |  |
|         | EXTEND |                |  |
|         | BZMF   | CALSAM1        |  |
|         | INDEX  | STARIND        |  |
|         | CA     | BESTI          |  |
|         | INDEX  | FIXLOC         |  |
|         | TS     | X1             |  |
|         | TC     | INTPRET        |  |
|         | VLOAD* | GOTO           |  |
|         |        | CATLOG,1       |  |
|         |        | CORPLAN        |  |
| CALSAM1 | TC     | INTPRET        |  |
|         | LXC,1  | DLOAD*         |  |
|         |        | STARIND        |  |
|         |        | BESTI,1        |  |
|         | LXC,1  | VLOAD*         |  |
|         |        | MPAC           |  |
|         |        | STARAD -228D,1 |  |
| CORPLAN | VAD    | UNIT           |  |
|         |        | VEL/C          |  |



July 12, 2016

1345

|            |      |           |
|------------|------|-----------|
|            | GOTO |           |
| # Page 784 |      |           |
|            |      | QMIN      |
| DEC227     | DEC  | 227       |
| VNPLANV    | VN   | 0688      |
| 1/SQR3     | 2DEC | .57735021 |

This code is written to file `src/P51-P53.s`.

## B.76 P61-P67

1346  $\langle \text{src}/P61-P67.s \text{ 1346} \rangle \equiv$

```
Copyright: Public domain.
Filename: P61-P67.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 789-818
Contact: Ron Burkey <info@sandroid.org>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-12 RSB Adapted from Colossus249 file of the same
name and Comanche 055 page images.
2009-05-20 RSB Corrections: V06N68 -> V06N74, added missing
definition of V06N74, in several
interpreter operands fixed stuff like
N-M,1 to N -M,1
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
#
Massachusetts Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 789
PROGRAM: P61
MOD NO.: 0 MAR. 13, 1967
MOD BY: R. HIRSCHKOP
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 22 JUN 67 RESTARTS
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 17 JAN 68 COLOSSUS GSOF
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 8 MAY 68 DELETE CMSM M
FUNCTION: TO CALCULATE AND DISPLAY EMS INITIALIZATION DATA
CALLING SEQUENCE: BY V37
```

July 12, 2016

1347

```
EXIT: TO P62
SUBROUTINE CALLS: S61.1, S61.3, GOFLASH, FLAGUP, RO2BOTH
ERASABLE INITIALIZATION:
EMSALT (-29) M .05G ALTITUDE ABOVE FISCHER ELLIPSOID PAD LOADED.
ALFAPAD /180 HYPERSONIC CM TRIM ANGLE OF ATTACK PAD LOADED
OUTPUT: THE FOLLOWING REGISTERS ARE WRITTEN IN FOR USE BY DISPLAYS
GMAX 100 GMAX (-14) G,S MAXIMUM ACCELERATION
VPRED (-7) M/CS PREDICTED VELOCITY AT 400K FT
GAMMAEI (GAMMA/360 PREDICTED GAMMA AT 400K FT
RTGO THETAH/360 RANGE ANGLE TO SPLASH FROM EMSALT
VIO (-7) M/CS INERTIAL VELOCITY AT EMSALT
TTE (-28) CS TIME TO EMSALT
LAT(SPL) /360 TARGET LOCATION
LNG(SPL) /360 TARGET LOCATION
HEADSUP (0) +1 = LIFT DOWN, -1 = LIFT UP
DEBRIS: SEE SUBROUTINES.

BANK 26
SETLOC P60S
BANK

EBANK= AOG

COUNT* $$/P61

P61 CA BIT14 # EXTENDED VERB SHOULD BE FREE THIS CLOSE
 TS EXTVBACT # TO V37
 # LOCK OUT EXTENDED VERBS SO CAN USE TFF
 # ROUTINES. EXT VERB ERASE IS USED

 CS ONE # REMOVE IF HEADSUP EVER ON UPLINK DATA
 TS HEADSUP # PRELOAD

 TC S61.1 # CHECK STATE VECTOR AND IMU ORIENTATION
 # RV 50GENRET. DOES PHASCHNG, GROUP 4.

 CA V06N61 # LAT(SPL) LNG(SPL) HEADSUP
 # XXX.XX DEG XXX.XX DEG XXXXX.

 TC BANKCALL
 CADR GOFLASHR
 TC GOTOP00H

Page 790

 TC P61.4
 TC -5

P61.3 TC PHASCHNG
```

|         |        |          |                                                              |
|---------|--------|----------|--------------------------------------------------------------|
|         | OCT    | 00014    |                                                              |
|         | TC     | ENDOFJOB |                                                              |
| P61.4   | ZL     |          |                                                              |
|         | CCS    | HEADSUP  | # C(HEADSUP)= +1/-1                                          |
|         | CA     | BIT14    | # IF HEADSUP POS,ROLLC =180 DEG.(LIFT DWN)                   |
|         | NOOP   |          | # IF HEADSUP NEG,ROLLC =0 (LIFT UP)                          |
|         | DXCH   | ROLLC    | # ROLLC IS USED BY S62.3: GIM ANG AT .05G                    |
|         | TC     | INTPRET  |                                                              |
| NEWRNVN | DLOAD  |          |                                                              |
|         |        | PIPTIME  | # SAVE TIME OF RN,VN TO DETERMINE IF AN                      |
|         | STCALL | MM       | # UPDATE HAS OCCURRED.                                       |
|         |        | STARTEN1 | # INITIALIZE                                                 |
|         | VLOAD  |          |                                                              |
|         |        | RN       |                                                              |
|         | STORE  | RONE     |                                                              |
|         | UNIT   |          |                                                              |
|         | STOVL  | URONE    |                                                              |
|         |        | VN       |                                                              |
|         | STORE  | VONE     |                                                              |
|         | VXV    | UNIT     |                                                              |
|         |        | URONE    |                                                              |
|         | STORE  | UNI      |                                                              |
| DUMPP61 | DLOAD  | DSU      |                                                              |
|         |        | MM       | # INITIAL VALUE OF PIPTIME                                   |
|         |        | PIPTIME  |                                                              |
|         | BMN    | CALRB    |                                                              |
|         |        | NEWRNVN  | # UPDATED... GO TRY AGAIN                                    |
|         |        | S61.2    | # GET DISPLAY DATA FOR N60 AND N63                           |
|         |        |          | # AND RETURN IN BASIC, BELOW.                                |
| P61.1   | TC     | CLEARMRK |                                                              |
|         | CA     | VO6N60   | # GMAX                   VPRED                   GAMMAEI     |
|         |        |          | # XXX.XX G               XXXXX. FPS               XXX.XX DEG |
|         | TC     | BANKCALL |                                                              |
|         | CADR   | GOFLASH  |                                                              |
|         | TC     | GOTOP00H |                                                              |
|         | TC     | P61.2    | # PROCEED                                                    |
|         | TC     | -5       |                                                              |
| P61.2   | TC     | INTPRET  | # CORRECT TTE FOR TIME LAPSE DURING                          |
|         |        |          | # ABOVE DISPLAY.                                             |
|         | RTB    | DSU      |                                                              |
|         |        | LOADTIME | # CURRENT TIME.                                              |

July 12, 2016

1349

# Page 791

```

 MM # PIPTIME FOR RONE & VONE.
DAD
 TTE1 # NEGATIVE OF FREE FALL TIME.
STORE TTE # DECREMENTED

EXIT

CA V06N63 # RTGO VIO TTE
 # XXXX.X NM XXXXX. FPS XXBXX M,S

TC BANKCALL
CADR GOFLASH
TC GOTOPOOH
TC +2
TC P61.2 # REDO
```

# .... THEN FALL INTO P62

# Page 792

```

PROGRAM: P62
MOD NO.: 0 MAR. 13, 1967
MOD BY: R. HIRSCHKOP
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 22 JUN 67 RESTARTS.
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 17 JAN 68 COLOSSUS GSOP CHANGES.
MOD NO: 4 MOD BY: RR BAIRNSFATHER DATE: 8 MAY 68 MOVE START OF DESIRED G
FUNCTION: 1) TO NOTIFY CREW WHEN GNC SYSTEM IS PREPARED FOR CM/SM SEPARATION.
2) TO ORIENT THE CM TO THE CORRECT ATTITUDE FOR ATMOSPHERIC ENTRY.
CALLING SEQUENCE: BY V37 OR DIRECTLY FROM P61
EXIT: TO P63
ERASABLE INITIALIZATION:
ALFAPAD LEFT BY PAD LOAD
LADPAD LEFT BY PAD LOAD
LODPAD LEFT BY PAD LOAD
LAT(SPL) (MAY BE CHANGED BELOW) LEFT BY DSKY, VIA P61
LNG(SPL) (MAY BE CHANGED BELOW) LEFT BY DSKY, VIA P61
HEADSUP (MAY BE CHANGED BELOW) LEFT BY DSKY, VIA P61
SUBROUTINE CALLS: NEWMODEX, S61.1, CM/DAPIC, CM/DAPON, R02BOTH, GOPERF1, GOFLASH, GODSPR.
```

COUNT\* \$\$/P62

```

TC NEWMODEX # MODE CHANGE IF CAME FROM P61.
MM 62 # MODE CHANGE AUTOMATIC VIA V 37.
CA ONE
TS DNLSTCOD
```

1350

July 12, 2016

|            |        |          |                                         |
|------------|--------|----------|-----------------------------------------|
| P62        | TC     | S61.1    | # CHECK STATE VECTOR AND IMU ORIENTA    |
|            | TC     | INTPRET  |                                         |
|            | SSP    | RTB      |                                         |
|            |        | POSEXIT  |                                         |
|            |        | P62.3    | # CALCULATE DESIRED .05G GIMBAL ANGLE   |
|            |        |          | # WITHOUT DISPLAY.                      |
|            |        | CM/DAPIC | # START CM/POSE AND BODY RATE CALC      |
|            |        |          | # DOES 2PHSCHNG, OCT 40116, OCT 0502    |
|            |        |          | # CM/DAPIC SETS EBANK = EBAOG           |
|            |        |          | # AND RETURNS IN BASIC TO P62.2.        |
| P62.2      | EXTEND |          |                                         |
|            | DCA    | POSECADR | # CONTINUE WITH CM/POSE AFTER AV G.     |
|            | DXCH   | AVEGEXIT |                                         |
|            | CAF    | OCT41    | # REQUEST SEPARATION                    |
|            | TC     | BANKCALL |                                         |
|            | CADR   | GOPERF1R |                                         |
|            | TC     | GOTOPOOH |                                         |
|            | TC     | +3       | # PROCEED                               |
| # Page 793 |        |          |                                         |
|            | TC     | -5       | # NOTE: NODOFLAG WILL BE SET IN CM/1    |
|            | TC     | P61.3    | # ENTER                                 |
|            |        |          | # FOR PHASCHNG AND ENDOFJOB             |
|            | +3     | TC       |                                         |
|            |        | POSTJUMP |                                         |
|            | CADR   | CM/DAPON | # DISABLE RCS DAP, ENABLE ENTRY DAP A   |
|            |        |          | # DO ATTITUDE HOLD.                     |
|            |        |          | # WILL IDLE UNTIL CM/POSE DOES ONE UP   |
|            |        |          | # CM/DAPON DOES NO PHASCHNG.            |
| P62.1      | CA     | V06N61   | # LAT(SPL)      LNG(SPL)      HEADS     |
|            |        |          | # XXX.XX DEG      XXX.XX DEG      00002 |
|            |        |          | # TERMINATE ATTITUDE HOLD. SET UP C     |
|            |        |          | # ROLLC, ALFACOM, BETACOM. BEGIN MA     |
|            |        |          | # ENTRY ATTITUDE.                       |
|            | TC     | BANKCALL |                                         |
|            | CADR   | GOFLASH  |                                         |
|            | TC     | -3       |                                         |
|            | TC     | +2       |                                         |
|            | TC     | -5       |                                         |

July 12, 2016

1351

```
TC PHASCHNG
OCT 04024 # USE ENTRYVN FOR DISPLAY BELOW.
 # EBANK WAS SET IN CM/DAPON TO EBAOG

CCS HEADSUP # C(HEADSUP) = +/- 1
CA BIT14 # IF HEADSUP POS, ROLLC=180 DEG (LIFT DWN)
NOOP ROLLC # IF HEADSUP NEG, ROLLC=0 DEG (LIFT UP)
TS ROLLC
CA ALFAPAD # NOMINAL ALFATRIM PAD LOADED, NEG. NO.
ZL
DXCH ALFACOM # SET ALFACOM = ALFA TRIM, BETACOM=0

CA ONE # PERMITS EXDAP2 TO CHANGE FLAG TO +0
TS P63FLAG # AS INDICATOR. STARTS UP P63.

CA V06N22 # SET UP DISPLAY FOR CDU DESIRED VALUES
TS ENTRYVN # FROM ENTRY ATTITUDE CALC, THAT IS
 # ALREADY GOING.
TC UPFLAG # TURN ON ENTRY DISPLAY
ADRES ENTRYDSP # ENTRYDSP = 92D BIT 13 FLAG 6

SKIP
Page 794

CS CMDAPMOD # GO DIRECTLY TO P63 IF BODY ATTITUDE
MASK ONE # IS SUCH THAT THE DELAY TASK: WAKEP62
EXTEND BZF # WILL BE OMITTED.
 P63.1 # DISABLE GRP 4, GO TO ENDOFJOB.
 P63 # (I.E., CONTINUE IF CMDAPMOD = -1, OR +0)

TC P63

PUT JOB TO SLEEP UNTIL VEHICLE MANEUVER HAS
REDUCED ALFA TO +/-45 DEG. CONSIDER REMAINING
65 DEG (25 DEG IF ALFA NEG) TO ALFA TRIM TO
OCCUR AT 3 DEG/SEC, AND TERMINATE P62 AT THAT
TIME.

TASK WAKEP62 IS CALLED FROM ENTRY DAP.

WAKEP62 CA PRI013
 TC NOVAC
 EBANK= AOG
 2CADR P63

TC TASKOVER

EACH 2 SEC, CALCULATE GIMBAL ANGLES FOR ENTRY
DITIONS THAT WILL HOLD IF REORIENTATION WERE
AT PRESENT RN, VN. COME HERE FROM CM/POSE AN
```

```

IN KEPLER PHASE OF ENTRY.

P62.3 SSP GOTO # SET RETURN ADDRESS SO THAT ROUTINE
 QPRET # GOES DIRECTLY TO ENTRY GUIDANCE EX.
 ENDEXIT # THAT DOES ENTRY DISPLAY, GRP 5.
 S62.3 # PUT DESIRED CDU VALUES IN CPHI'S F
 # N22 DISPLAY.

Page 795
P63
PROGRAM: P63
MOD NO: 0 MAR. 13, 1967
MOD BY: R. HIRSCHKOP
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 22 JUN 67 RESTARTS
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 14 JUL 67 REVISED RESTA
FUNCTION: 1) TO INITIALIZE THE ENTRY EQUATIONS.
2) TO CONTINUE TO HOLD THE CM TO THE CORRECT ATTITUDE WITH RESPECT TO
THE ONSET OF ENTRY DECELERATION. ROLL ANGLE IS LIFT UP/DOWN AS SP
3) TO SENSE .05G.
CALLING SEQUENCE: DIRECTLY FROM P62
EXIT: TO ENDOFJOB
SUBROUTINE CALLS: NEWMODEX, GODSPR

COUNT* $$/P63

P63 TC NEWMODEX
 MM 63

ARRIVE WITH EBANK = AOG.

CA ENTCADR # CONTINUE AT STARTENT AFTER CM/POSE

AT END OF STATEMENT, CHANGE ADDRESS
TO CONTINUE AT SCALEPOP THEREAFTER.

TS POSEXIT

CA V06N64 # G VI R TO
XXX.XX G XXXX. FPS XXXX
TS ENTRYVN # FOR DISPLAY CALL IN OVERNOUT

CS ONE # IN CASE FLAG IS LEFT AT +1 BY DAP.
TS P63FLAG # -1 ASSURES THAT EXO-ATM DAP WILL NO
CALL P63 OUT OF SEQUENCE IN P66.

TC PHASCHNG # THIS IS REQUIRED TO PRESERVE CLEAN

```



July 12, 2016

1353

```
OCT 00004 # RETURN IN EVENT OF AN EXTENDED VERB

TC BANKCALL # FLUSH 'N22' DISPLAY, IF ON, (OMIT
CADR CLEANDSP # DISPLAY DURING 'STARTENT' PASS.)

P63.1 TC PHASCHNG
OCT 00004 # DISABLE. DISPLAY RESTARTED VIA ENTRY.

TC ENDOFJOB

V06N60 VN 0660
V06N61 VN 0661
V06N63 VN 0663
Page 796
V06N64 VN 0664
ENTCADR CADR STARTENT

EBANK= RTINIT # TO CARRY OVER INTO ENTRY STEERING.
POSECADR 2CADR CM/POSE

Page 797
PROGRAM: P64
MOD NO: 1 SEPT. 19, 1967
MOD BY: R. HIRSCHKOP
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 8 MAY 68 REVISED COMMENTS FOR CO
FUNCTION: 1. TO START ENTRY GUIDANCE AT .05G SELECTING ROLL ATTITUDE, CONSTANT DRAG LEVE
DRAG THRESHHOLD, KA, WHICH ARE KEYED TO THE .05G POINT.
2. SELECT FINAL PHASE P67 IF V < 27000 FPS WHEN .2G OCCURS.
3. ITERATE FOR UP-CONTROL SOLUTION P65 IF V > 27000 FPS AND IF ALTITUDE RATE A
LEVEL CONDITIONS ARE SATISFIED. ENTER P65 WHEN CONSTANT DRAG CONTROLLER HA
AS PREDICTED TO WITHIN 25 NM OF DESIRED RANGE.
4. SELECT FINAL PHASE P67 IF NO UP-CONTROL SOLUTION EXISTS WITH VL > 18000 F
CALLING SEQUENCE: BY RTB FROM REENTRY CONTROL
EXIT: BACK TO REENTRY CONTROL.
SUBROUTINE CALLS: NEWMODEX

BANK 25
SETLOC P60S1
BANK

THIS DISPLAY IS CALLED EACH PASS THROUGH STEERING. RESTART PROTECTION IS VIA STEERING.

COUNT* $$/P64

P64 TC NEWMODEX # ENTER VIA RTB WHEN .05G IS EXCEEDED.
MM 64
```

```

CA V06N74 # ROLLC VI D
XXX.XX DEG XXXXX. FPS XXX.X
TS ENTRYVN # DISPLAY VIA OVERNOUT.

TC DANZIG # ... AND CONTINUE IN INITROLL ...

V06N74 VN 0674

Page 798
PROGRAM: P65
MOD NO: 0 MOD BY: RR BAIRNSFATHER DATE: 17 JAN 68 COLOSSUS GSOP
FUNCTION: TO CONTINUE ENTRY GUIDANCE, USING THE UP-CONTROL PHASE TO STEER TO A
CONDITION. THIS PHASE TERMINATES A) IF D < Q7 FPSS, GOTO TO P66.
B) IF RDOT NEG, AND IF V < VL +500
CALLING SEQUENCE: BY RTB FROM REENTRY CONTROL
EXIT: BACK TO REENTRY CONTROL, OR TO ENDOFJOB.
SUBROUTINE CALLS: NEWMODEX

COUNT* $$/P65

P65 TC NEWMODEX # ENTER VIA RTB WHEN RANGE < 25 N M 0
MM 65 # TARGET.

CA PRI013
TC NOVAC
EBANK= ENTRYVN
2CADR P65.1

TC 2PHSCHNG # 2 PHASE CHG REQUIRED TO PREVENT RE-
OCT 00554 # STARTING FLASHING DISPLAY TWICE.
OCT 10035 # 4.55 SPOT AND SERVICER, HERE.
TC INTPRET
SSP RTB
 GOTOADDR # CHANGE ENTRY MODE TO UPCONTRL.
 UPCONTRL
 REFAZE10 # GO HERE TO REESTABLISH ENTRY SEQUEL
 # AND CONTINUE AT UPCONTRL...

P65.1 TC DOWNFLAG
ADRES ENTRYDSP # ENTRYDSP = 92D BIT 13 FLAG 6

CA V16N69 # ROLLC DL (Q7) VL
TC BANKCALL # XXX.XX DEG XXX.XX G XXXXX
CADR GOFLASHR
TC -3 # NODOFLAG IS SET ...
TC +3

```

July 12, 2016

1355

```
TC -5
TC P61.3 # EST. GRP 4 FOR DISPLAY AND DO ENDOFJOB
 # IF PROCEED, CONTINUE

TC UPFLAG
ADRES ENTRYDSP # ENTRYDSP = 92D BIT 13 FLAG 6

TC P63.1 # DISABLE GRP 4, START UP ENTRY DISPLAY
 # NO6V68 VIA OVERNOUT, AS USED IN P64

V16N69 VN 1669
```

# Page 799

```
PROGRAM: P66
MOD NO: 0 MOD BY: RR BAIRNSFATHER DATE: 17 JAN 68 COLOSSUS GSOP ADDITIONS
FUNCTION: KEEP CM ATTITUDE IN TRIM TO THE RELATIVE VELOCITY VECTOR. ENTRY GUIDANCE STOPS
ROLL COMMANDS UNTIL DRAG BUILDS UP TO Q7+0.5 FPSS.
CALLING SEQUENCE: VIA RTB FROM REENTRY CONTROL.
EXIT: BACK TO REENTRY CONTROL.
SUBROUTINE CALLS: NEWMODEX
```

COUNT\* \$\$/P66

```
P66 TC NEWMODEX # ENTER VIA RTB WHEN D < Q7 FPSS
 MM 66

 CA V06N22 # OGA IGA MGA
 # XXX.XX DEG XXX.XX DEG XXX.XX DEG
 TC P66END # IN CASE CAME FROM P65, GO TO DISABLE GRP 4,
 # AND SET ENTRYDSP TO DO DISPLAY VIA
 # OVERNOUT.

 # ... AND CONTINUE AT KEP2
```

# Page 800

```
PROGRAM: P67
MOD NO: 0 MAR. 16, 1967
MOD BY: R. HIRSCHKOP
FUNCTION: TO TERMINATE STEERING WHEN THE CM VELOCITY WRT EARTH = 1000 FT/SEC
CALLING SEQUENCE:
EXIT: TO POOH
SUBROUTINE CALLS: GOFLASH
```

# THIS DISPLAY IS CALLED EACH PASS THROUGH STEERING. RESTART PROTECTION IS VIA STEERING.

COUNT\* \$\$/P67

```
P67 TC NEWMODEX # ENTER VIA RTB
```

1356

July 12, 2016

|            |        |          |                                       |                  |
|------------|--------|----------|---------------------------------------|------------------|
|            | MM     | 67       |                                       |                  |
|            | CA     | V06N66   | # ROLLC                               | XRNGERR DNRN     |
|            |        |          | # XXX.XX DEG                          | XXXX.X NM XXXX   |
| P66END     | TS     | ENTRYVN  | # DISPLAY VIA                         | OVERNOUT.        |
|            | TC     | UPFLAG   | # (IN CASE CAME FROM P65.             | ENTRY DIS        |
|            | ADRES  | ENTRYDSP | # WILL FLUSH FLASHING DISP.           | IF STILL         |
|            |        |          | # BIT 13 FLAG 6                       |                  |
| KILLGRP4   | TC     | PHASCHNG | # DISABLE GRP4, IN CASE CAME FROM HUI |                  |
|            | OCT    | 00004    | # (COME TO KILLGRP4 VIA RTB, RET TO C |                  |
|            | TC     | DANZIG   | # ... AND CONTINUE AT PREDICT3 ...    |                  |
| V06N66     | VN     | 0666     |                                       |                  |
|            | BANK   | 26       |                                       |                  |
|            | SETLOC | P60S2    |                                       |                  |
|            | BANK   |          |                                       |                  |
| P67.1      | CA     | V16N67   | # RTOGO                               | LAT LONG         |
|            |        |          | # XXXX.X NM                           | XXX.XX DEG XXX.X |
|            | TC     | BANKCALL |                                       |                  |
|            | CADR   | GOFFLASH |                                       |                  |
|            | TC     | +3       | # EFFECTIVE GOTOP00H                  |                  |
|            | TC     | +2       |                                       |                  |
|            | TC     | P67.1    | # REDO                                |                  |
|            | CS     | THREE    | # TURN OFF ENTRY DAP                  |                  |
|            | INHINT |          |                                       |                  |
|            | MASK   | CM/FLAGS | # CM/DSTBY, GAMDIFSW                  |                  |
|            | TS     | CM/FLAGS |                                       |                  |
|            | RELINT |          |                                       |                  |
|            | EXTEND |          |                                       |                  |
|            | DCA    | SERVCAD2 |                                       |                  |
| # Page 801 |        |          |                                       |                  |
|            | DXCH   | AVEGEXIT |                                       |                  |
|            | TCF    | GOTOP00H |                                       |                  |
| # Page 802 |        |          |                                       |                  |
| P67.2      | VLOAD  | CLEAR    | # CALC PRESENT LAT, LONG, ALT.        |                  |
|            |        | RN       |                                       |                  |
|            |        | ERADFLAG | # USE PAD RAD FOR ALT. (NOT SEEN ANY  |                  |
|            | STODL  | ALPHAV   |                                       |                  |
|            |        | PIPTIME  | # USE TIME OF RN                      |                  |

July 12, 2016

1357

```

 CLEAR CALL
 LUNAFLAG
 LAT-LONG
P67.3 RTB # ENTRY EXIT THAT OMITTS DISPLAY.
 SERVNOU

V16N67 VN 1667
OCT41 = 33DEC
SERVCAD2 = SERVCAD1

Page 803
SUBROUTINE NAME: S61.1
MOD NO: 0 DATE: 21 FEB 67
MOD BY: RR BAIRNSFATHER LOG SECTION: P61-P67
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 22 JUN 67 RESTART
#
FUNCTIONAL DESCRIPTION: CALLED BY BOTH P61 AND P62
FIRST, TEST TO SEE IF AVERAGEG IS ON. IF NOT, UPDATE THE STATE VECTOR TO PRESENT TIME
AND TURN ON AVERAGEG AT THAT TIME, AND CONTINUE. OTHERWISE CONTINUE: SEE IF IMU Y AXIS
WITHIN 30 DEG OF VAR. IF YES, EXIT SUBROUTINE S61.1. IF SO, SEE IF -Y AXIS OF IMU IS
30 DEG OF VAR. IF YES, DISPLAY ALARM: 01427 IMU REVERSED.
IF NO, DISPLAY ALARM: 01426 IMU UNSATISFACTORY.
IN EITHER OF THESE LAST 2 CASES, WAIT 10 SEC AND THEN EXIT SUBROUTINE S61.1.
#
REMARK: THERE WILL BE A SHORT 10 SEC DELAY IF AN ALARM EXIT IS TAKEN. THE DELAY FOR IMU
AS SHORT AS CAN BE MADE, BUT IS ARBITRARY SINCE IT DEPENDS ON THE AGE OF THE STATE VECTOR
#
CALLING SEQUENCE: CALL
S61.1
#
C(MPAC) UNSPECIFIED
PUSHLOC UNSPECIFIED
#
SUBROUTINES CALLED: LOADTIME, CSMPREC, TPAGTREE,
WAITLIST, JOBSLEEP, JOBWAKE, PREREAD, ALARM, GODSPR, BANKCALL, DELAYJOB
#
NORMAL EXIT MODES: RVQ
#
ALARMS: 01426 IMU UNSATISFACTORY
01427 IMU REVERSED
#
OUTPUT: POSSIBLE ALARMS
POSSIBLY TDEC1, RATT, VATT, RN, VN
#
ERASABLE INITIALIZATION REQUIRED:
AVEGFLAG AVERAGEG ON OR OFF LEFT BY SERVICE
```

```

PIPTIME (-28) CS TIME OF PIPA UPDATE LEFT
RN (-29) M STATE VECTOR LEFT
VN (-7) M/CS STATE VECTOR LEFT
REFSMMAT (-1) .5 REF TO SM MATRIX LEFT
#
DEBRIS: QPRET
POSSIBLY PIPTIME1, RATT, VATT, TDEC1, RN1, VN1, QTEMP, X1 IF U
PUSH LIST LOCS USED BY CSMPREC

EBANK= AOG # FOR 60GENRET, S61DT
BANK 26
SETLOC P60S3
BANK

COUNT* $$/S61.1

S61.1 EXTEND
 QXCH 60GENRET # SAVE RET ADDR IN EB 6
 TC BANKCALL
 CADR R02BOTH
 TC INTPRET

Page 804

BON CALRB
 AVEGFLAG # IS AVERAGEG ON
 S61.1A # YES
 MIDTOAV2 # GET FUTURE STATE VECTOR SOON AS CAN

CA MPAC +1 # RETURN INHINTED ***
TS S61DT # FOR RESTART.
TC WAITLIST
EBANK= DVCNTR
2CADR S61.1C

TC PHASCHNG
OCT 40434
TC ENDOFJOB

S61.1C CA PRI013
 TC FINDVAC
 EBANK= AOG
 2CADR S61.1A -1

 EXTEND
 DCA SERVCAD1 # HE WHO START AVERAGEG MUST SERVICE
 DXCH AVEGEXIT # THE EXIT.

```

July 12, 2016

1359

|            |       |           |                             |
|------------|-------|-----------|-----------------------------|
|            | TC    | 2PHSCHNG  |                             |
|            | OCT   | 00454     |                             |
|            | OCT   | 00415     |                             |
|            | CA    | EBENTRY   | # SET EB= 7 FOR PREREAD.    |
|            | TS    | EBANK     |                             |
|            | TC    | POSTJUMP  |                             |
|            | CADR  | PREREAD   | # PREREAD DOES TC TASKOVER. |
| S61.1A     | TC    | INTPRET   |                             |
|            | BOVB  | VLOAD     |                             |
|            |       | TCDANZIG  | # TURN OFF OVFind, IF ON    |
|            |       | VN        | # VN (-7) M/CS              |
|            | VXV   | MXV       |                             |
|            |       | RN        | # RN (-29) M                |
|            |       | REFSMMAT  | # .5 UNIT MATRIX            |
|            | UNIT  | DLOAD     |                             |
|            |       | MPAC +3   | # GET COS(THETA)/2          |
|            | BMN   | DAD       |                             |
|            |       | S61.1B    | # DO TEST ON -YSM           |
|            |       | C(30)LIM  | # = 1.0 -.5 COS(30)         |
|            | BOVB  | RTB       |                             |
|            |       | RETRN1    |                             |
|            |       | RETRN3    |                             |
| # Page 805 |       |           |                             |
| S61.1B     | DCOMP | DAD       |                             |
|            |       | C(30)LIM  | # = 1.0 - .5 COS(30)        |
|            | BOVB  | EXIT      |                             |
|            |       | RETRN2    |                             |
| RETRN3     | TC    | ALARM     |                             |
|            | OCT   | 01426     | # IMU UNSATISFACTORY        |
|            | TC    | RETRN2 +2 |                             |
| RETRN2     | TC    | ALARM     |                             |
|            | OCT   | 01427     | # IMU REVERSED              |
| +2         | CAF   | V05N09    |                             |
|            | TC    | BANKCALL  |                             |
|            | CADR  | GODSPR    | # DO DISPLAY                |
|            | CA    | 10SECS    |                             |
|            | TC    | BANKCALL  |                             |
|            | CADR  | DELAYJOB  |                             |
| RETRN1     | TC    | 60GENRET  |                             |

```

EBANK= DVCNTR
SERVCAD1 2CADR SERVEXIT

C(30)LIM 2DEC .566985 # = 1.0 - .5 COS(30)

10SECS DEC 1000 # 1000 CS
60SECDP 2DEC 6000 B-28 # 6000 CS

Page 806
PROGRAM NAME: S61.2 DATE: 14 FEB 67
MOD NO: 1 LOG SECTION: P61-P67
MOD BY: NORTH / BAIRNSFATHER
MOD NO: 2 MOD BY: NORTH/BAIRNSFATHER DATE: 11 MAY 67 ADD 2ND ITER
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 21 NOV 67 VARIABLE MU A
MOD NO: 4 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 68 DIFFERENT EAD
#
FUNCTIONAL DESCRIPTION: CALLED IN P61. PROVIDES DISPLAYS FOR NOUNS N60 AND N63
PROGRAM CALCULATES ENTRY DISPLAY OF MAXIMUM ACCELERATION EXPECTED (GMAX) AND
INERTIAL VELOCITY (VPRED) AND ENTRY ANGLE (GAMMAEI) THAT WILL OBTAIN AT 400
ELLIPSOID. PROGRAM ALSO CALCULATES A SECOND DISPLAY RELATIVE TO THE EMSALT
AND CONSISTS OF RANGE TO SPLASH FOM NOW (RTGO) , PREDICTED INERTIAL VELOCITY
GO FROM NOW (TTE) .
#
CALLING SEQUENCE: CALL
S61.2
C(MPAC) UNSPECIFIED
PUSHLOC WILL BE SET TO ZERO.
#
SUBROUTINES CALLED: TFFCONIC, CALCTFF, TFF/TRIG, FISHCALC, GETERAD, VGAMCALC
#
NORMAL EXIT MODES: RTB, P61.1
#
ALARMS: NONE
#
OUTPUT: THE FOLLOWING REGISTERS ARE WRITTEN IN FOR USE BY DISPLAYS
GMAX 100 GMAX (-14) G,S MAXIMUM ACCELERATION
VPRED (-7) M/CS PREDICTED VELOCITY AT 400K FT
GAMMAEI GAMMA/360 PREDICTED GAMMA AT 400K FT
FOR TM, DP(GAMMAEI) = (GAMMAEI, RTGO)
RTGO THETAH/360 RANGE ANGLE TO SPLASH FROM EMSALT
VIO (-7) M/CS INTERTIAL VELOCITY AT EMSALT
TTE (-28) CS TIME TO EMSALT
PUSHLOC = 0
CONIC PARAMETERS STORED IN VAC AREA (SEE TFF SUBROUTINES)
#

```



July 12, 2016

1361

# ERASABLE INITIALIZATION REQUIRED:

|   |        |            |                                                 |         |
|---|--------|------------|-------------------------------------------------|---------|
| # | RONE   | (-29) M    | STATE VECTOR                                    | LEFT BY |
| # | VONE   | (-7) M/CS  | STATE VECTOR                                    | LEFT BY |
| # | URONE  | UR/2       |                                                 | LEFT BY |
| # | UNI    | (-1)       | UNIT NORMAL V*R                                 | LEFT BY |
| # | THETAH | THETAH/360 | RANGE ANGLE                                     | LEFT BY |
| # | UNITW  | (0)        | UNIT POLAR VECTOR                               | LEFT BY |
| # | EMSALT | (-29) M    | EMS INTERFACE ALTITUDE                          | LEFT BY |
| # |        |            | ORBITAL REENTRY: 284843 FT., LUNAR REENTRY: 297 |         |

# DEBRIS: QPRET,  
# ALL PDL LOCATIONS ABOVE 12D, INCLUDING X1,X2,S1,S2  
# ALSO PDL+0 ... PDL+5, WHERE INITIAL PUSHLOC = PDL

# Page 807

# THE FOLLOWING PUSH LIST LOCATIONS HAVE BEEN RESERVED FOR TFF ROUTINES AND ARE REPEATED HERE F  
# OF COURSE FOR S61.2 USAGE, EARTH ORIGIN SCALING IS USED.

|   |           |   |       |                                    |              |              |
|---|-----------|---|-------|------------------------------------|--------------|--------------|
| # |           |   | BELOW | E: IS USED FOR EARTH ORIGIN SCALE  |              |              |
| # |           |   |       | M: IS USED FOR MOON ORIGIN SCALE   |              |              |
| # |           |   |       |                                    |              |              |
| # | RTERM     | = | 18D   | TERMINAL RADIUS M                  | E: (-29)     | M: (-27)     |
| # | NRTERM    | = | 16D   | TERMINAL RADIUS M                  | E: (-29+NR)  |              |
| # |           |   |       |                                    | M: (-27+NR)  |              |
| # | RMAG1     | = | 12D   | PRESENT RADIUS M                   | E: (-29)     | M: (-27)     |
| # | NRMAG     | = | 32D   | PRESENT RADIUS M                   | E: (-29+NR)  |              |
| # |           |   |       |                                    | M: (-27+NR)  |              |
| # | SDELF/2   |   |       | SIN(THETA) / 2                     |              |              |
| # | CDELF/2   | = | 14D   | COS(THETA) / 2                     |              |              |
| # | TFFX      | = | 34D   | X, ARGUMENT OF SERIES T(X)         |              |              |
| # | TFFTEM    | = | 36D   | ARG FOR TRANSFER ANGLE CALCULATION |              |              |
| # | TFFNP     | = | 28D   | LC P M                             | E: (-38+2NR) | M: (-36+2NR) |
| # | TFF/RTMU= |   | 30D   | 1/SQRT(MU)                         | E: (17)      | M: (14)      |
| # | TFFVSQ    | = | 20D   | -(VN.VN/MU) 1/M                    | E: (20)      | M: (18)      |

# Page 808

BANK 34  
SETLOC P60S2  
BANK  
COUNT\* \$\$/S61.2

# PDL LEFT AT ZERO BY TARGETING

S61.2 DLOAD DSU  
EMSALT  
290KFT

|         |                |                           |                                                                                                       |
|---------|----------------|---------------------------|-------------------------------------------------------------------------------------------------------|
|         | BPL            | DLOAD<br>LUNENT<br>1/RTMU | # ESTABLISH MU FOR ORBITAL ENTRIES                                                                    |
| CALLCON | CALL           | TFFCONIC                  | # FILL VAC AREA WITH CONIC PARAMETERS                                                                 |
|         | DLOAD          | CALL<br>RTRIAL<br>CALCTFF | # 1ST GUESS AT TERMINAL RADIUS (-29)<br># SAVES MPAC IN RTERM (18D)                                   |
|         | CALL           | TFF/TRIG                  | # CALC SDELFF/2, CDELFF/2<br># RETURN WITH S(THETA) IN MPAC                                           |
|         | CALL           | FISHCALC                  | # GET FISCHER RADIUS (-29) M<br># ANS IN MPAC AND IN ERADM                                            |
|         | DAD            | CALL<br>EMSALT<br>CALCTFF | # SAVES MPAC IN RTERM (18D)                                                                           |
|         | DCOMP<br>STORE | TTE1                      | # NEGATIVE AS IN COUNTDOWN<br># DECR TTE FROM BASB TTE1. (RESTART)<br># DNLIST AND DSKY WILL USE TTE. |
|         | STCALL         | TTE                       | # LET MISS CONTRL DECR BY ELAPSED TIME<br># TTE= TIME FROM NOW TO EMSALT +FISCHER                     |
|         |                | TFF/TRIG                  | # S(THETA) IN MPAC ON RETURNING<br># AND THETA= RANGE FROM NOW TO EMSALT                              |
|         | CALL           | FISHCALC                  |                                                                                                       |
|         | CALL           | VRCALC                    |                                                                                                       |
|         | CALL           | DISPTARG                  |                                                                                                       |
|         | CALL           | DISPTARG                  |                                                                                                       |
|         | STCALL         | RTGO                      |                                                                                                       |
|         |                | VGAMCALC                  |                                                                                                       |
|         | DMP            |                           | # MPAC = GAMMA<br># PDLO HAS VGAM.                                                                    |
|         | BDDV           | DAD<br>VEMSCON<br>0       | # -HS D 180/PI (-14)<br># VGAM FROM PDLO                                                              |

July 12, 2016

1363

```
STODL VIO # PREDICTED VELOCITY AT EMSALT.

 # GAMMA AND VGAM AT 300K FT ARE REQUIRED BY GMAX
 # ALGORITHM.

 ERADM # EARTH RADIUS FROM GETERAD (-29) M
 # = FISCHER RADIUS (-29)

DAD
 300KFT # M (-29)
STCALL RTERM # TERMINAL RADIUS M (-29)

 PREVGM # VGAMCALC WITH NEW RTERM

 # VBAR = (V(FPS) - 36KF/S) / 20 F/S
GMAX = (4/(1 + 4.8 VBARSQ))(GAM - 6.05 - 2.4 VBARSQ) - 10(L/D - .3) + 10 ASSUME L/D = 0.

GMAXCALC
PDDL DSU # GAM TO PDL2
 0 # VGAM IS IN PDL0 (-7)
 36KFT/S # (-7) M/CS
DDV DSQ #
 20KFT/S # (-6) M/CS
STORE 0 # VBARSQ (-2) TO PDL0

DMP DAD
 KR1
 # GAM, POS DOWN, FROM PDL2
DAD DMP
 -6.05DEG
 KR2
PDDL DAD # XCH PDL+0 FOR VBARSQ (-2)
DDV DAD
 KR4
 DP2(-4)
BDDV
 # NUM FROM PDL+0
DAD BPL
 KR3
 +3
DLOAD
 HI6ZEROS
STODL GMAX # 100 GMAX (-14)

Page 810
DISPLAY USES GMAX AS SP, SO LO WORD IS WRITTEN OVER BY VPRED.
 ERADM # = FISCHER RADIUS (-29) M
```

|          |        |                                             |                                             |
|----------|--------|---------------------------------------------|---------------------------------------------|
|          | DAD    | CALL                                        | # 2 ND ITERATION FOR FISCHER RADIUS         |
|          |        | 400KFT                                      |                                             |
|          |        | CALCTFF                                     | # ESTABLISH TRANSFER ANGLE DATA.            |
|          | CALL   |                                             |                                             |
|          |        | TFF/TRIG                                    | # GET SIN, COS DELF                         |
|          | CALL   |                                             |                                             |
|          |        | FISHCALC                                    | # GET CORRESPONDING FISCHER RADIUS.         |
|          | DAD    | LXA,2                                       | # SAVE HI-WORD FOR DOWNLIST.                |
|          |        | 400KFT                                      | # M (-29)                                   |
|          |        | RTGO                                        | # (RANGE ANGLE FROM EMSALT) / 360           |
|          | STCALL | RTERM                                       |                                             |
|          |        | PREVGAM                                     | # VGAMCALC WITH NEW RTERM                   |
|          | DCOMP  | SXA,2                                       | # HI-WORD OF EACH ON DOWNLIST.              |
|          |        | MPAC +1                                     |                                             |
|          | STODL  | GAMMAEI                                     | # CONIC GAMMA/360 AT 400K FT. (HI-WORD)     |
|          |        |                                             | # CONIC RTGO/360 FROM EMSALT (LOW-WORD)     |
|          |        |                                             | # FOR TM, DP(GAMMAEI) = (GAMMA, RTGO) / 360 |
|          |        |                                             | # VGAM FROM PDL+0 (-7)                      |
|          | STADR  |                                             |                                             |
|          | STORE  | VPRED                                       | # CONIC VELOCITY AT 400K FT                 |
|          | RTB    |                                             |                                             |
|          |        | P61.1                                       |                                             |
|          |        |                                             | # PDL BACK TO ZERO.                         |
| LUNENT   | DLOAD  | GOTO                                        |                                             |
|          |        | 1/RTMUE                                     | # ESTABLISH MU FOR LUNAR TYPE ENTRIES       |
|          |        | CALLCON                                     |                                             |
| 290KFT   | 2DEC   | 88392.0 B-29                                |                                             |
| KTETA1   | 2DEC*  | .421844723 E2 B-14* # 110 2PI/16384(163.84) |                                             |
| 36KFT/S  | 2DEC   | 109.728 B-7                                 | # (-7) M/CS = 36 KFT/S (-7)                 |
| 20KFT/S  | 2DEC   | 121.92 B-7                                  | # (-6) M/CS = 2 20KFT/S (-7)                |
| KR1      | 2DEC   | -.026666667                                 | # = -2.4 4 / 360                            |
| -6.05DEG | 2DEC   | -.016805556                                 | # = -6.05 / 360                             |
| KR2      | 2DEC   | .54931641                                   | # = (360/4) 100 (-14) = 9000 B-14           |
| KR3      | 2DEC   | 1000 B-14                                   | # = 100 (10.0) (-14) G,S                    |

July 12, 2016

1365

# Page 811

```
ASSUMES L/D = 0.3, BANK =0.
RTRIAL 2DEC 6460097.18 B-29 # RPAD +264643 FT =21 194 545 FT
400KFT 2DEC 121920 B-29 # RPAD DEFINED AS 20 909 901.57 FT =6 373 336 M
METERS
300KFT 2DEC 91440 B-29 # (-29) M
EMSALT 2DEC 86759.2 B-29 # 284643 FT (-29) M (ORBITAL REENTRY)
EMSALT 2DEC 90657 B-29 # 297431 FT (-29) M (LUNAR REENTRY)
KR4 2DEC .833333333
300KFT EQUALS MINPERE
VEMSCON 2DEC -.0389676 B-14 # = -HS D / 2 PI (-14) M SQ / CS SQ
= -16369 .05G 32.2 .3048 .3048/2 PI
```

# Page 812

```
SUBROUTINE NAME: FISHCALC (USED BY S61.2) DATE: 01.21.67
MOD NO: 0 LOG SECTION: P61-P67
MOD BY: MORTH / BAIRNSFATHER
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 11 MAY 67 INCLUDE
#
FUNCTIONAL DESCRIPTION: GIVEN THE PRESENT POSITION, UNITR, CALCULATE A NEW UNITR THAT IS ROT
TRANSFER ANGLE, THETA, ALONG THE TRAJECTORY. THEN CALCULATE SIN(LAT) AND USE TO OBTAIN
SINCE FISHCALC USED UNI (LEFT BY ENTRY) EARTH SCALING IS ASSUMED. (WILL IMPROVE FOR SU
#
CALLING SEQUENCE: CALL
FISHCALC
ENTER WITH .5 SIN(THETA) IN MPAC.
PUSHLOC IS AT PDL+0, AN ARBITRARY BASE VALUE IF LEQ 8D
#
SUBROUTINES CALLED: GET ERAD
#
NORMAL EXIT MODE: RVQ
#
EXIT MODES: NONE
#
OUTPUT: ERADM (-29) M IN MPAC ON RETURNING
NEW UNIT VECTOR NOT SAVED.
SIN(LAT) NOT SAVED.
PUSHLOC AT PDL+0
#
ERASABLE INITIALIZATION REQUIRED:
```

```

SDELF/2 =SIN(THETA) / 2, IN MPAC LEFT BY TFF/7
CDELF/2 =COS(THETA) / 2, STORED IN PDL 14D LEFT BY TFF/7
RONE (-29) M LEFT BY USER
VONE (-7) M/CS LEFT BY USER
URONE UR/2 LEFT BY USER
UNI .5 UNIT(V*R) LEFT BY ENTRY
UNITW UNIT NORTH POLE LEFT BY PAD 1
#
DEBRIS: QPRET, PDL+0 ... PDL+5
#
FISHCALC PDVL VXV # - - -
 URONE # URPR = UR CDELF + UHOR SDELF
 UNI
 VXSC VSL1
 PDVL VXSC # SIN(THETA) / 2 FROM PDL+0
 URONE # TO PDL+0, +5
 CDELF/2 # COS(THETA) / 2
 VAD STADR
 STORE URH # FOR USE IN RTGO FROM EMS DISPLAY
 DOT SL1
 UNITW # PULL UNIT VECTOR UNIT NORTH
 STORE ALPHAV +4 # = .5 SIN(LAT)
DUMPFISH GOTO
 GETERAD # SAVES FISCHER RAD (-29) M IN ERADM AND
 # IN MPAC. RETURNS TO CALLER VIO QPRET.

Page 813
SUBROUTINE NAME: VGAMCALC (USED BY S61.2)
MOD NO: 0
MOD BY: MORTH / BAIRNSFATHER
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 11 APR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 21 NOV 67 VARIABLE MU A
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 68 ACCEPT DIFFER

#
FUNCTIONAL DESCRIPTION: EARTH CENTERED VIS VIVA CALCULATION OF TERMINAL VELOCITY AND
HORIZONTAL) GIVEN THE SCALAR QUANTITIES: PRESENT RADIUS AND VELOCITY AND TH
THE USER MUST APPEND PROPER SIGN TO GAMMA, SINCE IT IS CALCULATED AS A POSIT
THE EQUATIONS ARE
#
VGAM = SQRT(VN VN/MU + 2(RN-RTERM)/(RN RTERM)) RTMU
#
COSGAM = H / RTERM VGAM = SQRT (LCP) / (RTERM VGAM/RTMU)
#
VGAMCALC ASSUMES THAT THE TERMINAL RADIUS IS LESS THAN THE PRESENT RADIUS. F
MAKE THIS ASSUMPTION.

```

```

#
CALLING SEQUENCE: CALL STCALL RTERM
VGAMCALC PREVGAM
PUSHLOC AT PDL+0, ARBITRARY IF LEQ 12D
C(MPAC) UNSPECIFIED C(MPAC)=NEW RTERM
#
SUBROUTINES CALLED: NONE
#
NORMAL EXIT MODE: RVQ
#
ALARMS: NONE
#
OUTPUT: GAMMA / 360 IN MPAC, POSITIVE NUMBER
VGAM E: (-7) M: (-5) M/CS IN PDL+0
PUSHLOC AT PDL+2
#
ERASABLE INITIALIZATION REQD:
TFF/RTMU E: (17) M: (14) 1/SQRT(MU) LEFT BY TFFCONI
RMAG1 E: (-29) M: (-27) M PRESENT RADIUS LENGTH LEFT BY TFFCONI
NRMAG E: (-29+NR) M NORM LENGTH OF PRESENT POSITION LEFT BY TFFCONI
M: (-27+NR)
RTERM E: (-29) M: (-27) M TERMINAL RADIUS LENGTH LEFT BY CALCTFF
NRTERM E: (-29+NR) M NORM LENGTH OF TERMINAL RADIUS LEFT BY CALCTFF
M: (-27+NR)
TFFVSQ E: (20) M: (18) 1/M -(V SQ/MU): PRESENT VELOCITY, NORM LEFT BY TFFCONI
TFFNP E: (-38+2NR) M LCP, SEMI-LATUS RECTUM, WEIGHT NR LEFT BY TFFCONI
M: (-36+2NR)
#
DEBRIS: QPRET, PDL+0 ... PDL+3
RTERM, NRTERM IF PREVGAM ENTERED.
Page 814

PREVGAM SL* # ENTER WITH NEW RTERM IN MPAC
E: (-29) M: (-27)
0,1 # X1 = -NR
STORE NRTERM # RTERM M E: (-29+NR) M: (-27+NR)

VGAMCALC DLOAD DMP
NRMAG # RMAG M E: (-29+NR) M: (-27+NR)
NRTERM # RTERM M E: (-29+NR) M: (-27+NR)
PDDL DSU # RMAG RTERM M E: (-58+2NR) M: (-54+2NR)
NRMAG # RMAG M E: (-29+NR) M: (-27+NR)
NRTERM # RTERM M E: (-29+NR) M: (-27+NR)
SL* DDV # 2(RN-RTERM) E: (-30+NR) M: (-28+NR)
0 -8D,1 # (-8+NR)
PUSH UP PRODUCT.

```

```

DSU
 TFFVSQ # -(V SQ/MU) E: (20) M: (
SQRT PUSH # SAVE VGAM/RT(MU) FOR NOW. E: (10) M: (
DDV PDDL # XCH PDL+0, LEAVING VGAM FOR OUTPUT.
 # VGAM TO PDL M/CS E: (-7) M: (
 TFF/RTMU # E: (17) M: (
DMP PDDL # RTERM VGAM/RTMU E: (-19+NR) M: (
 NRTERM # RTERM M E: (-29+NR) M: (
 TFFNP # LC P =H.H/MU M E: (-38+2NR) M: (
SQRT DDV # E: (-19+NR) M: (
 # PUSH UP DEN E: (-19+NR) M: (
 # USE DDV OVFL AS LIMITER (|COS| <1.0)

SR1 ACOS
DUMPVGAM RVQ

 # CALLER MUST SUPPLY OWN SIGN ...
 # 22W 27MS

Page 815
SUBROUTINE NAME: TFF/TRIG (USED BY S61.2) DATE: 01.17
MOD NO: 0 LOG SECTION: P61-1
MOD BY: RR BAIRNSFATHER
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 14 APR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 68 ACCEPT DIFFER
#
FUNCTIONAL DESCRIPTION: USED BY ENTRY DISPLAY TO CALCULATE SIN(THETA), COS(THETA)
PDL BY TFF SUBROUTINES. THE EQNS ARE
#
COS(THETA) = 1-2 ABS(ARG) / (RN RTERM (1+X))
2
SIN(THETA) = SGN(ARG) SQRT(1-COS (THETA))
#
WHERE THETA = TRANSFER ANGLE
AND ARG = P Z ABS(Z) IF ALFA ZZ LEQ 1
ARG = (P / ALFA) SGN(Q1 + R 1/Z) IF ALFA Z Z G 1
AND ARG HAS BEEN AFFIXED WITH THE SIGN OF SIN(THETA)
#
CALLING SEQUENCE: CALL
TFF/TRIG
PUSHLOC AT PDL+0, ARBITRARY IF NOT EQ 14D
C(MPAC) UNSPECIFIED
#
SUBROUTINES CALLED: NONE
#
NORMAL EXIT MODES: RVQ
#
ALARMS: NONE

```



July 12, 2016

1369

```
#
OUTPUT: C(MPAC) = .5 SIN(THETA)
CDELF/2 = .5 COS(THETA) (IN PDL 14D)
PUSHLOC AT PDL+0
#
ERASABLE INITIALIZATION REQUIRED:
TFFX X LEFT BY CALCTFF
TFFTEM E: (-59+2NR) ARG LEFT BY CALCTFF
M: (-55+2NR) WHERE ARG = LCF ZZ SGN(DELF) OR ARG = LCP/ALFA SGN(DELF)
NRTERM E: (-29+NR) M NORM LENGTH OF TERMINAL RADIUS LEFT BY CALCTFF
M: (-27+NR)
NRMAG E: (-29+NR) M NORM LENGTH OF PRESENT POSITION LEFT BY TFFCONI
M: (-27+NR)
#
DEBRIS: QPRET, CDELF/2
#
BANK 27
SETLOC P60S5
BANK
TFF/TRIG DLOAD SR1
TFFX
DAD DMP
HIDPHALF
NRMAG # RMAG M E: (-29+NR) M: (-27+NR)
DMP BDDV
NRTERM # RTERM M E: (-29+NR) M: (-27+NR)
TFFTEM # P ZSQ OR P/ALFA E: (-59+2NR) M: (-55+2NR)
ABS BDSU # THE SIGN IS FOR SDELF.
HIDPHALF
STORE CDELF/2 # .5 COS(THETA)
DSQ DCOMP # KEEP HONEST FOR SQRT.
#
Page 816
DAD SQRT
HIDP1/4
DUMPTRIG SIGN RVQ
TFFTEM # AFFIX SIGN(DELE/2)
RETURN WITH .5 SIN(THETA) IN MPAC
#
16W 15MS
#
DISPTARG STQ
60GENRET
DMP DSU
KTETA1
TTE1
STCALL DTEAROT
```

```

 EARROT2
 CALL
 VRCALC
 GOTO
 60GENRET
VRCALC VLOAD DOT
 URH
 RT
 SL2 ACOS
 RVQ

END OF PROGRAM S61.2

Page 817
PROGRAM DESCRIPTION S62.3 DATE 10JAN67
MOD NO 1: LOG SECTION P60-P67
MOD BY ZELDIN
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 15 MAY 67 CHANGED TO R
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 17 JAN 68 ALFAPAD CHAN
#
FUNCTIONAL DESCRIPTION
#
COMPUTE DESIRED GIMBAL ANGLES FOR ENTRY ATTITUDE
THE FOLLOWING TRAJECTORY TRIAD IS AVAILABLE IN MEMORY AND IS COMPUTED EACH 2
REFERENCE COORDINATES (V = VELOCITY RELATIVE TO EARTH):
#
UXA = -UNIT(V)
UYA = UNIT(V*R)
UZA = UXA*UYA
#
GENERATE A DESIRED BODY TRIAD FOR TRIMMED FLIGHT WITH RESPECT TO THE RELATIVE
ROLL COMMAND AND TRIM ANGLE OF ATTACK:
#
UXD = UNIT(UYD*UXA) SIN(ALFATRIM) + UXA COS(ALFATRIM)
UYD = UYA COS(ROLLC) + UZA SIN(ROLLC)
UZD = UXD * UYD
#
USE THE DESIRED SET (IN REFERENCE COORDS) AND REFSMMAT TO CALL CALCGA AND C
IN 2S, C IN MPAC, +2 AND THETAD, +2.
#
CALLING SEQUENCE
#
L CALL
L+1 S62.3
#
NORMAL EXIT MODE

```

```

#
RETURN VIA QPRET DIRECTLY FROM CALCGA.
#
SUBROUTINES CALLED
#
CALCGA
#
ALARM OR ABORT MODES
#
NONE
#
ERASABLE INITIALIZATION REQUIRED
#
ROLLC ROLL COMMAND DP 1'S COMP AT 1REV
ALFAPAD SP 1'S C / 180 LEFT BY PAD LOAD ALFATRIM IS NEGATIVE.
UXA/2 REF COORDS LEFT BY CM/POSE
UYA/2 REF COORDS LEFT BY CM/POSE
UZA/2 REF COORDS LEFT BY CM/POSE
#
OUTPUT
#
CPHI GIMBAL ANGLES (0,I,M) 2'S COMP TP (0,I,M)/180
#
DEBRIS
#
QTEMP, QPRET, PUSHLIST
#
BANK 10
SETLOC P60S4
BANK
#
Page 818
#
COUNT* $$/S62.3
#
S62.3
SETPD SLOAD
0
ALFAPAD # ALFATRIM / 180, ALFA IS NEG.
SR1 PUSH
COS PDDL # XCH PDL, COS TO PDLO
SIN PDDL # SIN TO PDL2
ROLLC
COS VXSC
UYA/2
PDDL # PUSH VECTOR INTO PDL4...9
SIN
ROLLC
VXSC VAD
UZA/2
REF COORDS

```

```

VECTOR FROM PDL4...9
VSL1
STORE YNB # = UYD REF COORDS

VXV VSL1
 UXA/2 # REF COORDS
VXSC PDDL

SIN TRIM FROM PDL2
XCH PDLO FOR COS TRIM
VXSC VAD
 UXA/2 # REF COORDS
FROM PDLO

VSL1
STORE XNB # X SC AXIS (.5 UNIT) REF COORDS

VXV VSL1
 YNB
STOVL ZNB # Z SC IN REF COOR. SCALED AT 2
 REFSMMAT
STOVL XSM
 REFSMMAT +6
STOVL YSM
 REFSMMAT +12D
STORE ZSM

CLEAR GOTO
 CPHIFLAG # CAUSE CALCGA TO STORE ANS IN TP CPHI
 CALCGA

CALCGA WILL RETURN TO ORIGINAL CALLER
VIA QPRET WITH 2'S COMP. ANGLES IN CPHI

```

This code is written to file `src/P61-P67.s`.

**B.77 P70-P71**

1373

*<src/P70-P71.s 1373>*≡

```

Copyright: Public domain.
Filename: P70-P71.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
Website: www.ibiblio.org/apollo.
Pages: 829-837
Mod history: 2009-05-23 HG Transcribed from page images.
2009-06-05 RSB Fixed a typo.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969

```

# Page 829

```

 BANK 21
 SETLOC R11
 BANK
 EBANK= DVCNTR
 COUNT* $$/R11

R10,R11 CS FLAGWRD7 # IS SERVICER STILL RUNNING?
 MASK AVEGFBIT
 CCS A
 TCF TASKOVER # LET AVGENG TAKE CARE OF GROUP 2.
 CCS PIPCTR
 TCF +2
 TCF LRHTASK # LAST PASS. CALL LRHTASK.
+2 TS PIPCTR1

```

|            |        |          |                                        |
|------------|--------|----------|----------------------------------------|
| PIPCTR1    | =      | LADQSAVE |                                        |
| PIPCTR     | =      | PHSPRDT2 |                                        |
|            | CAF    | OCT31    |                                        |
|            | TC     | TWIDDLE  |                                        |
|            | ADRES  | R10,R11  |                                        |
| R10,R11A   | CS     | IMODES33 | # IF LAMP TEST, DO NTO CHANGE LR LITES |
|            | MASK   | BIT1     |                                        |
|            | EXTEND |          |                                        |
|            | BZF    | 10,11    |                                        |
| FLASHH?    | MASK   | FLGWRD11 | # C(A) = 1 - HFLASH BIT                |
|            | EXTEND |          |                                        |
|            | BZF    | FLASHV?  | # H FLASH OFF, SO LEAVE ALONE          |
|            | CA     | HLITE    |                                        |
|            | TS     | L        |                                        |
|            | TC     | FLIP     | # FLIP H LITE                          |
| FLASHV?    | CA     | VFLSHBIT | # VLASHBIT MUST BE BIT 2.              |
|            | MASK   | FLGWRD11 |                                        |
|            | EXTEND |          |                                        |
|            | BZF    | 10,11    | # V FLASH OFF                          |
|            | CA     | VLITE    |                                        |
|            | TS     | L        |                                        |
|            | TC     | FLIP     | # FLIP V LITE                          |
| 10,11      | CA     | FLAGWRD9 | # IS THE LETABORT FLAG SET ?           |
|            | MASK   | LETABBIT |                                        |
|            | EXTEND |          |                                        |
|            | BZF    | LANDISP  | # NO. PROCEED TO R10.                  |
| P71NOW?    | CS     | MODREG   | # YES. ARE WE IN P71 NOW?              |
| # Page 830 | AD     | 1DEC71   |                                        |
|            | EXTEND |          |                                        |
|            | BZF    | LANDISP  | # YES. PROCEED TO R10.                 |
|            | EXTEND |          | # NO. IS AN ABORT STAGE COMMANDED?     |
|            | READ   | CHAN30   |                                        |
|            | COM    |          |                                        |
|            | TS     | L        |                                        |
|            | MASK   | BIT4     |                                        |
|            | CCS    | A        |                                        |
|            | TCF    | P71A     | # YES.                                 |

July 12, 2016

1375

|            |        |          |                              |
|------------|--------|----------|------------------------------|
| P7ONOW?    | CS     | MODREG   | # NO. ARE WE IN P70 NOW?     |
|            | AD     | 1DEC70   |                              |
|            | EXTEND |          |                              |
|            | BZF    | LANDISP  | # YES. PROCEED TO R10.       |
|            | CA     | L        | # NO. IS AN ABORT COMMANDED? |
|            | MASK   | BIT1     |                              |
|            | CCS    | A        |                              |
|            | TCF    | P70A     | # YES.                       |
|            | TCF    | LANDISP  | # NO. PROCEED TO R10.        |
|            | COUNT* | \$\$/P70 |                              |
| P70        | TC     | LEGAL?   |                              |
| P70A       | CS     | ZERO     |                              |
|            | TCF    | +3       |                              |
| P71        | TC     | LEGAL?   |                              |
| P71A       | CAF    | TWO      |                              |
| +3         | TS     | Q        |                              |
|            | INHINT |          |                              |
|            | EXTEND |          |                              |
|            | DCA    | CNTABTAD |                              |
|            | DTCB   |          |                              |
|            | EBANK= | DVCNTR   |                              |
| CNTABTAD   | 2CADR  | CONTABRT |                              |
| 1DEC70     | DEC    | 70       |                              |
| 1DEC71     | DEC    | 71       |                              |
|            | BANK   | 05       |                              |
|            | SETLOC | ABORTS1  |                              |
|            | BANK   |          |                              |
|            | COUNT* | \$\$/P70 |                              |
| CONTABRT   | CAF    | ABRTJADR |                              |
|            | TS     | BRUPT    |                              |
|            | RESUME |          |                              |
| # Page 831 |        |          |                              |
| ABRTJADR   | TCF    | ABRTJASK |                              |
| ABRTJASK   | CAF    | OCTAL27  |                              |
|            | AD     | Q        |                              |
|            | TS     | L        |                              |

```

COM
DXCH -PHASE4
INDEX Q
CAF MODE70
TS MODREG

TS DISPDEX # INSURE DISPDEX IS POSITIVE.

CCS Q # SET APSFLAG IF P71.
CS FLGWRD10 # SET APSFLAG PRIOR TO THE ENEMA.
MASK APSFLBIT
ADS FLGWRD10
CS DAPBITS # DAPBITS = OCT 640 = BITS 6, 8, 9
MASK DAPBOOLS # (TURN OFF ULLAGE, DRIFT, AND KOVINHIB
TS DAPBOOLS

CS FLAGWRD5 # SET ENGONFLG.
MASK ENGONBIT
ADS FLAGWRD5

CS PRI030 # INSURE THAT THE ENGINE IS ON, IF ARMED.
EXTEND
RAND DSALMOUT
AD BIT13
EXTEND
WRITE DSALMOUT

CAF LRBYBIT # TERMINATE R12.
TS FLGWRD11

CS FLAGWRD0 # SET R10FLAG TO SUPPRESS OUTPUTS TO THE
MASK R10FLBIT # CROSS-POINTER DISPLAY.
ADS FLAGWRD0 # THE FOLLOWING ENEMA WILL REMOVE THE
 # DISPLAY INERTIAL DATA OUTBIT.
TC CLRADMOD # INSURE RADMODES PROPERLY SET FOR R29.

EXTEND
DCA TIME2
DXCH TEVENT # LOAD TEVENT FOR THE DOWNLINK.

EXTEND
DCA SVEXITAD
DXCH AVGEXIT

```

EXTEND



July 12, 2016

1377

DCA      NEG0  
DXCH     -PHASE1

EXTEND  
DCA      NEG0  
DXCH     -PHASE3

EXTEND  
DCA      NEG0  
DXCH     -PHASE6

CAF      THREE                    # SET UP 4.3SPOT FOR GOABORT  
TS       L  
COM  
DXCH     -PHASE4

# the 3 in OCT37774 could be something else, garbled

CAF      OCT37774                # SET T5RUPT TO CALL DAPIDLER IN  
TS       TIME5                    # 40 MILLISECONDS.

TC       POSTJUMP  
CADR     ENEMA

SVEXITAD      EBANK= DVCNTR  
                 2CADR   SERVEXIT

MODE70        DEC      70  
OCTAL27       OCT      27  
MODE71        DEC      71

DAPBITS       OCT      00640

BANK        32  
SETLOC      ABORTS  
BANK

COUNT\*      \$\$/P70

GOABORT       TC       INTERPRET  
                 CALL  
                 INITCDUW  
                 EXIT  
CAF          FOUR  
TS           DVCNTR  
CAF          WHICHADR

|            |       |          |                                        |
|------------|-------|----------|----------------------------------------|
|            | TS    | WHICH    |                                        |
|            | TC    | DOWNFLAG |                                        |
|            | ADRES | FLRCS    |                                        |
| # Page 833 | TC    | DOWNFLAG |                                        |
|            | ADRES | FLUNDISP |                                        |
|            | TC    | DOWNFLAG |                                        |
|            | ADRES | IDLEFLAG |                                        |
|            | TC    | UPFLAG   | # INSURE 4-JET TRANSLATION CAPABILITY. |
|            | ADRES | ACC4-2FL |                                        |
| 70DEC      | TC    | CHECKMM  |                                        |
|            | DEC   | 70       |                                        |
|            | TCF   | P71RET   |                                        |
| P70INIT    | TC    | INTPRET  |                                        |
|            | CALL  |          |                                        |
|            |       | TGOCOMP  |                                        |
|            | DLOAD | SL       |                                        |
|            |       | MDOTDPS  |                                        |
|            |       | 4D       |                                        |
|            | BDDV  |          |                                        |
|            |       | MASS     |                                        |
|            | STODL | TBUP     |                                        |
|            |       | MASS     |                                        |
|            | DDV   | SR1      |                                        |
|            |       | K(1/DV)  |                                        |
|            | STORE | 1/DV1    |                                        |
|            | STORE | 1/DV2    |                                        |
|            | STORE | 1/DV3    |                                        |
|            | BDDV  |          |                                        |
|            |       | K(AT)    |                                        |
|            | STODL | AT       |                                        |
|            |       | DTDECAY  |                                        |
|            | DCOMP | SL       |                                        |
|            |       | 11D      |                                        |
|            | STORE | TTO      |                                        |
|            | SLOAD | DCOMP    |                                        |
|            |       | DPSVEX   |                                        |
|            | SR2   |          |                                        |
|            | STORE | VE       | # INITIALIZE DPS EXHAUST VELOCITY      |
|            | SET   | CALL     |                                        |
|            |       | FLAP     |                                        |

July 12, 2016

1379

```

 COMMIT
 AXC,1 GOTO # RETURN HERE IN P70, SET X1 FOR DPS COEFF.
 OD
 BOTHPOLY
INJTARG AXC,1 # RETURN HERE IN P71, SET X1 FOR APS COEFF.
 8D
BOTHPOLY DLOAD* DMP # TGO D
 ABTCOF,1
 TGO
Page 834
 DAD* DMP
 ABTCOF +2,1 # TGO(C+TGO)
 TGO
 DAD* DMP
 ABTCOF +4,1 # TGO(B+TGO d))
 TGO
 DAD*
 ABTCOF +6,1 # A+TGO(B+TGO(C+TGO D))
 STORE ZDOTD # STORE TENTATIVELY IN ZDOTD
 DSU BPL # CHECK AGAINST MINIMUM
 VMIN
 UPRATE # IF BIG ENOUGH, LEAVE ZDOTD AS IS .
 DLOAD
 VMIN
 STORE ZDOTD # IF TOO SMALL, REPLCAE WITH MINIMUM.
UPRATE DLOAD
 ABTRDOT
 STCALL RDOTD # INITIALZE RDOTD.
 YCOMP # COMPUTE Y
 ABS DSU
 YLIM # /Y/-DYMAX
 BMN SIGN # IF <0, XR<.5DEG, LEAVE YCO AT 0
 YOK # IF >0, FIX SIGN OF DEFICIT. THIS IS YCO.
 Y
 STORE YCO
 DLOAD DSU
 YCO
 Y # COMPUTE XRANGE IN CASE ASTRONAUT WANTS
 SR
 5D
 STORE XRANGE # TO LOOK.
UPTHROT SET EXIT
 FLVR
 TC UPFLAG # SET ROTFLAG
 ADRES ROTFLAG

```

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | TC     | THROTUP  |                                            |
|            | TC     | PHASCHNG |                                            |
|            | OCT    | 04024    |                                            |
| -3         | TC     | BANKCALL | # VERIFY THAT THE PANEL SWITCHES           |
|            | CADR   | P40AUTO  | # ARE PROPERLY SET.                        |
|            | TC     | THROTUP  |                                            |
| UPTHROT1   | EXTEND |          | # SET SERVICER TO CALL ASCENT GUIDANCE.    |
|            | DCA    | ATMAGAD  |                                            |
|            | DXCH   | AVGEXIT  |                                            |
| # Page 835 |        |          |                                            |
| GRP40FF    | TC     | PHASCHNG | # TERMINATE USE OF GROUP 4.                |
|            | OCT    | 00004    |                                            |
|            | TCF    | ENDOFJOB |                                            |
| P71RET     | TC     | DOWNFLAG |                                            |
|            | ADRES  | LETABORT |                                            |
|            | CAF    | THRESH2  | # SET DVMON THRESHOLD TO THE ASCENT VALUE. |
|            | TS     | DVTHRUSH |                                            |
|            | TC     | INTPRET  |                                            |
|            | BON    | CALL     |                                            |
|            |        | FLAP     |                                            |
|            |        | OLDTIME  |                                            |
|            |        | TGOCOMP  | # IF FLAP=0, TGO=T-TIG                     |
|            | SSP    | GOTO     |                                            |
|            |        | QPRET    |                                            |
|            | CADR   | INJTARG  |                                            |
|            |        | P12INIT  | # WILL EXIT P12INIT TO INJTARG             |
| OLDTIME    | DLOAD  | SL1      | # IF FLAP=1,GTO=2 TGO                      |
|            |        | TGO      |                                            |
|            | STCALL | TG01     |                                            |
|            |        | P12INIT  |                                            |
|            | EXIT   |          |                                            |
|            | TC     | PHASCHNG |                                            |
|            | OCT    | 04024    |                                            |
|            | EXTEND |          |                                            |
|            | DCA    | TG01     |                                            |
|            | DXCH   | TGO      |                                            |

July 12, 2016

1381

```
TCF UPTHROT1 -3

TGO1 = VGBODY

BANK 21
SETLOC R11
BANK
COUNT* $$/P70

LEGAL? CS MMNUMBER # IS THE DESIRED PGM ALREADY IN PROGRESS?
 AD MODREG
 EXTEND
 BZF ABORTALM

 CS FLAGWRD9 # ARE THE ABORTS ENABLED?
 MASK LETABBIT
 CCS A

Page 836
TCF ABORTALM

 CA FLAGWRD7 # IS SERVICER ON THE AIR?
 MASK AVEGFBIT
 CCS A
 TC Q # YES. ALL IS WELL.
ABORTALM TC FALTON
 TC RELDSP
 TC POSTJUMP
 CADR PINBRNCH

BANK 32
SETLOC ABORTS
BANK

COUNT* $$/P70

TGOCOMP RTB DSU
 LOADTIME
 TIG
 SL
 11D
 STORE TGO
 RVQ
```

```

```

```
THROTUP CAF BIT13
 TS THRUST
 CAF BIT4
 EXTEND
 WOR CHAN14
 TC Q
```

```

```

```
10SECS 2DEC 1000
HINJECT 2DEC 18288 B-24 # 60,000 FEET EXPRESSED IN METERS.
(TGO)A 2DEC 37000 B-17
K(AT) 2DEC .02 # SCALING CONSTANT
WHICHADR REMADR ABRTABLE
```

```

```

```
Page 837
```

```
EBANK= DVCNTR
ATMAGAD 2CADR ATMAG
ORBMANAD ADRES ORBMANUV
```

This code is written to file `src/P70-P71.s`.

**B.78 P76**

1383

*<src/P76.s 1383>*≡

```

Copyright: Public domain.
Filename: P76.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
Website: www.ibiblio.org/apollo.
Pages: pp 511-513
Mod history: 2009-05-08 HG Adapting from the Luminary131/ file
of the same name, using Comanche055 page
images 0511.jpg - 0513.jpg.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 511
1) PROGRAM NAME -- TARGET DELTA V PROGRAM (P76).
2) FUNCTIONAL DESCRIPTION -- UPON ENTRY BY ASTRONAUT ACTION, P76 FLASHES DSKY REQUESTS TO
TO PROVIDE VIA DSKY (1) THE DELTA V TO BE APPLIED TO THE OTHER VEHICLE STATE VECTOR AND
TIME (TIG) AT WHICH THE OTHER VEHICLE VELOCITY WAS CHANGED BY EXECUTION OF A THRUSTING
OTHER VEHICLE STATE VECTOR IS INTEGRATED TO TIG AND UPDATED BY THE ADDITION OF DELTA V
BEEN TRANSFORMED FROM LV TO REF COSYS). USING INTEGRVS, THE PROGRAM THEN INTEGRATES TH
VEHICLE STATE VECTOR TO THE STATE VECTOR OF THIS VEHICLE, THUS INSURING THAT THE W-MATR
STATES CORRESPOND TO THE SAME TIME.
3) ERASABLE INITIALIZATION REQUIRED -- NONE.
4) CALLING SEQUENCES AND EXIT MODES -- CALLED BY ASTRONAUT REQUEST THRU DSKY V 37 E 76E.
EXITS BY TCF ENDOFJOB.
5) OUTPUT -- OTHER VEHICLE STATE VECTOR INTEGRATED TO TIG AND INCREMENTED BY DELTA V IN RE

```

```

THE PUSHLIST CONTAINS THE MATRIX BY WHICH THE INPUT DELTA V MUST BE POST-MULT
TO REF COSYS.
6) DEBRIS -- OTHER VEHICLE STATE VECTOR.
7) SUBROUTINES CALLED -- BANKCALL, GOXDSPF, CSMPREC (OR LEMPREC), ATOPCSM (OR AT
INTPRET, INTEGRVS, AND MINIRECT.
8) FLAG USE -- MOONFLAG, CMOONFLG, INTYPFLG, RASFLAG, AND MARKCTR.

```

```

 BANK 30
 SETLOC P76LOC
 BANK

 COUNT* $$/P76

 EBANK= TIG

P76 TC UPFLAG
 ADRES TRACKFLG

 CAF V06N84 # FLASH LAST DELTA V.
 TC BANKCALL # AND WAIT FOR KEYBOARD ACTION.
 CADR GOFLASH
 TCF ENDP76
 TC +2 # PROCEED
 TC -5 # STORE DATA AND REPEAT FLASHING
 CAF V06N84 +1 # FLASH VERB 06 NOUN 33, DISPLAY LAST TIG,
 TC BANKCALL # AND WAIT FOR KEYBOARD ACTION.
 CADR GOFLASH
 TCF ENDP76
 TC +2
 TC -5
 TC INTPRET # RETURN TO INTERPRETIVE CODE
 DLOAD TIG # SET D(MPAC)=TIG IN CSEC B28
 STCALL TDEC1 # SET TDEC1=TIG FOR ORBITAL INTEGRATION
 OTHPREC
COMPMAT VLOAD UNIT
 RATT

Page 512

 VCOMP # U(-R)
 STORE 24D # U(-R) TO 24D
 VXV UNIT # U(-R) X V = U(V X R)
 VATT
 STORE 18D
 VXV UNIT # U(V X R) X U(-R) = U((R X V) X R)
 24D
 STOVL 12D

```



July 12, 2016

1385

```

 DELVOV
VXM VSL1 # V(MPAC)=DELTA V IN REFCOSYS
 12D
VAD
 VATT
STORE 6 # V(PD6)=VATT + DELTA V
CALL # PREVENT WOULD-BE USER OF ORBITAL
 INTSTALL # INTEG FROM INTERFERING WITH UPDATING
CALL
 P76SUB1
VLOAD VSR*
 6
 0,2
STOVL VCV
 RATT
VSR*
 0,2
STODL RCV
 TIG
STORE TET
CLEAR DLOAD
 INTYPFLG
 TETTHIS
INTOTHIS STCALL TDEC1
 INTEGRVS
CALL
 INTSTALL
CALL
 P76SUB1 # SET/CLEAR MOONFLAG
VLOAD
 RATT1
STORE RRECT
STODL RCV
 TAT
STOVL TET
 VATT1
CALL
 MINIRECT
EXIT
TC PHASCHNG
OCT 04024
Page 513
TC UPFLAG
ADRES REINTFLG
TC INTPRET
```

```

 CALL ATOPOTH
 SSP EXIT
 QPRET
 OUT
 TC BANKCALL # PERMIT USE OF ORBITAL INTEGRATION
 CADR INTWAKE1
OUT EXIT
ENDP76 CAF ZERO
 TS MARKCTR # CLEAR RR TRACKING MARK COUNTER
 TS VHFCNT

 CAF NEGONE
 TS MRKBUF2 # INVALIDATE MARK BUFFER

 TCF GOTOPOOH

V06N84 NV 0684
 NV 0633
P76SUB1 CLEAR SLOAD
 MOONFLAG
 X2
 BHIZ SET # X2=0...CLEAR MOONFLAG
 +2 # =2.....SET MOONFLAG
 MOONFLAG
 RVQ

```

This code is written to file `src/P76.s`.

## B.79 P-AXIS RCS AUTOPILOT

1387

*<src/P-AXIS-RCS-AUTOPILOT.s 1387>≡*

```
Copyright: Public domain.
Filename: P-AXIS_RCS_AUTOPILOT.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1421-1441
Mod history: 2009-05-27 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
2009-06-05 RSB Corrected a relative jump from
+8 to +8D.
2009-06-07 RSB Corrected a typo.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
```

# Page 1421

```
BANK 16
SETLOC DAPS1
BANK

EBANK= PERROR
COUNT* $$/DAPP
```

```
THE FOLLOWING T5RUPT ENTRY BEGINS THE PROGRAM WHICH CONTROLS THE P-AXIS ACTION OF THE LEM USI
THE NOMINAL TIME BETWEEN THE P-AXIS RUPTS IS 100 MS IN ALL NON-IDLING MODES OF THE DAP.
```

```
PAXIS CA MS100
```

```

ADS TIME5 # *** NECESSARY IN ORDER TO ALLOW
 # SYNCHRONIZATION WITH OTHER INTERRUPTS ***

LXCH BANKRUPT # INTERRUPT LEAD IN (CONTINUED)
EXTEND
QXCH QRUPT

CHECK IF DAP PASS IS PERMISSIBLE

CCS DAPZRUP # IF DAPZRUP POSITIVE, DAP (JASK) IS
TC BAILOUT # STILL IN PROGRESS AND A RESTART IS
OCT 02000 # CALLED FOR. IT IS NEVER ZERO

TC CHEKBITS # RETURN IS TC I+1 IF DAP SHOULD STAY ON.

CA CDUX # READ AND STORE CDU'S
TS DAPTREG4
CA CDUY
TS DAPTREG5
CA CDUZ
TS DAPTREG6

***** KALCMANU-DAP AND "RATE-HOLD"-DAP INTERFACE *****
#
THE FOLLOWING SECTION IS EXECUTED EVERY 100 MS (10 TIMES A SECOND) WITHIN THE P-AX.
AUTOPILOT (WHENEVER THE DAP IS IN OPERATION).

CA CDUXD
EXTEND
MSU DELCDUX
TC 1STOTWOS
TS CDUXD
CA CDUYD
EXTEND
MSU DELCDUY
TC 1STOTWOS
TS CDUYD
CA CDUZD
EXTEND
MSU DELCDUZ

Page 1422

TC 1STOTWOS
TS CDUZD
EXTEND
DIM TCP # DIMINISH MANUAL CONTROL DIRECT RATE
EXTEND # TIME COUNTERS.

```

DIM TCQR

```
RATFLOOP COMPUTES JETRATER, JRATER, AND 1JACC*NO. PJETS IN ITEMP1.
RETURNS TO BACKP.
#
JETRATER = 1JACC*NO.PJETS*TJP (NOTE TJ IS THE TIME FIRED DURING CSP)
JETRATER = 1JACCQ(TJU*NO.UJETS - TJV*NO.VJETS)
JETRATER = 1JACCR(TJU*NO.UJETS + TJV*NO.VJETS)
```

```
1STOTWOS TCF PAXFILT # PROCEEDS TO RATELOOP AFTER SUPERJOB
 CCS A
 AD ONE
 TC Q
 CS A
 TC Q
SUBDIVIDE EXTEND # OVERFLOW PROTECTION ROUTINE TO GIVE
 MP DAPTEMP3 # POSMAX OR NEGMAX IF THE DIVIDE WOULD
 DAS OMEGAU # OVERFLOW
```

```
+3 EXTEND
 DCA OMEGAU
 DXCH DAPTEMP5
 CCS OMEGAU
 TCF +2
 TCF DIVIDER
 AD -OCT630
 EXTEND
 BZMF DIVIDER

 CCS OMEGAU
 CA POSMAX # 45 DEG/SEC
 TC Q
 CS POSMAX
 TC Q
```

```
DIVIDER DXCH OMEGAU
 EXTEND
 DV DAPTREG4
 TC Q
```

```
OVERSUB TS 7 # RETURNS A UNCHANGED OR LIMITED TO
 TC Q # POSMAX OR NEGMAX IF A HAS OVERFLOW
 INDEX A
 CS BIT15 -1
```

# Page 1423

TC Q

```

-OCT630 OCT 77147

BACKP CA DAPTEMP1
 EXTEND
 MP 1JACC
 TS JETRATE

```

```

BEGINNING OF THE RATE DERIVATION
OMEGAP,Q,R BODY RATES SCALED AT PI/4
TRAPEDP,Q,R BODY ANGLE ERRORS FROM PREDICTED ANGLE (PI/40)
NP(QR)TRAPS NUMBER OF TIMES ANGLE ERROR HAS BEEN ACCUMULATED
AOSQ(R)TERM CHANGE IN RATE DUE TO OFFSET ACCELERATION. (PI/4)
JETRATE,Q,R CHANGE IN RATE DUE TO JET ACCELERATION. (PI/4)
TRAPSIZE NEGATIVE LIMIT OF MAGNITUDE OF TRAPEDP, ETC.
OMEGAU DP-TEMPORARY STORAGE
OMEGA = OMEGA + JETRATE + AOSTERM (+TRAPED/NTRAPS IF TRAPED BIG)

```

```

CAE DAPTREG4 # CDUX IS STORED HERE
TS L
EXTEND
MSU OLDXFORP # SCALED AT PI
LXCH OLDXFORP
TS DAPTEMP1
CA 1/40
TS DAPTREG4
CS JETRATE
EXTEND
MP BIT14
ADS TRAPEDP
CA JETRATEQ
AD AOSQTERM
EXTEND
MP -BIT14
ADS TRAPEDQ
CA JETRATER
AD AOSRTERM
EXTEND
MP -BIT14
ADS TRAPEDR

CA DAPTREG5 # CDUY IS STORED HERE
TS L
EXTEND
MSU OLDYFORP # SCALED AT PI
LXCH OLDYFORP

```

July 12, 2016

1391

# Page 1424

```
TS DAPTEMP2
EXTEND
MP M11 # M11 SCALED AT 1

AD DAPTEMP1
DXCH OMEGAU

TC SUBDIVIDE +3 # RETURNS WITH CDU-RATE AT PI/4

EXTEND
SU OMEGAP
ADS TRAPEDP
TC OVERSUB
TS TRAPEDP
EXTEND
DCA DAPTEMP5
DAS DXERROR
CS PLAST
EXTEND
MP 1/40
DAS DXERROR # MANUAL MODE X-ATTITUDE ERROR (DP)
CA DAPTREG6 # CDUZ IS STORED HERE
TS L
EXTEND
MSU OLDZFORQ
TS DAPTEMP3
LXCH OLDZFORQ
CA M21
EXTEND
MP DAPTEMP2
DXCH OMEGAU
CA M22
TC SUBDIVIDE

EXTEND
SU OMEGAQ
ADS TRAPEDQ
TC OVERSUB
TS TRAPEDQ
EXTEND
DCA DAPTEMP5
DAS DYERROR
CS QLAST
EXTEND
MP 1/40
DAS DYERROR # MANUAL MODE Y-ATTITUDE ERROR (DP)
```

```

CA M31
EXTEND
MP DAPTEMP2
DXCH OMEGAU
CA M32

TC SUBDIVDE

Page 1425
EXTEND
SU OMEGAR
ADS TRAPEDR
TC OVERSUB
TS TRAPEDR # TRAPEDS HAVE ALL BEEN COMPUTED

EXTEND
DCA DAPTEMP5
DAS DZERROR
CS RLAST
EXTEND
MP 1/40
DAS DZERROR # MANUAL MODE Z-ATTITUDE ERROR (DP)
CA DAPBOOLS # PICK UP PAD LOADED STATE ESTIMATOR GAINS
MASK CSMDOCKD
EXTEND
BZF LMONLY
EXTEND # DOCKED
DCA DKOMEGAN
DXCH DAPTREG4
CA DKTRAP
TCF +5
EXTEND # UNDOCKED
DCA LMOMEGAN
DXCH DAPTREG4
CA LMTRAP
TS DAPTREG6
CCS TRAPEDP
TCF +2
TCF SMALPDIF
AD DAPTREG6 # TRAPSIZE > ABOUT 77001 %-1.4DEG/SEC"
EXTEND
BZMF SMALPDIF
ZL
LXCH TRAPEDP
CA ZERO
EXTEND
DV NPTRAPS

```



July 12, 2016

1393

|             |        |          |                                        |
|-------------|--------|----------|----------------------------------------|
|             | ADS    | OMEGAP   |                                        |
|             | TC     | OVERSUB  |                                        |
|             | TS     | OMEGAP   |                                        |
|             | CA     | DAPTREG4 | ABOUT 10 OR 0 FOR DOCKED OR UNDOCKED   |
|             | TS     | NPTRAPS  |                                        |
| SMALPDIF    | INCR   | NPTRAPS  |                                        |
| P-RATE      | CA     | JETRATE  |                                        |
|             | ADS    | OMEGAP   |                                        |
|             | TC     | OVERSUB  |                                        |
|             | TS     | OMEGAP   |                                        |
|             | CCS    | TRAPEDQ  |                                        |
| # Page 1426 | TCF    | +2       |                                        |
|             | TCF    | Q-RATE   |                                        |
|             | AD     | DAPTREG6 | # TRAPSIZE > ABOUT 77001 %-1.4DEG/SEC" |
|             | EXTEND |          |                                        |
|             | BZMF   | Q-RATE   |                                        |
|             | ZL     |          |                                        |
|             | LXCH   | TRAPEDQ  |                                        |
|             | CA     | ZERO     |                                        |
|             | EXTEND |          |                                        |
|             | DV     | NQTRAPS  |                                        |
|             | TS     | DAPTEMP1 | # SAVE FOR OFFSET ESTIMATE             |
|             | ADS    | OMEGAQ   |                                        |
|             | TC     | OVERSUB  |                                        |
|             | TS     | OMEGAQ   |                                        |
|             | CA     | DAPTREG4 | # ABOUT 10 OR 0 FOR DOCKED OR UNDOCKED |
|             | XCH    | NQTRAPS  |                                        |
|             | AD     | DAPTREG5 | # KAOS > ABOUT 60D %N/N_60"            |
|             | XCH    | DAPTEMP1 |                                        |
|             | EXTEND |          |                                        |
|             | MP     | FIVE     |                                        |
|             | EXTEND |          |                                        |
|             | DV     | DAPTEMP1 |                                        |
|             | ADS    | AOSQ     |                                        |
| Q-RATE      | INCR   | NQTRAPS  |                                        |
|             | CA     | JETRATEQ |                                        |
|             | AD     | AOSQTERM |                                        |
|             | ADS    | OMEGAQ   |                                        |
|             | TC     | OVERSUB  |                                        |
|             | TS     | OMEGAQ   |                                        |
|             | CCS    | TRAPEDR  |                                        |
|             | TCF    | +2       |                                        |
|             | TCF    | R-RATE   |                                        |

```

AD DAPTRREG6 # TRAPSIZE > ABOUT 77001 % -1.4DEG/SEC"
EXTEND
BZMF R-RATE
ZL
LXCH TRAPEDR
CA ZERO
EXTEND
DV NRTRAPS
TS DAPTEMP2 # SAVE FOR OFFSET ESTIMATE
ADS OMEGAR
TC OVERSUB
TS OMEGAR
CA DAPTRREG4 # ABOUT 10 OR 0 FOR DOCKED OR UNDOCKED
XCH NRTRAPS
AD DAPTRREG5 # KAOS > ABOUT 60D %N/N_60"
XCH DAPTEMP2
EXTEND

Page 1427
MP FIVE
EXTEND
DV DAPTEMP2
ADS AOSR
R-RATE INCR NRTRAPS
CA JETRATER
AD AOSRTERM
ADS OMEGAR
TC OVERSUB
TS OMEGAR

END OF RATE DERIVATION
BEGIN OFFSET ESTIMATER
IN POWERED FLIGHT, AOSTASK WILL BE CALLED EVERY 2 SECONDS.
AOS = AOS + K*SUMPATE

CS DAPBOOLS
MASK DRIFTBIT
CCS A
TCF WORKTIME
TS ALPHAQ # ZERO THE OFFSET ACCELERATION VALUES.
TS ALPHAR
TS AOSQTERM
TS AOSRTERM
TS AOSQ
TS AOSR
TCF PRETIMCK
KAOS DEC 60

```

July 12, 2016

1395

```
WORKTIME CA QACCDOT
EXTEND
MP CALLCODE # OCTAL 00032 IS DECIMAL .1 AT 2(6).
DAS AOSQ
CA AOSQ
TS ALPHAQ
EXTEND
MP 200MS # .2 AT 1
TS AOSQTERM
CA RACCDOT
EXTEND
MP CALLCODE # OCTAL 00032 IS DECIMAL .1 AT 2(6).
DAS AOSR
CA AOSR
TS ALPHAR
EXTEND
MP 200MS # .2 AT 1
TS AOSRTERM
TCF PRETIMCK

Page 1428
PAXFILT CA CALLGMBL # EXECUTE ACDT+C12, IF NEEDED.
MASK RCSFLAGS
CCS A # CALLGMBL IS NOT BIT15, SO THIS TEST IS
TC ACDT+C12 # VALID.

DXCH ARUPT
DXCH DAPARUPT
CA SUPERJOB # SETTING UP THE SUPERJOB
XCH BRUPT
LXCH QRUPT
DXCH DAPBQRPT
CA SUPERADR
DXCH ZRUPT
DXCH DAPZRUPT
TCF NOQBRSM +1 # RELINT (JUST IN CASE) AND RESUME, IN THE
 # FORM OF A JASK, AT SUPERJOB.

SUPERADR GENADR SUPERJOB +1

COUNT DOWN GIMBAL DRIVE TIMERS AND TURN OFF DRIVES IF REQUIRED.

SUPERJOB TCF RATELOOP
PRETIMCK CCS QGIMTIMR
TCF DECQTIMR # POSITIVE -- COUNTING DOWN
TCF TURNOFFQ # NEGATIVE -- DRIVE SHOULD BE ENDED
```

|                   |        |             |                                            |
|-------------------|--------|-------------|--------------------------------------------|
| CHKRTIMR          | CCS    | RGIMTIMR    | # NEGATIVE -- INACTIVE                     |
|                   | TCF    | DECRTIMR    | # (NEG ZERO -- IMPOSSIBLE)                 |
|                   | TCF    | TURNOFFR    | # REPEATED (ABOVE) FOR R AXIS.             |
|                   | EXTEND |             | # DECREMENT DOCKED JET INHIBITION COUNTERS |
|                   | DIM    | PJETCTR     |                                            |
|                   | EXTEND |             |                                            |
|                   | DIM    | UJETCTR     |                                            |
|                   | EXTEND |             |                                            |
|                   | DIM    | VJETCTR     |                                            |
|                   | CA     | BIT12       |                                            |
|                   | MASK   | RCSFLAGS    |                                            |
|                   | EXTEND |             |                                            |
|                   | BZF    | SKIPPAXS    |                                            |
|                   | TC     | CHKVISFZ    |                                            |
| DECQTIMR          | TS     | QGIMTIMR    | # COUNT TIMERS DOWN TO POS ZERO.           |
|                   | TCF    | CHKRTIMR    |                                            |
| DECRTIMR          | TS     | RGIMTIMR    |                                            |
|                   | TCF    | CHKRTIMR +3 |                                            |
| TURNOFFQ          | TS     | NEGUQ       | # HALT DRIVES.                             |
|                   | TS     | QACCDOT     |                                            |
|                   | CS     | QGIMBITS    |                                            |
|                   | EXTEND |             |                                            |
| # Page 1429       |        |             |                                            |
|                   | WAND   | CHAN12      |                                            |
|                   | CAF    | NEGMAX      |                                            |
|                   | TS     | QGIMTIMR    |                                            |
|                   | TCF    | CHKRTIMR    |                                            |
| TURNOFFR          | TS     | NEGUR       |                                            |
|                   | TS     | RACCDOT     |                                            |
|                   | CS     | RGIMBITS    |                                            |
|                   | EXTEND |             |                                            |
|                   | WAND   | CHAN12      |                                            |
|                   | CAF    | NEGMAX      |                                            |
|                   | TS     | RGIMTIMR    |                                            |
|                   | TCF    | CHKRTIMR +3 |                                            |
| QGIMBITS          | EQUALS | OCT1400     | # BITS 9 AND 10 (OF CHANNEL 12).           |
| RGIMBITS          | EQUALS | PRI06       | # BITS 11 AND 12 (OF CHANNEL 12).          |
| SKIPPAXS          | CS     | RCSFLAGS    |                                            |
|                   | MASK   | BIT12       |                                            |
|                   | ADS    | RCSFLAGS    | # BIT 12 SET TO 1.                         |
|                   | TCF    | QRAXIS      | # GO TO QRAXIS OR TO CTS.                  |
| # Y-X TRANSLATION |        |             |                                            |

July 12, 2016

1397

```
#
INPUT: BITS 9-12 OF CH31 (FROM TRANSLATION CONTROLLER)
#
OUTPUT: NEXTP
#
NEXTP IS THE CHANNEL 6 CODE OF JETS FOR THE DESIRED TRANSLATION.
IF THERE ARE FAILURES IN THE DESIRED POLICY, THEN
(1) FOR DIAGONAL TRANS: UNFAILED PAIR
ALARM (IF NO PAIR)
(2) FOR PRINCIPAL TRANS: TRY TO TACK WITH DIAGONAL PAIRS
ALARM (IF DIAGONAL PAIRS ARE FAILED)
#
CHKVISFZ EXTEND
 READ CHAN31
 CS A
 MASK 074000CT
 EXTEND
 BZF TSNEXTP
 EXTEND
 MP BIT7
 INDEX A
 CA INDXYZ
 TS ROTINDEX
TRYUORV CA SIX
 TC SELECTYZ
 CS SIX
 AD NUMBERT
 EXTEND
Page 1430
 BZF TSNEXTP -1
 CS FIVE
 AD ROTINDEX
 EXTEND
 BZMF ALTERYZ
 CS NUMBERT
 AD FOUR
 EXTEND
 BZMF TSNEXTP -1
ABORTYZ TC ALARM
 OCT 02001
 CA BIT1 # INVERT BIT 1 OF RCSFLAGS.
 LXCH RCSFLAGS
 EXTEND
 RXOR 1
 TS RCSFLAGS
 CA ZERO
```

```

 ALTERYZ TCF TSNEXTP
 CA BIT1 # INVERT BIT 1 OF RCSFLAGS.
 LXCH RCSFLAGS
 EXTEND
 RXOR 1
 TS RCSFLAGS
 MASK BIT1
 AD FOUR
 ADS ROTINDEX
 TCF TRYUORV
 CA POLYTEMP
 TSNEXTP TS NEXTP

STATE LOGIC
CHECK IN ORDER: IF ON
LPDPHASE GO TO PURGENCY
PULSES MINIMUM PULSE LOTIC
DETENT(BIT15 CH31) RATE COMMAND
GOTO TO PURGENCY

 CA BIT13 # CHECK STICK IF IN ATT. HOLD.
 EXTEND
 RAND CHAN31
 EXTEND
 BZF MANMODE

 CA DAPBOOLS
 MASK XOVINHIB
 CCS A
 TCF PURGENCY # ATTITUDE STEER DURING VISIBILITY PHASE

 MANMODE TCF DETENTCK
 CA PULSES # PULSES IS ONE FOR PULSE MODE
 MASK DAPBOOLS

Page 1431

 EXTEND
 BZF DETENTCK # BRANCH FOR RATE COMMAND

 CA ZERO
 TS PERROR

MINIMUM IMPULSE MODE

 CA CDUX
 TS CDUXD

```

July 12, 2016

1399

```

 CCS OLDPMIN
 TCF CHECKP

FIREP CA BIT3
 EXTEND
 RAND CHAN31
 EXTEND
 BZF +XMIN

 CA BIT4
 EXTEND
 RAND CHAN31
 EXTEND
 BZF -XMIN

 TCF JETSOFF

CHECKP EXTEND
 READ CHAN31
 CS A
 MASK OCT14
 TS OLDPMIN
 TCF JETSOFF

-XMIN CS TEN # ANYTHING LESS THAN 14MS. CORRECTED
 TCF +2 # IN JET SELECTION ROUTINE
+XMIN CA TEN
 TS TJP
 CA ONE
 TS OLDPMIN
 TCF PJETSLEC -6
```

```

MANUAL RATE COMMAND MODE
=====
BY ROBERT F. STENGEL
```

```

THIS MODE PROVIDES RCAH MANUAL CONTROL THRU 2 CONTROL LAWS: 1) DIRECT RATE AND 2) PSEUDO-AUTO
THE DIRECT RATE MODE AFFORDS IMMEDIATE CONTROL WITHOUT OVERSHOOT. THE PSEUDO-AUTO MODE PROVIDES
RATE CONTROL AND ATTITUDE HOLD.
```

```
#
```

```
Page 1432
```

```

IN DIRECT RATE, JETS ARE FIRED WHEN STICK POSITION CHANGES BY A FIXED NUMBER OF INCREMENTS IN
THE 'BREAKOUT LEVEL' IS .6 D/S FOR LM-ONLY AND .3 D/S FOR CSM-DOCKED. THIS LAW NULLS THE RATE
THE 'TARGET DEADBAND', WHICH EQUALS THE BREAKOUT LEVEL.
```

```
#
```

```

IN PSEUDO-AUTO, BODY-FIXED RATE AND ATTITUDE ERRORS ARE SUPPLIED TO TJETLAW, WHICH EXERCISES
```

```

CONTROL SWITCHES FROM DIRECT RATE TO PSEUDO-AUTO IF THE TARGET DB IS ACHIEVED OR IF
IF THE INITIAL COMMAND DOES NOT EXCEED THE BREAKOUT LEVEL, CONTROL GOES TO PSEUDO-AUTO
#
SINCE P-AXIS CONTROL IS SEPARATE FROM Q,R AXES CONTROL, IT IS POSSIBLE TO USE (1) D/S
OR VICE VERSA. THIS ALLOWS A DEGREE OF ATTITUDE HOLD IN UNCONTROLLED AXES. DUE TO
R AXES ARE COUPLED AND MUST USE THE SAME CONTROL LAW.
#
HAND CONTROLLER COMMANDS ARE SCALED BY A LINEAR/QUADRATIC LAW. FOR THE LM-ALONE, M
AND 4 D/S IN NORMAL AND FINE SCALING; HOWEVER, STICK SENSITIVITY AT ZERO COUNTS (OR
OF 2 DEGREES FROM THE CENTERED POSITION) IS .5 OR .1 D/S PER DEGREE. NORMAL AND FINE
CASE IS AUTOMATICALLY SET TO 1/10 THE ABOVE VALUES. SCALING IS DETERMINED IN ROUTINE
#
ZEROENBL ENABLES COUNTERS SO THEY CAN BE READ NEXT TIME
JUSTOUT FIRST DETECTION OF OUT OF DETENT (BY OURRCBIT)

DETENTCK EXTEND
 READ CHAN31
 TS CH31TEMP
 MASK BIT15 # CHECK OUT-OF-DETENT BIT.
 EXTEND
 BZF RHCMOVED # BRANCH IF OUT OF DETENT.
 CAF OURRCBIT # IN DETENT. CHECK THE RATE COMMAND BIT.
 MASK DAPBOOLS
 EXTEND
 BZF PURGENCY # BRANCH IF NOT IN RATE COMMAND LAST PASS.

.....

 CA BIT9 # JUST IN DETENT??
 MASK RCSFLAGS
 EXTEND
 BZF RUTH
 CAF BIT13 # CHECK FOR ATTITUDE HOLD.
 EXTEND
 RAND CHAN31
 EXTEND
 BZF RATEDAMP # BRANCH IF IN ATTITUDE HOLD.

 CS BITS9,11 # IN AUTO.
 MASK RCSFLAGS # (X-AXIS OVERRIDE)
 TS RCSFLAGS # ZERO ORBIT (BIT 11) AND JUST-IN BIT (9).
 TCF RATEDAMP

RUTH CA RCSFLAGS
 MASK PBIT # IN ATTITUDE HOLD.
 EXTEND

```



July 12, 2016

1401

```
BZF +2 # BRANCH IF P-RATE DAMPING IS FINISHED.
TCF RATEDAMP
```

# Page 1433

```
CA RCSFLAGS
MASK QRBIT
EXTEND
BZF RATEDONE # BRANCH IF Q,R RATE DAMPING IS FINISHED.
TCF RATEDAMP
```

# =====

```
1/10SEC OCT 1
40CYC OCT 50
PQRBIT OCT 74777
BITS9,11 EQUALS EBANK5
LINRATP DEC 46
```

# =====

```
RATEDONE CS OURRCBIT # MANUAL COMMAND AND DAMPING COMPLETED IN
 INHINT # ALL AXES.
 MASK DAPBOOLS
 TS DAPBOOLS
```

# READ CDUS INTO CDU DESIRED REGISTERS

```
CAF BIT13
EXTEND
RAND CHAN31
EXTEND
BZF +4
CA CDUX # (X-AXIS OVERRIDE)
TS CDUXD
TC +3
TC IBNKCALL
FCADR ZATTEROR
RELINT
TCF PURGENCY
```

```
JUSTOUT TS PERROR
 CA OURRCBIT # INITIALIZATION -- FIRST MANUAL PASS.
 ADS DAPBOOLS
 CA ZERO
 TS DXERROR
 TS DXERROR +1
```

|             |        |            |                                            |
|-------------|--------|------------|--------------------------------------------|
|             | TS     | DYERROR    |                                            |
|             | TS     | DYERROR +1 |                                            |
|             | TS     | DZERROR    |                                            |
|             | TS     | DZERROR +1 |                                            |
|             | TS     | PLAST      |                                            |
|             | TS     | QLAST      |                                            |
|             | TS     | RLAST      |                                            |
|             | TS     | Q-RHCCTR   |                                            |
|             | TS     | R-RHCCTR   |                                            |
|             | CA     | PQRBIT     |                                            |
|             | MASK   | RCSFLAGS   |                                            |
|             | TS     | RCSFLAGS   | # BITS 10 AND 11 OF RCSFLAGS ARE 0.        |
| # Page 1434 |        |            |                                            |
|             | CS     | RCSFLAGS   | # SET 'JUST-IN' BIT TO 1.                  |
|             | MASK   | BIT9       |                                            |
|             | ADS    | RCSFLAGS   |                                            |
|             | TC     | ZEROENBL   |                                            |
| ZEROENBL    | TCF    | JETSOFF    |                                            |
|             | LXCH   | R-RHCCTR   |                                            |
|             | CA     | Q-RHCCTR   |                                            |
|             | DXCH   | SAVEHAND   |                                            |
|             | CA     | ZERO       |                                            |
|             | TS     | P-RHCCTR   |                                            |
|             | TS     | Q-RHCCTR   |                                            |
|             | TS     | R-RHCCTR   |                                            |
|             | CA     | BITS8,9    |                                            |
|             | EXTEND |            |                                            |
|             | WOR    | CHAN13     | # COUNTERS ZEROED AND ENABLED              |
|             | TC     | Q          |                                            |
| RATEDAMP    | CA     | ZERO       |                                            |
|             | TS     | P-RHCCTR   |                                            |
|             | TCF    | RATERERROR |                                            |
|             |        |            |                                            |
| RHCMOVED    | CA     | OURRCBIT   | # P CONTROL                                |
|             | MASK   | DAPBOOLS   |                                            |
|             | EXTEND |            |                                            |
|             | BZF    | JUSTOUT -1 |                                            |
| RATERERROR  | CA     | CDUX       | # FINDCDUW REQUIRES THAT CDUXD=CDUX DURING |
|             | TS     | CDUXD      | # X-AXIS OVERRIDE                          |
|             | CCS    | P-RHCCTR   |                                            |
|             | TCF    | +3         |                                            |
|             | TCF    | +2         |                                            |
|             | TCF    | +1         |                                            |
|             | DOUBLE |            | # LINEAR/QUADRATIC CONTROLLER SCALING      |
|             | DOUBLE |            | # (SEE EXPLANATION OF Q,R-AXES RCS         |
|             | AD     | LINRATP    | # AUTOPILOT)                               |

```

EXTEND
MP P-RHCCTR
CA L
EXTEND
MP STIKSENS
XCH PLAST
COM
AD PLAST
TS DAPTEMP1
TC ZEROENBL # INTERVAL. ZERO AND ENABLE ACA COUNTERS.
CS PLAST
AD OMEGAP
TS EDOTP
CCS DAPTEMP1 # IF P COMMAND CHANGE EXCEEDS BREAKOUT
TCF +3 # LEVEL, GO TO DIRECT RATE CONTROL. IF NOT
TCF +8D # CHECK FOR DIRECT RATE CONTROL LAST TIME.
TCF +1

Page 1435
AD -RATEDB
EXTEND
BZMF +4
CA 40CYC
TS TCP
TC PEGI
CA RCSFLAGS # CHECK FOR DIRECT RATE COMMAND LAST TIME.
MASK PBIT
EXTEND
BZF +2
TC PEGI # TO PURE RATE COMMAND
CA DXERROR # PSEUDO-AUTO CONTROL.
TS E # X-ATTITUDE ERROR (SP)
TS PERROR # LOAD P-AXIS ERROR FOR MODE1 FDAI DISPLAY
TC PURGENCY +4
CA CDUX # DIRECT RATE CONTROL.
TS CDUXD
CA ZERO
TS DXERROR
TS DXERROR +1
TS PERROR # ZERO P-AXIS ERROR FOR MODE1 FDAI DISPLAY
CCS EDOTP
TC +3
TC +2
TC +1
TS ABSEDOTP
AD TARGETDB
EXTEND # IF RATE ERROR IS LESS THAN DEADBANK,

```

```

 BZMF LAST # FIRE, AN SWITCH TO PSEUDO-AUTO.
 CA TCP
 EXTEND
 BZMF LAST # IF TIME IN RATE COMMAND EXCEEDS 4 SEC.
 CS RCSFLAGS
 MASK PBIT
 ADS RCSFLAGS # BIT 10 IS 1.
 TCF +4
 CS PBIT
 MASK RCSFLAGS
 TS RCSFLAGS # BIT 10 IS 0.
 CS EDOTP
 EXTEND
 MP 1/ANETP # 1/2JTACC SCALED AT 2EXP(7)/PI
 DAS A
 TC OVERSUB
 EXTEND
 MP 25/32 # A CONTAINS TJET SCALED AT 2EXP(4)(16/25)
 TS TJP # 4.JET TIME
 CA ABSEDOTP
 AD -2JETLIM # COMPARING DELTA RATE WITH 2 JET LIMIT
 EXTEND
Page 1436
 BZMF +3

 CA SIX
 TCF +8D
 CA TJP
 ADS TJP

GOES TO PJETSLEC FOR TWO JETS
P-JET-SELECTION-ROUTINE (ROTATION)
#
INPUT: NUMBERT 4,5,6 FOR WHICH PAIR OR 4 JETS
TJP + FOR +P ROTATION
#
OUTPUT: CHANNEL 6
PJUMPADR FOR P-AXIS SKIP
(JTLST CALL) (SMALL TJP)
#
ORDER OF POLICIES TRIED IN CASE OF FAILURE.
+P -P
7,15 8,16
4,12 3,11
4,7 8,11
7,12 11,16

```

July 12, 2016

1405

# 12,15 3,16  
# 4,15 3,8  
# ALARM ALARM

PJETSLEC  
CA AORBSYST  
MASK DAPBOOLS  
CCS A  
CA ONE  
AD FOUR  
TS NUMBERT  
CA ONE  
TS L  
CCS TJP  
TCF +5  
TCF JETSOFF  
TCF +2  
TCF JETSOFF  
ZL  
AD ONE  
TS ABSTJ  
LXCH ROTINDEX  
TC SELECTP  
CS SIX  
AD NUMBERT  
EXTEND  
BZF +2  
  
CS TWO

# Page 1437

AD FOUR  
TS NO.PJETS  
CA POLYTEMP  
TC WRITEP  
CS ABSTJ  
AD +150MST6  
EXTEND  
BZMF QRAXIS # GO TO QRAXIS OR TO GTS.  
  
AD -136MST6  
EXTEND  
BZMF +5  
  
ADS ABSTJ  
INDEX ROTINDEX  
CA MINTIMES

|                                                              |        |           |                                         |
|--------------------------------------------------------------|--------|-----------|-----------------------------------------|
|                                                              | TS     | TJP       |                                         |
|                                                              | CA     | ABSTJ     |                                         |
|                                                              | ZL     |           |                                         |
|                                                              | INHINT |           |                                         |
|                                                              | DXCH   | T6FURTHA  |                                         |
|                                                              | TC     | IBNKCALL  |                                         |
|                                                              | CADR   | JTLST     |                                         |
|                                                              | CS     | BIT12     |                                         |
|                                                              | MASK   | RCSFLAGS  |                                         |
|                                                              | TS     | RCSFLAGS  | # BIT 12 SET TO 0.                      |
|                                                              | TC     | ALTSYST   |                                         |
|                                                              | TCF    | QRAXIS    |                                         |
| ALTSYST                                                      | CA     | DAPBOOLS  | # ALTERNATE P-AXIS JETS                 |
|                                                              | TS     | L         |                                         |
|                                                              | CA     | AORBSYST  |                                         |
|                                                              | EXTEND |           |                                         |
|                                                              | RXOR   | LCHAN     |                                         |
|                                                              | TS     | DAPBOOLS  |                                         |
|                                                              | RELINT |           |                                         |
|                                                              | TC     | Q         |                                         |
| DKALT                                                        | TC     | ALTSYST   |                                         |
| JETSOFF                                                      | TC     | WRITEP -1 |                                         |
|                                                              | CA     | ZERO      |                                         |
|                                                              | TS     | TJP       |                                         |
|                                                              | TCF    | QRAXIS    |                                         |
| # (NOTE -- M13 = 1 IDENTICALLY IMPLIES NULL MULTIPLICATION.) |        |           |                                         |
| CALCPERR                                                     | CA     | CDUY      | # P-ERROR CALCULATION.                  |
|                                                              | EXTEND |           |                                         |
|                                                              | MSU    | CDUYD     | # CDU VALUE -- ANGLE DESIRED (Y-AXIS)   |
| # Page 1438                                                  | EXTEND |           |                                         |
|                                                              | MP     | M11       | # (CDUY-CDUYD)M11 SCALED AT PI RADIANS  |
|                                                              | XCH    | E         | # SAVE FIRST TERM (OF TWO)              |
|                                                              | CA     | CDUX      | # THIRD COMPONENT                       |
|                                                              | EXTEND |           |                                         |
|                                                              | MSU    | CDUXD     | # CDU VALUE -- ANGLE DESIRED (X-AXIS)   |
| #                                                            | EXTEND |           |                                         |
| #                                                            | MP     | M13       |                                         |
|                                                              | AD     | DELPERR   | # KALCMANU INTERFACE ERROR.             |
|                                                              | ADS    | E         | # SAVE SUM OF TERMS. COULD BE OVERFLOW. |

July 12, 2016

1407

```

 XCH PERROR # SAVE P-ERROR FOR EIGHT-BALL DISPLAY.
 TC Q # RETURN TO CALLER

P-AXIS URGENCY FUNCTION CALCULATION.

PURGENCY TC CALCPERR # CALCULATE P-AXIS ERRORS.
 CS OMEGAPD # THIS CODING IS COMMON TO BOTH LM DAP AND
 AD OMEGAP # SPS-BACKUP MODE.
 TS EDOTP # EDOTP = OMEGAP - OMEGAPD AT PI/4 RAD/SEC

 CS ONE
 TS AXISCTR
 CA DAPBOOLS
 MASK CSMDOCKD
 EXTEND
 BZF HEADTJET
 INHINT
 TC IBNKCALL # IF CSMDOCKD = 1, GOT TO DOCKED RCS LOGIC
 CADR SPSRCS

 CA TJP
 EXTEND
 BZF DKALT # IF TJP = ZERO, CHANGE AORBSYST.
 RELINT
 TCF PJETSLEC -6 # SELECT AORBSYST AND USE TWO JETS.
HEADTJET CA ZERO
 TS SENSETYP
 INHINT
 TC IBNKCALL
 CADR TJETLAW
 RELINT

 CS FIREFCT
 AD -FOURDEG
 EXTEND
 BZMF PJETSLEC -6
 CCS TJP
 TCF +2
 TCF JETSOFF

Page 1439
 AD -160MST6
 EXTEND
 BZMF PJETSLEC -6
 CA SIX
 TCF PJETSLEC -1
-160MST6 DEC -256
```

-FOURDEG            DEC        -.08888

# Page 1440

# JET POLICY CONTSTRUCTION SUBROUTINE

#

# INPUT:            ROTINDEX, NUMBERT

#

# OUTPUT:           POLYTEMP (JET POLICY)

#

# THIS SUBROUTINE SELECT A SUBSET OF THE DESIRED JETS WHICH HAS NO FAILURE

|          |       |            |                            |
|----------|-------|------------|----------------------------|
| SELECTP  | CA    | SIX        |                            |
|          | TS    | TEMPNUM    |                            |
|          | INDEX | NUMBERT    |                            |
|          | CA    | TYPEP      |                            |
|          | INDEX | ROTEXINDEX |                            |
|          | MASK  | JETSALL    |                            |
|          | TS    | POLYTEMP   |                            |
|          | MASK  | CH6MASK    |                            |
|          | CCS   | A          |                            |
|          | TCF   | +2         |                            |
|          | TC    | Q          |                            |
|          | CCS   | TEMPNUM    |                            |
|          | TCF   | +4         |                            |
|          | TC    | ALARM      |                            |
|          | OCT   | 02003      |                            |
|          | TCF   | JETSOFF    | # ***** TCF ALARMJET ***** |
| SELECTYZ | TS    | NUMBERT    |                            |
|          | TCF   | SELECTP +1 |                            |
| -1       | TCF   | ABORTYZ +2 |                            |
| JETSALL  | OCT   | 00252      |                            |
|          | OCT   | 00125      | # +P                       |
|          | OCT   | 00140      | # -Y                       |
|          | OCT   | 00006      | # -Z                       |
|          | OCT   | 00220      | # +Y                       |
|          | OCT   | 00011      | # +Z                       |
|          | OCT   | 00151      | # +V                       |
| TYPEP    | OCT   | 00146      | # -U                       |
|          | OCT   | 00226      | # -V                       |
|          | OCT   | 00231      | # +U                       |
|          | OCT   | 00151      | # +V                       |
|          | OCT   | 00132      | # 1-3                      |
|          | OCT   | 00245      | # 2-4                      |
|          | OCT   | 00377      | # ALL                      |
| INDXYZ   | =     | -136MST6   |                            |
| -136MST6 | DEC   | -218       |                            |



July 12, 2016

1409

```

DEC 4
DEC 2
OCT 07776
DEC 5
DEC 9
DEC 10
OCT 07776
DEC 3

Page 1441
DEC 8
DEC 7
OCT 07776
OCT 07776
OCT 07776
OCT 07776
OCT 07776
+150MST6 DEC 240
07400OCT OCT 07400

THESE INDEXES OF MASK JETSALL WILL
CHANGE THE INSTRUCTION AT SELECTP +4
TO BE TC JETSALL -1
ONLY USED FOR TRANSLATION FAILURE

T-JET LAW FIXED CONSTANTS

NORMSCL OCT 266
-100MS DEC -.1
200MS DEC .2
25/32 = PRI031
BITS8,9 OCTAL 00600
1/40 DEC .02500
MINTIMES DEC -22
 DEC 22
PSKIPADR GENADR SKIPPAXS

GOES TO Q,R-AXES RCS AUTOPILOT

QRAXIS CS OMEGARD
 AD OMEGAR
 TC OVERSUB
 TS EDOTR
 CS OMEGAQD
 AD OMEGAQ
 TC OVERSUB
 TS EDOTQ
 EXTEND
 DCA QERRCALL
 DTCB

 EBANK= AOSQ
```

1410

July 12, 2016

QERRCALL            2CADR    CALLQERR

This code is written to file `src/P-AXIS-RCS-AUTOPILOT.s`.

## B.80 PHASE TABLE MAINTENANCE

```

1411 <src/PHASE-TABLE-MAINTENANCE.s 1411>≡
Copyright: Public domain.
Filename: PHASE_TABLE_MAINTENANCE.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1404-1413
Mod history: 2009-05-10 SN (Sergio Navarro). Started adapting
from the Colossus249/ file of the same
name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 1404
SUBROUTINE TO UPDATE THE PROGRAM NUMBER DISPLAY ON THE DSKY.

COUNT 02/PHASE
BLOCK 02
SETLOC FFTAG1
BANK

NEWMODEX INDEX Q # UPDATE MODREG. ENTRY FOR MODE IN FIXED.
CAF 0
INCR Q

NEWMODEA TS MODREG # ENTRY FOR MODE IN A.

```

```

MMDSPY CAF +3 # DISPLAY MAJOR MODE.
PREBJUMP LXCH BBANK # PUTS BBANK IN L
 TCF BANKJUMP # PUTS Q INTO A
 CADR SETUPDSP

RETURN TO CALLER +3 IF MODE = THAT AT CALLER +1. OTHERWISE RETURN TO CALLER +2.

CHECKMM INDEX Q
 CS 0
 AD MODREG
 EXTEND
 BZF Q+2
 TCF Q+1 # NO MATCH

TCQ = Q+2 +1

 BANK 14
 SETLOC PHASETAB
 BANK

 COUNT 10/PHASE

SETUPDSP INHINT
 DXCH RUPTREG1 # SAVE CALLER'S RETURN 2CADR
 CAF PRI030 # EITHER A TASK OR JOB CAN COME TO
 TC NOVAC # NEWMODEX
 EBANK= MODREG
 2CADR DSPMMJOB

 DXCH RUPTREG1
 RELINT
 DXCH Z # RETURN

DSPMMJOB EQUALS DSPMMJB

 BLOCK 02

Page 1405
 SETLOC FFTAG1
 BANK

Page 1406
PHASCHNG IS THE MAIN WAY OF MAKING PHASE CHANGES FOR RESTARTS. THERE ARE THREE FOR
A, TYPE B, AND TYPE C. THEY ARE ALL CALLED AS FOLLOWS, WHERE OCT XXXXX CONTAINS TH
TC PHASCHNG
OCT XXXXX
TYPE A IS CONCERNED WITH FIXED PHASE CHANGES, THAT IS, PHASE INFORMATION THAT IS ST

```

```

OPTIONS ARE, WHERE G STANDS FOR A GROUP AND .X FOR THE PHASE,
G.0 INACTIVE, WILL NOT PERMIT A GROUP G RESTART
G.1 WILL CAUSE THE LAST DISPLAY TO BE REACTIVATED, USED MAINLY IN MANNED FL
G.EVEN A DOUBLE TABLE RESTART, CAN CAUSE ANY COMBINATION OF TWO JOBS, TASKS, A
LONGCALL TO BE RESTARTED.
G.ODD NOT .1 A SINGLE TABLE RESTART, CAN CAUSE EITHER A JOB, TASK, OR LONGCALL RESTA
#
THIS INFORMATION IS PUT INTO THE OCTAL WORD AFTER TC PHASCHNG AS FOLLOWS
TLO OOP PPP PPP GGG
WHERE EACH LETTER OR NUMBER STANDS FOR A BIT. THE G'S STAND FOR THE GROUP, OCTAL 1-7, THE P'
OCTAL 0 - 127. O'S MUST BE 0. IF ONE WISHES TO HAVE THE TBASE OF GROUP G TO BE SET AT THIS
T IS SET TO 1, OTHERWISE IT IS SET TO 0. SIMILARLY IF ONE WISHES TO SET LONGBASE, THEN L IS
IT IS SET TO 0. SOME EXAMLES,
#
TC PHASCHNG # THIS WILL CAUSE GROUP 3 TO BE SET TO 0,
OCT 00003 # MAKING GROUP 3 INACTIVE
#
TC PHASCHNG # IF A RESTART OCCURS THIS WOULD CAUSE
OCT 00012 # GROUP 2 TO RESTART THE LAST DISPLAY
#
TC PHASCHNG # THIS SETS THE TBASE OF GROUP 4 AND IN
OCT 40064 # CASE OF A RESTART WOULD START UP THE TWO
THINGS LOCATED IN THE DOUBLE 4.6 RESTART
LOCATION.
#
TC PHASCHNG # THIS SETS LONGBASE AND UPON A RESTART
OCT 20135 # CAUSES 5.13 TO BE RESTARTED (SINCE
LONGBASE WAS SET THIS SINGLE ENTRY
SHOULD BE A LONGCALL)
#
TC PHASCHNG # SINCE BOTH TBASE4 AND LONGBASE ARE SET,
OCT 60124 # 4.12 SHOULD CONTAIN BOTH A TASK AND A
LONGCALL TO BE RESTARTED
#
TYPE C PHASCHNG CONTAINS THE VARIABLE TYPE OF PHASCHNG INFORMATION. INSTEAD OF THE INFORMATI
PERMANENT FORM, ONE STORES THE DESIRED RESTART INFORMATION IN A VARIABLE LOCATION. THE BITS
TLO 1AD XXX CJW GGG
WHERE EACH LETTER OR NUMBER STANDS FOR A BIT. THE G'S STAND FOR THE GROUP, OCTAL 1 - 7. IF
BE BY WAITLIST, W IS SET TO 1, IF IT IS A JOB, J IS SET TO 1, IF IT IS A LONGCALL, C IS SET T
THESE THREE BITS MAY BE SET. X'S ARE IGNORED, 1 MUST BE 1, AND 0 MUST BE 0. AGAIN T STANDS
Page 1407
AND L FOR LONGBASE. THE BITS A AND D ARE CONCERNED WITH THE VARIABLE INFORMATION. IF D IS S
OR DELTA TIME WILL BE READ FROM THE NEXT LOCATION AFTER THE OCTAL INFORMATION., IF THIS IS TO
IS, THE NAME OF A LOCATION CONTAINING THE INFORMATION (DELTA TIME ONLY), THEN THIS IS GIVEN A
THAT LOCATION WHICH CONTAINS THE DELTA TIME. IF THE OLD PRIORITY OR DELTA TIME IS TO BE USED
ALREADY IN THE VARIABLE STORAGE, THEN D IS SET TO 0. NEXT THE A BIT IS USED. IF IT IS SET T
THAT WOULD BE RESTARTED DURING A RESTART IS THE NEXT LOCATION AFTER THE PHASE INFORMATION, TH

```

```

(TC PHASCHNG) +2 OR +3, DEPENDING ON WHETHER D HAD BEEN SET OR NOT. IF A IS SET T
WOULD BE RESTARTED IS THE 2CADR THAT IS READ FROM THE NEXT TWO LOCATION. EXAMPLES,
AD TC PHASCHNG # THIS WOULD CAUSE LOCATION AD +3 TO BE
AD+1 OCT 05023 # RESTARTED BY GROUP THREE WITH A PRIORITY
AD+2 OCT 23000 # OF 23. NOTE UPON RETURNING IT WOULD
AD+3 # ALSO GO TO AD+3
#
AD TC PHASCHNG # GROUP 1 WOULD CAUSE CALLCALL TO BE
AD+1 OCT 27441 # BE STARTED AS A LONGCALL FROM THE TIME
AD+2 -GENADR DELTIME # STORED IN LONGBASE (LONGBASE WAS SET) BY
AD+3 2CADR CALLCALL # A DELTATIME STORED IN DELTIME. THE
AD+4 # BBCON OF THE 2CADR SHOULD CONTAIN THE E
AD+5 # BANK OF DELTIME. PHASCHNG RETURNS TO
LOCATION AD+5
#
NOTE THAT IF A VARIABLE PRIORITY IS GIVEN FOR A JOB, THE JOB WILL BE RESTARTED AS A
NEGATIVE, AS A FINDVAC IF THE PRIORITY IS POSITIVE.
#
TYPE B PHASCHNG IS A COMBINATION OF VARIABLE AND FIXED PHASE CHANGES. IT WILL START
BELOW AND ALSO START UP ONE FIXED RESTART, THAT IS EITHER AN G.1 OR A G.ODD OR THE
DOUBLE ENTRY. THE BIT INFORMATION IS AS FOLLOW,
TL1 DAP PPP PPP GGG
WHERE EACH LETTER OR NUMBER STANDS FOR A BIT. THE G'S STAND FOR THE GROUP, OCTAL 1
PHASE INFORMATION, OCTAL 0 - 127. 1 MUST BE 1. AND AGAIN T STANDS FOR THE TBASE A
TIME STANDS ONLY FOR PRIORITY SINCE THIS WILL BE CONSIDERED A JOB, AND IT MUST BE C
AGAIN A STANDS FOR THE ADDRESS OF THE LOCATION TO BE RESTARTED, 1 IF THE 2CADR IS C
THE NEXT LOCATION. (THE RETURN LOCATION OF PHASCHNG) EXAMPLES,
AD TC PHASCHNG # TBASE IS SET AND A RESTART CAUSE GROUP 3
AD+1 OCT 56043 # TO START THE JOB AJOBAJOB WITH PRIORITY
AD+2 OCT 31000 # 31 AND THE FIRST ENTRY OF 3.4SPOT (WE CAN
AD+3 2CADR AJOBAJOB # ASSUME IT IS A TASK SINCE WE SET TBASE3)
AD+4 # UPON RETURN FROM PHASCHNG CONTROL WOULD
AD+5 # GO TO AD+5
#
AD TC PHASCHNG # UPON A RESTART THE LAST DISPLAY WOULD BE
AD+1 OCT 10015 # RESTARTED AND A JOB WITH THE PREVIOUSLY
AD+2 # STORED PRIORITY WOULD BE BEGUN AT AD+2
BY MEANS OF GROUP 5
#
Page 1408
THE NOVAC-FINDVAC CHOICE FOR JOBS HOLDS HERE ALSO -- NEGATIVE PRIORITY CAUSES A NOV
#
SUMMARY OF BITS:
TYPE A TL0 OOP PPP PPP GGG
TYPE B TL1 DAP PPP PPP GGG
TYPE C TL0 1AD XXX CJW GGG

```

July 12, 2016

1415

# Page 1409

# 2PHSCHNG IS USED WHEN ONE WISHES TO START UP A GROUP OR CHANGE A GROUP WHILE UNDER THE CONTROL OF A GROUP. FOR EXAMPLE, CHANGE THE PHASE OF GROUP 3 WHILE THE PORTION OF THE PROGRAM IS UNDER GROUP 3. CALLS ARE MADE IN THE FOLLOWING MANNER,

# TC 2PHSCHNG  
# OCT XXXXX  
# OCT YYYYY

# WHERE OCT XXXXX MUST BE OF TYPE A AND OCT YYYYY MAY BE OF EITHER TYPE A OR TYPE B OR TYPE C. # DIFFERENCE --- NOTE: IF LONGBASE IS TO BE SET THIS INFORMATION IS GIVEN IN THE OCT YYYYY INFORMATION. # BE DISREGARDED IF GIVEN WITH THE OCT XXXXX INFORMATION. A COUPLE OF EXAMPLES MAY HELP,

# AD TC 2PHACHNG # SET TBASE3 AND IF A RESTART OCCURS START  
# AD+1 OCT 40083 # THE TWO ENTRIES IN 3.8 TABLE LOCATION  
# AD+2 OCT 05025 # THIS IS OF TYPE C, SET THE JOB TO BE  
# AD+3 OCT 18000 # TO BE LOCATION AD+4, WITH A PRIORITY 18,  
# AD+4 # FOR GROUP 5 PHASE INFORMATION.

COUNT 02/PHASE

2PHSCHNG INHINT # THE ENTRY FOR A DOUBLE PHASE CHANGE

NDX Q  
CA 0  
INCR Q  
TS TEMPP2

MASK OCT7  
DOUBLE  
TS TEMPG2

CA TEMPP2  
MASK OCT17770  
EXTEND  
MP BIT12  
XCH TEMPP2

# NEED ONLY 1770, BUT WHY GET A NEW CONST.

MASK BIT15  
TS TEMPSW2

# INDICATES WHETHER TO SET TBASE OR NOT

TCF PHASCHNG +3

PHASCHNG

INHINT

CA ONE  
TS TEMPSW2

# INDICATES CAME FROM A PHASCHNG ENTRY

NDX Q  
CA 0  
INCR Q

```

Page 1410
TS TEMPSW
EXTEND
DCA ADRPCHN2 # OFF TO SWITCHED BANK
DTCB

EBANK= LST1
ADRPCHN2 2CADR PHSCHNG2

ONEORTWO LXCH TEMPBBCN
 LXCH BBANK
 LXCH TEMPBBCN

MASK OCT14000 # SEE WHAT KIND OF PHASE CHANGE IT IS
CCS A
TCF CHECKB # IT IS OF TYPE 'B'.

CA TEMPP
MASK BIT7
CCS A # SHALL WE USE THE OLD PRIORITY
TCF GETPRIO # NO GET A NEW PRIORITY (OR DELTA T)

OLDPRIO NDX TEMPG # USE THE OLD PRIORITY (OR DELTA T)
 CA PHSPRDT1 -2
 TS TEMPPR

CON1 CA TEMPP # SEE IF A 2CADR IS GIVEN
 MASK BIT8
 CCS A
 TCF GETNEWNM

 CA Q
 TS TEMPNM
 CA BB
 EXTEND
 ROR SUPERBNK # PICK UP USER'S SUPERBANK
 TS TEMPBB

TOCON2 CA CON2ADR # BACK TO SWITCHED BANK
 LXCH TEMPBBCN
 DTCB

CON2ADR GENADR CON2

GETPRIO NDX Q # DON'T CARE IF DIRECT OR INDIRECT
 CA 0 # LEAVE THAT DECISION TO RESTARTS

```



July 12, 2016

1417

|             |        |          |                         |
|-------------|--------|----------|-------------------------|
|             | INCR   | Q        | # OBTAIN RETURN ADDRESS |
|             | TCF    | CON1 -1  |                         |
| GETNEWNM    | EXTEND |          |                         |
| # Page 1411 |        |          |                         |
|             | INDEX  | Q        |                         |
|             | DCA    | 0        |                         |
|             | DXCH   | TEMPNM   |                         |
|             | CA     | TWO      |                         |
|             | ADS    | Q        | # OBTAIN RETURN ADDRESS |
|             | TCF    | TOCON2   |                         |
| OCT14000    | EQUALS | PRI014   |                         |
| TEMPG       | EQUALS | ITEMP1   |                         |
| TEMPP       | EQUALS | ITEMP2   |                         |
| TEMPNM      | EQUALS | ITEMP3   |                         |
| TEMPBB      | EQUALS | ITEMP4   |                         |
| TEMPSW      | EQUALS | ITEMP5   |                         |
| TEMPSW2     | EQUALS | ITEMP6   |                         |
| TEMPPR      | EQUALS | RUPTREG1 |                         |
| TEMPG2      | EQUALS | RUPTREG2 |                         |
| TEMPP2      | EQUALS | RUPTREG3 |                         |
| TEMPBBCN    | EQUALS | RUPTREG4 |                         |
| BB          | EQUALS | BBANK    |                         |
|             | BANK   | 14       |                         |
|             | SETLOC | PHASETAB |                         |
|             | BANK   |          |                         |
|             | EBANK= | PHSNAME1 |                         |
|             | COUNT  | 10/PHASE |                         |
| PHSCHNG2    | LXCH   | TEMPBBCN |                         |
|             | CA     | TEMPSW   |                         |
|             | MASK   | OCT7     |                         |
|             | DOUBLE |          |                         |
|             | TS     | TEMPG    |                         |
|             | CA     | TEMPSW   |                         |
|             | MASK   | OCT17770 |                         |
|             | EXTEND |          |                         |
|             | MP     | BIT12    |                         |
|             | TS     | TEMPP    |                         |

|             |        |            |                                          |
|-------------|--------|------------|------------------------------------------|
|             | CA     | TEMPSW     |                                          |
|             | MASK   | OCT60000   |                                          |
|             | XCH    | TEMPSW     |                                          |
|             | MASK   | OCT14000   |                                          |
|             | CCS    | A          |                                          |
|             | TCF    | ONEORTWO   |                                          |
| # Page 1412 |        |            |                                          |
|             | CA     | TEMPP      | # START STORING THE PHASE INFORMATION    |
|             | NDX    | TEMPPG     |                                          |
|             | TS     | PHASE1 -2  |                                          |
| BELOW1      | CCS    | TEMPSW2    | # IS IT A PHASCHNG OR A 2PHSCHNG         |
|             | TCF    | BELOW2     | # IT'S A PHASCHNG                        |
|             | TCF    | +1         | # IT'S A 2PHSCHNG                        |
|             | CS     | TEMPP2     |                                          |
|             | LXCH   | TEMPP2     |                                          |
|             | NDX    | TEMPPG2    |                                          |
|             | DXCH   | -PHASE1 -2 |                                          |
|             | CCS    | TEMPSW2    |                                          |
|             | NOOP   |            | # CAN'T GET HERE                         |
|             | TCF    | BELOW2     |                                          |
|             | CS     | TIME1      |                                          |
|             | NDX    | TEMPPG2    |                                          |
|             | TS     | TBASE1 -2  |                                          |
| BELOW2      | CCS    | TEMPSW     | # SEE IF WE SHOULD SET TBASE OR LONGBASE |
|             | TCF    | BELOW3     | # SET LONGBASE ONLY                      |
|             | TCF    | BELOW4     | # SET NEITHER                            |
|             | CS     | TIME1      | # SET TBASE TO BEGIN WITH                |
|             | NDX    | TEMPPG     |                                          |
|             | TS     | TBASE1 -2  |                                          |
|             | CA     | TEMPSW     | # SHALL WE NOW SET LONGBASE              |
|             | AD     | BIT14COM   |                                          |
|             | CCS    | A          |                                          |
|             | NOOP   |            | # ***** CAN'T GET HERE *****             |
| BIT14COM    | OCT    | 17777      | # ***** CAN'T GET HERE *****             |
|             | TCF    | BELOW4     | # NO WE NEED ONLY SET TBASE              |
| BELOW3      | EXTEND |            | # SET LONGBASE                           |
|             | DCA    | TIME2      |                                          |

```

 DXCH LONGBASE

BELOW4 CS TEMPP # AND STORE THE FINAL PART OF THE PHASE
 NDX TEMPG
 TS -PHASE1 -2

 CA Q
 LXCH TEMPBBCN
 RELINT
 DTCB
CON2 LXCH TEMPBBCN
Page 1413

 CA TEMPP
 NDX TEMPG
 TS PHASE1 -2

 CA TEMPPR
 NDX TEMPG
 TS PHSPRDT1 -2

 EXTEND
 DCA TEMPNM
 NDX TEMPG
 DXCH PHSNAME1 -2

 TCF BELOW1

 BLOCK 02
 SETLOC FFTAG1
 BANK

 COUNT 02/PHASE

CHECKB MASK BIT12 # SINCE THIS IS OF TYPE B, THIS BIT SHOULD
 CCS A # BE HERE IF WE ARE TO GET A NEW PRIORITY
 TCF GETPRIO # IT IS, SO GET NEW PRIORITY

 TCF OLDPRIO # IT ISN'T, USE THE OLD PRIORITY.

```

This code is written to file `src/PHASE-TABLE-MAINTENANCE.s`.

## B.81 PINBALL GAME BUTTONS AND LIGHTS

```

1420 <src/PINBALL-GAME-BUTTONS-AND-LIGHTS.s 1420>≡
Copyright: Public domain.
Filename: PINBALL_GAME_BUTTONS_AND_LIGHTS.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 307-389
Mod history: 2009-05-08 RSB Started adapting from the Colossus249/ file
of the same name, using Comanche055 page
images. Finished through page 329.
2009-05-09 RSB Finished first draft.
2009-05-20 RSB Corrected a CHKPOOH to CHKPOOH.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 307
PROGRAM NAME -- KEYBOARD AND DISPLAY PROGRAM
MOD NO -- 4 DATE -- 27 APRIL 1967 ASSEMBLY -- PINDISK REV 17
MOD BY -- FILENE
LOG SECTION -- PINBALL GAME BUTTONS AND LIGHTS
#
FUNCTIONAL DESCRIPTION
#
THE KEYBOARD AND DISPLAY SYSTEM PROGRAM OPERATES UNDER EXECUTIVE
CONTROL AND PROCESSES INFORMATION EXCHANGED BETWEEN THE AGC AND THE
COMPUTER OPERATOR. THE INPUTS TO THE PROGRAM ARE FROM THE KEYBOARD,

```

```

FROM INTERNAL PROGRAM, AND FROM THE UPLINK.
#
THE LANGUAGE OF COMMUNICATION WITH THE PROGRAM IS A PAIR OF WORDS
KNOWN AS VERB AND NOUN. EACH OF THESE IS REPRESENTED BY A 2 CHARACTER
DECIMAL NUMBER. THE VERB CODE INDICATES WHAT ACTION IS TO BE TAKEN, THE
NOUN CODE INDICATES TO WHAT THIS ACTION IS APPLIED. NOUNS USUALLY
REFER TO A GROUP OF ERASABLE REGISTERS.
#
VERBS ARE GROUPED INTO DISPLAYS, LOADS, MONITORS (DISPLAYS THAT ARE
UPDATED ONCE PER SECOND), SPECIAL FUNCTIONS, AND EXTENDED VERBS (THESE
ARE OUTSIDE OF THE DOMAIN OF PINBALL AND CAN BE FOUND UNDER LOG SECTION
'EXTENDED VERBS').
#
A LIST OF VERBS AND NOUNS IS GIVEN IN LOG SECTION 'ASSEMBLY AND
OPERATION INFORMATION'.
#
CALLING SEQUENCES --
#
KEYBOARD:
EACH DEPRESSION OF A MAIN (NAVIGATION) KEYBOARD BUTTON ACTIVATES
INTERRUPT KEYRUPT1 (KEYRUPT2) AND PLACES THE 5 BIT KEY CODE INTO
CHANNEL 15 (CHANNEL 16). KEYRUPT1 (KEYRUPT2) PLACES THE KEY
CODE INTO MPAC, ENTERS AN EXECUTIVE REQUEST FOR THE KEYBOARD AND DISPLAY
PROGRAM (AT 'CHARIN'), AND EXECUTES A RESUME.
#
UPLINK:
EACH WORD RECEIVED BY THE UPLINK ACTIVATES INTERRUPT UPRUPT, WHICH
PLACES THE 5 BIT KEY CODE INTO MPAC, ENTERS AN EXECUTIVE REQUEST FOR THE
KEYBOARD AND DISPLAY PROGRAM (AT 'CHARIN') AND EXECUTES A RESUME.
#
INTERNAL PROGRAMS:
INTERNAL PROGRAMS CALL PINBALL AT 'NVSUB' WITH THE DESIRED VERB/NOUN
CODE IN A (LOW 7 BITS FOR NOUN, NEXT 7 BITS FOR VERB). DETAILS
DESCRIBED ON REMARKS CARDS JUST BEFORE 'NVSUB' AND 'NVSBWAIT' (SEE
SYMBOL TABLE FOR PAGE NUMBERS).
#
NORMAL EXIT MODES --
#
IF PINBALL WAS CALLED BY EXTERNAL ACTION, THERE ARE FOUR EXITS:
1) ALL BUT (2), (3), AND (4) EXIT DIRECTLY TO ENDOFJOB.
Page 308
2) EXTENDED VERBS TO TO THE EXTENDED VERB FAN AS PART OF THE
PINBALL EXECUTIVE JOB WITH PRIORITY 30000. IT IS THE
RESPONSIBILITY OF THE EXTENDED VERB CALLED TO EVENTUALLY
CHANGE PRIORITY (IF NECESSARY) AND DO AN ENDOFJOB.
ALSO PINBALL IS A NOVAC JOB. EBANK SET FOR COMMON.

```

```

3) VERB 37. CHANGE OF PROGRAM (MAJOR MODE) CALLS 'V37' IN THE
SERVICE ROUTINES AS PART OF THE PINBALL EXEC JOB WITH PRIO
30000. THE NEW PROGRAM CODE (MAJOR MODE) IS LEFT IN A.
4) KEY RELEASE BUTTON CALLS 'PINBRNCH' IN THE DISPLAY INTERFACE
ROUTINES AS PART OF THE PINBALL EXEC JOB WITH PRIO 30000 IF
THE KEY RELEASE LIGHT IS OFF AND 'CADRSTOR' IS NOT 40.
#
IF PINBALL WAS CALLED BY INTERNAL PROGRAMS, EXIT FROM PINBALL IS BACK
TO CALLING ROUTINE. DETAILS DESCRIBED IN REMARKS CARDS JUST BEFORE
'NVSUB' AND 'NVSBWAIT' (SEE SYMBOL TABLE FOR PAGE NUMBERS).
#
ALARM OR ABORT EXIT MODES --
#
EXTERNAL INITIATION:
IF SOME IMPROPER SEQUENCE OF KEY CODES IS DETECTED, THE OPERATOR
ERROR LIGHT IS TURNED ON AND EXIT IS TO 'ENDOFJOB'.
#
INTERNAL PROGRAM INITIATION:
IF AN ILLEGAL V/N COMBINATION IS ATTEMPTED, AN ABORT IS CAUSED
(WITH OCTAL 01501).
IF A SECOND ATTEMPT IS MADE TO GO TO SLEEP IN PINBALL, AN ABORT IS
CAUSED (WITH OCTAL 01206). THERE ARE TWO WAYS TO GO TO SLEEP IN PINBALL:
1) ENDIDLE OR DATAWAIT.
2) NVSBWAIT, PRENVBSY, OR NVSUBUSY.
#
CONDITIONS LEADING TO THE ABOVE ARE DESCRIBED IN FORTHCOMING MIT/IL
E-REPORT DESCRIBING KEYBOARD AND DISPLAY OPERATION FOR 278.
#
OUTPUT --
#
INFORMATION TO BE SENT TO THE DISPLAY PANEL IS LEFT IN THE 'DSPTAB'
BUFFERS REGISTERS (UNDER EXEC CONTROL). 'DSPOUT' (A PART OF T4RUPT)
HANDLES THE PLACING OF THE 'DSPTAB' INFORMATION INTO OUTPUT CHANNEL 10
IN INTERRUPT.
#
ERASABLE INITIALIZATION --
#
FRESH START AND RESTART INITIALIZE THE NECESSARY E REGISTERS FOR
PINBALL IN 'STARTSUB'. REGISTERS ARE: DSPTAB BUFFER, CADRSTOR,
REQRET, CLPASS, DSPLOCK, MONSAVE, MONSAVE1, VERBREG, NOUNREG, DSPLIST,
DSPCOUNT, NOUT.
#
Page 309
A COMPLETE LIST OF ALL THE ERASABLES (BOTH RESERVED AND TEMPORARIES) FOR
PINBALL IS GIVEN BELOW.
#

```

```

THE FOLLOWING ARE OF GENERAL INTEREST --
#
REMARKS CARDS PRECEDE THE REFERENCED SYMBOL DEFINITION. SEE SYMBOL
TABLE TO FIND APPROPRIATE PACE NUMBERS.
#
NVSUB CALLING POINT FOR INTERNAL USE OF PINBALL.
OF RELATED INTEREST NVSBWAIT
NVSUBBUSY
PRENVBSY
#
ENDIDLE ROUTINE FOR INTERNAL PROGRAMS WISHING TO TO SLEEP WHILE
AWAITING OPERATOR'S RESPONSE.
#
DSPMM ROUTINE BY WHICH AN INTERNAL PROGRAM MAY DISPLAY A DECIMAL
PROGRAM CODE (MAJOR MODE) IN THE PROGRAM (MAJOR MODE) LIGHT
(DSPMM DOES NOT DISPLAY DIRECTLY BUT ENTERS EXEC REQUEST
FOR DSPMMJB WITH PRIO 30000 AND RETURNS TO CALLER.)
#
BLANKSUB ROUTINE BY WHICH AN INTERNAL PROGRAM MAY BLANK ANY
COMBINATION OF THE DISPLAY REGISTERS R1, R2, R3.
#
JAMTERM ROUTINE BY WHICH AN INTERNAL PROGRAM MAY PERFORM THE
JAMPROC TERMINATE (V 34) OR PROCEED (V33) FUNCTION.
#
MONITOR VERBS FOR PERIODIC (1 PER SEC) DISPLAY.
#
PLEASE PERFORM, PLEASE MARK SITUATIONS
REMARKS DESCRIBING HOW AN INTERNAL ROUTINE SHOULD HANDLE
THESE SITUATIONS CAN BE FOUND JUST BEFORE 'NVSUB' (SEE
SYMBOL TABLE FOR PAGE NUMBER).
#
THE NOUN TABLE FORMAT IS DESCRIBED ON A PAGE OF REMARKS CARDS JUST
BEFORE 'DSPABC' (SEE SYMBOL TABLE FOR PAGE NUMBER).
#
THE NOUN TABLES THEMSELVES ARE FOUND IN LOG SECTION 'PINBALL NOUN
TABLES'.
#
FOR FURTHER DETAILS ABOUT OPERATION OF THE KEYBOARD AND DISPLAY SYSTEM
PROGRAM, SEE THE MISSION PLAN AND/OR MIT/IL E-2129
DESCRIBING KEYBOARD AND DISPLAY OPERATION FOR 278.
(Note that this doc by Green and Filene is/was available online at
http://hrst.mit.edu/hrs/apollo/public/archive/1706.pdf --- RSB 6/2004.)
#
THE FOLLOWING QUOTATION IS PROVIDED THROUGH THE COURTESY OF THE AUTHORS.
#
"IT WILL BE PROVED TO THY FACE THAT THOU HAST MEN ABOUT THEE THAT

```

# Page 310

# USUALLY TALK OF A NOUN AND A VERB, AND SUCH ABOMINABLE WORDS AS NO  
# CHRISTIAN EAR CAN ENDURE TO HEAR."  
# HENRY 6, ACT 2, SCENE 4

# THE FOLLOWING ASSIGNMENTS FOR PINBALL ARE MADE ELSEWHERE

# RESERVED FOR PINBALL EXECUTIVE ACTION

#  
#DSPCOUNT ERASE # DISPLAY POSITION INDICATOR  
#DECBRNCH ERASE # +DEC, -DEC, OCT INDICATOR  
#VERBREG ERASE # VERB CODE  
#NOUNREG ERASE # NOUN CODE  
#XREG ERASE # R1 INPUT BUFFER  
#YREG ERASE # R2 INPUT BUFFER  
#ZREG ERASE # R3 INPUT BUFFER  
#XREGLP ERASE # LO PART OF XREG (FOR DEC CONV ONLY)  
#YREGLP ERASE # LO PART OF YREG (FOR DEC CONV ONLY)  
#HITEMOUT = YREGLP # TEMP FOR DISPLAY OF HRS,MIN,SEC  
# MUST = LOTEMOUT-1.  
#ZREGLP ERASE # LO PART OF ZREG (FOR DEC CONV ONLY)  
#LOTEMOUT = ZREGLP # TEMP FOR DISPLAY OF HRS,MIN,SEC  
# MUST = HITEMOUT+1  
#MODREG ERASE # MODE CODE  
#DSPLOCK ERASE # KEYBOARD/SUBROUTINE CALL INTERLOCK  
#REQRET ERASE # RETURN REGISTER FOR LOAD  
#LOADSTAT ERASE # STATUS INDICATOR FOR LOADTST  
#CLPASS ERASE # PASS INDICATOR FOR CLEAR  
#NOUT ERASE # ACTIVITY COUNTER FOR DSPTAB  
#NOUNCADR ERASE # MACHINE CADR FOR NOUN  
#MONSAVE ERASE # N/V CODE FOR MONITOR. (= MONSAVE1-1)  
#MONSAVE1 ERASE # NOUNCADR FOR MONITOR (MATBS1) = MONSAVE+1  
#MONSAVE2 ERASE # NVMONOPT OPTIONS  
#DSPTAB ERASE +13D # 0-10, DISPLAY PANEL BUFFER 11-13, C RELAYS  
#CADRSTOR ERASE # ENDIDLE STORAGE  
#NVQTEM ERASE # NVSUB STORAGE FOR CALLING ADDRESS  
# MUST = NVBNKTEM-1.  
#NVBNKTEM ERASE # NVSUB STORAGE FOR CALLING BANK  
# MUST = NVQTEM+1  
#VERBSAVE ERASE # NEEDED FOR RECYCLE  
#DSPLIST ERASE # WAITING REG FOR DSP SYST INTERNAL USE  
#EXTVBACT ERASE # EXTENDED VERB ACTIVITY INTERLOCK  
#DSPTEM1 ERASE +2 # BUFFER STORAGE AREA 1 (MOSTLY FOR TIME)  
#DSPTEM2 ERASE +2 # BUFFER STORAGE AREA 2 (MOSTLY FOR DEG)  
#  
# END OF ERASABLES RESERVED FOR PINBALL EXECUTIVE ACTION.



```

#
TEMPORARIES FOR PINBALL EXECUTIVE ACTION
Page 311
#DSEXIT = INTB15+ # RETURN FOR DSPIN
#EXITEM = INTB15+ # RETURN FOR SCALE FACTOR ROUTINE SELECT
#BLANKRET = INTB15+ # RETURN FOR 2BLANK
#WRDRET = INTBIT15 # RETURN FOR 5BLANK.
#WDRET = INTBIT15 # RETURN FOR DSPWD
#DECRET = INTBIT15 # RETURN FOR PUTCOM(DEC LOAD)
#21/22REG = INTBIT15 # TEMP FOR CHARIN
#UPDATRET = POLISH # RETURN FOR UPDATNN, UPDATVB
#CHAR = POLISH # TEMP FOR CHARIN
#ERCNT = POLISH # COUNTER FOR ERROR LIGHT RESET
#DECOUNT = POLISH # COUNTER FOR SCALING AND DISPLAY (DEC)
#SGNON = VBUF # TEMP FOR +,- ON
#NOUNTEM = VBUF # COUNTER FOR MIXNOUN FETCH
#DISTEM = VBUF # COUNTER FOR OCTAL DISPLAY VERB
#DECTEM = VBUF # COUNTER FOR FETCH (DEC DISPLAY VERBS)
#SGNOFF = VBUF +1 # TEMP FOR +,- ON
#NVTEMP = VBUF +1 # TEMP FOR NVSUB
#SFTEMP1 = VBUF +1 # STORAGE FOR SF CONST HI PART (=SFTEMP2-1)
#HITEMIN = VBUF +1 # TEMP FOR LOAD OF HRS,MIN,SEC
= MUST = LOTEMIN-1.
#CODE = VBUF +2 # FOR DSPIN
#SFTEMP2 = VBUF +2 # STORAGE FOR SF CONST LO PART (=SFTEMP1+1)
#LOTEMIN = VBUF +2 # TEMP FOR LOAD OF HRS,MIN,SEC
= MUST = HITEMIN+1
#MIXTEMP = VBUF +3 # FOR MIXNOUN DATA
#SIGNRET = VBUF +3 # RETURN FOR +,- ON
ALSO MIXTEMP+1 = VBUF+4, MIXTEMP+2 = VBUF+5
#ENTRET = DOTINC # EXIT FROM ENTER
#WDONT = DOTRET # CHAR COUNTER FOR DSPWD
#INREL = DOTRET # INPUT BUFFER SELECTOR (X,Y,Z, REG)
#DSPMMTEM = MATINC # DSPCOUNT SAVE FOR DSPMM
#MIXBR = MATINC # INDICATOR FOR MIXED OR NORMAL NOUN
#TEM1 ERASE # EXEC TEMP
#DSREL = TEM1 # REL ADDRESS FOR DSPIN
#TEM2 ERASE # EXEC TEMP
#DSMAG = TEM2 # MAGNITUDE STORE FOR DSPIN
#IDADDTEM = TEM2 # MIXNOUN INDIRECT ADDRESS STORAGE
#TEM3 ERASE # EXEC TEMP
#COUNT = TEM3 # FOR DSPIN
Page 312
#TEM4 ERASE # EXEC TEMP
#LSTPTR = TEM4 # LIST POINTER FOR GRABUSY
#RELRET = TEM4 # RETURN FOR RELDSP

```

```

#FREERET = TEM4 # RETURN FOR FREEDSP
#DSPWDRET = TEM4 # RETURN FOR DSPSIGN
#SEPSCRET = TEM4 # RETURN FOR SEPSEC
#SEPMNRET = TEM4 # RETURN FOR SEPMIN
#TEM5 ERASE # EXEC TEMP
#NOUNADD = TEM5 # TEMP STORAGE FOR NOUN ADDRESS
#NNADTEM ERASE # TEMP FOR NOUN ADDRESS TABLE ENTRY
#NNTYPTTEM ERASE # TEMP FOR NOUN TYPE TABLE ENTRY
#IDAD1TEM ERASE # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
MUST = IDAD2TEM-1, = IDAD3TEM-2
#IDAD2TEM ERASE # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
MUST = IDAD1TEM+1, IDAD3TEM-1.
#IDAD3TEM ERASE # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
MUST = IDAD1TEM+2, IDAD2TEM+1.
#RUTMXTEM ERASE # TEMP FOR SF ROUT TABLE ENTRY (MIXNN ONLY)
#
END OF TEMPORARIES FOR PINBALL EXECUTIVE ACTION.
#
ADDITIONAL TEMPORARIES FOR PINBALL EXECUTIVE ACTION
#
MPAC, THRU MPAC +6
BUF, +1, +2
BUF2, +1, +2
MPTEMP
ADDRWD
#
END OF ADDITIONAL TEMPS FOR PINBALL EXEC ACTION
#
RESERVED FOR PINBALL INTERRUPT ACTION
#
#DSPCNT ERASE # COUNTER FOR DSPOUT
#UPLOCK ERASE # BIT1 = UPLINK INTERLOCK (ACTIVATED BY
RECEPTION OF A BAD MESSAGE IN UPLINK)
#
END OF ERASABLES RESERVED FOR PINBALL INTERRUPT ACTION
#
TEMPORARIES FOR PINBALL INTERRUPT ACTION
#
#KEYTEMP1 = WAITEXIT # TEMP FOR KEYRUPT, UPRUPT
#DSRUPTTEM = WAITEXIT # TEMP FOR DSPOUT
#KEYTEMP2 = RUPTAGN # TEMP FOR KEYRUPT, UPRUPT
#
END OF TEMPORARIES FOR PINBALL INTERRUPT ACTION

Page 313
THE INPUT CODES ASSUMED FOR THE KEYBOARD ARE,

```

```

0 10000
1 00001
9 01001
VERB 10001
ERROR RES 10010
KEY RLSE 11001
+ 11010
- 11011
ENTER 11100
CLEAR 11110
NOUN 11111
#
OUTPUT FORMAT FOR DISPLAY PANEL. SET OUTO TO AAAABCCCCDDDDDD.
A'S SELECTS A RELAYWORD. THIS DETERMINES WHICH PAIR OF CHARACTERS ARE
ENERGIZED.
B FOR SPECIAL RELAYS SUCH AS SIGNS ETC.
C'S 5 BIT RELAY CODE FOR LEFT CHAR OF PAIR SELECTED BY RELAYWORD.
D'S 5 BIT RELAY CODE FOR RIGHT CHAR OF PAIR SELECTED BY RELAYWORD.
#
THE PANEL APPEARS AS FOLLOWS,
MD1 MD2 (MAJOR MODE)
VD1 VD2 (VERB)
R1D1 R1D2 R1D3 R1D4 R1D5 (R1)
R2D1 R2D2 R2D3 R2D4 R2D5 (R2)
R3D1 R3D2 R3D3 R3D4 R3D5 (R3)
#
EACH OF THESE IS GIVEN A DSPCOUNT NUMBER FOR USE WITHIN COMPUTATION ONLY
#
MD1 25 R2D1 11 ALL ARE OCTAL
MD2 24 R2D2 10
VD1 23 R2D3 7
VD2 22 R2D4 6
ND1 21 R2D5 5
ND2 20 R3D1 4
R1D1 16 R3D2 3
R1D2 15 R3D3 2
R1D3 14 R3D4 1
R1D4 13 R3D5 0
R1D5 12
#
THERE IS AN 11-REGISTER TABLE (DSPTAB) FOR THE DISPLAY PANEL.
#
DSPTAB RELAYWD BIT11 BITS 10-6 BITS 5-1
RELADD
10 1011 MD1 (25) MD2 (24)
9 1010 VD1 (23) VD2 (22)

```

|   |          |      |              |           |           |
|---|----------|------|--------------|-----------|-----------|
| # | 8        | 1001 |              | ND1 (21)  | ND2 (20)  |
| # | 7        | 1000 |              |           | R1D1 (16) |
| # | Page 314 |      |              |           |           |
| # | 6        | 0111 | +R1          | R1D2 (15) | R1D3 (14) |
| # | 5        | 0110 | -R1          | R1D4 (13) | R1D5 (12) |
| # | 4        | 0101 | +R2          | R2D1 (11) | R2D2 (10) |
| # | 3        | 0100 | -R2          | R2D3 (7)  | R2D4 (6)  |
| # | 2        | 0011 |              | R2D5 (5)  | R3D1 (4)  |
| # | 1        | 0010 | +R3          | R3D2 (3)  | R3D3 (2)  |
| # | 0        | 0001 | -R3          | R3D4 (1)  | R3D5 (0)  |
| # |          | 0000 | NO RELAYWORD |           |           |

#  
# THE 5-BIT OUTOUT RELAY CODES ARE:  
#

|   |       |       |
|---|-------|-------|
| # | BLANK | 00000 |
| # | 0     | 10101 |
| # | 1     | 00011 |
| # | 2     | 11001 |
| # | 3     | 11011 |
| # | 4     | 01111 |
| # | 5     | 11110 |
| # | 6     | 11100 |
| # | 7     | 10011 |
| # | 8     | 11101 |
| # | 9     | 11111 |

#  
# OUTPUT BITS USED BY PINBALL:  
#

|   |                      |    |                     |
|---|----------------------|----|---------------------|
| # | KEY RELEASE LIGHT    | -- | BIT 5 OF CHANNEL 11 |
| # | VERB/NOUN FLASH      | -- | BIT 6 OF CHANNEL 11 |
| # | OPERATOR ERROR LIGHT | -- | BIT 7 OF CHANNEL 11 |

# Page 315

# START OF EXECUTIVE SECTION OF PINBALL

BANK 40  
SETLOC PINBALL1  
BANK

COUNT 40/PIN

|        |     |          |                                           |
|--------|-----|----------|-------------------------------------------|
| CHARIN | CAF | ONE      | # BLOCK DISPLAY SYST                      |
|        | XCH | DSPLOCK  | # MAKE DSP SYST BUSY, BUT SAVE OLD        |
|        | TS  | 21/22REG | # C(DSPLOCK) FOR ERROR LIGHT RESET.       |
|        | CCS | CADRSTOR | # ALL KEYS EXCEPT ER TURN ON KR LITE IF   |
|        | TC  | +2       | # CADRSTOR IS FULL. THIS REMINDS OPERATOR |

July 12, 2016

1429

|            |        |          |   |                                        |                   |
|------------|--------|----------|---|----------------------------------------|-------------------|
|            | TC     | CHARIN2  | # | TO RE-ESTABLISH A FLASHING DISPLAY     |                   |
|            | CS     | ELRCODE1 | # | WHICH HE HAS OBSCURED WITH DISPLAYS OF |                   |
|            | AD     | MPAC     | # | HIS OWN (SEE REMARKS PRECEDING ROUTINE |                   |
|            | EXTEND |          | # | VBRELDSP).                             |                   |
|            | BZF    | CHARIN2  |   |                                        |                   |
|            | TC     | RELDSPON |   |                                        |                   |
| CHARIN2    | XCH    | MPAC     |   |                                        |                   |
|            | TS     | CHAR     |   |                                        |                   |
|            | INDEX  | A        |   |                                        |                   |
|            | TC     | +1       | # | INPUT CODE                             | FUNCTION          |
|            | TC     | CHARALRM | # | 0                                      |                   |
|            | TC     | NUM      | # | 1                                      |                   |
|            | TC     | NUM      | # | 2                                      |                   |
|            | TC     | NUM      | # | 3                                      |                   |
|            | TC     | NUM      | # | 4                                      |                   |
|            | TC     | NUM      | # | 5                                      |                   |
|            | TC     | NUM      | # | 6                                      |                   |
|            | TC     | NUM      | # | 7                                      |                   |
|            | TC     | 89TEST   | # | 10                                     | 8                 |
|            | TC     | 89TEST   | # | 11                                     | 9                 |
|            | TC     | CHARALRM | # | 12                                     |                   |
|            | TC     | CHARALRM | # | 13                                     |                   |
|            | TC     | CHARALRM | # | 14                                     |                   |
|            | TC     | CHARALRM | # | 15                                     |                   |
|            | TC     | CHARALRM | # | 16                                     |                   |
|            | TC     | CHARALRM | # | 17                                     |                   |
|            | TC     | NUM -2   | # | 20                                     | 0                 |
|            | TC     | VERB     | # | 21                                     | VERB              |
|            | TC     | ERROR    | # | 22                                     | ERROR LIGHT RESET |
|            | TC     | CHARALRM | # | 23                                     |                   |
|            | TC     | CHARALRM | # | 24                                     |                   |
|            | TC     | CHARALRM | # | 25                                     |                   |
|            | TC     | CHARALRM | # | 26                                     |                   |
|            | TC     | CHARALRM | # | 27                                     |                   |
|            | TC     | CHARALRM | # | 30                                     |                   |
|            | TC     | VBRELDSP | # | 31                                     | KEY RELEASE       |
| # Page 316 | TC     | POSGN    | # | 32                                     | +                 |
|            | TC     | NEGSGN   | # | 33                                     | -                 |
|            | TC     | ENTERJMP | # | 34                                     | ENTER             |
|            | TC     | CHARALRM | # | 35                                     |                   |
|            | TC     | CLEAR    | # | 36                                     | CLEAR             |
|            | TC     | NOUN     | # | 37                                     | NOUN              |
| ELRCODE1   | OCT    | 22       |   |                                        |                   |
| ENTERJMP   | TC     | POSTJUMP |   |                                        |                   |

|        |      |          |                                          |
|--------|------|----------|------------------------------------------|
|        | CADR | ENTER    |                                          |
| 89TEST | CCS  | DSPCOUNT |                                          |
|        | TC   | +4       | # +                                      |
|        | TC   | +3       | # +0                                     |
|        | TC   | ENDOFJOB | # - BLOCK DATA IN IF DSPCOUNT IS - OR -0 |
|        | TC   | ENDOFJOB | # -0                                     |
|        | CAF  | THREE    |                                          |
|        | MASK | DECBRNCH |                                          |
|        | CCS  | A        |                                          |
|        | TC   | NUM      | # IF DECBRNCH IS +, 8 OR 9 OK            |
|        | TC   | CHARALRM | # IF DECBRNCH IS +0, REJECT 8 OR 9       |

# NUM ASSEMBLES OCTAL 3 BITS AT A TIME. FOR DECIMAL IT CONVERTS INCOMING  
 # WORD AS A FRACTION, KEEPING RESULTS TO DP.  
 # OCTAL RESULTS ARE LEFT IN XREG, YREG, OR ZREG. HI PART OF DEC IN XREG,  
 # YREG, ZREG. THE LOW PARTS IN XREGLP, YREGLP, OR ZREGLP.  
 # DECBRNCH IS LEFT AT +0 FOR OCT, +1 FOR + DEC, +2 FOR - DEC.  
 # IF DSPCOUNT WAS LEFT -, NO MORE DATA IS ACCEPTED.

|     |       |          |                                     |
|-----|-------|----------|-------------------------------------|
|     | CAF   | ZERO     |                                     |
|     | TS    | CHAR     |                                     |
| NUM | CCS   | DSPCOUNT |                                     |
|     | TC    | +4       | # +                                 |
|     | TC    | +3       | # +0                                |
|     | TC    | +1       | # -BLOCK DATA IN IF DSPCOUNT IS -   |
|     | TC    | ENDOFJOB | # -0                                |
|     | TC    | GETINREL |                                     |
|     | CCS   | CLPASS   | # IF CLPASS IS + OR +0, MAKE IT +0. |
|     | CAF   | ZERO     |                                     |
|     | TS    | CLPASS   |                                     |
|     | TC    | +1       |                                     |
|     | INDEX | CHAR     |                                     |
|     | CAF   | RELTAB   |                                     |
|     | MASK  | LOW5     |                                     |
|     | TS    | CODE     |                                     |
|     | CA    | DSPCOUNT |                                     |
|     | TS    | COUNT    |                                     |
|     | TC    | DSPIN    |                                     |

# Page 317

|  |       |          |                                    |
|--|-------|----------|------------------------------------|
|  | CAF   | THREE    |                                    |
|  | MASK  | DECBRNCH |                                    |
|  | CCS   | A        | # +0, OCTAL. +1, + DEC. +2, - DEC. |
|  | TC    | DECTOBIN | # +                                |
|  | INDEX | INREL    | # +0 OCTAL                         |
|  | XCH   | VERBREG  |                                    |

July 12, 2016

1431

|            |        |           |                                              |
|------------|--------|-----------|----------------------------------------------|
|            | TS     | CYL       |                                              |
|            | CS     | CYL       |                                              |
|            | CS     | CYL       |                                              |
|            | XCH    | CYL       |                                              |
|            | AD     | CHAR      |                                              |
|            | TC     | ENDNMTST  |                                              |
| DECTOBIN   | INDEX  | INREL     |                                              |
|            | XCH    | VERBREG   |                                              |
|            | TS     | MPAC      | # SUM X 2EXP-14 IN MPAC                      |
|            | CAF    | ZERO      |                                              |
|            | TS     | MPAC +1   |                                              |
|            | CAF    | TEN       | # 10 X 2EXP-14                               |
|            | TC     | SHORTMP   | # 10SUM X 2EXP-28 IN MPAC, MPAC+1            |
|            | XCH    | MPAC +1   |                                              |
|            | AD     | CHAR      |                                              |
|            | TS     | MPAC +1   |                                              |
|            | TC     | ENDNMTST  | # NO OF                                      |
|            | ADS    | MPAC      | # OF MUST BE 5TH CHAR                        |
|            | TC     | DECEND    |                                              |
| ENDNMTST   | INDEX  | INREL     |                                              |
|            | TS     | VERBREG   |                                              |
|            | CS     | DSPCOUNT  |                                              |
|            | INDEX  | INREL     |                                              |
|            | AD     | CRITCON   |                                              |
|            | EXTEND |           |                                              |
|            | BZF    | ENDNUM    | # -0, DSPCOUNT = CRITCON                     |
|            | TC     | MORNUM    | # -, DSPCOUNT G/ CRITCON                     |
| ENDNUM     | CAF    | THREE     |                                              |
|            | MASK   | DECBRNCH  |                                              |
|            | CCS    | A         |                                              |
|            | TC     | DECEND    |                                              |
| ENDALL     | CS     | DSPCOUNT  | # BLOCK NUMIN BY PLACING DSPCOUNT            |
|            | TC     | MORNUM +1 | # NEGATIVELY                                 |
| DECEND     | CS     | ONE       |                                              |
|            | AD     | INREL     |                                              |
|            | EXTEND |           |                                              |
|            | BZMF   | ENDALL    | # IF INREL=0,1 (VBREG,NNREG) LEAVE WHOLE     |
|            | TC     | DMP       | # IF INREL=2,3,4 (R1,R2,R3), CONVERT TO FRAC |
|            |        |           | # MULT SUM X 2EXP-28 IN MPAC, MPAC+1 BY      |
|            | ADRES  | DECON     | # 2EXP14/10EXP5, GIVES (SUM/10EXP5)X2EXP-14  |
|            | CAF    | THREE     | # IN MPAC, +1, +2.                           |
|            | MASK   | DECBRNCH  |                                              |
|            | INDEX  | A         |                                              |
|            | TC     | +0        |                                              |
| # Page 318 | TC     | +DECSGN   |                                              |

```

 EXTEND # - CASE
 DCS MPAC +1
 DXCH MPAC +1
+DECSGN XCH MPAC +2
 INDEX INREL
 TS XREGLP -2
 XCH MPAC +1
 INDEX INREL
 TS VERBREG
 TC ENDALL
MORNUM CCS DSPCOUNT # DECREMENT DSPCOUNT
 TS DSPCOUNT
 TC ENDOFJOB

CRITCON OCT 22 # (DEC 18)
 OCT 20 # (DEC 16)
 OCT 12 # (DEC 10)
 OCT 5
 OCT 0

DECON 2DEC 1 E-5 B14 # 2EXP14/10EXP5 = .16384 DEC

GETINREL GETS PROPER DATA REG REL ADDRESS FOR CURRENT C(DSPCOUNT) AND
PUTS IN INTO INREL. +0 VERBREG, 1 NOUNREG, 2 XREG, 3 YREG, 4 ZREG.

GETINREL INDEX DSPCOUNT
 CAF INRELTAB
 TS INREL # (A TEMP. REG)
 TC Q

INRELTAB OCT 4 # R3D5 (DSPCOUNT = 0)
 OCT 4 # R3D4 =(1)
 OCT 4 # R3D3 =(2)
 OCT 4 # R3D2 =(3)
 OCT 4 # R3D1 =(4)
 OCT 3 # R2D5 =(5)
 OCT 3 # R2D4 =(6)
 OCT 3 # R2D3 =(7)
 OCT 3 # R2D2 =(8D)
 OCT 3 # R2D1 =(9D)
 OCT 2 # R1D5 =(10D)
 OCT 2 # R1D4 =(11D)
 OCT 2 # R1D3 =(12D)
 OCT 2 # R1D2 =(13D)
 OCT 2 # R1D1 =(14D)
 TC CCSHOLE # NO DISCOUNT NUMBER = 15D

```



July 12, 2016

1433

|            |       |           |                                           |        |
|------------|-------|-----------|-------------------------------------------|--------|
| # Page 319 | OCT   | 1         | # ND2                                     | =(16D) |
|            | OCT   | 1         | # ND1                                     | =(17D) |
|            | OCT   | 0         | # VD2                                     | =(18D) |
|            | OCT   | 0         | # VD1                                     | =(19D) |
| VERB       | CAF   | ZERO      |                                           |        |
|            | TS    | VERBREG   |                                           |        |
|            | CAF   | VD1       |                                           |        |
| NVCOM      | TS    | DSPCOUNT  |                                           |        |
|            | TC    | 2BLANK    |                                           |        |
|            | CAF   | ONE       |                                           |        |
|            | TS    | DECBRNCH  | # SET FOR DEC V/N CODE                    |        |
|            | CAF   | ZERO      |                                           |        |
|            | TS    | REQRET    | # SET FOR ENTPASO                         |        |
|            | CAF   | ENDINST   | # IF DSPALARM OCCURS BEFORE FIRST ENTPASO |        |
|            | TS    | ENTRET    | # OR NVSUB, ENTRET MUST ALREADY BE SET    |        |
|            |       |           | # TO TC ENDOFJOB                          |        |
|            | TC    | ENDOFJOB  |                                           |        |
| NOUN       | CAF   | ZERO      |                                           |        |
|            | TS    | NOUNREG   |                                           |        |
|            | CAF   | ND1       | # ND1, OCT 21 (DEC 17)                    |        |
|            | TC    | NVCOM     |                                           |        |
| NEGSGN     | TC    | SIGNTEST  |                                           |        |
|            | TC    | -ON       |                                           |        |
|            | CAF   | TWO       |                                           |        |
| BOTHSGN    | INDEX | INREL     | # SET DEC COMP BIT TO 1 (IN DECBRNCH)     |        |
|            | AD    | BIT7      | # BIT 5 FOR R1. BIT 4 FOR R2.             |        |
|            | ADS   | DECBRNCH  | # BIT 3 FOR R3.                           |        |
| FIXCLPAS   | CCS   | CLPASS    | # IF CLPASS IS + OR +0. MAKE IT +0.       |        |
|            | CAF   | ZERO      |                                           |        |
|            | TS    | CLPASS    |                                           |        |
|            | TC    | +1        |                                           |        |
|            | TC    | ENDOFJOB  |                                           |        |
| POSGN      | TC    | SIGNTEST  |                                           |        |
|            | TC    | +ON       |                                           |        |
|            | CAF   | ONE       |                                           |        |
|            | TC    | BOTHSGN   |                                           |        |
| +ON        | LXCH  | Q         |                                           |        |
|            | TC    | GETINREL  |                                           |        |
|            | INDEX | INREL     |                                           |        |
|            | CAF   | SGNTAB -2 |                                           |        |
|            | TS    | SGNOFF    |                                           |        |

|            |        |           |                                        |
|------------|--------|-----------|----------------------------------------|
|            | AD     | ONE       |                                        |
|            | TS     | SGNON     |                                        |
| SGNCOM     | CAF    | ZERO      |                                        |
|            | TS     | CODE      |                                        |
| # Page 320 |        |           |                                        |
|            | XCH    | SGNOFF    |                                        |
|            | TC     | 11DSPIN   |                                        |
|            | CAF    | BIT11     |                                        |
|            | TS     | CODE      |                                        |
|            | XCH    | SGNON     |                                        |
|            | TC     | 11DSPIN   |                                        |
|            | TC     | L         |                                        |
| -ON        | LXCH   | Q         |                                        |
|            | TC     | GETINREL  |                                        |
|            | INDEX  | INREL     |                                        |
|            | CAF    | SGNTAB -2 |                                        |
|            | TS     | SGNON     |                                        |
|            | AD     | ONE       |                                        |
|            | TS     | SGNOFF    |                                        |
|            | TC     | SGNCOM    |                                        |
| SGNTAB     | OCT    | 5         | # -R1                                  |
|            | OCT    | 3         | # -R2                                  |
|            | OCT    | 0         | # -R3                                  |
| SIGNTTEST  | LXCH   | Q         | # ALLOWS +,- ONLY WHEN DSPCOUNT=R1D1,  |
|            | CAF    | THREE     | # R2D1, OR R3D1. ALLOWS ONLY FIRST OF  |
|            | MASK   | DECBRNCH  | # CONSECUTIVE +/- CHARACTERS.          |
|            | CCS    | A         | # IF LOW2 BITS OF DECBRNCH NOT 0. SIGN |
|            | TC     | ENDOFJOB  | # FOR THIS WORD ALREADY IN. REJECT.    |
|            | CS     | R1D1      |                                        |
|            | TC     | SGNTST1   |                                        |
|            | CS     | R2D1      |                                        |
|            | TC     | SGNTST1   |                                        |
|            | CS     | R3D1      |                                        |
|            | TC     | SGNTST1   |                                        |
|            | TC     | ENDOFJOB  | # NO MATCH FOUND. SIGN ILLEGAL         |
| SGNTST1    | AD     | DSPCOUNT  |                                        |
|            | EXTEND |           |                                        |
|            | BZF    | +2        | # MATCH FOUND                          |
|            | TC     | Q         |                                        |
|            | TC     | L         | # SIGN LEGAL                           |

# CLEAR BLANKS WHICH R1, R2, R3 IS CURRENT OR LAST TO BE DISPLAYED (PERTINENT  
# XREG, YREG, ZREG IS CLEARED). SUCCESSIVE CLEARS TAKE CARE OF EACH RX  
# L/ RC UNTIL R1 IS DONE. THEN NO FURTHER ACTION.

```

#
THE SINGLE COMPONENT LOAD VERBS ALLOW ONLY THE SINGLE RC THAT IS
APPROPRIATE TO BE CLEARED.
#
CLPASS +0 PASSO, CAN BE BACKED UP
+NZ HIPASS, CAN BE BACKED UP
-NZ PASSO, CANNOT BE BACKED UP
Page 321
CLEAR CCS DSPCOUNT
 AD ONE
 TC +2
 AD ONE
 INDEX A # DO NOT CHANGE DSPCOUNT BECAUSE MAY LATER
 CAF INRELTAB # FAIL LEGALTST.
 TS INREL # MUST SET INREL, EVEN FOR HIPASS.
 CCS CLPASS
 TC CLPASHI # +
 TC +2 # +0 IF CLPASS IS +0 OR -, IT IS PASSO
 TC +1 # -
 CA INREL
 TC LEGALTST
 TC CLEAR1
CLPASHI CCS INREL
 TS INREL
 TC LEGALTST
 CAF DOUBLK +2 # +3 TO - NUMBER, BACKS DATA REQUESTS.
 ADS REQRET
 CA INREL
 TS MIXTEMP # TEMP STORAGE FOR INREL
EXTEND
DIM VERBREG # DECREMENT VERB AND RE-DISPLAY
TC BANKCALL
CADR UPDATVB
CA MIXTEMP
TS INREL # RESTORE INREL
CLEAR1 TC CLR5
INCR CLPASS # ONLY IF CLPASS IS + OR +0
TC ENDOFJOB # SET FOR HIGHER PASS.
CLR5 LXCH Q # USED 5BLANK BUT AVOIDS ITS TC GETINREL
TC 5BLANK +2
LEGALTST AD NEG2
CCS A
TC Q # LEGAL INREL G/2
TC CCSHOLE
TC ENDOFJOB # ILLEGAL INREL=0,1
TC Q # LEGAL INREL=2

```

# 5BLANK BLANKS 5 CHAR DISPLAY WORD IN R1, R2, OR R3. IT ALSO ZEROES XREG,  
 # YREG, OR ZREG. PLACE ANY + DSPCOUNT NUMBER FOR PERTINENT RC INTO DSPCOUNT.  
 # DSPCOUNT IS LEFT SET TO LEFT MOST DSP NUMB FOR RC JUST BLANKED.

|            |       |             |                                        |
|------------|-------|-------------|----------------------------------------|
|            | TS    | DSPCOUNT    | # NEEDED FOR BLANKSUB                  |
| 5BLANK     | LXCH  | Q           |                                        |
|            | TC    | GETINREL    |                                        |
|            | CAF   | ZERO        |                                        |
|            | INDEX | INREL       |                                        |
|            | TS    | VERBREG     | # ZERO X, Y, Z, REG.                   |
| # Page 322 |       |             |                                        |
|            | INDEX | INREL       |                                        |
|            | TS    | XREGLP -2   |                                        |
|            | TS    | CODE        |                                        |
|            | INDEX | INREL       | # ZERO PERTINENT DEC COMP BIT.         |
|            | CS    | BIT7        | # PROTECT OTHERS                       |
|            | MASK  | DECBRNCH    |                                        |
|            | MASK  | BRNCHCON    | # ZERO LOW 2 BITS.                     |
|            | TS    | DECBRNCH    |                                        |
|            | INDEX | INREL       |                                        |
|            | CAF   | SINBLANK -2 | # BLANK ISOLATED CHAR SEPARATELY       |
|            | TS    | COUNT       |                                        |
|            | TC    | DSPIN       |                                        |
| 5BLANK1    | INDEX | INREL       |                                        |
|            | CAF   | DOUBLK -2   |                                        |
|            | TS    | DSPCOUNT    |                                        |
|            | TC    | 2BLANK      |                                        |
|            | CS    | TWO         |                                        |
|            | ADS   | DSPCOUNT    |                                        |
|            | TC    | 2BLANK      |                                        |
|            | INDEX | INREL       |                                        |
|            | CAF   | R1D1 -2     |                                        |
|            | TS    | DSPCOUNT    | # SET DSPCOUNT TO LEFT MOST DSP NUMBER |
|            | TC    | L           | # OF REG. JUST BLANKED                 |
| SINBLANK   | OCT   | 16          | # DEC 14                               |
|            | OCT   | 5           |                                        |
|            | OCT   | 4           |                                        |
| DOUBLK     | OCT   | 15          | # DEC 13                               |
|            | OCT   | 11          | # DEC 9                                |
|            | OCT   | 3           |                                        |
| BRNCHCON   | OCT   | 77774       |                                        |

# 2BLANK BLANKS TWO CHAR. PLACE DSP NUMBER OF LEFT CHAR OF THE PAIR INTO

# DSPCOUNT. THIS NUMBER IS LEFT IN DSPCOUNT

|          |        |          |                                 |
|----------|--------|----------|---------------------------------|
| 2BLANK   | CA     | DSPCOUNT |                                 |
|          | TS     | SR       |                                 |
|          | CS     | BLANKCON |                                 |
|          | INHINT |          |                                 |
|          | INDEX  | SR       |                                 |
|          | XCH    | DSPTAB   |                                 |
|          | EXTEND |          |                                 |
|          | BZMF   | +2       | # IF OLD CONTENTS -, NOUT OK    |
|          | INCR   | NOUT     | # IF OLD CONTENTS +, +1 TO NOUT |
|          | RELINT |          | # IF -, NOUT OK                 |
|          | TC     | Q        |                                 |
| BLANKCON | OCT    | 4000     |                                 |

# Page 323

# ENTER PASS 0 IS THE EXECUTE FUNCTION. HIGHER ORDER ENTERS ARE TO LOAD  
# DATA. THE SIGN OF REQRET DETERMINES THE PASS, + FOR PASS 0, - FOR HIGHER  
# PASSES

#

# MACHINE CADR TO BE SPECIFIED (MCTBS) NOUNS DESIRE AN ECADR TO BE LOADED  
# WHEN USED WITH LOAD VERBS, MONITOR VERBS, OR DISPLAY VERBS (EXCEPT  
# VERB = FIXED MEMORY DISPLAY, WHICH REQUIRES AN FCADR).

|        |          |
|--------|----------|
| BANK   | 41       |
| SETLOC | PINBALL2 |
| BANK   |          |

|       |        |
|-------|--------|
| COUNT | 41/PIN |
|-------|--------|

|         |    |        |                                  |
|---------|----|--------|----------------------------------|
| NVSUBB  | TC | NVSUB1 | # STANDARD LEAD INS. DON'T MOVE. |
| LOADLV1 | TC | LOADLV |                                  |

# END OF STANDARD LEAD INS.

|          |        |         |                                      |
|----------|--------|---------|--------------------------------------|
| ENTER    | CAF    | ZERO    |                                      |
|          | TS     | CLPASS  |                                      |
|          | CAF    | ENDINST |                                      |
|          | TS     | ENTRET  |                                      |
|          | CCS    | REQRET  |                                      |
|          | TC     | ENTPASO | # IF +, PASS 0                       |
|          | TC     | ENTPASO | # IF +, PASS 0                       |
|          | TC     | +1      | # IF -, NOT PASS 0                   |
| ENTPASHI | CAF    | MMADREF |                                      |
|          | AD     | REQRET  | # IF L/ 2 CHAR IN FOR MM CODE, ALARM |
|          | EXTEND |         | # AND RECYCLE (DECIDE AT MCHANG+1).  |

|            |        |             |                                           |
|------------|--------|-------------|-------------------------------------------|
|            | BZF    | ACCEPTWD    |                                           |
|            | CAF    | THREE       | # IF DEC, ALARM IF L/ 5 CHAR IN FOR DATA, |
|            | MASK   | DECBRNCH    | # BUT LEAVE REQRET - AND FLASH ON, SO     |
|            | CCS    | A           | # OPERATOR CAN SUPPLY MISSING NUMERICAL   |
|            | TC     | +2          | # CHARACTERS AND CONTINUE.                |
|            | TC     | ACCEPTWD    | # OCTAL. ANY NUMBER OF CHAR OK.           |
|            | CCS    | DSPCOUNT    |                                           |
|            | TC     | GODSPALM    | # LESS THAN 5 CHAR DEC(DSPCOUNT IS +)     |
|            | TC     | GODSPALM    | # LESS THAN 5 CHAR DEC(DSPCOUNT IS +)     |
|            | TC     | +1          | # 5 CHAR IN (DSPCOUNT IS -)               |
| ACCEPTWD   | CS     | REQRET      | # 5 CHAR IN (DSPCOUNT IS -)               |
|            | TS     | REQRET      | # SET REQRET +.                           |
|            | TC     | FLASHOFF    |                                           |
|            | TC     | REQRET      |                                           |
| ENTEXIT    | =      | ENTRET      |                                           |
| MMADREF    | ADRES  | MMCHANG +1  | # ASSUMES TC REQMM AT MMCHANG.            |
| # Page 324 |        |             |                                           |
| LOWVERB    | DEC    | 28          | # LOWER VERB THAT AVOIDS NOUN TEST.       |
| ENTPASO    | CAF    | ZERO        | # NOUN VERB SUB ENTERS HERE               |
|            | TS     | DECBRNCH    |                                           |
|            | CS     | VD1         | # BLOCK FURTHER NUM CHAR, SO THAT STRAY   |
|            | TS     | DSPCOUNT    | # CHAR DO NOT GET INTO VERB OR NOUN LTS.  |
| TESTVB     | CS     | VERBREG     | # IF VERB IS G/E LOWVB, SKIP NOUN TEST.   |
|            | TS     | VERBSAVE    | # SAVE VERB FOR POSSIBLE RECYCLE.         |
|            | AD     | LOWVERB     | # LOWVERB - VB                            |
|            | EXTEND |             |                                           |
|            | BZMF   | VERBFAN     | # VERB G/ E LOWVERB                       |
| TESTNN     | EXTEND |             | # VERB L/ LOWVERB                         |
|            | DCA    | LODNNLOC    | # SWITCH BANKS TO NOUN TABLE READING      |
|            | DXCH   | Z           | # ROUTINE.                                |
|            | INDEX  | MIXBR       |                                           |
|            | TC     | +0          |                                           |
|            | TC     | +2          | # NORMAL                                  |
|            | TC     | MIXNOUN     | # MIXED                                   |
|            | CCS    | NNADTEM     | # NORMAL                                  |
|            | TC     | VERBFAN -2  | # NORMAL IF +                             |
|            | TC     | GODSPALM    | # NOT IN USE IF +0                        |
|            | TC     | REQADD      | # SPECIFY MACHINE CADR IF -               |
|            | INCR   | NOUNCADR    | # AUGMENT MACHINE CADR IF -0              |
|            | TC     | SETNADD     | # ECADR FROM NOUNCADR, SETS ED, NOUNADD.  |
|            | TC     | INTMCTBS +2 |                                           |
| REQADD     | CAF    | BIT15       | # SET CLPASS FOR PASS 0 ONLY              |

|            |        |          |                                           |
|------------|--------|----------|-------------------------------------------|
|            | TS     | CLPASS   |                                           |
|            | CS     | ENDINST  | # TEST IF REACHED HERE FROM INTERNAL OR   |
|            | AD     | ENTEXIT  | # FROM EXTERNAL                           |
|            | EXTEND |          |                                           |
|            | BZF    | +2       | # EXTERNAL MACH CADR TO BE SPECIFIED      |
|            | TC     | INTMCTBS |                                           |
|            | TC     | REQDATZ  | # EXTERNAL MACH CADR TO BE SPECIFIED      |
|            | CCS    | DECBRNCH | # ALARM AND RECYCLE IF DECIMAL USED       |
|            | TC     | ALMCYCLE | # FOR MCTBS.                              |
|            | CS     | VD1      | # OCTAL USED OK                           |
|            | TS     | DSPCOUNT | # BLOCK NUM CHAR IN                       |
|            | CCS    | CADRSTOR |                                           |
|            | TC     | +3       | # EXTERNAL MCTBS DISPLAY WILL LEAVE FLASH |
|            | TC     | USEADD   | # ON IF ENDIDLE NOT = +0.                 |
|            | TC     | +1       |                                           |
|            | TC     | FLASHON  |                                           |
| USEADD     | XCH    | ZREG     |                                           |
|            | TC     | SETNCADR | # ECADR INTO NOUNCADR. SET EB, NOUNADD.   |
|            | EXTEND |          |                                           |
|            | DCA    | LODNNLOC | # SWITCH BANKS TO NOUN TAB E READING      |
|            | DXCH   | Z        | # ROUTINE.                                |
|            | TC     | VERBFAN  |                                           |
|            | EBANK= | DSPCOUNT |                                           |
| # Page 325 |        |          |                                           |
| LODNNLOC   | 2CADR  | LODNNTAB |                                           |
| NEG5       | OCT    | 77772    |                                           |
| INTMCTBS   | CA     | MPAC +2  | # INTERNAL MACH CADR TO BE SPECIFIED.     |
|            | TC     | SETNCADR | # ECADR INTO NOUNCADR. SET EB, NOUNADD.   |
|            | CS     | FIVE     | # NVSUB CALL LEFT CADR IN MPAC+2 FOR MACH |
|            | AD     | VERBREG  | # CADR TO BE SPECIFIED.                   |
|            | EXTEND |          |                                           |
|            | BZF    | VERBFAN  | # DON'T DISPLAY CADR IF VB = 05.          |
|            | CAF    | R3D1     | # VB NOT = 05. DISPLAY CADR.              |
|            | TS     | DSPCOUNT |                                           |
|            | CA     | NOUNCADR |                                           |
|            | TC     | DSPOCTWO |                                           |
|            | TC     | VERBFAN  |                                           |
|            | AD     | ONE      |                                           |
|            | TC     | SETNCADR | # ECADR INTO NOUNCADR. SETS EB, NOUNADD.  |
| VERBFAN    | CS     | LST2CON  |                                           |
|            | AD     | VERBREG  | # VERB -- LST2CON                         |
|            | CCS    | A        |                                           |

|            |       |          |                                          |
|------------|-------|----------|------------------------------------------|
|            | AD    | ONE      | # VERB G/ LST2CON                        |
|            | TC    | +2       |                                          |
|            | TC    | VBFANDIR | # VERB L/ LST2CON                        |
|            | TS    | MPAC     |                                          |
|            | TC    | RELDSP   | # RELEASE DISPLAY SYST                   |
|            | TC    | POSTJUMP | # GO TO GOEXTVB WITH VB=40 IN MPAC.      |
|            | CADR  | GOEXTVB  |                                          |
| LST2CON    | DEC   | 40       | # FIRST LIST2 VERB (EXTENDED VERB)       |
| VBFANDIR   | INDEX | VERBREG  |                                          |
|            | CAF   | VERBTAB  |                                          |
|            | TC    | BANKJUMP |                                          |
| VERBTAB    | CADR  | GODSPALM | # VB00 ILLEGAL                           |
|            | CADR  | DSPA     | # VB01 DISPLAY OCT COMP 1 (R1)           |
|            | CADR  | DSPB     | # VB02 DISPLAY OCT COMP 2 (R1)           |
|            | CADR  | DSPC     | # VB03 DISPLAY OCT COMP 3 (R1)           |
|            | CADR  | DSPAB    | # VB04 DISPLAY OCT COMP 1,2 (R1,R2)      |
|            | CADR  | DSPABC   | # VB05 DISPLAY OCT COMP 1,2,3 (R1,R2,R3) |
|            | CADR  | DECDSP   | # VB06 DECIMAL DISPLAY                   |
|            | CADR  | DSPDPDEC | # VB07 DP DECIMAL DISPLAY (R1,R2)        |
|            | CADR  | GODSPALM | # VB08 SPARE                             |
|            | CADR  | GODSPALM | # VB09 SPARE                             |
|            | CADR  | DSPALARM | # VB10 SPARE                             |
|            | CADR  | MONITOR  | # VB11 MONITOR OCT COMP 1 (R1)           |
|            | CADR  | MONITOR  | # VB12 MONITOR OCT COMP 2 (R1)           |
|            | CADR  | MONITOR  | # VB13 MONITOR OCT COMP 3 (R1)           |
|            | CADR  | MONITOR  | # VB14 MONITOR OCT COMP 1,2 (R1,R2)      |
| # Page 326 |       |          |                                          |
|            | CADR  | MONITOR  | # VB15 MONITOR OCT COMP 1,2,3 (R1,R2,R3) |
|            | CADR  | MONITOR  | # VB16 MONITOR DECIMAL                   |
|            | CADR  | MONITOR  | # VB17 MONITOR DP DEC (R1,R2)            |
|            | CADR  | GODSPALM | # VB18 SPARE                             |
|            | CADR  | GODSPALM | # VB19 SPARE                             |
|            | CADR  | GODSPALM | # VB20 SPARE                             |
|            | CADR  | ALOAD    | # VB21 LOAD COMP 1 (R1)                  |
|            | CADR  | BLOAD    | # VB22 LOAD COMP 2 (R2)                  |
|            | CADR  | CLOAD    | # VB23 LOAD COMP 3 (R3)                  |
|            | CADR  | ABLOAD   | # VB24 LOAD COMP 1,2 (R1,R2)             |
|            | CADR  | ABCLOAD  | # VB25 LOAD COMP 1,2,3 (R1,R2,R3)        |
|            | CADR  | GODSPALM | # VB26 SPARE                             |
|            | CADR  | DSPFMEM  | # VB27 FIXED MEMORY DISPLAY              |
|            |       |          | # THE FOLLOWING VERBS MAKE NO NOUN TEST  |
|            | CADR  | GODSPALM | # VB28 SPARE                             |
|            | CADR  | GODSPALM | # VB29 SPARE                             |
| REQEXLQC   | CADR  | VBRQEXEC | # VB30 REQUEST EXECUTIVE                 |



|      |          |                                           |
|------|----------|-------------------------------------------|
| CADR | VRQWAIT  | # VB31 REQUEST WAITLIST                   |
| CADR | VBRESEQ  | # VB32 RESEQUENCE                         |
| CADR | VBPROC   | # VB33 PROCEED WITHOUT DATA               |
| CADR | VBTERM   | # VB34 TERMINATE CURRENT TEST OR LOAD REQ |
| CADR | VBSTLTS  | # VB35 TEST LIGHTS                        |
| CADR | SLAP1    | # VB36 FRESH START                        |
| CADR | MMCHANG  | # VB37 CHANGE MAJOR MODE                  |
| CADR | GODSPALM | # VB38 SPARE                              |
| CADR | GODSPALM | # VB39 SPARE                              |

# THE LIST2 VERBFAN IS LOCATED IN THE EXTENDED VERB BANK.

# Page 327

# NNADTAB CONTAINS A RELATIVE ADDRESS, IDADDREL (IN LOW 10 BITS), REFERRING

# TO WHERE 3 CONSECUTIVE ADDRESSES ARE STORED (IN IDADDTAB).

# MIXNOUN GETS DATA AND STORES IN MIXTEMP,+1,+2. IT SETS NOUNADD FOR

# MIXTEMP.

|         |        |          |                                             |
|---------|--------|----------|---------------------------------------------|
| MIXNOUN | CCS    | NNADTEM  |                                             |
|         | TC     | +4       | # + IN USE                                  |
|         | TC     | GODSPALM | # +0 NOT IN USE                             |
|         | TC     | +2       | # - IN USE                                  |
|         | TC     | +1       | # -0 IN USE                                 |
|         | CS     | SIX      |                                             |
|         | AD     | VERBREG  |                                             |
|         | EXTEND |          |                                             |
|         | BZMF   | +2       | # VERB L/E 6                                |
|         | TC     | VERBFAN  | # AVOID MIXNOUN SWAP IF VB NOT = DISPLAY    |
|         | CAF    | TWO      |                                             |
| MIXNN1  | TS     | DECOUNT  |                                             |
|         | AD     | MIXAD    |                                             |
|         | TS     | NOUNADD  | # SET NOUNADD TO MIXTEMP +K                 |
|         | INDEX  | DECOUNT  | # GET IDADDTAB ENTRY FOR COMPONENT K        |
|         | CA     | IDAD1TEM | # OF NOUN.                                  |
|         | TS     | NOUNTEM  |                                             |
|         |        |          | # TEST FOR DP (FOR OCT DISPLAY). IF SO, GET |
|         |        |          | # MINOR PART ONLY.                          |
|         | TC     | SFRUTMIX | # GET SF ROUT NUMBER IN A                   |
|         | TC     | DPTEST   |                                             |
|         | TC     | MIXNN2   | # NO DP                                     |
|         | INCR   | NOUNTEM  | # DP GET MINOR PART                         |
| MIXNN2  | CA     | NOUNTEM  |                                             |
|         | MASK   | LOW11    | # ESUBK (NO DP) OR (ESUBK)+1 FOR DP.        |
|         | TC     | SETEBANK | # SET EBANK, LEAVE EADRES IN A.             |
|         | INDEX  | A        | # PICK UP C(ESUBK) NOT DP.                  |
|         | CA     | 0        | # OR C((ESUBK)+1) FOR DP MINOR PART         |
|         | INDEX  | NOUNADD  |                                             |

|            |       |                                 |                                        |
|------------|-------|---------------------------------|----------------------------------------|
|            | XCH   | 0                               | # STORE IN MIXTEM + K                  |
|            | CCS   | DECOUNT                         |                                        |
|            | TC    | MIXNN1                          |                                        |
|            | TC    | VERBFAN                         |                                        |
| MIXAD      | TC    | MIXTEMP                         |                                        |
| # DPTEST   |       | ENTER WITH SF ROUT NUMBER IN A. |                                        |
| #          |       | RETURNS TO L+1 IF NO DP.        |                                        |
| #          |       | RETURNS TO L+2 IF DP.           |                                        |
| DPTEST     | INDEX | A                               |                                        |
|            | TCF   | +1                              |                                        |
|            | TC    | Q                               | # OCTAL ONLY NO DP                     |
|            | TC    | Q                               | # FRACT NO DP                          |
| # Page 328 |       |                                 |                                        |
|            | TC    | Q                               | # DEG NO DP                            |
|            | TC    | Q                               | # ARITH NO DP                          |
|            | TCF   | DPTEST1                         | # DP1OUT                               |
|            | TCF   | DPTEST1                         | # DP2OUT                               |
|            | TC    | Q                               | # OPDEG NO DP                          |
|            | TCF   | DPTEST1                         | # DP3OUT                               |
|            | TC    | Q                               | # HMS NO DP                            |
|            | TC    | Q                               | # M/S NO DP                            |
|            | TCF   | DPTEST1                         | # DP4OUT                               |
|            | TC    | Q                               | # ARITH1 NO DP                         |
|            | TC    | Q                               | # 2INTOUT NO DP TO GET HI PART IN MPAC |
|            | TCF   | DPTEST1                         | # DPFRACOT                             |
| DPTEST1    | INDEX | Q                               |                                        |
|            | TC    | 1                               | # RETURN TO L+2                        |
| REQDATX    | CAF   | R1D1                            |                                        |
|            | TCF   | REQCOM                          |                                        |
| REQDATY    | CAF   | R2D1                            |                                        |
|            | TCF   | REQCOM                          |                                        |
| REQDATZ    | CAF   | R3D1                            |                                        |
| REQCOM     | TS    | DSPCOUNT                        |                                        |
|            | CS    | Q                               |                                        |
|            | TS    | REQRET                          |                                        |
|            | TC    | BANKCALL                        |                                        |
|            | CADR  | 5BLANK                          |                                        |
|            | TC    | FLASHON                         |                                        |
| ENDRQDAT   | TC    | ENTEXIT                         |                                        |
|            | TS    | NOUNREG                         |                                        |
| UPDATNN    | XCH   | Q                               |                                        |

|            |        |           |                                             |
|------------|--------|-----------|---------------------------------------------|
|            | TS     | UPDATRET  |                                             |
|            | EXTEND |           |                                             |
|            | DCA    | LODNNLOC  | # SWITCH BANKS TO NOUN TABLE READING        |
|            | DXCH   | Z         | # ROUTINE.                                  |
|            | CCS    | NNADTEM   |                                             |
|            | AD     | ONE       | # NORMAL                                    |
|            | TCF    | PUTADD    |                                             |
|            | TCF    | PUTADD +1 | # MCTBS DON'T CHANGE NOUNADD                |
|            | TCF    | PUTADD +1 | # MCTBI DON'T CHANGE NOUNADD                |
| PUTADD     | TC     | SETNCADR  | # ECADR INTO NOUNCADR. SETS EB, NOUNADD.    |
|            | CAF    | ND1       |                                             |
|            | TS     | DSPCOUNT  |                                             |
|            | CA     | NOUNREG   |                                             |
|            | TCF    | UPDAT1    |                                             |
|            |        |           |                                             |
|            | TS     | VERBREG   |                                             |
| UPDATVB    | XCH    | Q         |                                             |
|            | TS     | UPDATRET  |                                             |
|            | CAF    | VD1       |                                             |
|            |        |           |                                             |
| # Page 329 |        |           |                                             |
|            | TS     | DSPCOUNT  |                                             |
|            | CA     | VERBREG   |                                             |
| UPDAT1     | TC     | POSTJUMP  | # CAN'T USE SWCALL TO GO TC DSPDECVN, SINCE |
|            | CADR   | GOVNUPDT  | # UPDATVB CAN ITSELF BE CALLED BY SWCALL.   |
|            | TC     | UPDATRET  |                                             |
|            |        |           |                                             |
| GOALMCYC   | TC     | ALMCYCLE  | # NEEDED BECAUSE BANKJUMP CAN'T HANDLE F/F. |
|            |        |           |                                             |
| GODSPALM   | TC     | POSTJUMP  |                                             |
|            | CADR   | DSPALARM  |                                             |

# Page 330

# NOUN TABLES

#

# NOUN CODE L/40, NORMAL NOUN CASE. NOUN CODE G/E 40, MIXED NOUN CASE.

# FOR NORMAL CASE, NNADTAB CONTAINS ONE ECADR FOR EACH NOUN.

# +0 INDICATES NOUN NOT USED. - ENTRY INDICATES MACHINE CADR (E OR F) TO BE SPECIFIED. -1 INDICATES CHANNEL TO BE SPECIFIED. -0 INDICATES AUGMENT OF LAST MACHINE CADR SUPPLIED.

#

# FOR MIXED CASE, NNADTAB CONTAINS ONE INDIRECT ADDRESS (IDADDREL) IN LOW 10 BITS, AND THE COMPONENT CODE NUMBER IN THE HIGH 5 BITS.

#

# NNTYPTAB IS A PACKED TABLE OF THE FORM MMMMMNNNNPPPPPP.

#

# FOR THE NORMAL CASE, M'S ARE THE COMPONENT CODE NUMBER.

```

N'S ARE THE SF ROUTINE CODE NUMBER.
P'S ARE THE SF CONSTANT CODE NUMBER.
#
MIXED-CASE, M'S ARE THE SF CONSTANT3 CODE NUMBER 3 COMPONENT CASE
N'S ARE THE SF CONSTANT2 CODE NUMBER
P'S ARE THE SF CONSTANT1 CODE NUMBER
N'S ARE THE SF CONSTANT2 CODE NUMBER 2 COMPONENT CASE
P'S ARE THE SF CONSTANT1 CODE NUMBER
P'S ARE THE SF CONSTANT1 CODE NUMBER 1 COMPONENT CASE
#
THERE IS ALSO AN INDIRECT ADDRESS TABLE (IDADDTAB) FOR MIXED CASE ONLY
EACH ENTRY CONTAINS ONE ECADR. IDADDREL IS THE RELATIVE ADDRESS OF
THE FIRST OF THESE ENTRIES.
#
THERE IS ONE ENTRY IN THIS TABEL FOR EACH COMPONENT OF A MIXED NOUN
THEY ARE LISTED IN ORDER OF ASCENDING K.
#
THERE IS ALSO A SCALE FACTOR ROUTINE NUMBER TABLE (RUTMXTAB) FOR MIXED
CASE ONLY. THERE IS ONE ENTRY PER MIXED NOUN. THE FORM IS,
#
QQQQRRRRRSSSSS
#
Q'S ARE THE SF ROUTINE 3 CODE NUMBER 3 COMPONENT CASE
R'S ARE THE SF ROUTINE 2 CODE NUMBER
S'S ARE THE SF ROUTINE 1 CODE NUMBER
R'S ARE THE SF ROUTINE 2 CODE NUMBER 2 COMPONENT CASE
S'S ARE THE SF ROUTINE 1 CODE NUMBER
#
IN OCTAL DISPLAY AND LOAD (OCT OR DEC) VERBS, EXCLUDE USE OF VERBS WHOSE
COMPONENT NUMBER IS GREATER THAN THE NUMBER OF COMONENTS IN NOUN.
(ALL MACHINE ADDRESS TO BE SPECIFIED NOUNS ARE 3 COMPONENT.)
#
IN MULTI-COMPONENT LOAD VERBS, NO MIXING OF OCTAL AND DECIMAL DATA
COMPONENT WORDS IS ALLOWED. ALARM IF VIOLATION.
#
IN DECIMAL LOADS OF DATA, 5 NUMERICAL CHARACTERS MUST BE KEYED IN
BEFORE EACH ENTER. IF NOT, ALARM.

Page 331
DISPLAY VERBS

```

```

DSPABC CS TWO
 TC COMPTST
 INDEX NOUNADD
 CS 2
 XCH BUF +2

```

|         |       |          |                                 |
|---------|-------|----------|---------------------------------|
| DSPAB   | CS    | ONE      |                                 |
|         | TC    | COMPTST  |                                 |
|         | INDEX | NOUNADD  |                                 |
|         | CS    | 1        |                                 |
|         | XCH   | BUF      | +1                              |
| DSPA    | TC    | DECTEST  |                                 |
|         | TC    | TSTFORDP |                                 |
|         | INDEX | NOUNADD  |                                 |
|         | CS    | 0        |                                 |
| DSPCOM1 | XCH   | BUF      |                                 |
|         | TC    | DSPCOM2  |                                 |
| DSPB    | CS    | ONE      |                                 |
|         | TC    | DCOMPTST |                                 |
|         | INDEX | NOUNADD  |                                 |
|         | CS    | 1        |                                 |
|         | TC    | DSPCOM1  |                                 |
| DSPC    | CS    | TWO      |                                 |
|         | TC    | DCOMPTST |                                 |
|         | INDEX | NOUNADD  |                                 |
|         | CS    | 2        |                                 |
|         | TC    | DSPCOM1  |                                 |
| DSPCOM2 | CS    | TWO      | # A B C AB ABC                  |
|         | AD    | VERBREG  | # -1 -0 +1 +2 +3 IN A           |
|         | CCS   | A        | # +0 +0 +0 +1 +2 IN A AFTER CCS |
|         | TC    | DSPCOM3  |                                 |
|         | TC    | ENTEXIT  |                                 |
|         | TC    | +1       |                                 |
| DSPCOM3 | TS    | DISTEM   | # +0 +1 +2 INTO DISTEM          |
|         | INDEX | A        |                                 |
|         | CAF   | R1D1     |                                 |
|         | TS    | DSPCOUNT |                                 |
|         | INDEX | DISTEM   |                                 |
|         | CS    | BUF      |                                 |
|         | TC    | DSPCOM2  |                                 |
|         | XCH   | DISTEM   |                                 |
|         | TC    | DSPCOM2  | +2                              |

# COMPTST ALARMS IF COMPONENT NUMBER OF VERB (LOAD OR OCT DISPLAY) IS  
# GREATER THAN THE HIGHEST COMPONENT NUMBER OF NOUN.

|          |      |         |             |
|----------|------|---------|-------------|
| COMPTST  | TS   | SFTEMP1 | # VERB COMP |
|          | LXCH | Q       |             |
| COMPTST1 | TC   | GETCOMP |             |
|          | TC   | LEFT5   |             |
|          | MASK | THREE   | # NOUN COMP |

|          |     |          |                          |
|----------|-----|----------|--------------------------|
|          | AD  | SFTEMP1  | # NOUN COMP -- VERB COMP |
|          | CCS | A        |                          |
|          | TC  | L        | # NOUN COMP G/ VERB COMP |
|          | TC  | CCSHOLE  |                          |
|          | TC  | GODSPALM | # NOUN COMP L/ VERB COMP |
| NDCMPTST | TC  | L        | # NOUN COMP = VERB COMP  |

# DCOMPTST ALARMS IF DECIMAL ONLY BIT (BIT4 OF COMP CODE NUMBER) = 1.  
 # IF NOT, IT PERFORMS REGULAR COMPTST.

|          |      |          |               |
|----------|------|----------|---------------|
| DCOMPTST | TS   | SFTEMP1  | # - VERB COMP |
|          | LXCH | Q        |               |
|          | TC   | DECTEST  |               |
|          | TC   | COMPTST1 |               |

|         |        |          |                                            |
|---------|--------|----------|--------------------------------------------|
| DECTEST | EXTEND |          | # ALARMS IF DEC ONLY BIT = 1 (BIT4 OF COMP |
|         | QXCH   | MPAC +2  | # CODE NUMBER). RETURNS IF NOT.            |
|         | TC     | GETCOMP  |                                            |
|         | MASK   | BIT14    |                                            |
|         | CCS    | A        |                                            |
|         | TC     | GODSPALM |                                            |
|         | TC     | MPAC +2  |                                            |

|          |      |          |                                           |
|----------|------|----------|-------------------------------------------|
| DCTSTCYC | LXCH | Q        | # ALARMS AND RECYCLES IF DEC ONLY BIT = 1 |
|          | TC   | GETCOMP  | # (BIT4 OF COMP CODE NUMBER). RETURNS     |
|          | MASK | BIT14    | # IF NOT. USED BY LOAD VERBS.             |
|          | CCS  | A        |                                           |
|          | TC   | ALMCYCLE |                                           |
|          | TC   | L        |                                           |

# NOUNTEST ALARMS IF NO-LOAD BIT (BIT5 OF COMP CODE NUMBER) = 1.  
 # IF NOT, IT RETURNS.

|          |      |          |
|----------|------|----------|
| NOUNTEST | LXCH | Q        |
|          | TC   | GETCOMP  |
|          | CCS  | A        |
|          | TC   | L        |
|          | TC   | L        |
|          | TC   | GODSPALM |

|          |        |         |                                            |
|----------|--------|---------|--------------------------------------------|
| TSTFORDP | LXCH   | Q       | # TEST FOR DP. IF SO, GET MINOR PART ONLY. |
|          | CA     | NNADTEM |                                            |
|          | AD     | ONE     | # IF NNADTEM = -1, CHANNEL TO BE SPECIFIED |
|          | EXTEND |         |                                            |
|          | BZF    | CHANDSP |                                            |
|          | INDEX  | MIXBR   |                                            |

July 12, 2016

1447

```
Page 333
TC +0
TC +2 # NORMAL

TC L # MIXED CASE ALREADY HANDLED IN MIXNOUN
TC SFRUTNOR
TC DPTEST
TC L # NO DP
INCR NOUNADD # DP E+1 INTO NOUNADD FOR MINOR PART.
TC L

CHANDSP CA NOUNCADR
 MASK LOW9
 EXTEND
 INDEX A
 READ 0
 CS A
 TCF DSPCOM1

COMPICK ADRES NNTYPTM
 ADRES NNADTEM

GETCOMP INDEX MIXBR # NORMAL MIXED
 CAF COMPICK -1 # ADRES NNTYPTM ADRES NNADTEM
 INDEX A
 CA 0 # C(NNTYPTM) C(NNADTEM)
 MASK HI5 # GET HI5 OF NNTYPTAB (NORM) OF NNADTAB (MIX
 TC Q

DECDSP TC GETCOMP
 TC LEFT5
 MASK THREE
 TS DECOUNT # COMP NUMBER INTO DECOUNT
 TS DECTEM # PICKS UP DATA
 AD NOUNADD # DECTEM 1COMP +0, 2COMP +1, 3COMP +2
 INDEX A
 CS 0
 INDEX DECTEM
 XCH XREG # CAN'T USE BUF SINCE DMP USES IT.
 CCS DECTEM
 TC DSPDCGET # MORE TO GET
 CAF ZERO # DISPLAYS DATA
 TS MPAC +1 # DECOUNT 1COMP +0, 2COMP +1, 3COMP +2
 TS MPAC +2
 INDEX DECOUNT
 CAF R1D1
 TS DSPCOUNT
```

```

Page 334
INDEX DECOUNT
CS XREG
TS MPAC
TC SFCONUM # 2X (SF CON NUMB) IN A

TS SFTEMP1
EXTEND
DCA GTSFOUTL # SWITCH BANKS TO SF CONSTANT TABLE
DXCH Z # READING ROUTINE.
INDEX MIXBR
TC +0
TC DSPSFNOR
TC SFRUTMIX
TC DECDSP3

DSPSFNOR TC SFRUTNOR
TC DECDSP3

EBANK= DSPCOUNT
GTSFOUTL 2CADR GTSFOUT

DSPDCEND TC BANKCALL # ALL SFOUT ROUTINES END HERE
CADR DSPDECWD
CCS DECOUNT
TC +2
TC ENTEXIT
TS DECOUNT
TC DSPDCPUT # MORE TO DISPLAY

DECDSP3 INDEX A
CAF SFOUTABR
TC BANKJUMP

SFOUTABR CADR PREDSPAL # ALARM IF DEC DISP WITH OCTAL ONLY NOUN
CADR DSPDCEND
CADR DEGOUTSF
CADR ARTOUTSF
CADR DP1OUTSF
CADR DP2OUTSF
CADR OPDEGOUT
CADR DP3OUTSF
CADR HMSOUT
CADR M/SOUT
CADR DP2OUTSF
CADR AROUT1SF
CADR 2INTOUT

```



CADR DPFACOT  
ENDRTOUT EQUALS

# THE FOLLOWING IS ATYPICAL SF ROUTINE. IT USES MPAC. LEAVES RESULTS  
# IN MPAC, MPAC+1. ENDS WITH TC DSPDCEND

# Page 335

SETLOC BLANKCON +1

COUNT 40/PIN

# DEGOUTSF SCALES BY .18 THE LOW 14 BITS OF ANGLE, ADDING .18 FOR  
# NUMBERS IN THE NEGATIVE (AGC) RANGE.

|          |     |          |                                           |
|----------|-----|----------|-------------------------------------------|
| DEGOUTSF | CAF | ZERO     |                                           |
|          | TS  | MPAC +2  | # SET INDEX FOR FULL SCALE.               |
|          | TC  | FIXRANGE |                                           |
|          | TC  | +2       | # NO AUGMENT NEEDED (SFTEMP1 AND 2 ARE 0) |
|          | TC  | SETAUG   | # SET AUGMENTER ACCORDING TO C(MPAC +2)   |
|          | TC  | DEGCOM   |                                           |

# OPDEGOUT SCALES BY .45 (THE RANGE IS 90 DEGREES) AND ADDS A 20 DEG BIAS.

|          |        |             |                                           |
|----------|--------|-------------|-------------------------------------------|
| OPDEGOUT | CCS    | MPAC        | # RANGE IS 90 DEG                         |
|          | XCH    | MPAC        | # IF POS OR POS 0 THEN ADD BIAS AND       |
|          | TC     | +3          | # CORRECT FOR POSSIBLE OVERFLOW           |
|          | TC     | NEGOPT      | # IF NEG NON ZERO                         |
|          | AD     | NEG1        | # IF NEG ZERO SUBTRACT 1                  |
|          | AD     | 20BIAS      |                                           |
| BIASCOM  | TS     | MPAC        | # TEST FOR OVEFLOW                        |
|          | TC     | +3          | # NO OVFLOW                               |
|          | CAF    | BIT15       | # IF OVFLOW                               |
|          | ADS    | MPAC        |                                           |
|          | CAF    | TWO         | # SET MULTIPLIER TO .45                   |
|          | TC     | DEGOUTSF +1 |                                           |
| NEGOPT   | XCH    | MPAC        | # NEGATIVE CASE                           |
|          | AD     | 20BIAS      |                                           |
|          | CCS    | A           |                                           |
|          | TC     | BIASCOM     | # IF POS THEN SUBTRACT 1 BECASUE OF 2SCOM |
|          | TC     | CCSHOLE     |                                           |
|          | AD     | ONE         | # IF NEG RESTORE SUM                      |
|          | COM    |             | # IF NEG 0 LEAVE NEG 0                    |
|          | TC     | BIASCOM     |                                           |
| SETAUG   | EXTEND |             | # LOADS SFTEMP1 AND SFTEMP2 WITH THE      |

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | INDEX  | MPAC +2  | # DP AUGMENTER CONSTANT                    |
|            | DCA    | DEGTAB   |                                            |
|            | DXCH   | SFTEMP1  |                                            |
|            | TC     | Q        |                                            |
| FIXRANGE   | CCS    | MPAC     | # IF MPAC IS + RETURN TO L+1               |
|            | TC     | Q        | # IF MPAC IS - RETURN TO L+2 AFTER         |
|            | TC     | Q        | # MASKING OUT THE SIGN BIT                 |
|            | TCF    | +1       |                                            |
|            | CS     | BIT15    |                                            |
|            | MASK   | MPAC     |                                            |
| # Page 336 | TS     | MPAC     |                                            |
|            | INDEX  | Q        |                                            |
|            | TC     | 1        |                                            |
| DEGCOM     | EXTEND |          | # LOADS MULTIPLIER, DOES SHORTMP, AND      |
|            | INDEX  | MPAC +2  | # ADDS AUTMENTER.                          |
|            | DCA    | DEGTAB   |                                            |
|            | DXCH   | MPAC     | # ADJUSTED ANGLE IN A                      |
|            | TC     | SHORTMP  |                                            |
|            | DXCH   | SFTEMP1  |                                            |
|            | DAS    | MPAC     |                                            |
|            | TC     | SCOUTEND |                                            |
| DEGTAB     | OCT    | 05605    | # HI PART OF .18                           |
|            | OCT    | 03656    | # LOW PART OF .18                          |
|            | OCT    | 16314    | # HI PART OF .45                           |
|            | OCT    | 31463    | # LO PART OF .45                           |
| 20BIAS     | OCT    | 16040    | # 20 DEG BIAS FOR OPTICS                   |
| ARTOUTSF   | DXCH   | SFTEMP1  | # ASSUMES POINT AT LEFT OF DP SFCON        |
|            | DXCH   | MPAC     |                                            |
|            | TC     | PRSHRTMP | # IF C(A) = -0, SHORTMP FAILS TO GIVE -0.  |
| SCOUTEND   | TC     | POSTJUMP |                                            |
|            | CADR   | DSPDCEND |                                            |
| AROUT1SF   | DXCH   | SFTEMP1  | # ASSUMES POINT BETWEEN HI AND LO PARTS OF |
|            | DXCH   | MPAC     | # DP SFCON. SHIFTS RESULTS LEFT 14, BY     |
|            | TC     | PRSHRTMP | # TAKING RESULTS FROM MPAC+1, MPAC+2.      |
|            | TC     | L14/OUT  |                                            |
| DP10OUTSF  | TC     | DPOUT    | # SCALES MPAC, MPAC +1 BY DP SCALE FACTOR  |
| L14/OUT    | XCH    | MPAC +2  | # IN SFTEMP1, SFTEMP2. THEN SCALE RESULT   |
|            | XCH    | MPAC +1  | # BY B14                                   |

July 12, 2016

1451

|          |     |          |                                           |
|----------|-----|----------|-------------------------------------------|
|          | TS  | MPAC     |                                           |
|          | TC  | SCOUTEND |                                           |
| DP2OUTSF | TC  | DPOUT    | # SCALES MPAC, MPAC +1 BY DP SCALE FACTOR |
|          | TC  | SCOUTEND |                                           |
| DP3OUTSF | TC  | DPOUT    | # ASSUMES POINT BETWEEN BITS 7-8 OF HIGH  |
|          | CAF | SIX      | # LEFT BY 7, ROUNDS MPAC+2 INTO MPAC+1    |
|          | TC  | TPLEFTN  | # SHIFT LEFT 7.                           |
|          | TC  | SCOUTEND |                                           |

# Page 337

|        |   |         |                                |
|--------|---|---------|--------------------------------|
| MPAC+6 | = | MPAC +6 | # USE MPAC +6 INSTEAD OF OVFIN |
|--------|---|---------|--------------------------------|

|       |       |         |                                      |
|-------|-------|---------|--------------------------------------|
| DPOUT | XCH   | Q       |                                      |
|       | TS    | MPAC+6  |                                      |
|       | TC    | READLO  | # GET FRESH DATA FOR BOTH HI AND LO. |
|       | TC    | TPAGREE | # MAKE DP DATA AGREE                 |
|       | TC    | DMP     |                                      |
|       | ADRES | SFTEMP1 |                                      |
|       | TC    | MPAC+6  |                                      |

# THE FOLLOWING ROUTINE DISPLAYS TWO CONTIGUOUS SP POSITIVE INTEGERS  
# AS TWO POSITIVE DECIMAL INTEGERS IN RXD1-RXD2 AND RXD4-RXD5 (RXD3 IS  
# BLANKED). THE INTEGER IN THE LOWER NUMBERED ADDRESS IS DISPLAYED IN  
# RXD1-RXD2.

|          |       |             |                                            |
|----------|-------|-------------|--------------------------------------------|
| 2INTOUT  | TC    | 5BLANK      | # TO BLANK RXD3                            |
|          | TC    | +ON         | # TURN ON + SIGN                           |
|          | CA    | MPAC        |                                            |
|          | TC    | DSPDECVN    | # DISPLAY 1ST INTEGER (LIKE VERB AND NOUN) |
|          | CS    | THREE       |                                            |
|          | INDEX | DECOUNT     |                                            |
|          | AD    | R1D1        | # RXD4                                     |
|          | TS    | DSPCOUNT    |                                            |
|          | TC    | READLO      | # GET 2ND INTEGER                          |
|          | CA    | MPAC +1     |                                            |
|          | TC    | DSPDECVN    | # DISPLAY 2ND INTEGER (LIKE VERB AND NOUN) |
|          | TC    | POSTJUMP    |                                            |
|          | CADR  | DSPDCEND +2 |                                            |
| DPFRACOT | TC    | READLO      | # DP FRACTION TO MPAC,+1                   |
|          | TC    | SCOUTEND    |                                            |

# READLO PICKS UP FRESHDATA FOR BOTH HI AND LO AND LEAVES IT IN

# MPAC, MPAC+1. THIS IS NEEDED FOR TIME DISPLAY. IT ZEROES MPAC+2, BUT  
# DOES NOT FORCE TPAGREE.

|            |        |            |                                             |
|------------|--------|------------|---------------------------------------------|
| READLO     | XCH    | Q          |                                             |
|            | TS     | TEM4       |                                             |
|            | INDEX  | MIXBR      |                                             |
|            | TC     | +0         |                                             |
|            | TC     | RDLONOR    |                                             |
|            | INDEX  | DECOUNT    |                                             |
|            | CA     | IDAD1TEM   | # GET IDADDTAB ENTRY FOR COMP K OF NOUN.    |
|            | MASK   | LOW11      | # E SUBK                                    |
|            | TC     | SETEBANK   | # SET EB, LEAVE EADRES IN A                 |
| READLO1    | EXTEND |            | # MIXED NORMAL                              |
|            | INDEX  | A          | # C(ESUBK) C(E)                             |
|            | DCA    | 0          | # C(E SUBK)+1) C(E+1)                       |
|            | DXCH   | MPAC       |                                             |
| # Page 338 |        |            |                                             |
|            | CAF    | ZERO       |                                             |
|            | TS     | MPAC       | +2                                          |
|            | TC     | TEM4       |                                             |
| RDLONOR    | CA     | NOUNADD    | # E                                         |
| ENDRDLO    | TC     | READLO1    |                                             |
|            | BANK   | 42         |                                             |
|            | SETLOC | PINBALL3   |                                             |
|            | BANK   |            |                                             |
|            | COUNT  | 42/PIN     |                                             |
| HMSOUT     | TC     | BANKCALL   | # READ FRESH DATA FOR HI AND LO INTO MPAC.  |
|            | CADR   | READLO     | # MPAC+1.                                   |
|            | TC     | TPAGREE    | # MAKE DP DATA AGREE.                       |
|            | TC     | SEPSECNR   | # LEAVE FRACT SEC/60 IN MPAC, MPAC+1. LEAVE |
|            |        |            | # WHOLE MIN IN BIT13 OF LOTEMOUT AND ABOVE  |
|            | TC     | DMP        | # USE ONLY FRACT SEC/60 MOD 60              |
|            | ADRES  | SECON2     | # MULT BY .06                               |
|            | CAF    | R3D1       | # GIVES CENTI-SEC/10EXP5 MOD 60             |
|            | TS     | DSPCOUNT   |                                             |
|            | TC     | BANKCALL   | # DISPLAY SEC MOD 60                        |
|            | CADR   | DSPDECWD   |                                             |
|            | TC     | SEPMIN     | # REMOVE REST OF SECONDS                    |
|            | CAF    | MINCON2    | # LEAVE FRACT MIN/60 IN MPAC+1. LEAVE       |
|            | XCH    | MPAC       | # WHOLE HOURS IN MPAC.                      |
|            | TS     | HITEMOUT   | # SAVE WHOLE HOURS.                         |
|            | CAF    | MINCON2 +1 |                                             |

July 12, 2016

1453

|            |        |             |          |                                            |
|------------|--------|-------------|----------|--------------------------------------------|
|            | XCH    | MPAC        | +1       | # USE ONLY FRACT MIN/60 MOD 60             |
|            | TC     | PRSHRTMP    |          | # IF C(A) = -0, SHORTMP FAILS TO GIVE -0.  |
|            |        |             |          | # MULT BY .0006                            |
|            | CAF    | R2D1        |          | # GIVE MIN/10EXP5 MOD 60                   |
|            | TS     | DSPCOUNT    |          |                                            |
|            | TC     | BANKCALL    |          | # DISPLAY MIN MOD 60                       |
|            | CADR   | DSPDECWD    |          |                                            |
|            | EXTEND |             |          | # MINUTES, SECONDS HAVE BEEN REMOVED       |
|            | DCA    | HRCON1      |          |                                            |
|            | DXCH   | MPAC        |          |                                            |
|            | CA     | HITEMOUT    |          | # USE WHOLE HOURS                          |
|            | TC     | PRSHRTMP    |          | # IF C(A) = -0, SHORTMP FAILS TO GIVE -0.  |
|            |        |             |          | # MULT BY .16384                           |
|            | CAF    | R1D1        |          | # GIVES HOURS/10EXP5                       |
|            | TS     | DSPCOUNT    |          |                                            |
|            | TC     | BANKCALL    |          | # USE REGULAR DSPDECWD WITH ROUND OFF.     |
|            | CADR   | DSPDECWD    |          |                                            |
|            | TC     | ENTEXIT     |          |                                            |
| SECON1     | 2DEC*  | 1.666666666 | E-4 B12* | # 2EXP12/6000                              |
| # Page 339 |        |             |          |                                            |
| SECON2     | OCT    | 01727       |          | # .06 FOR SECONDS DISPLAY                  |
|            | OCT    | 01217       |          |                                            |
| MINCON2    | OCT    | 00011       |          | # .0006 FOR MINUTES DISPLAY                |
|            | OCT    | 32445       |          |                                            |
| MINCON1    | OCT    | 02104       |          | # .06..66 UPPED BY 2EXP-2B                 |
|            | OCT    | 10422       |          |                                            |
| HRCON1     | 2DEC   | .16384      |          |                                            |
|            | OCT    | 00000       |          |                                            |
| RNDCON     | OCT    | 00062       |          | # .5 SEC                                   |
| M/SOUT     | TC     | BANKCALL    |          | # READ FRESH DATA FOR HI AND LO INTO MPAC. |
|            | CADR   | READLO      |          | # MPAC+1.                                  |
|            | TC     | TPAGREE     |          | # MAKE DP DATA AGREE                       |
|            | CCS    | MPAC        |          | # IF MAG OF (MPAC, MPAC+1) G/ 59 M 59 S.   |
|            | TC     | +2          |          | # DISPLAY 59B59, WITH PROPER SIGN.         |
|            | TC     | M/SNORM     |          | # MPAC = +0. L/ 59M58.5S                   |
|            | AD     | M/SCON1     |          | # - HI PART OF (59M58.5) +1 FOR CCS        |
|            | CCS    | A           |          | # MAG OF MPAC - HI PART OF (59M58.5S)      |
|            | TC     | M/SLIMIT    |          | # G/ 59M58.5S                              |
|            | TC     | M/SNORM     |          | # ORIGINAL MPAC = -0. L/59M58.5S           |
|            | TC     | M/SNORM     |          | # L/ 59M58.5S                              |
|            | CCS    | MPAC +1     |          | # MAG OF MPAC = HI PART OF 59M58.5S        |
|            | TC     | +2          |          |                                            |
|            | TC     | M/SNORM     |          | # MPAC+1 = +0. L/ 59M58.5S                 |
|            | AD     | M/SCON2     |          | # - LO PART OF (59M58.5S) +1 FOR CCS       |

|            |        |             |                                            |
|------------|--------|-------------|--------------------------------------------|
|            | CCS    | A           | # MAG OF MPAC+1 - LO PART OF (59M58.5S)    |
|            | TC     | M/SLIMIT    | # G/ 59M58.5S                              |
|            | TC     | M/SNORM     | # ORIGINAL MPAC+1 = -0. L/ 49M58.5S        |
|            | TC     | M/SNORM     | # L/ 59M58.5S                              |
| M/SLIMIT   | CCS    | MPAC        | # = 59M58.5S LIMIT                         |
|            | CAF    | M/SCON3     | # MPAC CANNOT BE +/- 0 AT THIS POINT.      |
|            | TC     | +LIMIT      | # FORCE MPAC, MPAC+1 TO +/- 59M58.5S       |
|            | CS     | M/SCON3     |                                            |
|            | TS     | MPAC        | # WILL DISPLAY 59M59S IN DSPDECNR          |
| LIMITCOM   | CS     | M/SCON3 +1  |                                            |
|            | TS     | MPAC +1     |                                            |
|            | CAF    | NORMADR     | # SET RETURN TO M/SNORM+1.                 |
|            | TC     | SEPSECNR +1 |                                            |
| +LIMIT     | TS     | MPAC        |                                            |
|            | CAF    | M/SCON3 +1  |                                            |
|            | TC     | LIMITCOM    |                                            |
| M/SNORM    | TC     | SEPSEC      | # LEAVE FRACT SEC/60 IN MPAC,MPAC+1. LEAVE |
|            |        |             | # WHOLE MIN IN BIT13 OF LOTEMOUT AND ABOVE |
|            | CAF    | HISECON     | # USE ONLY FRACT SEC/60 MOD 60             |
|            | TC     | SHORTMP     | # MULT BY .6 + 2EXP-14                     |
|            | CS     | THREE       | # GIVES SEC/100 MOD 60                     |
|            | ADS    | DSPCOUNT    | # DSPCOUNT ALREADY SET TO RXD1             |
|            | TC     | BANKCALL    | # DISPLAY SEC MOD 60 IN D4D5.              |
| # Page 340 |        |             |                                            |
|            | CADR   | DSPDC2NR    |                                            |
|            | CAF    | ZERO        |                                            |
|            | TS     | CODE        |                                            |
|            | CS     | TWO         |                                            |
|            | INDEX  | DECOUNT     |                                            |
|            | AD     | R1D1        | # RXD3                                     |
|            | TS     | COUNT       |                                            |
|            | TC     | BANKCALL    | # BLANK MIDDLE CHAR                        |
|            | CADR   | DSPIN       |                                            |
|            | TC     | SEPMIN      | # REMOVE REST OF SECONDS                   |
|            | XCH    | MPAC +1     | # LEAVE FRACT MIN/60 IN MPAC+1             |
|            | EXTEND |             | # USE ONLY FRACT MIN/60 MOD 60             |
|            | MP     | HIMINCON    | # MULT BY .6 + 2EXP-7                      |
|            | DXCH   | MPAC        | # GIVES MIN/100 MOD 60                     |
|            | INDEX  | DECOUNT     |                                            |
|            | CAF    | R1D1        | # RXD1                                     |
|            | TS     | DSPCOUNT    |                                            |
|            | TC     | BANKCALL    | # DISPLAY MIN MOD 60 IN D1D2.              |
|            | CADR   | DSPDC2NR    |                                            |
|            | TC     | POSTJUMP    |                                            |
|            | CADR   | DSPDCEND +2 |                                            |

July 12, 2016

1455

|            |        |            |                                         |
|------------|--------|------------|-----------------------------------------|
| HISECON    | OCT    | 23147      | # .6 + 2EXP-14                          |
| HIMINCON   | OCT    | 23346      | # .6 + 2EXP-7                           |
| M/SCON1    | OCT    | 77753      | # - HI PART OF (59M58.5S)               |
| M/SCON2    | OCT    | 41126      | # - LO PART OF (59M58.5S)               |
| NORMADR    | ADRES  | M/SNORM +1 |                                         |
| M/SCON3    | OCT    | 00025      | # 59M 59.5S                             |
|            | OCT    | 37016      |                                         |
| SEPSEC     | CCS    | MPAC +1    | # IF +, ROUND BY ADDING .5 SEC          |
|            | TCF    | POSEC      | # IF -, ROUND BY SUBTRACING .5 SEC      |
|            | TCF    | POSEC      | # FINDS TIME IN MPAC, MPAC+1            |
|            | TCF    | +1         | # ROUNDS OFF BY +/- .5 SEC              |
|            | EXTEND |            | # LEAVES WHOLE MIN IN BIT13 OF          |
|            | DCS    | RNDCON -1  | # LOTEMOUT AND ABOVE.                   |
| SEPSEC1    | DAS    | MPAC       | # LEAVES FRACT SEC/60 IN MPAC, MPAC+1.  |
|            | TCF    | SEPSECNR   |                                         |
| POSEC      | EXTEND |            |                                         |
|            | DCA    | RNDCON -1  |                                         |
|            | TCF    | SEPSEC1    |                                         |
| SEPSECNR   | XCH    | Q          | # THIS ENTRY AVOIDS ROUNDING BY .5 SEC  |
|            | TS     | SEPSECRET  |                                         |
|            | TC     | DMP        | # MULT BY 2EXP12/6000                   |
|            | ADRES  | SECON1     | # GIVES FRACT SEC/60 IN BIT12 OF MPAC+1 |
|            | EXTEND |            | # AND BELOW.                            |
|            | DCA    | MPAC       | # SAVE MINUTES AND HOURS                |
|            | DXCH   | HITEMOUT   |                                         |
| # Page 341 |        |            |                                         |
|            | TC     | TPSL1      |                                         |
|            | TC     | TPSL1      | # GIVES FRACT SEC/60 IN MPAC+1, MPAC+2. |
|            | CAF    | ZERO       |                                         |
|            | XCH    | MPAC +2    | # LEAVE FRACT SEC/60 IN MPAC, MPAC+1.   |
|            | XCH    | MPAC +1    |                                         |
|            | XCH    | MPAC       |                                         |
|            | TC     | SEPSECRET  |                                         |
| SEPMIN     | XCH    | Q          | # FIND WHOLE MINUTES IN BIT13           |
|            | TS     | SEPMNRET   | # OF LOTEMOUT AND ABOVE.                |
|            | CA     | LOTEMOUT   | # REMOVES REST OF SECONDS.              |
|            | EXTEND |            | # LEAVES FRACT MIN/60 IN MPAC+1.        |
|            | MP     | BIT3       | # LEAVES WHOLE HOURS IN MPAC.           |
|            | EXTEND |            | # SR 12, THROW AWAY LP.                 |
|            | MP     | BIT13      | # SR 2, TAKE FROM LP. = SL 12.          |
|            | LXCH   | MPAC +1    | # THIS FORCES BITS 12-1 TO 0 IF +.      |
|            |        |            | # FORCES BITS 12-1 TO 1 IF -.           |
|            | CA     | HITEMOUT   |                                         |

|          |       |          |                                 |
|----------|-------|----------|---------------------------------|
|          | TS    | MPAC     |                                 |
|          | TC    | DMP      | # MULT BY 1/15                  |
|          | ADRES | MINCON1  | # GIVES FRACT MIN/60 IN MPAC+1. |
| ENDSPMIN | TC    | SEPMNRET | # GIVES WHOLE HOURS IN MPAC.    |

# THIS IS A SPECIAL PURPOS VERB FOR DISPLAYING A DOUBLE PRECISION AGC  
 # WORD AS 10 DECIMAL DIGITS ON THE AGC DISPLAY PANEL. IT CAN BE USED WITH  
 # ANY NOUN, EXCEPT MIXED NOUNS. IT DISPLAYS THE CONTENTS  
 # OF THE REGISTER NOUNADD IS POINTING TO. IF USED WITH NOUNS WHICH ARE  
 # INHERENTLY NOT DP SUCH AS THE CDU COUNTERS THE DISPLAY WILL BE GARBAGE.  
 # DISPLAY IS IN R1 AND R2 ONLY WITH THE SIGN IN R1.

SETLOC ENDRDLO +1

COUNT 40/PIN

|          |        |          |               |
|----------|--------|----------|---------------|
| DSPDPDEC | INDEX  | MIXBR    |               |
|          | TC     | +0       |               |
|          | TC     | +2       | # NORMAL NOUN |
|          | TC     | DSPALARM |               |
|          | EXTEND |          |               |
|          | INDEX  | NOUNADD  |               |
|          | DCA    | 0        |               |
|          | DXCH   | MPAC     |               |
|          | CAF    | R1D1     |               |
|          | TS     | DSPCOUNT |               |
|          | CAF    | ZERO     |               |
|          | TS     | MPAC +2  |               |
|          | TC     | TPAGREE  |               |

# Page 342

|          |    |         |
|----------|----|---------|
|          | TC | DSP2DEC |
| ENDDPDEC | TC | ENTEXIT |

# Page 343

# LOAD VERBS IF ALARM CONDITION IS DETECTED DURING EXECUTE,  
 # CHECK FAIL LIGHT IS TURNED ON AND ENDOFJOB. IF ALARM CONDITION IS  
 # DETECTED DURING ENTER OF DATA, CHECK FAIL IS TURNED ON AND IT RECYCLES  
 # TO EXECUTE OF ORIGINAL LOAD VERB. RECYCLE CAUSED BY 1) DECIMAL MACHINE  
 # CADR 2) MIXTURE OF OCTAL/DECIMAL DATA 3) OCTAL DATA INTO DECIMAL  
 # ONLY NOUN 4) DEC DATA INTO OCT ONLY NOUN 5) DATA TOO LARGE FOR SCALE  
 # 6) FEWER THAN 3 DATA WORDS LOADED FOR HRS, MIN, SEC NOUN.8 (2)-(6) ALARM  
 # AND RECYCLE OCCUR AT FINAL ENTER OF SET. (1) ALARM AND RECYCLE OCCUR AT  
 # ENTER OF CADR.

SETLOC ENDRTOUIT



|            |        |             |                                            |
|------------|--------|-------------|--------------------------------------------|
|            | COUNT  | 41/PIN      |                                            |
| ABCLOAD    | CS     | TWO         |                                            |
|            | TC     | COMPTST     |                                            |
|            | TC     | NOUNTEST    | # TEST IF NOUN CAN BE LOADED.              |
|            | CAF    | VBSP1LD     |                                            |
|            | TC     | UPDATVB -1  |                                            |
|            | TC     | REQDATX     |                                            |
|            | CAF    | VBSP2LD     |                                            |
|            | TC     | UPDATVB -1  |                                            |
|            | TC     | REQDATY     |                                            |
|            | CAF    | VBSP3LD     |                                            |
|            | TC     | UPDATVB -1  |                                            |
|            | TC     | REQDATZ     |                                            |
| PUTXYZ     | CS     | SIX         | # TEST THAT THE 3 DATA WORDS LOADED ARE    |
|            | TC     | ALLDC/OC    | # ALL DEC OR ALL OCT.                      |
|            | EXTEND |             |                                            |
|            | DCA    | LODNNLOC    | # SWITCH BANKS TO NOUN TABLE READING       |
|            | DXCH   | Z           | # ROUTINE.                                 |
|            | CAF    | ZERO        | # X COMP                                   |
|            | TC     | PUTCOM      |                                            |
|            | INDEX  | NOUNADD     |                                            |
|            | TS     | 0           |                                            |
|            | CAF    | ONE         | # Y COMP                                   |
|            | TC     | PUTCOM      |                                            |
|            | INDEX  | NOUNADD     |                                            |
|            | TS     | 1           |                                            |
|            | CAF    | TWO         | # Z COMP                                   |
|            | TC     | PUTCOM      |                                            |
|            | INDEX  | NOUNADD     |                                            |
|            | TS     | 2           |                                            |
|            | CS     | SEVEN       | # IF NOUN 7 HAS JUST BEEN LOADED, SET      |
|            | AD     | NOUNREG     | # FLAG BITS AS SPECIFIED.                  |
|            | EXTEND |             |                                            |
|            | BZF    | +2          |                                            |
| # Page 344 | TC     | LOADLV      |                                            |
|            | CA     | XREG        | # ECADR OF ERASABLE CELL                   |
|            | TC     | SETNCADR +1 | # SET EBANK, NOUNADD                       |
|            | CA     | ZREG        | # ZERO TO RESET BITS. NON-ZERO TO SET BITS |
|            | INHINT |             |                                            |
|            | EXTEND |             |                                            |
|            | BZF    | BITSOFF     |                                            |
|            | INDEX  | NOUNADD     |                                            |

|            |        |            |                                         |
|------------|--------|------------|-----------------------------------------|
|            | CS     | 0          |                                         |
|            | MASK   | YREG       | # BITS TO BE PROCESSED                  |
|            | INDEX  | NOUNADD    |                                         |
|            | ADS    | 0          | # SET BITS.                             |
|            | TC     | BITSOFF1   |                                         |
| BITSOFF    | CS     | YREG       | # BITS TO BE PROCESSED                  |
|            | INDEX  | NOUNADD    |                                         |
|            | MASK   | 0          |                                         |
|            | INDEX  | NOUNADD    |                                         |
|            | TS     | 0          | # RESET BITS                            |
| BITSOFF1   | RELINT |            |                                         |
|            | TC     | LOADLV     |                                         |
| ABLOAD     | CS     | ONE        |                                         |
|            | TC     | COMPTST    |                                         |
|            | TC     | NOUNTEST   | # TEST IF NOUN CAN BE LOADED            |
|            | CAF    | VBSP1LD    |                                         |
|            | TC     | UPDATVB -1 |                                         |
|            | TC     | REQDATX    |                                         |
|            | CAF    | VBSP2LD    |                                         |
|            | TC     | UPDATVB -1 |                                         |
|            | TC     | REQDATY    |                                         |
| PUTXY      | CS     | FIVE       | # TEST THAT THE 2 DATA WORDS LOADED ARE |
|            | TC     | ALLDC/OC   | # ALL DEC OR ALL OCT.                   |
|            | EXTEND |            |                                         |
|            | DCA    | LODNNLOC   | # SWITCH BANKS TO NOUN TABLE READING    |
|            | DXCH   | Z          | # ROUTINE.                              |
|            | CAF    | ZERO       | # X COMP                                |
|            | TC     | PUTCOM     |                                         |
|            | INDEX  | NOUNADD    |                                         |
|            | TS     | 0          |                                         |
|            | CAF    | ONE        | # Y COMP                                |
|            | TC     | PUTCOM     |                                         |
|            | INDEX  | NOUNADD    |                                         |
|            | TS     | 1          |                                         |
|            | TC     | LOADLV     |                                         |
| ALOAD      | TC     | REQDATX    |                                         |
|            | EXTEND |            |                                         |
|            | DCA    | LODNNLOC   | # SWITCH BANKS TO NOUN TABLE READING    |
|            | DXCH   | Z          | # ROUTINE.                              |
|            | CAF    | ZERO       | # X COMP                                |
| # Page 345 | TC     | PUTCOM     |                                         |
|            | INDEX  | NOUNADD    |                                         |

July 12, 2016

1459

|          |        |          |                                           |
|----------|--------|----------|-------------------------------------------|
|          | TS     | 0        |                                           |
|          | TC     | LOADLV   |                                           |
| BLOAD    | CS     | ONE      |                                           |
|          | TC     | COMPTST  |                                           |
|          | CAF    | BIT15    | # SET CLPASS FOR PASSO ONLY               |
|          | TS     | CLPASS   |                                           |
|          | TC     | REQDATY  |                                           |
|          | EXTEND |          |                                           |
|          | DCA    | LODNNLOC | # SWITCH BANKS TO NOUN TABLE READING      |
|          | DXCH   | Z        | # ROUTINE.                                |
|          | CAF    | ONE      |                                           |
|          | TC     | PUTCOM   |                                           |
|          | INDEX  | NOUNADD  |                                           |
|          | TS     | 1        |                                           |
|          | TC     | LOADLV   |                                           |
| CLOAD    | CS     | TWO      |                                           |
|          | TC     | COMPTST  |                                           |
|          | CAF    | BIT15    | # SET CLPASS FOR PASSO ONLY               |
|          | TS     | CLPASS   |                                           |
|          | TC     | REQDATZ  |                                           |
|          | EXTEND |          |                                           |
|          | DCA    | LODNNLOC | # SWITCH BANKS TO NOUN TABLE READING      |
|          | DXCH   | Z        | # ROUTINE.                                |
|          | CAF    | TWO      |                                           |
|          | TC     | PUTCOM   |                                           |
|          | INDEX  | NOUNADD  |                                           |
|          | TS     | 2        |                                           |
|          | TC     | LOADLV   |                                           |
| LOADLV   | CAF    | ZERO     |                                           |
|          | TS     | DECBRNCH |                                           |
|          | CS     | ZERO     |                                           |
|          | TS     | LOADSTAT |                                           |
|          | TC     | RELDSP   | # RELEASE FOR PRIORITY DISPLAY PROBLEM.   |
|          | CS     | VD1      | # TO BLOCK NUMERICAL CHARACTERS AND       |
|          | TS     | DSPCOUNT | # CLEARS AFTER A COMPLETED LOAD           |
|          | TC     | POSTJUMP | # AFTER COMPLETED LOAD, GO TO RECALTST    |
|          | CADR   | RECALTST | # TO SEE IF THERE IS RECALL FROM ENDIDLE. |
| VBSP1LD  | DEC    | 21       | # VB21 = ALOAD                            |
| VBSP2LD  | DEC    | 22       | # VB22 = BLOAD                            |
| VBSP3LD  | DEC    | 23       | # VB23 = CLOAD                            |
| ALLDC/OC | TS     | DECOUNT  | # TESTS THAT DATA WORDS LOADED ARE EITHER |

```

Page 346
CS DECBRNCH # ALL DEC OR ALL OCT. ALARMS IF NOT.

TS SR
CS SR
CS SR # SHIFTED RIGHT 2
CCS A # DEC COMP BITS IN LOW 3
TCF +2 # SOME ONES IN LOW 3
TC Q # ALL ZEROS. ALL OCTAL. OK
AD DECOUNT # DEC COMP = 7 FOR 3COMP, =6 FOR 2COMP
EXTEND
BZF +2 # MUST MATCH 6 FOR 3COMP, 5 FOR 2COMP.
TC ALMCYCLE # ALARM AND RECYCLE.
GOQ TC Q # ALL REQUIRED ARE DEC. OK

SFRUTNOR XCH Q # GETS SF ROUTINE NUMBER FOR NORMAL CASE
 TS EXITEM # CAN'T USE L FOR RETURN. TSTFORDP USES L.
 CAF MID5
 MASK NNTYPTM
 TC RIGHT5
 TC EXITEM # SF ROUTINE NUMBER IN A

SFRUTMIX XCH Q # GETS SF ROUTINE NUMBER FOR MIXED CASE
 TS EXITEM
 INDEX DECOUNT
 CAF DISPLACE # PUT TC GOQ, TC RIGHT5, OR TC LEFT5 IN L
 TS L
 INDEX DECOUNT
 CAF LOW5 # LOW5, MID5, OR HI5 IN A
 MASK RUTMXTEM # GET HI5, MID5, OR LOW5 OF RUTMXTAB ENTRY
 INDEX L
 TC 0

DO TC GOQ(DECOUNT=0), DO TC RIGHT5(DECOUNT=1), DO TC LEFT5(DECOUNT=2).

SFRET1 TC EXITEM # SF ROUTINE NUMBER IN A

SFCONUM XCH Q # GETS 2X(SF CONSTANT NUMBER)
 TS EXITEM
 INDEX MIXBR
 TC +0
 TC CONUMNOR # NORMAL NOUN
 INDEX DECOUNT # MIXED NOUN
 CAF DISPLACE
 TS L # PUT TC GOQ, TC RIGHT5, OR TC LEFT5 IN L
 INDEX DECOUNT
 CAF LOW5

```

July 12, 2016

1461

MASK NNTYPTM  
INDEX L  
TC 0

# DO TC GOQ(DECOUNT=0), DO TC RIGHT5(DECOUNT=1), DO TC LEFT5(DECOUNT=2).

SFRET DOUBLE # 2X(SF CONSTANT NUMBER) IN A  
TC EXITEM

# Page 347

DISPLACE TC GOQ  
TC RIGHT5  
TC LEFT5

CONUMNOR CAF LOW5 # NORMAL NOUN ALWAYS GETS LOW 5 OF  
MASK NNTYPTM # NNTYPTAB FOR SF CONUM.  
DOUBLE  
TC EXITEM # 2X(SF CONSTANT NUMBER) IN A

PUTCOM TS DECOUNT  
XCH Q  
TS DECRET  
CAF ZERO  
TS MPAC+6  
INDEX DECOUNT  
XCH XREGLP  
TS MPAC +1  
INDEX DECOUNT  
XCH XREG  
TS MPAC  
INDEX MIXBR  
TC +0  
TC PUTNORM # NORMAL NOUN

# IF MIXNOUN, PLACE ADDRESS FOR COMPONENT K INTO NOUNADD, SET EBANK BITS.

INDEX DECOUNT # GET IDADDTAB ENTRY FOR COMPONENT K  
CA IDAD1TEM # OF NOUN.  
MASK LOW11 # (ECADR)SUBK FOR CURRENT COMP OF NOUN  
TC SETNCADR # ECADR INTO NOUNCADR. SETS EB, NOUNADD.  
EXTEND # C(NOUNADD) IN A UPON RETURN  
SU DECOUNT # PLACE (ESUBK)-K INTO NOUNADD  
TS NOUNADD  
CCS DECBRNCH  
TC PUTDECSF # + DEC

|            |        |            |                                            |
|------------|--------|------------|--------------------------------------------|
|            | TC     | DCTSTCYC   | # +0 OCTAL                                 |
|            | TC     | SFRUTMIX   | # TEST IF DEC ONLY BIT = 1. IF SO,         |
|            | TC     | DPTEST     | # ALARM AND RECYCLE. IF NOT, CONTINUE.     |
|            | TC     | PUTCOM2    | # NO DP                                    |
|            |        |            | # TEST FOR DP SCALE FOR OCT LOAD. IF SO,   |
|            |        |            | # +0 INTO MAJOR PART. SET NOUNADD FOR      |
|            |        |            | # LOADING OCTAL WORD INTO MINOR PART.      |
| PUTDPCOM   | INCR   | NOUNADD    | # DP (ESUBK)-K+1 OR E+1                    |
|            | CA     | NOUNADD    | # NOUNADD NOW SET FOR MINOR PART           |
|            | ADS    | DECOUNT    | # (ESUBK)+1 OR E+1 INTO DECOUNT            |
|            | CAF    | ZERO       | # NOUNADD SET FOR MINOR PART               |
|            | INDEX  | DECOUNT    |                                            |
|            | TS     | 0 -1       | # ZERO MAJOR PART(ESUBK OR E)              |
|            | TC     | PUTCOM2    |                                            |
| PUTNORM    | TC     | SETNADD    | # ECADR FROM NOUNCADR. SETS EB, NOUNADD.   |
| # Page 348 |        |            |                                            |
|            | CCS    | DECBRNCH   |                                            |
|            | TC     | PUTDECSF   | # + DEC                                    |
|            | TC     | DCTSTCYC   | # +0 OCTAL                                 |
|            | TC     | SFRUTNOR   | # TEST IF DEC ONLY BIT (garbled). IF SO,   |
|            | TC     | DPTEST     | # ALARM AND RECYCLE. IF NOT, CONTINUE.     |
|            | TC     | PUTCOM2 -4 | # NO DP                                    |
|            | CAF    | ZERO       | # DP                                       |
|            | TS     | DECOUNT    |                                            |
|            | TC     | PUTDPCOM   |                                            |
|            | CA     | NNADTEM    |                                            |
|            | AD     | ONE        | # IF NNADTEM = -1, CHANNEL TO BE SPECIFIED |
|            | EXTEND |            |                                            |
|            | BZF    | CHANLOAD   |                                            |
| PUTCOM2    | XCH    | MPAC       |                                            |
|            | TC     | DECRET     |                                            |
|            | EBANK= | DSPCOUNT   |                                            |
| GTSFINLC   | 2CADR  | GTSFIN     |                                            |
| CHANLOAD   | CS     | SEVEN      | # DON'T LOAD CHAN 7. (IT'S SUPERBANK).     |
|            | AD     | NOUNCADR   |                                            |
|            | EXTEND |            |                                            |
|            | BZF    | LOADLV     |                                            |
|            | CA     | NOUNCADR   |                                            |
|            | MASK   | LOW9       |                                            |
|            | XCH    | MPAC       |                                            |
|            | EXTEND |            |                                            |
|            | INDEX  | MPAC       |                                            |

July 12, 2016

1463

WRITE 0  
TC LOADLV

# PUTDECSF FINDS MIXBR AND DECOUNT STILL SET FROM PUTCOM.

PUTDECSF TC SFCONUM # 2X(SF CON NUMB) IN A  
TS SFTEMP1  
EXTEND # SWITCH BANKS TO SF CONSTANT TABLE  
DCA GTSFINLC # READING ROUTINE.  
DXCH Z # LOADS SFTEMP1, SFTEMP2.  
INDEX MIXBR  
TC +0  
TC PUTSFNOR  
TC SFRUTMIX  
TC PUTDCSF2  
PUTSFNOR TC SFRUTNOR

PUTDCSF2 INDEX A

# Page 349

CAF SFINTABR  
TC BANKJUMP # SWITCH BANKS FOR EXPANSION ROOM  
SFINTABR CADR GOALMCYC # ALARM AND RECYCLE IF DEC LOAD  
# WITH OCTAL ONLY NOUN.  
CADR BINROUND  
CADR DEGINSF  
CADR ARTHINSF  
CADR DPINSF  
CADR DPINSF2  
CADR OPTDEGIN  
CADR DPINSF # SAME AS ARITHDP1  
CADR HMSIN  
CADR DSPALARM # MIN/SEC CAN'T BE LOADED.  
CADR DPINSF4  
CADR ARTIN1SF  
CADR DSPALARM # 2INTOUT CAN'T BE LOADED.  
CADR DPFRACIN  
ENDROUTIN EQUALS

# SCALE FACTORS FOR THOSE ROUTINES NEEDING THEM ARE AVAILABLE IN SFTEMP1.

# ALL SFIN ROUTINES USE MPAC MPAC+1. LEAVE RESULT IN A. END WITH TC DECRET.

SETLOC ENDDPDEC +1

COUNT 40/PIN

# DEGINSF APPLIES  $1000/180 * 5.5555(10) = 5.43434(8)$

```

DEGINSF TC DMP # SF ROUTINE FOR DEC DEGREES
 ADRES DEGCON1 # MULT BY 5.5 5(10)X2EXP-3
 CCS MPAC +1 # THIS ROUNDS OFF MPAC+1 BEFORE SHIFT
 CAF BIT11 # LEFT 3, AND CAUSES 360.00 TO OF/UF
 TC +2 # WHEN SHIFTED LEFT AND ALARM.
 CS BIT11
 AD MPAC +1
 TC 2ROUND +2
 TC TPSL1 # LEFT 1
DEGINSF2 TC TPSL1 # LEFT 2
 TC TESTOFUF
 TC TPSL1 # RETURNS IF NO OF/UF (LEFT3)
 CCS MPAC
 TC SIGNFIX # IF +, GO TO SIGNFIX
 TC SIGNFIX # IF +0, GO TO SIGNFIX
 COM # IF -, USE -MAGNITUDE +1
 TS MPAC # IF -0, USE +0
SIGNFIX CCS MPAC+6
 TC SGNT01 # IF OVERFLOW
 TC ENDSALE # NO OVERFLOW/UNDERFLOW
Page 350
 CCS MPAC # IF UF FORCE SIGN TO 0 EXCEPT -180
 TC CCSHOLE
 TC NEG180
 TC +1
 XCH MPAC
 MASK POSMAX
 TS MPAC
ENDSCALE TC POSTJUMP
 CADR PUTCOM2

NEG180 CS POSMAX
 TC ENDSALE -1

SGNT01 CS MPAC # IF OF FORCE SIGN TO 1
 MASK POSMAX
 CS A
 TC ENDSALE -1

DEGCON1 2DEC 5.55555555 B-3
DEGCON2 2DEC 2.22222222 B-2

NEG.2 OCT -06250 # = .197753906 I.E., THE BIAS SCALED

```



July 12, 2016

1465

|            |        |           |                                           |
|------------|--------|-----------|-------------------------------------------|
| ARTHINSF   | TC     | DMP       | # SCALES MPAC, +1 BY SFTEMP1, SFTEMP2.    |
|            | ADRES  | SFTEMP1   | # ASSUMES POINT BETWEEN HI AND LO PARTS   |
|            | XCH    | MPAC +2   | # OF SFCON. SHIFTS RESULTS LEFT BY 14.    |
|            | XCH    | MPAC +1   | # (BY TAKING RESULTS FROM MPAC+1, MPAC+2) |
|            | XCH    | MPAC      |                                           |
|            | EXTEND |           |                                           |
|            | BZF    | BINROUND  |                                           |
|            | TC     | ALMCYCLE  | # TOO LARGE A LOAD. ALARM AND RECYCLE.    |
| BINROUND   | TC     | 2ROUND    |                                           |
|            | TC     | TESTOFUF  |                                           |
|            | TC     | ENDSCALE  | # RETURNS IF NO OF/UF                     |
| ARTIN1SF   | TC     | DMP       | # SCALES MPAC, +1 BY SFTEMP1, SFTEMP2.    |
|            | ADRES  | SFTEMP1   | # ROUNDS MPAC+1 INTO MPAC.                |
|            | TC     | BINROUND  |                                           |
| OPTDEGIN   | CCS    | MPAC      | # OPTICS SCALING ROUTINE                  |
|            | TC     | +4        |                                           |
|            | TC     | +3        |                                           |
|            | TC     | ALMCYCLE  | # REJECT -- INPUT. ALARM AND RECYCLE.     |
|            | TC     | ALMCYCLE  | # REJECT -- INPUT. ALARM AND RECYCLE.     |
| OPDEGIN2   | CAF    | NEG.2     | # RANGE IS 90 DEG                         |
|            | ADS    | MPAC      | # SUBTRACT BIAS                           |
| # Page 351 |        |           |                                           |
|            | TC     | DMP       | # MULT BY 100 / 45 B-2                    |
|            | ADRES  | DEGCON2   |                                           |
|            | CAF    | BIT12     | # ROUND AS IN DEGINSF                     |
|            | AD     | MPAC +1   |                                           |
|            | TC     | 2ROUND +2 |                                           |
|            | TC     | DEGINSF2  |                                           |
| DPINSF     | TC     | DMP       | # SCALES MPAC, MPAC +1 BY SFTEMP1,        |
|            | ADRES  | SFTEMP1   | # SFTEMP. STORES LOW PART OF RESULT       |
|            | XCH    | MPAC +2   | # IN (E SUBK) +1 OR E+1                   |
|            | DOUBLE |           |                                           |
|            | TS     | MPAC +2   |                                           |
|            | CAF    | ZERO      |                                           |
|            | AD     | MPAC +1   |                                           |
|            | TC     | 2ROUND +2 |                                           |
|            | TC     | TESTOFUF  |                                           |
| DPFRACIN   | INDEX  | MIXBR     | # RETURNS IF NO OF/UF                     |
|            | TC     | +0        |                                           |
|            | TC     | DPINORM   |                                           |
|            | CA     | DECOUNT   | # MIXED NOUN                              |
| DPINCOM    | AD     | NOUNADD   | # MIXED NORMAL                            |
|            | TS     | Q         | # E SUBK E                                |

|            |        |             |                                            |
|------------|--------|-------------|--------------------------------------------|
|            | XCH    | MPAC +1     |                                            |
|            | INDEX  | Q           |                                            |
|            | TS     | 1           | # PLACE LOW PART IN                        |
|            | TC     | ENDSCALE    | # (E SUBK) +1 MIXED                        |
| DPINORM    | CAF    | ZERO        | # E +1 NORMAL                              |
|            | TC     | DPINCOM     |                                            |
| DPINSF2    | TC     | DMP         | # ASSUMES POINT BETWEEN BITS 7-8 OF HIGH   |
|            | ADRES  | SFTEMP1     | # PART OF SF CONST. DPINSF2 SHIFTS RESULTS |
|            | CAF    | SIX         | # LEFT BY 7, ROUNDS MPAC+2 INTO MPAC+1     |
|            | TC     | TPLEFTN     | # SHIFT LEFT 7.                            |
|            | TC     | DPINSF +2   |                                            |
| DPINSF4    | TC     | DMP         | # ASSUMES POINT BETWEEN BITS 11-12 OF HIGH |
|            | ADRES  | SFTEMP1     | # PART OF SF CONST. DPINSF2 SHIFTS RESULTS |
|            | CAF    | TWO         | # LEFT BY 3, ROUNDS MPAC+2 INTO MPAC+1.    |
|            | TC     | TPLEFTN     | # SHIFT LEFT 3.                            |
|            | TC     | DPINSF +2   |                                            |
| TPLEFTN    | XCH    | Q           | # SHIFTS MPAC, +1, +2 LEFT N. SETS OVFIN   |
|            | TS     | SFTEMP2     | # TO +1 FOR OF, -1 FOR UF.                 |
|            | XCH    | Q           | # CALL WITH N-1 IN A.                      |
| LEFTNCOM   | TS     | SFTEMP1     | # LOOP TIME .37 MSEC.                      |
|            | TC     | TPSL1       |                                            |
|            | CCS    | SFTEMP1     |                                            |
| # Page 352 |        |             |                                            |
|            | TC     | LEFTNCOM    |                                            |
|            | TC     | SFTEMP2     |                                            |
| 2ROUND     | XCH    | MPAC +1     |                                            |
|            | DOUBLE |             |                                            |
|            | TS     | MPAC +1     |                                            |
|            | TC     | Q           | # IF MPAC+1 DOES NOT OF/UF                 |
|            | AD     | MPAC        |                                            |
|            | TS     | MPAC        |                                            |
|            | TC     | Q           | # IF MPAC DOES NOT OF/UF                   |
|            | TS     | MPAC+6      |                                            |
| 2RNDEND    | TC     | Q           |                                            |
| TESTOFUF   | CCS    | MPAC+6      | # RETURNS IF NO OF/UF                      |
|            | TC     | ALMCYCLE    | # OF ALARM AND RECYCLE.                    |
|            | TC     | Q           |                                            |
|            | TC     | ALMCYCLE    | # UF ALARM AND RECYCLE.                    |
|            | SETLOC | ENDSPMIN +1 |                                            |

|            | COUNT  | 42/PIN   |                                          |
|------------|--------|----------|------------------------------------------|
| HMSIN      | TC     | ALL3DEC  | # IF ALL 3 WORDS WERE NOT LOADED, ALARM. |
|            | TC     | DMP      | # XREG, XREGLP (=HOURS) WERE ALREADY PUT |
|            | ADRES  | WHOLECON | # INTO MPAC, MPAC+1.                     |
|            | TC     | RND/TST  | # ROUND OFF TO WHOLE HRS IN MPAC+1.      |
|            | CAF    | ZERO     | # ALARM IF MPAC NON ZERO (G/ 16383).     |
|            | TS     | MPAC +2  |                                          |
|            | CAF    | HRCN     |                                          |
|            | TS     | MPAC     |                                          |
|            | CAF    | HRCN +1  |                                          |
|            | XCH    | MPAC +1  |                                          |
|            | TC     | SHORTMP  |                                          |
|            | TC     | MPACTST  | # ALARM IF MPAC NON ZERO (G/ 745)        |
|            | DXCH   | MPAC +1  | # STORE HOURS CONTRIBUTION               |
|            | DXCH   | HITEMIN  |                                          |
|            | CA     | YREG     | # PUT YREG, YREGLP INTO MPAC, +1.        |
|            | LXCH   | YREGLP   |                                          |
|            | DXCH   | MPAC     |                                          |
|            | TC     | DMP      |                                          |
|            | ADRES  | WHOLECON |                                          |
|            | TC     | RND/TST  | # ROUND OFF TO WHOLE MIN IN MPAC+1       |
|            | CS     | 59MIN    | # ALARM IF MPAC NON ZERO (G/16383)       |
|            | TC     | SIZETST  | # ALARM IF MPAC+1 G/ 59MIN               |
|            | XCH    | MPAC +1  |                                          |
|            | EXTEND |          |                                          |
|            | MP     | MINCON   | # LEAVES MINUTES CONTRIBUTION IN A,L     |
| # Page 353 | DAS    | HITEMIN  | # ADD IN MINUTES CONTRIBUTION            |
|            | EXTEND |          | # IF THIS DAS OVEFLOWS, G/ 745 HR, 39MIN |
|            | BZF    | +2       |                                          |
|            | TC     | ALMCYCLE |                                          |
|            | CA     | ZREG     | # PUT ZREG, ZREGLP INTO MPAC +1.         |
|            | LXCH   | ZREGLP   |                                          |
|            | DXCH   | MPAC     |                                          |
|            | TC     | DMP      |                                          |
|            | ADRES  | WHOLECON |                                          |
|            | TC     | RND/TST  | # ROUND OFF TO WHOLE CENTI-SEC IN MPAC+1 |
|            | CS     | 59.99SEC | # ALARM IF MPAC NON ZERO (G/163.83 SEC)  |
|            | TC     | SIZETST  | # ALARM IF MPAC+1 G/59.99 SEC            |
|            | DXCH   | HITEMIN  | # ADD IN SECONDS CONTRIBUTION            |
|            | DAS    | MPAC     | # IF THIS DAS OVERFLOWS,                 |
|            | EXTEND |          | # G/ 745 HR, 39 MIN, 14.59 SEC.          |
|            | BZF    | +2       |                                          |
|            | TC     | ALMCYCLE | # ALARM AND RECYCLE                      |

|                                                                 |        |          |                                            |
|-----------------------------------------------------------------|--------|----------|--------------------------------------------|
|                                                                 | CAF    | ZERO     |                                            |
|                                                                 | TS     | MPAC +2  |                                            |
|                                                                 | TC     | TPAGREE  |                                            |
|                                                                 | DXCH   | MPAC     |                                            |
|                                                                 | INDEX  | NOUNADD  |                                            |
|                                                                 | DXCH   | 0        |                                            |
|                                                                 | TC     | POSTJUMP |                                            |
|                                                                 | CADR   | LOADLV   |                                            |
| WHOLECON                                                        | OCT    | 00006    | # (10EXP5/2EXP14)2EXP14                    |
|                                                                 | OCT    | 03240    |                                            |
| HRCN                                                            | OCT    | 00025    | # 1 HOUR IN CENTI-SEC                      |
|                                                                 | OCT    | 37100    |                                            |
| MINCON                                                          | OCT    | 13560    | # 1 MINUTE IN CENTI-SEC                    |
| 59MIN                                                           | OCT    | 00073    | # 59 AS WHOLE                              |
| 59.99SEC                                                        | OCT    | 13557    | # 5999 SENTI-SEC                           |
| RND/TST                                                         | XCH    | MPAC +2  | # ROUNDS MPAC+2 INTO MPAC+1.               |
|                                                                 | DOUBLE |          | # ALARMS IF MPAC NOT 0                     |
|                                                                 | TS     | MPAC +2  |                                            |
|                                                                 | CAF    | ZERO     |                                            |
|                                                                 | AD     | MPAC +1  |                                            |
|                                                                 | TS     | MPAC +1  |                                            |
|                                                                 | CAF    | ZERO     |                                            |
|                                                                 | AD     | MPAC     | # CAN'T OVFLOW                             |
|                                                                 | XCH    | MPAC     |                                            |
| MPACTST                                                         | CCS    | MPAC     | # ALARM IF MPAC NON ZERO                   |
|                                                                 | TC     | ALMCYCLE | # ALARM AND RECYCLE                        |
|                                                                 | TC     | Q        |                                            |
|                                                                 | TC     | ALMCYCLE | # ALARM AND RECYCLE                        |
|                                                                 | TC     | Q        |                                            |
| # Page 354                                                      |        |          |                                            |
| SIZETST                                                         | TS     | MPAC +2  | # CALLED WITH - CON IN A                   |
|                                                                 | CCS    | MPAC +1  | # GET MAG OF MPAC+1                        |
|                                                                 | AD     | ONE      |                                            |
|                                                                 | TCF    | +2       |                                            |
|                                                                 | AD     | ONE      |                                            |
|                                                                 | AD     | MPAC +2  |                                            |
|                                                                 | EXTEND |          | # MAG OF MPAC+1 - CON                      |
|                                                                 | BZMF   | +2       |                                            |
|                                                                 | TC     | ALMCYCLE | # MAG OF MPAC+1 G/ CON. ALARM AND RECYCLE. |
|                                                                 | TC     | Q        | # MAG OF MPAC+1 L/= CON                    |
| # ALL3DEC TESTS THAT ALL 3 WORDS ARE LOADED IN DEC (FOR HMSIN). |        |          |                                            |

July 12, 2016

1469

# ALARM IF NOT. (TEST THAT BITS 3,4,5 OF DECBRNCH ARE ALL = 1).

|          |        |          |                                           |
|----------|--------|----------|-------------------------------------------|
| ALL3DEC  | CS     | OCT34BAR | # GET BITS 3,4,5 IN A                     |
|          | MASK   | DECBRNCH | # GET BITS 3,4,5 OF DECBRNCH IN A         |
|          | AD     | OCT34BAR | # BITS 3,4,5 OF DECBRNCH MUST ALL = 1     |
|          | CCS    | A        |                                           |
|          | TC     | FORCEV25 |                                           |
| OCT34BAR | OCT    | 77743    |                                           |
|          | TC     | FORCEV25 |                                           |
|          | TC     | Q        |                                           |
| FORCEV25 | CS     | OCT31    | # FORCE VERB 25 TO BE EXECUTED BY RECYCLE |
|          | TS     | VERBSAVE | # IN CASE OPERATOR EXECUTED A LOWER LOAD  |
|          | TC     | ALMCYCLE | # VERB. ALARM AND RECYCLE.                |
| ENDHMSS  | EQUALS |          |                                           |

# Page 355

# MONITOR ALLOWS OTHER KEYBOARD ACTIVITY. IT IS ENDED BY VERB TERMINATE,  
# VERB PROCEED WITHOUT DATA, VERB RESEQUENCE,  
# ANOTHER MONITOR, OR ANY NVSUB CALL THAT PASSES THE DSPLOCK (PROVIDED  
# THAT THE OPERATOR HAS SOMEHOW ALLOWED THE ENDING OF A MONITOR WHICH  
# HE HAS INITIATED THROUGH THE KEYBOARD).  
#  
# MONITOR ACTION IS SUSPENDED, BUT NOT ENDED, BY ANY KEYBOARD ACTION.  
# EXCEPT ERROR LIGHT RESET. IT BEGINS AGAIN WHEN KEY RELEASE IS PERFORMED.  
# MONITOR SAVES THE NOUN AND APPROPRIATE DISPLAY VERB IN MONSAVE. IT SAVES  
# NOUNCADR IN MONSAVE1, IF NOUN = MACHINE CADR TO BE SPECIFIED. BIT 15 OF  
# MONSAVE1 IS THE KILL MONITOR SIGNAL (KILLER BIT). BIT 14 OF MONSAVE1  
# INDICATES THE CURRENT MONITOR WAS EXTERNALLY INITIATED (EXTERNAL  
# MONITOR BIT). IT IS TURNED OFF BY RELDSP AND KILMONON.  
#  
# MONSAVE INDICATES IF MONITOR IS ON (+=ON, +0=OFF)  
# IF MONSAVE IS +, MONITOR ENTERS NO REQUEST, BUT TURNS KILLER BIT OFF.  
# IF MONSAVE IS +0, MONITOR ENTERS REQUEST AND TURNS KILLER BIT OFF.  
#  
# NVSUB (IF EXTERNAL MONITOR BIT IS OFF), VB=PROCEED WITHOUT DATA,  
# VB=RESEQUENCE, AND VB=TERMINATE TURN KILL MONITOR BIT ON.  
#  
# IF KILLER BIT IS ON, MONREQ ENTERS NO FURTHER REQUESTS, ZEROS MONSAVE  
# AND MONSAVE1 (TURNING OFF KILLER BIT AND EXTERNAL MONITOR BIT).  
#  
# MONITOR DOESN'T TEST FOR MATBS SINCE NVSUB CAN HANDLE INTERNAL MATBS NOW.

SETLOC ENDRUTIN

COUNT 41/PIN

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
| MONITOR    | CS     | BIT15/14 |                                            |
|            | MASK   | NOUNCADR |                                            |
| MONIT1     | TS     | MPAC +1  | # TEMP STORAGE                             |
|            | CS     | ENTEXIT  |                                            |
|            | AD     | ENDINST  |                                            |
|            | CCS    | A        |                                            |
|            | TC     | MONIT2   |                                            |
| BIT15/14   | OCT    | 60000    |                                            |
|            | TC     | MONIT2   |                                            |
|            | CAF    | BIT14    | # EXTERNALLY INITIATED MONITOR.            |
|            | ADS    | MPAC +1  | # SET BIT 14 FOR MONSAVE1.                 |
|            | CAF    | ZERO     |                                            |
|            | TS     | MONSAVE2 | # ZERO NVMONOPT OPTIONS                    |
| MONIT2     | CAF    | LOW7     |                                            |
|            | MASK   | VERBREG  |                                            |
|            | TC     | LEFT5    |                                            |
|            | TS     | CYL      |                                            |
|            | CS     | CYL      |                                            |
|            | XCH    | CYL      |                                            |
|            | AD     | NOUNREG  |                                            |
|            | TS     | MPAC     | # TEMP STORAGE                             |
| # Page 356 |        |          |                                            |
|            | CAF    | ZERO     |                                            |
|            | TS     | DSPLOCK  | # +0 INTO DSPLOCK SO MONITOR CAN RUN.      |
|            | CCS    | CADRSTOR | # TURN OFF KR LITE IF CADRSTOR AND DSPLIST |
|            | TC     | +2       | # ARE BOTH EMPTY. (LITE COMES ON IF NEW    |
|            | TC     | RELDSP1  | # MONITOR IS KEYED IN OVER OLD MONITOR.)   |
|            | INHINT |          |                                            |
|            | CCS    | MONSAVE  |                                            |
|            | TC     | +5       | # IF MONSAVE WAS +, NO REQUEST             |
|            | CAF    | ONE      | # IF MONSAVE WAS 0, REQUEST MONREQ         |
|            | TC     | WAITLIST |                                            |
|            | EBANK= | DSPCOUNT |                                            |
|            | 2CADR  | MONREQ   |                                            |
|            |        |          |                                            |
|            | DXCH   | MPAC     | # PLACE MONITOR VERB AND NOUN INTO MONSAVE |
|            | DXCH   | MONSAVE  | # ZERO THE KILL MONITOR BIT                |
|            | RELINT |          | # SET UP EXTERNAL MONITOR BIT              |
|            | TC     | ENTRET   |                                            |
| MONREQ     | TC     | LODSAMPT | # CALLED BY WAITLIST                       |
|            | CCS    | MONSAVE1 | # TIME IS SNATCHED N RUPT FOR NOUN 65      |
|            | TC     | +4       | # IF KILLER BIT = 0, ENTER REQUESTS        |
|            | TC     | +3       | # IF KILLER BIT = 0, ENTER REQUESTS        |
|            | TC     | KILLMON  | # IF KILLER BIT = 1, NO REQUESTS.          |

July 12, 2016

1471

```

TC KILLMON # IF KILLER BIT = 1, NO REQUESTS.
CAF MONDEL
TC WAITLIST # ENTER WAITLIST REQUEST FOR MONREQ
EBANK= DSPCOUNT
2CADR MONREQ

CAF CHRPRIO
TC NOVAC # ENTER EXEC REQUEST FOR MONDO
EBANK= DSPCOUNT
2CADR MONDO

TC TASKOVER

KILLMON CAF ZERO # ZERO MONSAVE AND TURN KILLER BIT OFF
 TS MONSAVE
 TS MONSAVE1 # TURN OFF KILL MONITOR BIT.
 TC TASKOVER # TURN OFF EXTERNAL MONITOR BIT.
MONDEL OCT 144 # FOR 1 SEC MONITOR INTERVALS.

MONDO CCS MONSAVE1 # CALLED BY EXEC
 TC +4 # IF KILLER BIT = 0, CONTINUE
 TC +3 # IF KILLER BIT = 0, CONTINUE
 TC ENDOFJOB # IN CASE TERMINATE CAME SINCE LAST MONREQ
 TC ENDOFJOB # IN CASE TERMINATE CAME SINCE LAST MONREQ
 CCS DSPLOCK

Page 357

TC MONBUSY # NVSUB IS BUSY
CAF LOW7
MASK MONSAVE
TC UPDATNN -1 # PLACE NOUN INTO NOUNREG AND DISPLAY IT
CAF MID7
MASK MONSAVE # CHANGE MONITOR VERB TO DISPLAY VERB
AD MONREF # -DEC10, STARTING IN BIT8
TS EDOP # RIGHT 7
CA EDOP
TS VERBREG
CAF MONBACK # SET RETURN TO PASTEVB AFTER DATA DISPLAY
TS ENTRET
CS BIT15/14
MASK MONSAVE1 # PUT ECADR INTO MPAC +2. INTMCTBS WILL
TS MPAC +2 # DISPLAY IT AND SET NOUNCADR, NOUNADD,
ENDMONDO TC TESTNN # EBANK.

BLOCK 2

SETLOC FFTAG8
```

```

 BANK
 COUNT 02/PIN

PASTEVB CAF MID7
 MASK MONSAVE2 # NVMONOPT PASTE OPTION
 EXTEND
 BZF +2
 TC PASTEOPT # PASTE PLEASE VERB FOR NVMONOPT
 CA MONSAVE # PASTE MONITOR VERB -- PASTE OPTION IS O
PASTEOPT TS EDOP # RIGHT 7
 CA EDOP # PLACE MONITOR VERB OR PLEASE VERB INTO
 TC BANKCALL # VERBREG AND DISPLAY IT.
 CADR UPDATVB -1
 CAF ZERO # ZERO REQRET SO THAT PASTED VERBS CAN
 TS REQRET # BE EXECUTED BY OPERATOR.
 CA MONSAVE2
 TC BLANKSUB # PROCESS NVMONOPT BLANK OPTION IF ANY
 TC +1
ENDPASTE TC ENDOFJOB

MID7 OCT 37600

 SETLOC ENDMONDO +1
 COUNT 41/PIN

MONREF OCT 75377 # -DEC10, STARTING IN BIT8
MONBACK ADRES PASTEVB

Page 358
MONBUSY TC RELDSPON # TURN KEY RELEASE LIGHT
 TC ENDOFJOB

DSPFMEM IS USED TO DISPLAY (IN OCTAL) ANY FIXED REGISTER.
IT IS USED WITH NOUN = MACHINE CADR TO BE SPECIFIED. THE FCADR OF THE
DESIRED LOCATION IS THEN PUNCHED IN. IT HANDLES F/F (FCADR 4000-7777)
#
FOR BANKS L/E 27, THIS IS ENOUGH.
#
FOR BANKS G/E 30, THE THIRD COMPONENT OF NOUN 26 (PRIO, ADRES, BBCON)
MUST BE PRELOADED WITH THE DESIRED SUPERBANK BITS (BITS 5,6,7).
V23N26 SHOULD BE USED.
#
SUMMARY
FOR BANKS L/E 27, V27N01E(FCADR)E
FOR BANKS G/E 30, V23N26E(SUPERBITS)E V27N01E(FCADR)E

```



July 12, 2016

1473

```
DSPFMEM CAF R1D1 # IF F/F, DATACALL USES BANK 02 OR 03.
 TS DSPCOUNT
 CA DSPTM1 +2 # SUPERBANK BITS WERE PRELOADED INTO
 TS L # 3RD COMPONENT OF NOUN 26.
 CA NOUNCADR # ORIGINAL FCADR LOADED STILL IN NOUNCADR.
 TC SUPDACAL # CALL WITH FCADR IN A, SUPERBITS IN L.
 TC DSPOCTWO
ENDSPF TC ENDOFJOB
```

# Page 359

# WORD DISPLAY ROUTINES

```
SETLOC TESTOFUF +4
COUNT 40/PIN
```

```
DSPSIGN XCH Q
 TS DSPWDRET
 CCS MPAC
 TC +8D
 TC +7
 AD ONE
 TS MPAC
 TC -ON
 CS MPAC +1
 TS MPAC +1
 TC DSPWDRET
 TC +ON
 TC DSPWDRET
```

```
DSPRND EXTEND # ROUND BY 5 EXP-6
 DCA DECROUND -1
 DAS MPAC
 EXTEND
 BZF +4
 EXTEND
 DCA DPOSMAX
 DXCH MPAC
 TC Q
```

# DSPDECWD CONVERTS C(MPAC,MPAC+1) INTO A SIGN AND 5 CHAR DECIMAL  
# STARTING IN LOC SPECIFIED IN DSPCOUNT. IT ROUNDS BY 5 EXP-6.

```
DSPDECWD XCH Q
 TS WDRET
 TC DSPSIGN
 TC DSPRND
```

|            |       |          |                                   |
|------------|-------|----------|-----------------------------------|
|            | CAF   | FOUR     |                                   |
| DSPDCWD1   | TS    | WDCNT    |                                   |
|            | CAF   | BINCON   |                                   |
|            | TC    | SHORTMP  |                                   |
| TRACE1     | INDEX | MPAC     |                                   |
|            | CAF   | RELTAB   |                                   |
|            | MASK  | LOW5     |                                   |
|            | TS    | CODE     |                                   |
|            | CAF   | ZERO     |                                   |
|            | XCH   | MPAC +2  |                                   |
|            | XCH   | MPAC +1  |                                   |
|            | TS    | MPAC     |                                   |
|            | XCH   | DSPCOUNT |                                   |
| TRACE1S    | TS    | COUNT    |                                   |
| # Page 360 |       |          |                                   |
|            | CCS   | A        | # DECREMENT DSPCOUNT EXCEPT AT +0 |
|            | TS    | DSPCOUNT |                                   |
|            | TC    | DSPIN    |                                   |
|            | CCS   | WDCNT    |                                   |
|            | TC    | DSPDCWD1 |                                   |
|            | CS    | VD1      |                                   |
|            | TS    | DSPCOUNT |                                   |
|            | TC    | WDRET    |                                   |
|            | OCT   | 00000    |                                   |
| DECROUND   | OCT   | 02476    |                                   |

# DSPDECNR CONVERTS C(MPAC,MPAC+1) INTO A SIGN AND 5 CHAR DECIMAL  
# STARTING IN LOC SPECIFIED IN DSPCOUNT. IT DOES NOT ROUND

|          |     |             |
|----------|-----|-------------|
| DSPDECNR | XCH | Q           |
|          | TS  | WDRET       |
|          | TC  | DSPSIGN     |
|          | TC  | DSPDCWD1 -1 |

# DSPDC2NR CONVERTS C(MPAC,MPAC+1) INTO A SIGN AND 2 CHAR DECIMAL  
# STARTING IN LOC SPECIFIED IN DSPCOUNT. IT DOES NOT ROUND

|          |     |          |
|----------|-----|----------|
| DSPDC2NR | XCH | Q        |
|          | TS  | WDRET    |
|          | TC  | DSPSIGN  |
|          | CAF | ONE      |
|          | TC  | DSPDCWD1 |

# DSP2DEC CONVERTS C(MPAC) AND C(MPAC+1) INTO A SIGN AND 10 CHAR DECIMAL  
# STARTING IN THE LOC SPECIFIED IN DSPCOUNT.

July 12, 2016

1475

```
DSP2DEC XCH Q
 TS WDRET
 CAF ZERO
 TS CODE
 CAF THREE
 TC 11DSPIN # -R2 OFF
 CAF FOUR
 TC 11DSPIN # +R2 OFF
 TC DSPSIGN
 CAF R2D1
END2DEC TC DSPDCWD1
```

```
DSPDECVN DISPLAYS C(A) UPON ENTRY AS A 2 CHAR DECIMAL BEGINNING IN THE
DSP LOC SPECIFIED IN DSPCOUNT.
#
C(A) SHOULD BE IN FORM N X 2EXP-14. THIS IS SCALED TO FORM N/100 BEFORE
DISPLAY CONVERSION.
Page 361
```

```
DSPDECVN EXTEND
 MP VNDSPCON # MULT BY .01
 LXCH MPAC # TAKE RESULTS FROM L. (MULT BY 2EXP14).
 CAF ZERO
 TS MPAC +1
 XCH Q
 TS WDRET
 TC DSPDC2NR +3 # NO SIGN, NO ROUND, 2 CHAR

VNDSPCON OCT 00244 # .01 ROUNDED UP

GOVNUPDT TC DSPDECVN # THIS IS NOT FOR GENERAL USE. REALLY PART
 TC POSTJUMP # OF UPDATVB.
 CADR UPDAT1 +2

ENDECVN EQUALS

 SETLOC ENDSPF +1
 COUNT 41/PIN
```

```
DSPOCTWD DISPLAYS C(A) UPON ENTRY AS A 5 CHAR OCT STARTING IN THE DSP
CHAR SPECIFIED IN DSPCOUNT. IT STOPS AFTER 5 CHAR HAVE BEEN DISPLAYED.
```

```
DSPOCTWO TS CYL
 XCH Q
 TS WDRET # MUST USE SAME RETURN AS DSP2BIT.
```

|            |       |          |                                          |
|------------|-------|----------|------------------------------------------|
|            | CAF   | BIT14    | # TO BLANK SIGNS                         |
|            | ADS   | DSPCOUNT |                                          |
|            | CAF   | FOUR     |                                          |
| WDAGAIN    | TS    | WDCNT    |                                          |
|            | CS    | CYL      |                                          |
|            | CS    | CYL      |                                          |
|            | CS    | CYL      |                                          |
|            | CS    | A        |                                          |
|            | MASK  | DSPMSK   |                                          |
|            | INDEX | A        |                                          |
|            | CAF   | RELTAB   |                                          |
|            | MASK  | LOW5     |                                          |
|            | TS    | CODE     |                                          |
|            | XCH   | DSPCOUNT |                                          |
|            | TS    | COUNT    |                                          |
|            | CCS   | A        | # DECREMENT DSPCOUNT EXCEPT AT +0        |
|            | TS    | DSPCOUNT |                                          |
|            | TC    | POSTJUMP |                                          |
|            | CADR  | DSPOCTIN |                                          |
| OCTBACK    | CCS   | WDCNT    |                                          |
|            | TC    | WDAGAIN  | # +                                      |
| DSPLV      | CS    | VD1      | # TO BLOCK NUMERICAL CHARACTERS, CLEARS, |
| # Page 362 |       |          |                                          |
|            | TS    | DSPCOUNT | # AND SIGNS AFTER A COMPLETED DISPLAY.   |
|            | TC    | WDRET    |                                          |
| DSPMSK     | =     | SEVEN    |                                          |

# DSP2BIT DISPLAYS C(A) UPON ENTRY AS A 2 CHAR OCT BEGINNING IN THE DSP  
 # LOC SPECIFIED IN DSPCOUNT BY PRE CYCLING RIGHT C(A) AND USING THE LOGIC  
 # OF THE 5 CHAR OCTAL DISPLAY

|         |     |            |
|---------|-----|------------|
| DSP2BIT | TS  | CYR        |
|         | XCH | Q          |
|         | TS  | WDRET      |
|         | CAF | ONE        |
|         | TS  | WDCNT      |
|         | CS  | CYR        |
|         | CS  | CYR        |
|         | XCH | CYR        |
|         | TS  | CYL        |
|         | TC  | WDAGAIN +5 |

# FOR DSPIN PLACE 0/25 OCT INTO COUNT, 5 BIT RELAY CODE INTO CODE. BOTH  
 # ARE DESTROYED. IF BIT14 OF COUNT IS 1, SIGN IS BLANKED WITH LEFT CHAR.  
 # FOR DSPIN1 PLACE 0,1 INTO BIT11 OF CODE, 2 INTO COUNT, REL ADDRESS OF

# DSPTAB ENTRY INTO DSREL.

```

 SETLOC ENDECVN

 COUNT 40/PIN

DSPIN XCH Q # CAN'T USE L FOR RETURN, SINCE MANY OF THE
 TS DSEXIT # ROUTINES CALLING DSPIN USE L AS RETURN.
 CAF LOW5
 MASK COUNT
 TS SR
 XCH SR
 TS DSREL
 CAF BIT1
 MASK COUNT
 CCS A
 TC +2 # LEFT IF COUNT IS ODD
 TC DSPIN1 -1 # RIGHT IF COUNT IS EVEN
 XCH CODE
 TC SLEFT5 # DOES NOT USE CYL
 TS CODE
 CAF BIT14
 MASK COUNT
 CCS A
 CAF TWO # BIT14 = 1, BLANK SIGN

Page 363
 AD ONE # BIT14 = 0, LEAVE SIGN ALONE
 TS COUNT # +0 INTO COUNT FOR RIGHT
 # +1 INTO COUNT FOR LEFT (SIGN LEFT ALONE)
 # +3 INTO COUNT FOR LEFT (TO BLANK SIGN)

DSPIN1 INHINT
 INDEX DSREL
 CCS DSPTAB
 TC +2 # IF +
 TC CCSHOLE
 AD ONE # IF -
 TS DSMAG
 INDEX COUNT
 MASK DSMSK
 EXTEND
 SU CODE
 EXTEND
 BZF DSLV # SAME
DFRNT INDEX COUNT
 CS DSMSK # MASK WITH 77740, 76037, 76777, OR 74037
 MASK DSMAG

```

```

 AD CODE
 CS A
 INDEX DSREL
 XCH DSPTAB
 EXTEND
 BZMF DSLV # DSPTAB ENTRY WAS -
 INCR NOUT # DSPTAB ENTRY WAS +
DSLV RELINT
 TC DSEXIT

DSMSK OCT 37
 OCT 1740
 OCT 2000
 OCT 3740

FOR 11DSPIN, PUT REL ADDRESS OF DSPTAB ENTRY INTO A, 1 IN BIT11 OR 0 IN
BIT11 OF CODE.

11DSPIN TS DSREL
 CAF TWO
 TS COUNT
 XCH Q # MUST USE SAME RETURN AS DSPIN
 TS DSEXIT
 TC DSPIN1

DSPOCTIN TC DSPIN # SO DSPOCTWD DOESN'T USE SWCALL
 CAF +2
 TC BANKJUMP

Page 364
ENDSPOCT CADR OCTBACK

DSPALARM FINDS TC NVSUBEND IN ENTRET FOR NVSUB INITIATED ROUTINES
ABORT WITH 01501.
#
DSPALARM FINDS TC ENDOFJOB IN ENTRET FOR KEYBOARD INITIATED ROUTINES.
DC TC ENTRET.

PREDSPAL CS VD1
 TS DSPCOUNT
DSPALARM CS NVSBENDL
 AD ENTEXTIT
 EXTEND
 BZF CHARALRM +2
 CS MONADR # IF THIS IS A MONITOR, KILL IT
 AD ENTEXTIT
 EXTEND

```

July 12, 2016

1479

```

 BZF +2
 TC CHARALRM
 TC KILMONON
 TC FALTON
 TC PASTEVB # PUT MONITOR VERB BACK IN VERBREG
CHARALRM TC FALTON # NOT NVSUB INITATED TURN ON OPR ERROR
 TC ENDOFJOB
 TC POODOO
 OCT 01501
MONADR GENADR PASTEVB
NVSBENDL TC NVSUBEND
```

```

ALMCYCLE TURNS ON CHECK FAIL LIGHT, REDISPLAYS THE ORIGINAL VERB THAT
WAS EXECUTED, AND RECYCLES TO EXECUTE THE ORIGINAL VERB/NOUN COMBINATION
THAT WAS LAST EXECUTED. USED FOR BAD DATA DURING LOAD VERBS AND BY
MCTBS. ALSO BY MMCHANG IF 2 NUMERICAL CHARACTERS WERE NOT PUNCHED IN
FOR MM CODE
```

```

 SETLOC MID7 +1
 COUNT 02/PIN
```

```

ALMCYCLE TC FALTON # TURN ON CHECK FAIL LIGHT.
 CS VERBSAVE # GET ORIGINAL VERB THAT WAS EXECUTED
 TS REQRET # SET FOR ENTPASO
 TC BANKCALL # PUTS ORIGINAL VERB INTO VERBREG AND
CADR UPDATVB -1 # DISPLAYS IT IN VERB LIGHTS.
 TC POSTJUMP
ENDALM CADR ENTER
```

```

MMCHANG USES NOUN DISPLAY UNTIL ENTER. THEN IT USES MODE DISP.
IT GOES TO MODROUT WITH THE NEW M M CODE IN A, BUT NOT DISPLAYED IN
Page 365
MM LIGHTS.
#
IT DEMANDS 2 NUMERICAL CHARACTERS BE PUNCHED IN FOR NEW MM CODE.
IF NOT, IT RECYCLES.
```

```

 SETLOC DSP2BIT +10D
 COUNT 41/PIN
```

```

MMCHANG TC REQMM # ENTPASHI ASSUMES THE TC REQMM AT MMCHANG
 # IF THIS MOVES AT ALL, MUST CHANGE
 # MMADREF AT ENTPASHI.
 CAF BIT5 # OCT20 = ND2.
 AD DSPCOUNT # DSPCOUNT MUST = -ND2.
```

```

EXTEND # DEMAND THAT 2 NUM CHAR WERE PUNCHED IN.
BZF +2
TC ALMCYCLE # DSPCOUNT NOT= -ND2. ALARM AND RECYCLE.
CAF ZERO # DSPCOUNT = -ND2.
XCH NOUNREG
TS MPAC
CAF ND1
TS DSPCOUNT
TC BANKCALL
CADR 2BLANK
CS VD1 # BLOCK NUM CHAR IN
TS DSPCOUNT
CA MPAC
TC POSTJUMP
CADR MODROUTB # GO THRU STANDARD LOC.

MODROUTB = V37
REQMM CS Q
 TS REQRET
 CAF ND1
 TS DSPCOUNT
 CAF ZERO
 TS NOUNREG
 TC BANKCALL
 CADR 2BLANK
 TC FLASHON
 CAF ONE
 TS DECBRNCH # SET FOR DEC
 TC ENTEXTIT

VBRQEXEC ENTERS REQUEST TO EXEC FOR ANY ADDRESS WITH ANY PRIORITY.
IT DOES ENDOFJOB AFTER ENTERING REQUEST. DISPLAY SYST IS RELEASED.
IT ASSUMES NOUN 26 HAS BEEN PRELOADED WITH
COMPONENT 1 PRIORITY (BITS 10-14) BIT1=0 FOR NOVAC, BIT1=1 FOR FINDVAC.
COMPONENT 2 JOB ADRES (12 BIT)
Page 366
COMPONENT 3 BBCON

VBRQEXEC CAF BIT1
 MASK DSPTEM1
 CCS A
 TC SETVAC # IF BIT1 = 1, FINDVAC
 CAF TCNOVAC # IF BIT1 = 0, NOVAC
REQEX1 TS MPAC # TC NOVAC OR TC FINDVAC INTO MPAC
 CS BIT1
 MASK DSPTEM1

```



July 12, 2016

1481

```
REQUESTC TS MPAC +4 # PRIO INTO MPAC+4 AS A TEMP
 TC RELDSP
 CA ENDINST
 TS MPAC +3 # TC ENDOFJOB INTO MPAC+3
 EXTEND
 DCA DSPTM1 +1 # JOB ADRES INTO MPAC+1
 DXCH MPAC +1 # BBCON INTO MPAC+2
 CA MPAC +4 # PRIO IN A
 INHINT
 TC MPAC

SETVAC CAF TCFINDVC
 TC REQEX1
```

```
VBRQWAIT ENTERS REQUEST TO WAITLIST FOR ANY ADDRESS WITH ANY DELAY.
IT DOES ENDOFJOB AFTER ENTERING REQUEST. DISPLAY SYST IS RELEASED.
IT ASSUMES NOUN 26 HAS BEEN PRELOADED WITH
COMPONENT 1 DELAY (LOW BITS)
COMPONENT 2 TASK ADRES (12 BIT)
COMPONENT 3 BBCON
```

```
VBRQWAIT CAF TCWAIT
 TS MPAC # TC WAITLIST INTO MPAC
 CA DSPTM1 # TIME DELAY
ENDRQWT TC REQUESTC -1
```

```
REQUESTC WILL PUT TASK ADRES INTO MPAC+1, BBCON INTO MPAC+2,
TC ENDOFJOB INTO MPAC+3. IT WILL TAKE TIME DELAY OUT OF MPAC+4 AND
LEAVE IT IN A, INHINT AND TC MPAC.
```

```
SETLOC NVSBENDL +1
COUNT 40/PIN
```

```
VBPROC CAF ONE # PROCEED WITHOUT DATA
 TS LOADSTAT
 TC KILMONON # TURN ON KILL MONITOR BIT
 TC RELDSP
 TC FLASHOFF
 TC RECALTST # SEE IF THERE IS ANY RECALL FROM ENDIDLE
```

# Page 367

```
VBTERM CS ONE
 TC VBPROC +1 # TERM VERB SETS LOADSTAT NEG
```

```
PROCKEY PERFORMS THE SAME FUNCTION AS VBPROC. IT MUST BE CALLED UNDER
EXECUTIVE CONTROL, WITH CHRPRIO.
```

```

PROCKEY CAF ZERO # SET REQRET FOR ENTER PASS 0.
 TS REQRET
 CS VD1 # BLOCK NUMERICAL CHARACTERS, SIGNS, CLEAR
 TS DSPCOUNT
 TC VBPROC

```

```

VBRESEQ WAKES ENDIDLE AT SAME LINE AS FINAL ENTER OF LOAD (L+3).
(MAIN USE IS INTENDED AS RESPONSE TO INTERNALLY INITIATED FLASHING
DISPLAYS IN ENDIDLE. SHOULD NOT BE USED WITH LOAD VERBS, PLEASE PERFORM,
OR PLEASE MARK VERBS BECAUSE THEY ALREADY USE L+3 IN ANOTHER CONTEXT.)

```

```

VBRESEQ CS ZERO # MAKE IT LOOK LIKE DATA IN.
 TC VBPROC +1

```

```

FLASH IS TURNED OFF BY PROCEED WITHOUT DATA, TERMINATE, RESEQUENCE,
END OF LOAD.
Page 368

```

```

KEY RELEASE ROUTINE
#
THIS ROUTINE ALWAYS TURNS OFF THE UPACT LIGHT AND ALWAYS CLEARS DSPLOCK.
#
THE HIGHEST PRIORITY FUNCTION OF THE KEY RELEASE BUTTON IS THE
UNSUSPENDING OF A SUSPENDED MONITOR WHICH WAS EXTERNALLY INITIATED.
THIS FUNCTION IS ACCOMPLISHED BY CLEARING DSPLOCK AND TURNING OFF
THE KEY RELEASE LIGHT IF BOTH DSPLIST AND CADRSTOR ARE EMPTY.
#
IF NO SUCH MONITOR EXISTS, THEN RELDSP IS EXECUTED TO CLEAR DSPLOCK
AND THE EXTERNAL MONITOR BIT (FREEING THE DISPLAY SYSTEM FOR INTERNAL
USE), TURN OFF THE KEY RELEASE LIGHT, AND WAKE UP ANY JOB IN DSPLIST.
#
IN ADDITION IF THERE IS A JOB IN ENDIDLE, THEN CONTROL IS TRANSFERRED
TO PINBRNCH (IN DISPLAY INTERFACE ROUTINE) TO RE-EXECUTE THE SERIES OF
NVSUB CALLS ETC. THAT PRECEDED THE ENDIDLE CALL STILL AWAITING RESPONSE.
THIS FEATURE IS INTENDED FOR USE WHEN THE OPERATOR HAS BEEN REQUESTED TO
RESPOND TO SOME INTERNAL ACTION THAT USED ENDIDLE, BUT HE HAS WRITTEN
OVER THE INFORMATION ON THE DISPLAY PANEL BY SOME DISPLAYS OF HIS OWN
INITIATION WHICH DO NOT SERVE AS RESPONSES. HITTING KEYRLSE WILL
RE-ESTABLISH THE DISPLAYS TO THE STATE THEY WERE IN BEFORE HE OBSCURED
THEM, SO THAT HE CAN SEE THE WAITING REQUEST. THIS WORKS ONLY FOR
INTERNAL PROGRAMS THAT USED ENDIDLE THROUGH MARGARET'S DISPLAY
SUBROUTINES.

```

```

VBRELDSP CS BIT3
 EXTEND

```

|          |        |          |                                          |
|----------|--------|----------|------------------------------------------|
|          | WAND   | DSALMOUT | # TURN OF UPACT LITE                     |
|          | CCS    | 21/22REG | # OLD DSPLOCK                            |
|          | CAF    | BIT14    |                                          |
|          | MASK   | MONSAVE1 | # EXTERNAL MONITOR BIT (EMB)             |
|          | CCS    | A        |                                          |
|          | TC     | UNSUSPEN | # OLD DSPLOCK AND EMB BOTH 1, UNSUSPEND, |
| TSTLTS4  | TC     | RELDSP   | # NOT UNSUSPENDING EXTERNAL MONITOR      |
|          | CCS    | CADRSTOR | # RELEASE DISPLAY SYSTEM AND             |
|          | TC     | +2       | # DO RE-ESTABLISH IF CADRSTOR IS FULL.   |
|          | TC     | ENDOFJOB |                                          |
|          | TC     | POSTJUMP |                                          |
|          | CADR   | PINBRNCH |                                          |
| UNSUSPEN | CAF    | ZERO     | # EXTERNAL MONITOR IS SUSPENDED,         |
|          | TS     | DSPLOCK  | # JUST UNSUSPEND IT BY CLEARING DSPLOCK. |
|          | CCS    | CADRSTOR | # TURN KEY RELEASE LIGHT OFF IF BOTH     |
|          | TC     | ENDOFJOB | # CADRSTOR AND DSPLIST ARE EMPTY.        |
|          | TC     | RELDSP1  |                                          |
|          | TC     | ENDOFJOB |                                          |
| ENDRELDS | EQUALS |          |                                          |

# Page 369

# NVSUB IS USED FOR SUBROUTINE CALLS FROM WITHIN COMPUTER. IT CAN BE

# USED TO CALL THE COMBINATION OF ANY DISPLAY, LOAD, OR MONITOR VERB

# TOGETHER WITH ANY NOUN AVAILABLE TO THE KEYBOARD.

# PLACE OVVVVVVVNNNNNNN INTO A.

# V'S ARE THE 7-BIT VERB CODE. N'S ARE THE 7-BIT NOUN CODE.

#

# IF NVSUB IS CALLED WITH THE FOLLOWING NEGATIVE NUMBERS (RATHER THAN THE

# VERB-NOUN CODE) IN A, THEN THE DISPLAY IS BLANKED AS FOLLOWS ---

# -4 FULL BLANK, -3 LEAVE MODE, -2 LEAVE MODE AND VERB, -1 BLANK R'S ONLY.

#

# NVSUB CAN BE USED WITH MACHINE CADR TO BE SPECIFIED BY PLACING THE CADR INTO

# MPAC+2 BEFORE THE STANDARD NVSUB CALL.

#

# NVSUB RETURNS TO 2+ CALLING LOC AFTER PERFORMING TASK, IF DISPLAY

# SYSTEM IS AVAILABLE. THE NEW NOUN AND VERB CODES ARE DISPLAYED.

# IF V'S =0, THE NEW NOUN CODE IS DISPLAYED ONLY (RETURN WITH NO FURTHER

# ACTION). IF N'S =0, THE NEW VERB CODE IS DISPLAYED ONLY (RETURN WITH NO

# FURTHER ACTION).

#

# IT RETURNS TO 1+ CALLING LOC WITHOUT PERFORMING TASK, IF DISPLAY

# SYSTEM IS BLOCKED (NOTHING IS DISPLAYED IN THIS CASE).

# IT DOES TC ABORT (WITH OCT 01501) IF IT ENCOUNTERS A DISPLAY PROGRAM

# ALARM CONDITION BEFORE RETURN TO CALLER.

#

```

THE DISPLAY SYSTEM IS BLOCKED BY THE DEPRESSION OF ANY
KEY, EXCEPT ERROR LIGHT RESET.
IT IS RELEASED BY THE KEY RELEASE BUTTON, ALL EXTENDED VERBS,
PROCED WITOHOUT DATA, TERMINATE, RESEQUENCE, INITIALIZE EXECUTIVE,
RECALL PART OF RECALTST IF ENDIDLE WAS USED,
VB = REQUEST EXECUTIVE, VB = REQUEST WAITLIST,
MONITOR SET UP.
#
THE DISPLAY SYSTEM IS ALSO BLOCKED BY THE EXTERNAL MONITOR BIT, WHICH
INDICATES AND EXTERNALLY INITIATED MONITOR IS RUNNING (SEE MONITOR).
#
A NVSUB CALL THAT PASSES DSPLOCK AND THE EXTERNAL MONITOR BIT ENDS OLD
MONITOR.
#
DSPLOCK IS THE INTERLOCK FOR USE OF KEYBOARD AND DISPLAY SYSTEM WHICH
LOCKS OUT INTERNAL USE WHENEVER THERE IS EXTERNAL KEYBOARD ACTION.
#
NVSUB SHOULD BE USED TWICE IN SUCCESSION FOR 'PLEASE PERFORM' SITUATIONS
(SIMILARLY FOR PLEASE MARK). FIRST PLACE THE CODED NUMBER FOR WHAT
ACTION IS DESIRED OF OPERATOR INTO THEREGISTERS REFERRED TO BY THE
'CHECKLIST' NOUN. GO TO NVSUB WITH A DISPLAY VERB AND THE 'CHECKLIST'
NOUN. GO TO NVSUB AGAIN WTIH THE 'PLEASE PERFORM' VERB AND ZEROS IN THE
LOW 7 BITS. THIS 'PASTES UP' THE 'PLEASE PERFORM' VERB INTO THE VERB
LIGHTS.
#
NVMONOPT IS AN ENTRY SIMILAR TO NVSUB, BUT REQUIRING AN ADDITIONAL
Page 370
PARAMETER IN L. IT SHOULD BE USED ONLY WITH A MONITOR VERB-NOUN CODE IN
A. AFTER EACH MONITOR DISPLAY A *PLEASE* VERB WILL BE PASED INT THE VERB
LIGHTS OR DATA WILL BE BLANKED (OR BOTH) ACCORDING TO THE OPTIONS
SPECIFIED IN L. IF BITS 8-14 OF L ARE OTHER THAN ZERO, THEN THEY WILL
BE INTERPRETED AS A VERB CODE AND PASTED IN THE VERB LIGHTS. (THIS VERB
CODE SHOULD DESIGNATE ONE OF THE *PLEASE* VERBS.) IF BITS 1-3 OF L ARE
OTHER THAN ZERO, THEN THEY WILL BE USED BO BLANK DATA BY BEING FED TO
BLANKSUB. IF NVMONOPT IS USED WITH A VERB OTHER THAN A MONITOR VERB,
THE PARAMETER IN L HAS NO EFFECT.
#
NVSUB IN FIXED-FIXED PLACES 2+CALLING LOC INTO NVQTEM, TC NVSUBEND INTO
ENTRET. (THIS WILL RESTORE OLD CALLING BANK BITS)

```

SETLOC ENDALM +1

COUNT 02/PIN

|          |      |        |                         |
|----------|------|--------|-------------------------|
| NVSUB    | LXCH | 7      | # ZERO NVMONOPT OPTIONS |
| NVMONOPT | TS   | NVTEMP |                         |

July 12, 2016

1485

|          |        |            |                                           |
|----------|--------|------------|-------------------------------------------|
|          | CAF    | BIT14      |                                           |
|          | MASK   | MONSAVE1   | # EXTERNAL MONITOR BIT                    |
|          | AD     | DSPLOCK    |                                           |
|          | CCS    | A          |                                           |
|          | TC     | Q          | # DSP SYST BLOCKED. RET TO 1. CALLING LOC |
|          | CAF    | ONE        | # DSP SYST AVAILABLE.                     |
| NVSBCOM  | AD     | Q          |                                           |
|          | TS     | NVQTEM     | # 2+ CALLING LOC INTO NVQTEM              |
|          | LXCH   | MONSAVE2   | # STORE NVMONOPT OPTIONS                  |
|          | TC     | KILMONON   | # TURN ON KILL MONITOR BIT                |
| NVSUBCOM | CAF    | NVSBBBNK   |                                           |
|          | XCH    | BBANK      |                                           |
|          | EXTEND |            | # SAVE OLD SUPERBITS                      |
|          | ROR    | SUPERBNK   |                                           |
|          | TS     | NVBNKTEM   |                                           |
|          | CAF    | PINSUPBT   |                                           |
|          | EXTEND |            |                                           |
|          | WRITE  | SUPERBNK   |                                           |
|          | TC     | NVSUBB     | # GO TO NVSUB1 THRU STANDARD LOC          |
|          | EBANK= | DSPCOUNT   |                                           |
| NVSBBBNK | BBCON  | NVSUB1     |                                           |
| PINSUPBT | =      | NVSBBBNK   | # CONTAINS THE PINBALL SUPERBIT.          |
| NVSUBEND | DXCH   | NVQTEM     | # NVBNKTEM MUST = NVQTEM+1                |
|          | TC     | SUPDXCHZ   | # DTCB WITH SUPERBIT SWITCHING            |
|          | SETLOC | ENDRQWT +1 |                                           |
|          | COUNT  | 41/PIN     |                                           |

# Page 371

# BLANKDSP BLANKS DISPLAY ACCORDING TO OPTION NUMBER IN NVTEMP AS FOLLOWS:  
# -4 FULL BLANK, -3 LEAVE MODE, -2 LEAVE MODE AND VERB, -1 BLANK R'S ONLY.

|          |        |        |                                   |
|----------|--------|--------|-----------------------------------|
| BLANKDSP | AD     | SEVEN  | # 7,8,9, OR 10 (A HAD 0,1,2,OR 3) |
|          | INHINT |        |                                   |
|          | TS     | CODE   | # BLANK SPECIFIED DSPTABS         |
|          | CS     | BIT12  |                                   |
|          | INDEX  | CODE   |                                   |
|          | XCH    | DSPTAB |                                   |
|          | CCS    | A      |                                   |
|          | INCR   | NOUT   |                                   |
|          | TC     | +1     |                                   |
|          | CCS    | CODE   |                                   |

|            |        |             |                                             |
|------------|--------|-------------|---------------------------------------------|
|            | TC     | BLANKDSP +2 |                                             |
|            | RELINT |             |                                             |
|            | INDEX  | NVTEMP      |                                             |
|            | TC     | +5          |                                             |
|            | TC     | +1          | # NVTEMP HAS -4 (NEVER TOUCH MODREG)        |
|            | TS     | VERBREG     | # -3                                        |
|            | TS     | NOUNREG     | # -2                                        |
|            | TS     | CLPASS      | # -1                                        |
|            | CS     | VD1         |                                             |
|            | TS     | DSPCOUNT    |                                             |
|            | TC     | FLASHOFF    | # PROTECT AGAINST INVISIBLE FLASH           |
|            | TC     | ENTSET -2   | # ZEROS REQRET                              |
| NVSUB1     | CAF    | ENTSET      | # IN BANK                                   |
|            | TS     | ENTRET      | # SET RETURN TO NVSUBEND                    |
|            | CCS    | NVTEMP      | # WHAT NOW                                  |
|            | TC     | +4          | # NORMAL NVSUB CALL (EXECUTE VN OR PASTE)   |
|            | TC     | GODSPALM    |                                             |
|            | TC     | BLANKDSP    | # BLANK DISPLAY AS SPECIFIED                |
|            | TC     | GODSPALM    |                                             |
|            | CAF    | LOW7        |                                             |
|            | MASK   | NVTEMP      |                                             |
|            | TS     | MPAC +3     | # TEMP FOR NOUN (CAN'T USE MPAC. DSPDECVN   |
|            | CA     | NVTEMP      | # USES MPAC, +1, +2).                       |
|            | TS     | EDOP        | # RIGHT 7                                   |
|            | CA     | EDOP        |                                             |
|            | TS     | MPAC +4     | # TEMP FOR VERB (CAN'T USE MPAC+1. DSPDECVN |
|            |        |             | # USES MPAC, +1, +2)                        |
|            | CCS    | MPAC +3     | # TEST NOUN                                 |
|            | TC     | NVSUB2      | # IF NOUN NOT +0, GO ON                     |
|            | CA     | MPAC +4     |                                             |
|            | TC     | UPDATVB -1  | # IF NOUN = +0, DISPLAY VERB. THEN RETURN   |
|            | CAF    | ZERO        | # ZERO REQRET SO THAT PASTED VERBS CAN      |
|            | TS     | REQRET      | # BE EXECUTED BY OPERATOR.                  |
| ENTSET     | TC     | NVSUBEND    |                                             |
| NVSUB2     | CCS    | MPAC +4     | # TEST VERB                                 |
|            | TC     | +4          | # IF VERB NOT +0, GO ON                     |
| # Page 372 | CA     | MPAC +3     |                                             |
|            | TC     | UPDATNN -1  | # IF VERB = +0, DISPLAY NOUN. THEN RETURN   |
|            | TC     | NVSUBEND    |                                             |
|            | CA     | MPAC +2     | # TEMP FOR MACH CADR TO BE SPEC. (DSPDECVN  |
|            | TS     | MPAC +5     | # USES MPAC, +1, +2)                        |
|            | CA     | MPAC +4     |                                             |
|            | TC     | UPDATVB -1  | # IF BOTH NOUN AND VERB NOT +0, DISPLAY     |
|            | CA     | MPAC +3     | # BOTH AND GO TO ENTPAS0.                   |

July 12, 2016

1487

```
TC UPDATNN -1
CAF ZERO
TS LOADSTAT # SET FOR WAITING FOR DATA CONDITION
TS CLPASS
TS REQRET # SET REQRET FOR PASS 0.
CA MPAC +5 # RESTORES MACH CADR TO BE SPEC TO MPAC+2
TS MPAC +2 # FOR USE IN INTMCTBS (IN ENTPASO).
ENDNVSBI TC ENTPASO
```

```
IF INTERNAL MACH CADR TO BE SPECIFIED, MPAC+2 WILL BE PLACED INTO
NOUNCADR IN ENTPASO (INTMCTBS).
```

```
SETLOC NVSUBEND +2
COUNT 02/PIN
KILMONON CAF BIT15 # FORCE BIT 15 OF MONSAVE1 TO .
TS MONSAVE1 # THIS IS THE KILL MONITOR BIT.
TC Q # TURN OFF BIT 14, THE EXTERNAL
 # MONITOR BIT.
```

```
LOADSTAT +0 INACTIVE (WAITING FOR DATA). SET BY NVSUB
+1 PROCEED NO DATA. SET BY SPECIAL VERB
-1 TERMINATE. SET BY SPECIAL VERB.
-0 DATA IN. SET BY END OF LOAD ROUTINE.
OR RESEQUENCE. SET BY VERB 32
```

```
L TO ENDIDLE (FIXED FIXED).
```

```
ROUTINES THAT REQUEST LOADS THROUGH NVSUB SHOULD USE ENDIDLE WHILE
WAITING FOR THE DATA TO BE LOADED. ENDIDLE PUTS CURRENT JOB TO SLEEP.
ENDIDLE CANNOT BE CALLED FROM ERASABLE OR F/F MEMORY,
SINCE JOB SLEEP AND JOBWAKE CAN HANDLE ONLY FIXED BANKS.
RECALSTST TESTS LOADSTAT AND WAKES JOB UP TO,
```

```
L+1 FOR TERMINATE
L+2 FOR PROCEED WITHOUT DATA
L+3 FOR DATA IN, OR RESEQUENCE
```

```
IT DOES NOTHING IF LOADSTAT INDICATES WAITING FOR DTA.
```

```
Page 373
```

```
#
ENDIDLE ABORTS (WITH CODE 1206) IF A SECOND JOB ATTEMPTS TO GO TO SLEEP
IN PINBALL. IN PARTICULAR, IF AN ATTEMPT IS MADE TO GO TO ENDIDLE WHEN
1) CADRSTOR NOT= +0. THIS IS THE CASE WHERE THE CAPACITY OF ENDIDLE IS
EXCEEDED. (+-NZ INDICATES A JOB IS ALREADY ASLEEP DUE TO ENDIDLE.)
2) DSPLIST NOT= +0. THIS INDICATES A JOB IS ALREADY ASLEEP DUE TO
NVSUBUSY.
```

```
ENDIDLE LXCH Q # RETURN ADDRESS INTO L.
```

|                                                                       |        |          |                                            |
|-----------------------------------------------------------------------|--------|----------|--------------------------------------------|
|                                                                       | TC     | ISCADR+0 | # ABORT IF CADRSTOR NOT= +0.               |
|                                                                       | TC     | ISLIST+0 | # ABORT IF DSPLIST NOT= +0                 |
|                                                                       | CA     | L        | # DON'T SET DSPLOC TO 1 SO CAN USE         |
|                                                                       | MASK   | LOW10    | # ENDIDLE WITH NVSUB INITIATE MONITOR.     |
|                                                                       | AD     | FBANK    | # SAME STRATEGY FOR CADR AS MAKECADR.      |
|                                                                       | TS     | CADRSTOR |                                            |
|                                                                       | TC     | JOBSLEEP |                                            |
| ENDINST                                                               | TC     | ENDOFJOB |                                            |
| ISCADR+0                                                              | CCS    | CADRSTOR | # ABORTS (CODE 01206) IF CADRSTOR NOT= +0. |
|                                                                       | TC     | DSPABORT | # RETURNS IF CADRSTOR = +0.                |
|                                                                       | TC     | Q        |                                            |
|                                                                       | TC     | DSPABORT |                                            |
| ISLIST+0                                                              | CCS    | DSPLIST  | # ABORTS (CODE 01206) IF DSPLIST NOT= +0.  |
|                                                                       | TC     | DSPABORT | # RETURNS IF DSPLIST = +0.                 |
|                                                                       | TC     | Q        |                                            |
| DSPABORT                                                              | TC     | P00D00   |                                            |
|                                                                       | OCT    | 01206    |                                            |
| # JAMTERM ALLOWS PROGRAMS TO PERFORM THE TERMINATE FUNCTION.          |        |          |                                            |
| # IT DOES ENDOFJOB.                                                   |        |          |                                            |
| JAMTERM                                                               | CAF    | PINSUPBT |                                            |
|                                                                       | EXTEND |          |                                            |
|                                                                       | WRITE  | SUPERBNK |                                            |
|                                                                       | CAF    | 34DEC    |                                            |
|                                                                       | TS     | REQRET   | # LEAVE ENTER SET FOR ENTPASSO.            |
|                                                                       | CS     | VD1      |                                            |
|                                                                       | TS     | DSPCOUNT |                                            |
|                                                                       | TC     | POSTJUMP |                                            |
|                                                                       | CADR   | VBTERM   |                                            |
| 34DEC                                                                 | DEC    | 34       |                                            |
| # JAMPROC ALLOWS PROGRAMS TO PERFORM THE PROCEED/PROCEED WITHOUT DATA |        |          |                                            |
| # FUNCTION. IT DOES ENDOFJOB.                                         |        |          |                                            |
| # Page 374                                                            |        |          |                                            |
| JAMPROC                                                               | CAF    | PINSUPBT |                                            |
|                                                                       | EXTEND |          |                                            |
|                                                                       | WRITE  | SUPERBNK |                                            |
|                                                                       | CAF    | 33DEC    |                                            |
|                                                                       | TS     | REQRET   | # LEAVE ENTER SET FOR ENTPASSO.            |
|                                                                       | CS     | VD1      |                                            |



|      |          |
|------|----------|
| TS   | DSPCOUNT |
| TC   | POSTJUMP |
| CADR | VBPROC   |

|       |     |    |
|-------|-----|----|
| 33DEC | DEC | 33 |
|-------|-----|----|

```
BLANKSUB BLANKS ANY COMBINATION OF R1, R2, R3.
CALL WITH BLANKING CODE IN A.
BIT1=1 BLANKS R1, BIT2=1 BLANKS R2, BIT3=1 BLANKS R3.
ANY COMBINATION OF THESE BITS IS ACCEPTED.
#
DSPCOUNT IS RESTORED TO STATE IT WAS IN BEFORE BLANKSUB WAS EXECUTED.
```

|          |        |          |                                           |
|----------|--------|----------|-------------------------------------------|
| BLANKSUB | MASK   | SEVEN    |                                           |
|          | TS     | NVTEMP   | # STORE BLANKING CODE IN NVTEMP.          |
|          | CAF    | BIT14    |                                           |
|          | MASK   | MONSAVE1 | # EXTERNAL MONITOR BIT                    |
|          | AD     | DSPLOCK  |                                           |
|          | CCS    | A        |                                           |
|          | TC     | Q        | # DSP SYST BLOCKED. RET TO 1+ CALLING LOC |
|          | INCR   | Q        | # DSP SYST AVAILABLE                      |
|          |        |          | # SET RETURN FOR 2+ CALLING LOC           |
|          | CCS    | NVTEMP   |                                           |
|          | TCF    | +2       |                                           |
|          | TC     | Q        | # NOTHING TO BLANK. RET TO 2+ CALLING LOC |
|          | LXCH   | Q        | # SET RETURN FOR 2 + CALLING LOC          |
|          | CAF    | BLNKBBNK |                                           |
|          | XCH    | BBANK    |                                           |
|          | EXTEND |          |                                           |
|          | ROR    | SUPERBNK | # SAVE OLD SUPERBITS.                     |
|          | DXCH   | BUF      |                                           |
|          | CAF    | PINSUPBT |                                           |
|          | EXTEND |          |                                           |
|          | WRITE  | SUPERBNK |                                           |
|          | TC     | BLNKSUB1 |                                           |
|          | EBANK= | DSPCOUNT |                                           |
| BLNKBBNK | BBCON  | BLNKSUB1 |                                           |
| ENDBLFF  | EQUALS |          |                                           |
|          | SETLOC | ENDRELDS |                                           |
|          | COUNT  | 40/PIN   |                                           |

# Page 375

|          |    |          |                                           |
|----------|----|----------|-------------------------------------------|
| BLNKSUB1 | CA | DSPCOUNT | # SAVE OLD DSPCOUNT FOR LATER RESTORATION |
|          | TS | BUF +2   |                                           |

```

CAF BIT1 # TEST BIT1. SEE IF R1 TO BE BLANKED.
TC TESTBIT
CAF R1D1
TC 5BLANK -1
CAF BIT2 # TEST BIT2. SEE IF R2 TO BE BLANKED.
TC TESTBIT
CAF R2D1
TC 5BLANK -1
CAF BIT3 # TEST BIT3. SEE IF R3 TO BE BLANKED.
TC TESTBIT
CAF R3D1
TC 5BLANK -1
CA BUF +2 # RESTORE DSPCOUNT TO STATE IT HAD
TS DSPCOUNT # BEFORE BLANKSUB.
DXCH BUF # CALL L+2 DIRECTORY.
TC SUPDXCHZ +1 # DTCB WITH SUPERBIT SWITCHING

TESTBIT MASK NVTEMP # NVTEMP CONTAINS BLANKING CODE
 CCS A
 TC Q # IF CURRENT BIT = 1, RETURN TO L+1.
 INDEX Q # IF CURRENT BIT = 0, RETURN TO L+3.
 TC 2

ENDBSUB1 EQUALS

DSPMM DOES NOT DISPLAY MODREG DIRECTLY. IT PUTS IN EXEC REQUEST WITH
PRIO 30000 FOR DSPMMJB AND RETURNS TO CALLER.
#
IF MODREG CONTAINS -0, DSPMMJB BLANKS THE MODE LIGHTS.
#
DSPMM MUST BE IN BANK 27 OR LOWER, SO IT CAN BE CALLED VIA BANKCALL.

 BANK 7
 SETLOC PINBALL4
 BANK

 COUNT 07/PIN

DSPMM XCH Q
 TS MPAC
 INHINT
 CAF CHRPRIO
 TC NOVAC
 EBANK= DSPCOUNT
 2CADR DSPMMJB

```

```

RELINT
Page 376
ENDSPMM TC MPAC

DSPMM PLACE MAJOR MODE CODE INTO MODREG

 SETLOC ENDBSUB1

 COUNT 40/PIN

DSPMMJB CAF MD1 # GETS HERE THRU DSPMM
 XCH DSPCOUNT
 TS DSPMMTEM # SAVE DSPCOUNT
 CCS MODREG
 AD ONE
 TC DSPDECVN # IF MODREG IS + OR +0, DISPLAY MODREG
 TC +2 # IF MODREG IS -NZ, DO NOTHING
 TC 2BLANK # IF MODREG IS -0, BLANK MM
 XCH DSPMMTEM # RESTORE DSPCOUNT
 TS DSPCOUNT
 TC ENDOFJOB

RECALTST IS ENTERED DIRECTLY AFTER DATA IS LOADED (OR RESEQUENCE VERB IS
EXECUTED), TERMINATE VERB IS EXECUTED, OR PROCEED WITHOUT DATA VERB IS
EXECUTED. IT WAKES UP JOB THAT DID TC ENDIDLE.
#
IF CADRSTOR NOT= +0, IT PUTS +0 INTO DSPLOCK, AND TURNS OFF KEY RLSE
LIGHT IF DSPLIST IS EMPTY (LEAVES KEY RLSE LIGHT ALONE IF NOT EMPTY).

RECALTST CCS CADRSTOR
 TC RECAL1
 TC ENDOFJOB # NORMAL EXIT IF KEYBOARD INITIATED
RECAL1 CAF ZERO
 XCH CADRSTOR
 INHINT
 TC JOBWAKE
 CCS LOADSTAT
 TC DOPROC # + PROCEED WITHOUT DATA
 TC ENDOFJOB # PATHOLOGICAL CASE EXIT
 TC DOTERM # - TERMINATE
 CAF TWO # -0 DATA IN OR RESEQUENCE
RECAL2 INDEX LOCCTR
 AD LOC # LOC IS + FOR BASIC JOBS
 INDEX LOCCTR
 TS LOC
 CA NOUNREG # SAVE VERB IN MPAC, NOUN IN MPAC+1 AT

```

|                                                 |                                   |                                          |                                          |
|-------------------------------------------------|-----------------------------------|------------------------------------------|------------------------------------------|
|                                                 | TS                                | L                                        | # TIME OF RESPONSE TO ENDIDLE FOR        |
|                                                 | CA                                | VERBREG                                  | # POSSIBLE LATER TESTING BY JOB THAT HAS |
|                                                 | INDEX                             | LOCCTR                                   | # BEEN WAKED UP.                         |
|                                                 | DXCH                              | MPAC                                     |                                          |
|                                                 | RELINT                            |                                          |                                          |
| # Page 377                                      |                                   |                                          |                                          |
| RECAL3                                          | TC                                | RELDSP                                   |                                          |
|                                                 | TC                                | ENDOFJOB                                 |                                          |
| DOTERM                                          | CAF                               | ZERO                                     |                                          |
|                                                 | TC                                | RECAL2                                   |                                          |
| DOPROC                                          | CAF                               | ONE                                      |                                          |
|                                                 | TC                                | RECAL2                                   |                                          |
|                                                 | SBANK=                            | LOWSUPER                                 |                                          |
| # Page 378                                      |                                   |                                          |                                          |
| # MISCELLANEOUS SERVICE ROUTINES IN FIXED/FIXED |                                   |                                          |                                          |
|                                                 | SETLOC                            | ENDBLFF                                  |                                          |
|                                                 | COUNT                             | 02/PIN                                   |                                          |
| # SETNCADR                                      | E CADR ARRIVES IN A.              | IT IS STORED IN NOUNCADR.                | EBANK BITS                               |
| #                                               | ARE SET.                          | E ADRES IS DERIVED AND PUT INTO NOUNADD. |                                          |
| SETNCADR                                        | TS                                | NOUNCADR                                 | # STORE ECADR                            |
|                                                 | TS                                | EBANK                                    | # SET EBANK BITS                         |
|                                                 | MASK                              | LOW8                                     |                                          |
|                                                 | AD                                | OCT1400                                  |                                          |
|                                                 | TS                                | NOUNADD                                  | # PUT E ADRES INTO NOUNADD               |
|                                                 | TC                                | Q                                        |                                          |
| # SETNADD                                       | GETS E CADR FROM NOUNCADR,        | SETS EBANK BITS,                         | DERIVES                                  |
| #                                               | E ADRES AND PUTS IT INTO NOUNADD. |                                          |                                          |
| SETNADD                                         | CA                                | NOUNCADR                                 |                                          |
|                                                 | TCF                               | SETNCADR +1                              |                                          |
| # SETEBANK                                      | E CADR ARRIVES IN A.              | EBANK BITS ARE SET.                      | E ADRES IS                               |
| #                                               | DERIVED AND LEFT IN A.            |                                          |                                          |
| SETEBANK                                        | TS                                | EBANK                                    | # SET EBANK BITS                         |
|                                                 | MASK                              | LOW8                                     |                                          |
|                                                 | AD                                | OCT1400                                  | # E ADRES LEFT IN A                      |
|                                                 | TC                                | Q                                        |                                          |

July 12, 2016

1493

|            |        |          |                                          |
|------------|--------|----------|------------------------------------------|
| R1D1       | OCT    | 16       | # THESE 3 CONSTANTS FORM A PACKED TABLE. |
| R2D1       | OCT    | 11       | # DON'T SEPARATE.                        |
| R3D1       | OCT    | 4        |                                          |
| RIGHT5     | TS     | CYR      |                                          |
|            | CS     | CYR      |                                          |
|            | CS     | CYR      |                                          |
|            | CS     | CYR      |                                          |
|            | CS     | CYR      |                                          |
|            | XCH    | CYR      |                                          |
|            | TC     | Q        |                                          |
| LEFT5      | TS     | CYL      |                                          |
|            | CS     | CYL      |                                          |
|            | CS     | CYL      |                                          |
|            | CS     | CYL      |                                          |
| # Page 379 | CS     | CYL      |                                          |
|            | XCH    | CYL      |                                          |
|            | TC     | Q        |                                          |
| SLEFT5     | DOUBLE |          |                                          |
|            | DOUBLE |          |                                          |
|            | DOUBLE |          |                                          |
|            | DOUBLE |          |                                          |
|            | DOUBLE |          |                                          |
|            | TC     | Q        |                                          |
| LOW5       | OCT    | 37       | # THESE 3 CONSTANTS FORM A PACKED TABLE. |
| MID5       | OCT    | 1740     | # DON'T SEPARATE.                        |
| HI5        | OCT    | 76000    | # MUST STAY HERE                         |
| TCNOVAC    | TC     | NOVAC    |                                          |
| TCWAIT     | TC     | WAITLIST |                                          |
| TCTSKOVR   | TC     | TASKOVER |                                          |
| TCFINDVC   | TC     | FINDVAC  |                                          |
| CHRPRI0    | OCT    | 30000    | # EXEC PRIORITY OF CHARIN                |
| LOW11      | OCT    | 3777     |                                          |
| B12-1      | EQUALS | LOW11    |                                          |
| LOW8       | OCT    | 377      |                                          |
| VD1        | OCT    | 23       | # THESE 3 CONSTANTS FORM A PACKED TABLE. |
| ND1        | OCT    | 21       | # DON'T SEPARATE.                        |

1494

July 12, 2016

|                                                                          |        |          |                                         |
|--------------------------------------------------------------------------|--------|----------|-----------------------------------------|
| MD1                                                                      | OCT    | 25       |                                         |
| BINCON                                                                   | DEC    | 10       |                                         |
| FALTON                                                                   | CA     | BIT7     | # TURN ON OPERATOR ERROR LIGHT.         |
|                                                                          | EXTEND |          |                                         |
|                                                                          | WOR    | DSALMOUT | # BIT 7 OF CHANNEL 11                   |
|                                                                          | TC     | Q        |                                         |
| FALTOF                                                                   | CS     | BIT7     | # TURN OFF OPERATOR ERROR LIGHT         |
|                                                                          | EXTEND |          |                                         |
|                                                                          | WAND   | DSALMOUT | # BIT 7 OF CHANNEL 11                   |
|                                                                          | TC     | Q        |                                         |
| RELDSPON                                                                 | CAF    | BIT5     | # TURN ON KEY RELEASE LIGHT             |
|                                                                          | EXTEND |          |                                         |
|                                                                          | WOR    | DSALMOUT | # BIT 5 OF CHANNEL 11                   |
|                                                                          | TC     | Q        |                                         |
| # Page 380                                                               |        |          |                                         |
| LODSAMPT                                                                 | EXTEND |          |                                         |
|                                                                          | DCA    | TIME2    |                                         |
|                                                                          | DXCH   | SAMPTIME |                                         |
|                                                                          | TC     | Q        |                                         |
| TPSL1                                                                    | EXTEND |          | # SHIFTS MPAC, +1, +2 LEFT 1            |
|                                                                          | DCA    | MPAC +1  | # LEAVES OVFINDD SET TO +/- 1 FOR OF/UF |
|                                                                          | DAS    | MPAC +1  |                                         |
|                                                                          | AD     | MPAC     |                                         |
|                                                                          | ADS    | MPAC     |                                         |
|                                                                          | TS     | 7        | # TS A DOES NOT CHANGE A ON OF/UF.      |
|                                                                          | TC     | Q        | # NO NET OF/UF                          |
|                                                                          | TS     | MPAC+6   | # MPAC +6 SET TO +/- 1 FOR OF/UF        |
|                                                                          | TC     | Q        |                                         |
| # IF MPAC, +1 ARE EACH +NZ OR +0 AND C(A)=-0, SHORTMP WRONGLY GIVES +0.  |        |          |                                         |
| # IF MPAC, +1 ARE EACH -NZ OR -0 AND C(A)=+0, SHORTMP WRONGLY GIVES +0.  |        |          |                                         |
| # PRSHRTMP FIXES FIRST CASE ONLY, BY MERELY TESTING C(A) AND IF IT = -0, |        |          |                                         |
| # SETTING RESULT TO -0.                                                  |        |          |                                         |
| #                                                                        |        |          |                                         |
| # (DO NOT USE PRSHRTMP UNLESS MPAC, +1 ARE EACH +NZ OR +0, AS THEY ARE   |        |          |                                         |
| # WHEN THEY CONTAIN THE SF CONSTANTS.)                                   |        |          |                                         |
| PRSHRTMP                                                                 | TS     | MPTMP    |                                         |
|                                                                          | CCS    | A        |                                         |
|                                                                          | CA     | MPTMP    | # C(A) +, DO REGULAR SHORTMP            |

July 12, 2016

1495

```
TCF SHORTMP +1 # C(A) +0, DO REGULAR SHORTMP
TCF -2 # C(A) -, DO REGULAR SHORTMP
CS ZERO # C(A) -0, FORCE RESULT TO -0 AND RETURN.
TS MPAC
TS MPAC +1
TS MPAC +2
TC Q

FLASHON CAF BIT6 # TURN ON V/N FLASH
 EXTEND # BIT 6 OF CHANNEL 11
 WOR DSALMOUT
 TC Q

FLASHOFF CS BIT6 # TURN OFF V/N FLASH
 EXTEND
 WAND DSALMOUT # BIT 6 OF CHANNEL 11
 TC Q
```

# Page 381

# INTERNAL USE OF KEYBOARD AND DISPLAY PROGRAM.

#

# USER MUST SCHEDULE CALLS TO NVSUB SO THAT THERE IS NO CONFLICT OF USE OR  
# CONFUSION TO OPERATOR. THE OLD CRABLOCK (INTERNAL/INTERNAL INTERLOCK)  
# HAS BEEN REMOVED AND THE INTERNAL USER NO LONGER HAS THE PROTECTION THIS  
# OFFERED.

#

# THERE ARE TWO WAYS A JOB CAN BE PUT TO SLEEP BY THE KEYBOARD + DISPLAY  
# PROGRAM. 1) BY ENDIDLE

# 2) BY NVSUBUSY

# THE BASIC CONVENTION IS THAT ONLY ONE JOB WILL BE PERMITTED ASLEEP VIA  
# THE KEYBOARD + DISPLAY PROGRAM AT A TIME. IF A JOB ATTEMPTS TO GO TO  
# SLEEP BY MEANS OF (1) OR (2) AND THERE IS ALREADY A JOB ASLEEP THAT WAS  
# PUT TO SLEEP BY (1) OR (2), THEN AN ABORT IS CAUSED.

#

# THE CALLING SEQUENCE FOR NVSUB IS

```
CAF V/N
L TC NVSUB
L+1 RETURN HERE IF OPERATOR HAS INTERVENED
L+2 RETURN HERE AFTER EXECUTION
```

#

# A ROUTINE CALLED NVSUBUSY IS PROVIDED (USE IS OPTIONAL) TO PUT  
# YOUR JOB TO SLEEP UNTIL THE OPERATOR RELEASES THE KEYBOARD + DISPLAY  
# SYSTEM. NVSUBUSY ALSO TURNS ON THE KEY RELEASE LIGHT.  
# NVSUBUSY CANNOT BE CALLED FROM ERASABLE OR F/F MEMORY,  
# SINCE JOBSLEEP AND JOBWAKE CAN HANDLE ONLY FIXED BANKS.

#

```

THE CALLING SEQUENCE IS
CAF WAKEFCADR
TC NVSUBUSY
.
#
NVSUBUSY IS INTENDED FOR USE WHEN AN INTERNAL PROGRAM FINDS THE OPERATOR
IS NOT USING THE KEYBOARD + DISPLAY PROGRAM (BY HIS OWN INITIATION). IT IS
NOT INTENDED FOR USE WHEN ONE INTERNAL PROGRAM FINDS ANOTHER INTERNAL
PROGRAM USING THE KEYBOARD + DISPLAY PROGRAM.
#
NVSUBUSY ABORTS (WITH CODE 01206) IF A SECOND JOB ATTEMPTS TO GO TO
SLEEP IN PINBALL. IN PARTICULAR, IF AN ATTEMPT IS MAKE TO GO TO NVSUBUSY
WHEN
#
1) DSPLIST NOT= +0. THIS IS THE CASE WHERE THE CAPACITY FO THE DSPLIST
IS EXCEEDED.
2) CADRSTOR NOT= +0. THIS INDICATES THAT A JOB IS ALREADY USING
Page 382
ENDIDLE. (+-NZ INDICATES A JOB IS ALREADY ASLEEP DUE TO ENDIDLE.)

PRENVBSY CS 2K+3 # SPECIAL ENTRANCE FOR ROUTINES IN FIXED
 AD Q # BANKS ONLY DESIRING THE FCADR OF (LOC
 AD FBANK # FROM WHICH THE TC PRENVBSY WAS DONE) -2
NVSUBUSY TC POSTJUMP # TO BE ENTERED.
 CADR NVSUBSY1
2K+3 OCT 2003

NVSUBSY1 MUST BE IN BANK 27 OR LOWER, SO IT WILL PUT CALLER TO SLEEP
WITH HIS PROPER SUPERBITS.

 SETLOC ENDSPMM +1
 COUNT 07/PIN

NVSUBSY1 TS L
 TC ISCADR+0 # ABORT IF CADRSTOR NOT= +0.
 TC ISLIST+0 # ABORT IF DSPLIST NOT= +0.
 TC RELDSPON
 CA L
 TS DSPLIST
ENDNVBSY TC JOBSLEEP

NVSBWAIT IS A SPECIAL ENTRANCE FOR ROUTINES IN FIXED BANKS ONLY. IF
SYSTEM IS NOT BUSY, IT EXECUTES V/N AND RETURNS TO L+1 (L= LOC FROM
WHICH THE TC NVSBWAIT WAS DONE). IF SYSTEM IS BUSY, IT PUTS CALLING JOB
TO SLEEP WITH L-1 GOING INTO LIST FOR EVENTUAL WAKING UP WHEN SYSTEM
IS NOT BUSY.

```



```

SETLOC NVSUBUSY +3
COUNT 02/PIN

NVSBWAIT LXCH 7 # ZERO NVMONOPT OPTIONS
TS NVTEMP
CAF BIT14
MASK MONSAVE1 # EXTERNAL MONITOR BIT
AD DSPLOCK
CCS A
TCF NVSBWT1 # BUSY
TCF NVSB COM # FREE. NVSUB WILL SAVE L+1 FOR RETURN
AFTER EXECUTION.
NVSBWT1 INCR Q # L+2. PRENVBSY WILL PUT L-1 INTO LIST AND
TCF PRENVBSY # GO TO SLEEP.

RELDSP IS USED BY VBPROC, VBTERM, VBRQEXEC, VBRQWAIT, VBRELDSP, EXTENDED
VERB DISPATCHER, VBRESEQ, RECALST.
#
RELDSP1 IS USED BY MONITOR SET UP, VBRELDSP.

RELDSP XCH Q # SET DSPLOCK TO +0, TURN RELDSP LIGHT
Page 383
TS RELRET # OFF, SEARCH DSPLIST
CS BIT14
INHINT
MASK MONSAVE1
TS MONSAVE1 # TURN OFF EXTERNAL MONITOR BIT
CCS DSPLIST
TC +2
TC RELDSP2 # LIST EMPTY
CAF ZERO
XCH DSPLIST
TC JOBWAKE
RELDSP2 RELINT
CS BIT5 # TURN OFF KEY RELEASE LIGHT
EXTEND # (BIT 5 OF CHANNEL 11)
WAND DSALMOUT
CAF ZERO
TS DSPLOCK
TC RELRET
RELDSP1 XCH Q # SET DSPLOCK TO +0. NO DSPLIST SEARCH.
TS RELRET # TURN KEY RLSE LIGHT OFF IF DSPLIST IS
EMPTY. LEAVE KEY RLSE LIGHT ALONE IF
DSPLIST IS NOT EMPTY.
CCS DSPLIST

```

|     |         |      |                                      |
|-----|---------|------|--------------------------------------|
| TC  | +2      | # +  | NOT EMPTY. LEAVE KEY RLSE LIGHT ALON |
| TC  | RELDSP2 | # +0 | EMPTY. TURN OFF KEY RLSE LIGHT       |
| CAF | ZERO    | # -  | NOT EMPTY. LEAVE KEY RLSE LIGHT ALON |
| TS  | DSPLOCK |      |                                      |
| TC  | RELRET  |      |                                      |

ENDPINBF        EQUALS

# Page 384

# PINTEST IS NEEDED FOR AUTO CHECK OF PINBALL.

PINTEST        EQUALS    LST2FAN

# Page 385

# VBTSTLTS TURNS ON ALL DISPLAY PANEL LIGHTS. AFTER 5 SEC, IT TURNS

# OFF THE CAUTION AND STATUS LIGHTS.

SETLOC    ENDNVSB1 +1

COUNT    41/PIN

|          |      |          |
|----------|------|----------|
| VBTSTLTS | TC   | BANKCALL |
|          | CADR | CHKPOOH  |

INHINT

|    |      |                                         |
|----|------|-----------------------------------------|
| CS | BIT1 | # SET BIT 1 OF IMODES33 SO IMUMON WON'T |
|----|------|-----------------------------------------|

|      |          |                       |
|------|----------|-----------------------|
| MASK | IMODES33 | # TURN OUT ANY LAMPS. |
|------|----------|-----------------------|

|    |      |
|----|------|
| AD | BIT1 |
|----|------|

|    |          |
|----|----------|
| TS | IMODES33 |
|----|----------|

|     |         |                                            |
|-----|---------|--------------------------------------------|
| CAF | TSTCON1 | # TURN ON UPLINK ACTIVITY, TEMP, KEY RLSE, |
|-----|---------|--------------------------------------------|

|        |  |                              |
|--------|--|------------------------------|
| EXTEND |  | # V/N FLASH, OPERATOR ERROR. |
|--------|--|------------------------------|

|     |          |
|-----|----------|
| WOR | DSALMOUT |
|-----|----------|

|     |         |                                         |
|-----|---------|-----------------------------------------|
| CAF | TSTCON2 | # TURN ON NO ATT, GIMBAL LOCK, TRACKER, |
|-----|---------|-----------------------------------------|

|    |             |             |
|----|-------------|-------------|
| TS | DSPTAB +11D | # PROG ALM. |
|----|-------------|-------------|

|     |       |                             |
|-----|-------|-----------------------------|
| CAF | BIT10 | # TURN ON TEST ALARM OUTBIT |
|-----|-------|-----------------------------|

EXTEND

|     |        |
|-----|--------|
| WOR | CHAN13 |
|-----|--------|

|     |     |
|-----|-----|
| CAF | TEN |
|-----|-----|

TSTLTS1

|    |       |
|----|-------|
| TS | ERCNT |
|----|-------|

|    |         |
|----|---------|
| CS | FULLDSP |
|----|---------|

|       |       |
|-------|-------|
| INDEX | ERCNT |
|-------|-------|

|    |        |
|----|--------|
| TS | DSPTAB |
|----|--------|

|     |       |
|-----|-------|
| CCS | ERCNT |
|-----|-------|

|    |         |
|----|---------|
| TC | TSTLTS1 |
|----|---------|

|    |          |
|----|----------|
| CS | FULLDSP1 |
|----|----------|

July 12, 2016

1499

|            |        |             |                                                                                                                |
|------------|--------|-------------|----------------------------------------------------------------------------------------------------------------|
|            | TS     | DSPTAB +1   | # TURN ON 3 PLUS SIGNS                                                                                         |
|            | TS     | DSPTAB +4   |                                                                                                                |
|            | TS     | DSPTAB +6   |                                                                                                                |
|            | CAF    | ELEVEN      |                                                                                                                |
|            | TS     | NOUT        |                                                                                                                |
|            | CAF    | SHOLTS      |                                                                                                                |
|            | TC     | WAITLIST    |                                                                                                                |
|            | EBANK= | DSPTAB      |                                                                                                                |
|            | 2CADR  | TSTLTS2     |                                                                                                                |
|            | TC     | ENDOFJOB    | # DSPLOCK IS LEFT BUSY (FROM KEYBOARD<br># ACTION) UNTIL TSTLTS3 TO INSURE THAT<br># LIGHTS TEST WILL BE SEEN. |
| FULLDSP    | OCT    | 05675       | # DISPLAY ALL 8'S                                                                                              |
| FULLDSP1   | OCT    | 07675       | # DISPLAY ALL 8'S AND +                                                                                        |
| TSTCON1    | OCT    | 00175       |                                                                                                                |
| # Page 386 |        |             |                                                                                                                |
|            |        |             | # UPLINK ACTIVITY, TEMP, KEY RLSE,<br># V/N FLASH, OPERATOR ERROR.                                             |
| TSTCON2    | OCT    | 40650       | # DSPTAB+11D BITS 4,6,8,9,<br># NO ATT, GIMBAL LOCK, TRACKER, PROG ALM.                                        |
| TSTCON3    | OCT    | 00115       | # CHAN 11 BITS 1, 3, 4, 7.<br># UPLINK ACITIVY, TEMP, OPERATOR ERROR.                                          |
| SHOLTS     | OCT    | 764         | # 5 SEC                                                                                                        |
| TSTLTS2    | CAF    | CHRPRI0     | # CALLED BY WAITLIST                                                                                           |
|            | TC     | NOVAC       |                                                                                                                |
|            | EBANK= | DSPTAB      |                                                                                                                |
|            | 2CADR  | TSTLTS3     |                                                                                                                |
|            | TC     | TASKOVER    |                                                                                                                |
| TSTLTS3    | CS     | TSTCON3     | # CALLED BY EXECUTIVE                                                                                          |
|            | INHINT |             |                                                                                                                |
|            | EXTEND |             | # TURN OFF UPLINK ACTIVITY, TEMP,                                                                              |
|            | WAND   | DSALMOUT    | # OPERATOR ERROR.                                                                                              |
|            | CS     | BIT10       | # TURN OFF TEST ALARM OUTBIT                                                                                   |
|            | EXTEND |             |                                                                                                                |
|            | WAND   | CHAN13      |                                                                                                                |
|            | CAF    | BIT4        | # MAKE NO ATT FOLLOW BIT 4 OF CHANNEL 12                                                                       |
|            | EXTEND |             | # (NO TT LIGHT ON IF IN COARSE ALIGN)                                                                          |
|            | RAND   | CHAN12      |                                                                                                                |
|            | AD     | BIT15       | # TURN OFF AUTO, HOLD, FREE, SPARE,                                                                            |
|            | TS     | DSPTAB +11D | # GIMBAL LOCK, SPARE, TRACKER, PROG ALM                                                                        |
|            | CS     | 13-11,1     | # SET BITS TO INDICATE ALL LAMPS OUT. TEST                                                                     |

|  |        |          |                                          |
|--|--------|----------|------------------------------------------|
|  | MASK   | IMODES33 | # LIGHTS COMPLETE.                       |
|  | AD     | PRI016   |                                          |
|  | TS     | IMODES33 |                                          |
|  | CS     | OCT55000 |                                          |
|  | MASK   | IMODES30 |                                          |
|  | AD     | PRI015   | # 15000.                                 |
|  | TS     | IMODES30 |                                          |
|  | CS     | OPTMODES |                                          |
|  | MASK   | BIT7     |                                          |
|  | ADS    | OPTMODES |                                          |
|  | RELINT |          |                                          |
|  | TC     | BANKCALL | # REDISPLAY C(MODREG)                    |
|  | CADR   | DSPMM    |                                          |
|  | TC     | KILMONON | # TURN ON KILL MONITOR BIT.              |
|  | TC     | FLASHOFF | # TURN OFF V/N FLASH.                    |
|  | TC     | POSTJUMP | # DOES RELDSP AND GOES TO PINBRNCH IF    |
|  | CADR   | TSTLTS4  | # ENDIDLE IS AWAITING OPERATOR RESPONSE. |

# Page 387

|          |        |       |  |
|----------|--------|-------|--|
| 13-11,1  | OCT    | 16001 |  |
| OCT55000 | OCT    | 55000 |  |
| ENDPINS2 | EQUALS |       |  |

# Page 388

# ERROR LIGHT RESET (RSET) TURNS OFF:

# UPLINK ACTIVITY, AUTO, HOLD, FREE, OPERATOR ERROR,

# PROG ALM, TRACKER FAIL.

#

# LEAVES GIMBAL LOCK AND NO ATT ALONE.

#

# IT ALSO ZEROS THE 'TEST ALARM' OUT BIT, WHICH TURNS OFF STBY, RESTART.

# IT ALSO SETS 'CAUTION RESET' TO 1.

# IT ALSO FORCES BIT 12 OF ALL DSPTAB ENTRIES TO 1.

|  |        |           |  |
|--|--------|-----------|--|
|  | SETLOC | DOPROC +2 |  |
|  | COUNT  | 40/PIN    |  |

|       |        |          |                                           |
|-------|--------|----------|-------------------------------------------|
| ERROR | XCH    | 21/22REG | # RESTORE ORIGINAL C(DSPLOCK). THUS ERROR |
|       | TS     | DSPLOCK  | # LIGHT RESET LEAVES DSPLOCK CHANGED.     |
|       | INHINT |          |                                           |
|       | CAF    | BIT10    | # TURN ON 'CAUTION RESET' OUTBIT          |
|       | EXTEND |          |                                           |
|       | WOR    | DSALMOUT | # BIT10 CHAN 11                           |
|       | CAF    | GL+NOATT | # LEAVE GIMBAL LOCK AND NO ATT INTACT,    |

July 12, 2016

1501

```

MASK DSPTAB +11D # TURNING OFF AUTO, HOLD, FREE,
AD BIT15 # PROG ALARM, AND TRACKER.
TS DSPTAB +11D
CS PRI016
MASK IMODES33 # RESET FAIL BITS WHICH GENERATE PROG
AD PRI016 # ALARM SO THAT IF THE FAILURE STILL
TS IMODES33 # EXISTS, THE ALARM WILL COME BACK.
CS BIT10
MASK IMODES30
AD BIT10
TS IMODES30

CS OPTMODES
MASK BIT7
ADS OPTMODES
CS BIT10 # TURN OFF 'TEST ALARM' OUTBIT
EXTEND
WAND CHAN13
CS ERCON # TURN OFF UPLINK ACTIVITY,
EXTEND # OPERATOR ERROR.
WAND DSALMOUT
CAF BINCON # (DEC 10)
TS ERCNT # ERCNT = COUNT
INHINT
INDEX ERCNT
CCS DSPTAB
AD ONE
TC ERPLUS
AD ONE
ERMINUS CS A
MASK NOTBIT12
TC ERCOM

Page 389
ERPLUS CS A
MASK NOTBIT12
CS A # MIGHT WANT TO RESET CLPASS, DECBRNCH,
ERCOM INDEX ERCNT # ETC.
TS DSPTAB
RELINT
CCS ERCNT
TC TSTAB +1
CAF ZERO
TS FAILREG
TS FAILREG +1
TS FAILREG +2
TS SFAIL
```

1502

July 12, 2016

```
TC ENDOFJOB

ERCON OCT 104 # CHAN 11 BITS 3,7.
 # UPLINK ACTIVITY, AND OPERATOR ERROR.

BITS6,8 OCT 240
GL+NOATT OCT 00050 # NO ATT AND GIMBAL LOCK LAMPS
NOTBIT12 OCT 73777

ENDPINS1 EQUALS

SBANK= LOWSUPER
```

This code is written to file `src/PINBALL-GAME-BUTTONS-AND-LIGHTS.s`.

## B.82 PINBALL NOUN TABLES

```

1503 <src/PINBALL-NOUN-TABLES.s 1503>≡
Copyright: Public domain.
Filename: PINBALL_NOUN_TABLES.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 268-284
Contact: Ron Burkey <info@sandroid.org>,
Fabrizio Bernardini <fabrizio@spacecraft.it>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-18 FB Transcription Batch 3 Assignment.
2009-05-23 RSB In NNTYPTAB, corrected former 13 SPARE.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.
#
Page 268
THE FOLLOWING REFERS TO THE NOUN TABLES
#
COMPONENT CODE NUMBER INTERPRETATION
#
00000 1 COMPONENT
00001 2 COMPONENT
00010 3 COMPONENT
X1XXX BIT 4 = 1. DECIMAL ONLY
1XXXX BIT 5 = 1. NO LOAD
#
END OF COMPONENT CODE NUMBERS

```

```

#
SF ROUTINE CODE NUMBER INTERPRETATION
#
00000 OCTAL ONLY
00001 STRAIGHT FRACTIONAL
00010 CDU DEGREES (XXX.XX)
00011 ARITHMETIC SF
00100 ARITH DP1 OUT (MULT BY 2EXP14 AT END) IN (S
00101 ARITH DP2 OUT (STRAIGHT) IN (S
00110 Y OPTICS DEGREES (XX.XXX MAX 89.999)
00111 ARITH DP3 OUT (SL 7 AT END) IN (S
01000 WHOLE HOURS IN R1, WHOLE MINUES (MOD 60) IN R2,
SECONDS (MOD 60) OXX.XX IN R3. *** ALARMS IF
01001 MINUTES (MOD 60) IN D1D2, D3 BLANK, SECONDS (MOD 60)
LIMITS TO 59B59 IF MAG EXCEEDS THIS VALUE.
ALARMS IF USED WITH OCTAL ***** IN (ALARM)
01010 ARITH DP4 OUT (STRAIGHT) IN (S
01011 ARITH1 SF OUT (MULT BY 2EXP14 AT END) IN (S
01100 2 INTEGERS IN D1D2, D4D5, D3 BLANK.
ALARMS IF USED WITH OCTAL ***** IN (ALARM)
01101 DP STRAIGHT FRACTIONAL
#
END OF SF ROUTINE CODE NUMBERS

SF CONSTANT CODE NUMBER INTERPRETATION
#
00000 WHOLE USE ARITH
00000 DP TIME SEC (XXX.XX SEC) USE ARITHDP1
00001 SPARE
00010 CDU DEGREES USE CDU DEGREES
00010 Y OPTICS DEGREES USE Y OPTICS DEGREES
00011 DP DEGREES (90) XX.XXX DEG USE ARITHDP3
00100 DP DEGREES (360) XXX.XX DEG USE ARITHDP4
00101 DEGREES (180) XXX.XX DEG USE ARITH
00110 WEIGHT2 (XXXXX. LBS) USE ARITH1
00111 POSITION5 (XXX.XX NAUTICAL MILES)
USE ARITHDP3
01000 POSITION4 (XXXX.X NAUTICAL MILES)
Page 269
#
01001 VELOCITY2 (XXXXX. FT/SEC) USE ARITHDP4
01010 VELOCITY3 (XXXX.X FT/SEC) USE ARITHDP3
01011 ELEVATION DEGREES (89.999 MAX) USE ARITH
01100 TRIM DEGREES (XXX.XX DEG) USE ARITH
01101 INERTIA (XXXXXXBB. SLUG FT FT) USE ARITH
01101 THRUST MOMENT (XXXXXXBB. FT LBS) USE ARITH

```



July 12, 2016

1505

```
01110 VELOCITY/2VS (XXXXX. FT/SEC) USE ARITHDP4
01111 POSITION6 (XXXX.X NAUT MI) USE ARITHDP3
10000 DRAG ACCELERATION (XXX.XX) USE ARITHDP2
10001 POSITION8 (XXXX.X NAUT MI) USE ARITHDP?
10010 POSITION9 (XXX.XX NAUT MI) USE ARITHDP4
10011 VELOCITY4 (XXXX.X FT/SEC) USE ARITHDP2
#
END OF SF CONSTANT CODE NUMBERS
```

```
FOR GREATER THAN SINGLE PRECISION SCALES, PUT ADDRESS IN MAJOR PART INTO
NOUN TABLES.
```

```
OCTAL LOADS PLACE +0 INTO MAJOR PART, DATA INTO MINOR PART.
```

```
OCTAL DISPLAYS SHOW MINOR PART ONLY.
```

```
TO GET AT BOTH MAJOR AND MINOR PARTS (IN OCTAL), USE NOUN 01.
```

```
A NOUN MAY BE DECLARED "DECIMAL ONLY" BY MAKING BIT4=1 OF ITS COMPONENT
CODE NUMBER. IF THIS NOUN IS USED WITH ANY OCTAL DISPLAY VERB, OR IF
DATA IS LOADED IN OCTAL, IT ALARMS.
```

```
IN LOADING AN "HOURS, MINUTES, SECONDS" NOUN, ALL 3 WORDS MUST BE
LOADED, OR ALARM.
```

```
ALARM IF AN ATTEMPT IS MADE TO LOAD "SPLIT MINUTES/SECONDS" (MMBSS).
THIS IS USED FOR DISPLAY ONLY.
```

```
Page 270
```

```
THE FOLLOWING ROUTINES ARE FOR READING THE NOUN TABLES AND THE SF TABLES
(WHICH ARE IN A SEPARATE BANK FROM THE REST OF PINBALL). THESE READING
ROUTINES ARE IN THE SAME BANK AS THE TABLES. THEY ARE CALLED BY DXCH Z.
```

```
LODNNTAB LOADS NNADTEM WITH THE NNADTAB ENTRY, NNTYPTTEM WITH THE
NNTYPTAB ENTRY. IF THE NOUN IS MIXED, IDADITEM IS LOADED WITH THE FIRST
IDADDTAB ENTRY, IDAD2TEM THE SECOND IDADDTAB ENTRY, IDAD3TEM THE THIRD
IDADDTAB ENTRY, RUTMXTEM WITH THE RUTMXTAB ENTRY. MIXBR IS SET FOR
MIXED OR NORMAL NOUN.
```

```
BANK 06
SETLOC PINBALL3
BANK
```

```
COUNT 42/NOUNS
```

```
LODNNTAB DXCH IDAD2TEM # SAVE RETURN INFO IN IDAD2TEM, IDAD3TEM.
```

```

INDEX NOUNREG
CAF NNADTAB
TS NNADTEM
INDEX NOUNREG
CAF NNTYPTAB
TS NNTYPTM
CS NOUNREG
AD MIXCON
EXTEND
BZMF LODMIXNN # NOUN NUMBER G/E FIRST MIXED NOUN
CAF ONE # NOUN NUMBER L/ FIRST MIXED NOUN
TS MIXBR # NORMAL. +1 INTO MIXBR
TC LODNLV
LODMIXNN CAF TWO # MIXED. +2 INTO MIXBR.
TS MIXBR
INDEX NOUNREG
CAF RUTMXTAB -40D # FIRST MIXED NOUN = 40.
TS RUTMXTEM
CAF LOW10
MASK NNADTEM
TS Q # TEMP
INDEX A
CAF IDADDTAB
TS IDAD1TEM # LOAD IDAD1TEM WITH FIRST IDADDTAB EN
EXTEND
INDEX Q # LOAD IDAD2TEM WITH 2ND IDADDTAB EN
DCA IDADDTAB +1 # LOAD IDAD3TEM WITH 3RD IDADDTAB EN
LODNLV DXCH IDAD2TEM # PUT RETURN INFO INTO A, L.
DXCH Z

MIXCON = OCT50 # FIRST MIXED NOUN =40. (DEC 40)

GTSFOUT LOADS SFTEMP1, SFTEMP2 WITH THE DP SFOUTAB ENTRIES.
Page 271

GTSFOUT DXCH SFTEMP1 # 2X (SFCONUM) ARRIVES IN SFTEMP1.
EXTEND
INDEX A
DCA SFOUTAB
SFCOM DXCH SFTEMP1
DXCH Z

GTSFIN LOADS SFTEMP1, SFTEMP2 WITH THE DP SFINTAB INTRIES.

GTSFIN DXCH SFTEMP1 # 2X (SFCONUM) ARIVES IN SFTEMP1.
EXTEND

```

July 12, 2016

1507

INDEX    A  
DCA      SFINTAB  
TCF      SFCOM

|         |       | #         | NN   | NORMAL NOUNS                           |
|---------|-------|-----------|------|----------------------------------------|
| NNADTAB | OCT   | 00000     | # 00 | NOT IN USE                             |
|         | OCT   | 40000     | # 01 | SPECIFY MACHINE ADDRESS (FRACTIONAL)   |
|         | OCT   | 40000     | # 02 | SPECIFY MACHINE ADDRESS (WHOLE)        |
|         | OCT   | 40000     | # 03 | SPECIFY MACHINE ADDRESS (DEGREES)      |
|         | OCT   | 0         | # 04 | SPARE                                  |
|         | ECADR | DSPTM1    | # 05 | ANGULAR ERROR/DIFFERENCE               |
|         | ECADR | OPTION1   | # 06 | OPTION CODE                            |
|         | ECADR | XREG      | # 07 | ECADR OF WORD TO BE MODIFIED           |
|         |       |           | #    | ONES FOR BITS TO BE MODIFIED           |
|         |       |           | #    | 1 TO SET OR 0 TO RESET SELECTED BITS   |
|         | ECADR | ALMCADR   | # 08 | ALARM DATA                             |
|         | ECADR | FAILREG   | # 09 | ALARM CODES                            |
|         | OCT   | 77776     | # 10 | CHANNEL TO BE SPECIFIED                |
|         | ECADR | TCSI      | # 11 | TIG OF CSI (HRS,MIN,SEC)               |
|         | ECADR | OPTIONX   | # 12 | OPTION CODE                            |
|         |       |           | #    | (USED BY EXTENDED VERBS ONLY)          |
|         | ECADR | TCDH      | # 13 | TIG OF CDH (HRS,MIN,SEC)               |
|         | OCT   | 0         | # 14 | SPARE                                  |
|         | OCT   | 77777     | # 15 | INCREMENT MACHINE ADDRESS              |
|         | ECADR | DSPTMX    | # 16 | TIME OF EVENT (HRS,MIN,SEC)            |
|         | ECADR | CPHIX     | # 17 | ASTRONAUT TOTAL ATTITUDE               |
|         | ECADR | THETAD    | # 18 | AUTO MANEUVER BALL ANGLES              |
|         | ECADR | THETAD    | # 19 | BYPASS ATTITUDE TRIM MANEUVER          |
|         | ECADR | CDUX      | # 20 | ICDU ANGLES                            |
|         | ECADR | PIPAX     | # 21 | PIPAS                                  |
|         | ECADR | THETAD    | # 22 | NEW ICDU ANGLES                        |
|         | OCT   | 00000     | # 23 | SPARE                                  |
|         | ECADR | DSPTM2 +1 | # 24 | DELTA TIME FOR AGC CLOCK (HRS,MIN,SEC) |
|         | ECADR | DSPTM1    | # 25 | CHECKLIST                              |
|         |       |           | #    | (USED WITH PLEASE PERFORM ONLY)        |
|         | ECADR | DSPTM1    | # 26 | PRIO/DELAY, ADRES, BBCON               |
|         | ECADR | SMODE     | # 27 | SELF TEST ON/OFF SWITCH                |
|         | OCT   | 0         | # 28 | SPARE                                  |
|         | ECADR | DSPTM1    | # 29 | XSM LAUNCH AZIMUTH                     |
|         | ECADR | DSPTM1    | # 30 | TARGET CODES                           |
|         | ECADR | DSPTM1    | # 31 | TIME OF LANDING SITE (HRS,MIN,SEC)     |
|         | ECADR | -TPER     | # 32 | TIME TO PERIGEE (HRS,MIN,SEC)          |
|         | ECADR | TIG       | # 33 | TIME OF IGNITION (HRS,MIN,SEC)         |
|         | ECADR | DSPTM1    | # 34 | TIME OF EVENT (HRS,MIN,SEC)            |

ECADR TTOGO  
 ECADR TIME2  
 ECADR TTPI  
 ECADR TET  
 ECADR T3TOT4

# 35 TIME TO GO TO EVENT (HRS,MIN,SEC)  
 # 36 TIME OF AGC CLOCK (HRS,MIN,SEC)  
 # 37 TIG OF TPI (HRS,MIN,SEC)  
 # 38 TIME OF STATE VECTOR  
 # 39 DELTA TIME TO TRANSFER (HRS,MIN,SEC)

# END OF NNADTAB FOR NORMAL NOUNS

# NN MIXED NOUNS

OCT 64000

# 40 TIME TO IGNITION/CUTOFF  
 # VG

OCT 02003

# DELTA V (ACCUMULATED)

OCT 24006

# 41 TARGET AZIMUTH  
 # ELEVATION

OCT 24011

# 42 APOGEE  
 # PERIGEE  
 # DELTA V (REQUIRED)

OCT 64014

# 43 LATITUDE  
 # LONGITUDE  
 # ALTITUDE

OCT 64017

# 44 APOGEE  
 # PERIGEE  
 # TFF

OCT 02022

# 45 MARKS (VHF - OPTIC)  
 # TTI OF NEXT BURN  
 # MGA

OCT 22025

# 46 AUTOPILOT CONFIGURATION  
 # 47 THIS VEHICLE WEIGHT

OCT 22030

# OTHER VEHICLE WEIGHT  
 # 48 PITCH TRIM

OCT 24033

# YAW TRIM

OCT 64036

# 49 DELTA R  
 # DELTA V  
 # VHF OR OPTICS CODE

OCT 22041

# 50 SPLASH ERROR  
 # PERIGEE

OCT 00044

# TFF  
 # 51 S-BAND ANTENNA PITCH

OCT 24047

# YAW

# 52 CENTRAL ANGLE OF ACTIVE VEHICLE

# 53 RANGE

# RANGE RATE

# PHI

# Page 273

OCT 24052

# 54 RANGE

July 12, 2016

1509

|     |       |      |                                |
|-----|-------|------|--------------------------------|
|     |       | #    | RANGE RATE                     |
|     |       | #    | THETA                          |
| OCT | 24055 | # 55 | PERIGEE CODE                   |
|     |       | #    | ELEVATION ANGLE                |
|     |       | #    | CENTRAL ANGLE                  |
| OCT | 22060 | # 56 | REENTRY ANGLE,                 |
|     |       | #    | DELTA V                        |
| OCT | 20063 | # 57 | DELTA R                        |
| OCT | 24066 | # 58 | PERIGEE ALT                    |
|     |       | #    | DELTA V TPI                    |
|     |       | #    | DELTA V TPF                    |
| OCT | 24071 | # 59 | DELTA VELOCITY LOS             |
| OCT | 24074 | # 60 | GMAX                           |
|     |       | #    | VPRED                          |
|     |       | #    | GAMMA EI                       |
| OCT | 24077 | # 61 | IMPACT LATITUDE                |
|     |       | #    | IMPACT LONGITUDE               |
|     |       | #    | HEADS UP/DOWN                  |
| OCT | 24102 | # 62 | INERTIAL VEL MAG (V1)          |
|     |       | #    | ALT RATE CHANGE (HDOT)         |
|     |       | #    | ALT ABOVE PAD RADIUS (H)       |
| OCT | 64105 | # 63 | RANGE 297,431 TO SPLASH (RTGO) |
|     |       | #    | PREDICTED INERT VEL (VIO)      |
|     |       | #    | TIME TO GO TO 297,431 (TTE)    |
| OCT | 24110 | # 64 | DRAG ACCELERATION              |
|     |       | #    | INERTIAL VELOCITY (VI)         |
|     |       | #    | RANGE TO SPLASH                |
| OCT | 24113 | # 65 | SAMPLED AGC TIME (HRS,MIN,SEC) |
|     |       | #    | (FETCHED IN INTERRUPT)         |
| OCT | 24116 | # 66 | COMMAND BANK ANGLE (BETA)      |
|     |       | #    | CROSS RANGE ERROR              |
|     |       | #    | DOWN RANGE ERROR               |
| OCT | 24121 | # 67 | RANGE TO TARGET                |
|     |       | #    | PRESENT LATITUDE               |
|     |       | #    | PRESENT LONGITUDE              |
| OCT | 24124 | # 68 | COMMAND BANK ANGLE (BETA)      |
|     |       | #    | INERTIAL VELOCITY (VI)         |
|     |       | #    | ALT RATE CHANGE (RDOT)         |
| OCT | 24127 | # 69 | BETA                           |
|     |       | #    | DL                             |
|     |       | #    | VL                             |
| OCT | 04132 | # 70 | STAR CODE                      |
|     |       | #    | LANDMARK DATA                  |
|     |       | #    | HORIZON DATA                   |
| OCT | 04135 | # 71 | STAR CODE                      |
|     |       | #    | LANDMARK DATA                  |

# Page 274

|  |  |      |                                 |
|--|--|------|---------------------------------|
|  |  | #    | HORIZON DATA                    |
|  |  | # 72 | DELT ANG                        |
|  |  | #    | DELT ALT                        |
|  |  | #    | SEARCH OPTION                   |
|  |  | # 73 | ALTITUDE                        |
|  |  | #    | VELOCITY                        |
|  |  | #    | FLIGHT PATH ANGLE               |
|  |  | # 74 | COMMAND BANK ANGLE (BETA)       |
|  |  | #    | INERTIAL VELOCITY (VI)          |
|  |  | #    | DRAG ACCELERATION               |
|  |  | # 75 | DELTA ALTITUDE CDH              |
|  |  | #    | DELTA TIME (CDH-CSI OR TPI-CDH) |
|  |  | #    | DELTA TIME (TPI-CDH OR TPI-NO)  |
|  |  | # 76 | SPARE                           |
|  |  | # 77 | SPARE                           |
|  |  | # 78 | SPARE                           |
|  |  | # 79 | SPARE                           |
|  |  | # 80 | TIME TO IGNITION/CUTOFF         |
|  |  | #    | VG                              |
|  |  | #    | DELTA V (ACCUMULATED)           |
|  |  | # 81 | DELTA V (LV)                    |
|  |  | # 82 | DELTA V (LV)                    |
|  |  | # 83 | DELTA V (BODY)                  |
|  |  | # 84 | DELTA V (OTHER VEHICLE)         |
|  |  | # 85 | VG (BODY)                       |
|  |  | # 86 | DELTA V (LV)                    |
|  |  | # 87 | MARK DATA            SHAFT      |
|  |  | #    | TRUNION                         |
|  |  | # 88 | HALF UNIT SUN OR PLANET VECTO   |
|  |  | # 89 | LANDMARK            LATITUDE    |
|  |  | #    | LONGITUDE/2                     |
|  |  | #    | ALTITUDE                        |
|  |  | # 90 | Y                               |
|  |  | #    | Y DOT                           |
|  |  | #    | PSI                             |
|  |  | # 91 | OCDU ANGLES            SHAFT    |
|  |  | #    | TRUNION                         |
|  |  | # 92 | NEW OPTICS ANGLES SHAFT         |
|  |  | #    | TRUNION                         |
|  |  | # 93 | DELTA GYRO ANGLES               |
|  |  | # 94 | NEW OPTICS ANGLES SHAFT         |
|  |  | #    | TRUNION                         |
|  |  | # 95 | PREFERRED ATTITUDE ICDU ANGLE   |
|  |  | # 96 | +X-AXIS ATTITUDE ICDU ANGLES    |
|  |  | # 97 | SYSTEM TEST INPUTS              |

July 12, 2016

1511

OCT 04256  
OCT 24261

# 98 SYSTEM TEST RESULTS  
# 99 RMS IN POSITION  
# RMS IN VELOCITY  
# RMS OPTION

# END OF NNADTAB FOR MIXED NOUNS

# Page 275

NNTYPTAB

OCT 00000  
OCT 04040  
OCT 04140  
OCT 04102  
OCT 0  
OCT 00504  
OCT 02000  
OCT 04000  
OCT 04000  
OCT 04000  
OCT 00000  
OCT 24400  
OCT 02000  
OCT 24400  
OCT 0  
OCT 00000  
OCT 24400  
OCT 04102  
OCT 04102  
OCT 04102  
OCT 04102  
OCT 04140  
OCT 04102  
OCT 00000  
OCT 24400  
OCT 04140  
OCT 04000  
OCT 00140  
OCT 00000  
OCT 20102  
OCT 04140  
OCT 24400  
OCT 24400  
OCT 24400  
OCT 24400  
OCT 24400  
OCT 24400  
OCT 24400

# NN NORMAL NOUNS

# 00 NOT IN USE  
# 01 3COMP FRACTIONAL  
# 02 3COMP WHOLE  
# 03 3COMP CDU DEGREES  
# 04 SPARE  
# 05 1COMP DPDEG(360)  
# 06 2COMP OCTAL ONLY  
# 07 3COMP OCTAL ONLY  
# 08 3COMP OCTAL ONLY  
# 09 3COMP OCTAL ONLY  
# 10 1COMP OCTAL ONLY  
# 11 3COMP HMS (DEC ONLY)  
# 12 2COMP OCTAL ONLY  
# 13 3COMP HMS (DEC ONLY)  
# 14 SPARE  
# 15 1COMP OCTAL ONLY  
# 16 3COMP HMS (DEC ONLY)  
# 17 3COMP CDU DEG  
# 18 3COMP CDU DEG  
# 19 3COMP CDU DEG  
# 20 3COMP CDU DEGREES  
# 21 3COMP WHOLE  
# 22 3COMP CDU DEGREES  
# 23 SPARE  
# 24 3COMP HMS (DEC ONLY)  
# 25 3COMP WHOLE  
# 26 3COMP OCTAL ONLY  
# 27 1COMP WHILE  
# 28 SPARE  
# 29 1COMP CDU DEG (DEC ONLY)  
# 30 3COMP WHOLE  
# 31 3COMP HMS (DEC ONLY)  
# 32 3COMP HMS (DEC ONLY)  
# 33 3COMP HMS (DEC ONLY)  
# 34 3COMP HMS (DEC ONLY)  
# 35 3COMP HMS (DEC ONLY)  
# 36 3COMP HMS (DEC ONLY)

OCT 24400  
 OCT 24400  
 OCT 24400

# 37 3COMP HMS (DEC ONLY)  
 # 38 3COMP HMS (DEC ONLY)  
 # 39 3COMP HMS (DEC ONLY)

# END OF NNTYPTAB FOR NORMAL NOUNS

# Page 276

OCT 24500  
 OCT 00542  
 OCT 24410  
 OCT 20204  
 OCT 00410  
 OCT 10000  
 OCT 00000  
 OCT 00306  
 OCT 00614  
 OCT 00510  
 OCT 00417  
 OCT 00204  
 OCT 00004  
 OCT 10507  
 OCT 10507  
 OCT 10200  
 OCT 00444  
 OCT 00010  
 OCT 24510  
 OCT 24512

# NN MIXED NOUNS

# 40 3COMP MIN/SEC, VEL3, VEL3  
 # (NO LOAD, DEC ONLY)  
 # 41 2COMP CDU DEG, ELEV DEG  
 # 42 3COMP POS4, POS4, VEL3  
 # (DEC ONLY)  
 # 43 3COMP DPDEG(360), DPDEG(360)  
 # (DEC ONLY)  
 # 44 3COMP POS4, POS4, MIN/SEC  
 # (NO LOAD, DEC ONLY)  
 # 45 3COMP 2INT, MIN/SEC, DPDEG  
 # (NO LOAD, DEC ONLY)  
 # 46 2COMP OCTAL ONLY FOR EACH  
 # 47 2COMP WEIGHT2 FOR EACH  
 # (DEC ONLY)  
 # 48 2COMP TRIM DEG, TRIM DEG  
 # (DEC ONLY)  
 # 49 3COMP POS4, VEL3, WHOLE  
 # (DEC ONLY)  
 # 50 3COMP POS6, POS4, MIN/SEC  
 # (NO LOAD, DEC ONLY)  
 # 51 2COMP DPDEG(360), DPDEG(360)  
 # (DEC ONLY)  
 # 52 1COMP DPDEG(360)  
 # 53 3COMP POS5, VEL3, DPDEG(360)  
 # (DEC ONLY)  
 # 54 3COMP POS5, VEL3, DPDEG(360)  
 # (DEC ONLY)  
 # 55 3COMP WHOLE, DPDEG(360), DPDEG(360)  
 # (DEC ONLY)  
 # 56 2COMP DPDEG(360), VEL2  
 # (DEC ONLY)  
 # 57 1COMP POS4  
 # (DEC ONLY)  
 # 58 3COMP POS4, VEL3, VEL3  
 # (DEC ONLY)  
 # 59 3COMP VEL3 FOR EACH  
 # (DEC ONLY)



July 12, 2016

1513

|            |     |       |      |       |                                 |
|------------|-----|-------|------|-------|---------------------------------|
|            | OCT | 10440 | # 60 | 3COMP | WHOLE, VEL2, DPDEG(360)         |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 00204 | # 61 | 3COMP | DPDEG(360), DPDEG(360), WHOLE   |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 20451 | # 62 | 3COMP | VEL2, VEL2, POS4                |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 00457 | # 63 | 3COMP | POS6, VEL2, MIN/SEC             |
|            |     |       | #    |       | (NO LOAD, DEC ONLY)             |
|            | OCT | 36460 | # 64 | 3COMP | DRAG ACCEL, VEL2, POS6          |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 00000 | # 65 | 3COMP | HMS (DEC ONLY)                  |
|            | OCT | 37044 | # 66 | 3COMP | DPDEG(360), POS8, POS6          |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 10217 | # 67 | 3COMP | POS6, DPDEG(360), DPDEG(360)    |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 34444 | # 68 | 3COMP | DPDEG(360), VEL2, VEL/2VS       |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 35004 | # 69 | 3COMP | DPDEG(360), DRAG ACCEL, VEL/2VS |
|            |     |       | #    |       | (DEC ONLY)                      |
| # Page 277 | OCT | 00000 | # 70 | 3COMP | OCTAL ONLY FOR EACH             |
|            | OCT | 0     | # 71 | 3COMP | OCTAL ONLY FOR EACH             |
|            | OCT | 00404 | # 72 | 3COMP | DPDEG(360), POS4, WHOLE         |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 10450 | # 73 | 3COMP | POS4, VEL2, DPDEG(360)          |
|            | OCT | 40444 | # 74 | 3COMP | DPDEG(360), VEL2, DRAG ACCEL    |
|            | OCT | 00010 | # 75 | 3COMP | POS4, MIN/SEC, MIN/SEC          |
| #          |     |       | #    |       | (NO LOAD, DEC ONLY)             |
|            | OCT | 0     | # 76 | SPARE |                                 |
|            | OCT | 0     | # 77 | SPARE |                                 |
|            | OCT | 0     | # 78 | SPARE |                                 |
|            | OCT | 0     | # 79 | SPARE |                                 |
|            | OCT | 22440 | # 80 | 3COMP | MIN/SEC, VEL2, VEL2             |
|            |     |       | #    |       | (NO LOAD, DEC ONLY)             |
|            | OCT | 24512 | # 81 | 3COMP | VEL3 FOR EACH                   |
|            |     |       | #    |       | (NO LOAD, DEC ONLY)             |
|            | OCT | 24512 | # 82 | 3COMP | VEL3 FOR EACH                   |
|            |     |       | #    |       | (NO LOAD, DEC ONLY)             |
|            | OCT | 24512 | # 83 | 3COMP | VEL3 FOR EACH                   |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 24512 | # 84 | 3COMP | VEL3 FOR EACH                   |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 24512 | # 85 | 3COMP | VEL3 FOR EACH                   |
|            |     |       | #    |       | (DEC ONLY)                      |
|            | OCT | 22451 | # 86 | 3COMP | VEL2 FOR EACH                   |
|            |     |       | #    |       | (DEC ONLY)                      |

|                                   |     |       |      |                                    |
|-----------------------------------|-----|-------|------|------------------------------------|
|                                   |     |       | #    | (DEC ONLY)                         |
|                                   | OCT | 00102 | # 87 | 2COMP CDU DEG, Y OPTICS DEG        |
|                                   | OCT | 0     | # 88 | 3COMP FRAC FOR EACH                |
|                                   |     |       | #    | (DEC ONLY)                         |
|                                   | OCT | 16143 | # 89 | 3COMP DPDEG(90), DPDEG(90)         |
|                                   |     |       | #    | (DEC ONLY)                         |
|                                   | OCT | 10507 | # 90 | 3COMP POS5, VEL3, DEPDEG(30)       |
|                                   |     |       | #    | (DEC ONLY)                         |
|                                   | OCT | 00102 | # 91 | 2COMP CDUDEG, YOPTICS DEG          |
|                                   | OCT | 00102 | # 92 | 2COMP CDUDEG, YOPTICS DEG          |
|                                   | OCT | 06143 | # 93 | 3COMP DPDEG(90) FOR EACH           |
|                                   | OCT | 00102 | # 94 | 2COMP CDUDEG, YOPTICS DEG          |
|                                   | OCT | 04102 | # 95 | 3COMP CDU DEG FOR EACH             |
|                                   | OCT | 04102 | # 96 | 3COMP CDU DEG FOR EACH             |
|                                   | OCT | 00000 | # 97 | 3COMP WHOLE FOR EACH               |
|                                   | OCT | 00000 | # 98 | 3COMP WHOLE, FRAC, WHOLE           |
|                                   | OCT | 01162 | # 99 | 3COMP POS9, VEL4, WHOLE            |
|                                   |     |       | #    | (DEC ONLY)                         |
| # END OF NNTYPTAB FOR MIXED NOUNS |     |       |      |                                    |
| SFINTAB                           | OCT | 00006 | #    | WHOLE, DP TIME (SEC)               |
|                                   | OCT | 03240 |      |                                    |
|                                   | OCT | 00000 | #    | SPARE                              |
|                                   | OCT | 00000 |      |                                    |
| # Page 278                        |     |       |      |                                    |
|                                   | OCT | 00000 | #    | CDU DEGREES, Y OPTICS DEGREES      |
|                                   | OCT | 00000 | #    | (SFCONS IN DEGINSF, OPTDEGIN)      |
|                                   | OCT | 10707 | #    | DP DEGREES (90)                    |
|                                   | OCT | 03435 | #    | UPPED BY 1                         |
|                                   | OCT | 13070 | #    | DP DEGREES (360) (POINT BETWN BITS |
|                                   | OCT | 34345 | #    | UPPED BY 1                         |
|                                   | OCT | 00005 | #    | DEGREES (180)                      |
|                                   | OCT | 21616 |      |                                    |
|                                   | OCT | 26113 | #    | WEIGHT2                            |
|                                   | OCT | 31713 |      |                                    |
|                                   | OCT | 00070 | #    | POSITION5                          |
|                                   | OCT | 20460 |      |                                    |
|                                   | OCT | 01065 | #    | POSITION4                          |
|                                   | OCT | 05740 |      |                                    |
|                                   | OCT | 11414 | #    | VELOCITY2 (POINT BETWN BITS 11-    |
|                                   | OCT | 31463 |      |                                    |
|                                   | OCT | 07475 | #    | VELOCITY3                          |
|                                   | OCT | 16051 |      |                                    |
|                                   | OCT | 00001 | #    | ELEVATION DEGREES                  |
|                                   | OCT | 03434 |      |                                    |

July 12, 2016

1515

|                  |      |            |                                            |
|------------------|------|------------|--------------------------------------------|
|                  | OCT  | 00002      | # TRIM DEGREES                             |
|                  | OCT  | 22245      |                                            |
|                  | OCT  | 00014      | # INERTIA, THRUST MOMENT                   |
|                  | OCT  | 35607      |                                            |
|                  | OCT  | 07606      | # VELOCITY/2VS                             |
|                  | OCT  | 06300      |                                            |
|                  | OCT  | 16631      | # POSITION 6                               |
|                  | OCT  | 11307      |                                            |
|                  | OCT  | 12000      | # DRAG ACCELERATION (POINT BETWN BITS 7-8) |
|                  | OCT  | 00000      |                                            |
|                  | OCT  | 27176      | # POSITION 8                               |
|                  | OCT  | 14235      |                                            |
|                  | 2DEC | 30480 B-19 | # POSITION 9                               |
|                  |      |            |                                            |
|                  | 2DEC | 30.48 B-7  | # VELOCITY4                                |
|                  |      |            |                                            |
| # END OF SFINTAB |      |            |                                            |
| SFOUTAB          | OCT  | 05174      | # WHOLE, DP TIME (SEC)                     |
|                  | OCT  | 13261      |                                            |
|                  | OCT  | 00000      | # SPARE                                    |
|                  | OCT  | 00000      |                                            |
|                  | OCT  | 00000      | # CDU DEGREES, Y OPTICS DEGREES            |
|                  | OCT  | 00000      | # (SFCONS IN DEGOUTSF, OPTDEGOUT)          |
|                  | OCT  | 00714      | # DP DEGREES (90) (POINT BETWN BITS 7-8)   |
|                  | OCT  | 31463      |                                            |
|                  | OCT  | 13412      | # DP DEGREES (360)                         |
|                  | OCT  | 07534      |                                            |
|                  | OCT  | 05605      | # DEGREES (180)                            |
| # Page 279       |      |            |                                            |
|                  | OCT  | 03656      |                                            |
|                  | OCT  | 00001      | # WEIGHT2                                  |
|                  | OCT  | 16170      |                                            |
|                  | OCT  | 00441      | # POSITION5                                |
|                  | OCT  | 34306      |                                            |
|                  | OCT  | 07176      | # POSITION4                                |
|                  | OCT  | 21603      |                                            |
|                  | OCT  | 15340      | # VELOCITY2                                |
|                  | OCT  | 15340      |                                            |
|                  | OCT  | 01031      | # VELOCITY3 (POINT BETWN BITS 7-8)         |
|                  | OCT  | 21032      |                                            |
|                  | OCT  | 34631      | # ELEVATION DETREES                        |
|                  | OCT  | 23146      |                                            |
|                  | OCT  | 14340      | # TRIM DEGREES                             |
|                  | OCT  | 24145      |                                            |
|                  | OCT  | 02363      | # INERTIA, THRUST MOMENT                   |

|      |                |                                     |
|------|----------------|-------------------------------------|
| OCT  | 03721          |                                     |
| OCT  | 20373          | # VELOCITY/ZVS                      |
| OCT  | 02122          |                                     |
| OCT  | 00424          | # POSITION 6 (POINT BETWN BITS 7-8) |
| OCT  | 30446          |                                     |
| OCT  | 00631          | # DRAG ACCELERATION                 |
| OCT  | 23146          |                                     |
| OCT  | 00260          | # POSITION 8                        |
| OCT  | 06213          |                                     |
| 2DEC | 17.2010499 B-7 | # POSITION 9                        |
| 2DEC | .032808399     | # VELOCITY4                         |

# END OF SFOUTAB

|            |                 | # NN | SF CONSTANT     | SF RO |
|------------|-----------------|------|-----------------|-------|
| IDADDTAB   | ECADR TTOGO     | # 40 | MIN/SEC         | M/S   |
|            | ECADR VGDISP    | # 40 | VEL3            | DP3   |
|            | ECADR DVTOTAL   | # 40 | VEL3            | DP3   |
|            | ECADR DSPTM1    | # 41 | CDU DEG         | CDU   |
|            | ECADR DSPTM1 +1 | # 41 | ELEV DEG        | ARTH  |
|            | OCT 0           | # 41 | SPARE COMPONENT |       |
|            | ECADR HAPO      | # 42 | POS4            | DP3   |
|            | ECADR HPER      | # 42 | POS4            | DP3   |
|            | ECADR VGDISP    | # 42 | VEL3            | DP3   |
|            | ECADR LAT       | # 43 | DPDEG(360)      | DP4   |
|            | ECADR LONG      | # 43 | DPDEG(360)      | DP4   |
|            | ECADR ALT       | # 43 | POS4            | DP3   |
|            | ECADR HAPOX     | # 44 | POS4            | DP3   |
|            | ECADR HPERX     | # 44 | POS4            | DP3   |
|            | ECADR TFF       | # 44 | MIN/SEC         | M/S   |
|            | ECADR VHFCNT    | # 45 | 2INT            | 2INT  |
| # Page 280 | ECADR TTOGO     | # 45 | MIN/SEC         | M/S   |
|            | ECADR +MGA      | # 45 | DPDEG(360)      | DP4   |
|            | ECADR DAPDATR1  | # 46 | OCTAL ONLY      | OCT   |
|            | ECADR DAPDATR2  | # 46 | OCATAL ONLY     | OCT   |
|            | OCT 0           | # 46 | SPARE COMPONENT |       |
|            | ECADR CSMMASS   | # 47 | WEIGHT2         | ARTH  |
|            | ECADR LEMMASS   | # 47 | WEIGHT2         | ARTH  |
|            | OCT 00000       | # 47 | SPARE COMPONENT |       |
|            | ECADR PACTOFF   | # 48 | TRIM DEG2       | ARTH  |
|            | ECADR YACTOFF   | # 48 | TRIM DEG2       | ARTH  |
|            | OCT 00000       | # 48 | SPARE COMPONENT |       |
|            | ECADR N49DISP   | # 49 | POS4            | DP3   |

July 12, 2016

1517

|       |            |      |                 |      |
|-------|------------|------|-----------------|------|
| ECADR | N49DISP +2 | # 49 | VEL3            | DP3  |
| ECADR | N49DISP +4 | # 49 | WHOLE           | ARTH |
| ECADR | RSP-RREC   | # 50 | POS6            | DP3  |
| ECADR | HPERX      | # 50 | POS4            | DP3  |
| ECADR | TFF        | # 50 | MIN/SEC         | M/S  |
| ECADR | RHOSB      | # 51 | DPDEG(360)      |      |
| ECADR | GAMMASB    | # 51 | DPDEG(360)      | DP4  |
| OCT   | 0          | # 51 | SPARE COMPONENT |      |
| ECADR | ACTCENT    | # 52 | DPDEG(360)      | DP4  |
| OCT   | 00000      | # 52 | SPARE COMPONENT |      |
| OCT   | 00000      | # 52 | SPARE COMPONENT |      |
| ECADR | RANGE      | # 53 | POS5            | DP1  |
| ECADR | RRATE      | # 53 | VEL3            | DP3  |
| ECADR | RTHETA     | # 53 | DPDEG(360)      | DP4  |
| ECADR | RANGE      | # 54 | POS5            | DP1  |
| ECADR | RRATE      | # 54 | VEL3            | DP3  |
| ECADR | RTHETA     | # 54 | DPDEG(360)      | DP4  |
| ECADR | NN1        | # 55 | WHOLE           | ARTH |
| ECADR | ELEV       | # 55 | DPDEG(360)      | DP4  |
| ECADR | CENTANG    | # 55 | DPDEG(360)      | DP4  |
| ECADR | RTEGAM2D   | # 56 | DPDEG(360)      | DP4  |
| ECADR | RTEDVD     | # 56 | VEL2            | DP4  |
| OCT   | 0          | # 56 | SPARE COMPONENT |      |
| ECADR | DELTAR     | # 57 | POS4            | DP3  |
| OCT   | 0          | # 57 | SPARE COMPONENT |      |
| OCT   | 0          | # 57 | SPARE COMPONENT |      |
| ECADR | POSTTPI    | # 58 | POS4            | DP3  |
| ECADR | DELVTPI    | # 58 | VEL3            | DP3  |
| ECADR | DELVTPI    | # 58 | VEL3            | DP3  |
| ECADR | DVLOS      | # 59 | VEL3            | DP3  |
| ECADR | DVLOS +2   | # 59 | VEL3            | DP3  |
| ECADR | DVLOS +4   | # 59 | VEL3            | DP3  |
| ECADR | GMAX       | # 60 | WHOLE           | ARTH |
| ECADR | VPRED      | # 60 | VEL2            | DP4  |
| ECADR | GAMMAEI    | # 60 | DPDEG(360)      | DP4  |
| ECADR | LAT(SPL)   | # 61 | DPDEG(360)      | DP4  |
| ECADR | LNG(SPL)   | # 61 | DPDEG(360)      | DP4  |
| ECADR | HEADSUP    | # 61 | WHOLE           | ARTH |
| ECADR | VMAGI      | # 62 | VEL2            | DP4  |
| ECADR | HDOT       | # 62 | VEL2            | DP4  |
| ECADR | ALTI       | # 62 | POS4            | DP3  |
| ECADR | RTGO       | # 63 | POS6            | DP3  |
| ECADR | VIO        | # 63 | VEL2            | DP4  |
| ECADR | TTE        | # 63 | MIN/SEC         | M/S  |
| ECADR | D          | # 64 | DRAG ACCEL      | DP2  |

|            |          |      |                              |      |
|------------|----------|------|------------------------------|------|
| ECADR      | VMAGI    | # 64 | VEL2                         | DP4  |
| ECADR      | RTGON64  | # 64 | POS6                         | DP3  |
| ECADR      | SAMPTIME | # 65 | HMS (MIXED ONLY TO KEEP CODE |      |
| ECADR      | SAMPTIME | # 65 | HMS                          | HMS  |
| ECADR      | SAMPTIME | # 65 | HMS                          | HMS  |
| ECADR      | ROLLC    | # 66 | DPDEG(360)                   | DP4  |
| ECADR      | XRNGERR  | # 66 | POS8                         | DP3  |
| ECADR      | DNRNGERR | # 66 | POS6                         | DP3  |
| ECADR      | RTGON67  | # 67 | POS6                         | DP3  |
| ECADR      | LAT      | # 67 | DPDEG(360)                   | DP4  |
| ECADR      | LONG     | # 67 | DPDEG(360)                   | DP4  |
| ECADR      | ROLLC    | # 68 | DPDEG(360)                   | DP4  |
| ECADR      | VMAGI    | # 68 | VEL2                         | DP4  |
| ECADR      | RDOT     | # 68 | VEL/2VS                      | DP4  |
| ECADR      | ROLLC    | # 69 | DPDEG(360)                   | DP4  |
| ECADR      | Q7       | # 69 | DRAG ACCEL                   | DP2  |
| ECADR      | VL       | # 69 | VEL/2VS                      | DP4  |
| ECADR      | STARCODE | # 70 | OCTAL ONLY                   | OCT  |
| ECADR      | LANDMARK | # 70 | OCTAL ONLY                   | OCT  |
| ECADR      | HORIZON  | # 70 | OCTAL ONLY                   | OCT  |
| ECADR      | STARCODE | # 71 | OCTAL ONLY                   | OCT  |
| ECADR      | LANDMARK | # 71 | OCTAL ONLY                   | OCT  |
| ECADR      | HORIZON  | # 71 | OCTAL ONLY                   | OCT  |
| ECADR      | THETZERO | # 72 | DPDEG(360)                   | DP4  |
| ECADR      | DELHITE  | # 72 | POS4                         | DP3  |
| ECADR      | OPTION2  | # 72 | WHOLE                        | ARTH |
| ECADR      | P21ALT   | # 73 | POS4                         | DP3  |
| ECADR      | P21VEL   | # 73 | VEL2                         | DP4  |
| ECADR      | P21GAM   | # 73 | DPDEG(360)                   | DP4  |
| ECADR      | ROLLC    | # 74 | DPDEG(360)                   | DP4  |
| ECADR      | VMAGI    | # 74 | VEL 2                        | DP4  |
| ECADR      | D        | # 74 | DRAG ACCEL                   | DP2  |
| ECADR      | DIFFALT  | # 75 | POS4                         | DP3  |
| ECADR      | T1TOT2   | # 75 | MIN/SEC                      | M/S  |
| ECADR      | T2TOT3   | # 75 | MIN/SEC                      | M/S  |
| OCT        | 0        | # 76 | SPARE                        |      |
| OCT        | 0        | # 76 | SPARE                        |      |
| OCT        | 0        | # 76 | SPARE                        |      |
| OCT        | 0        | # 77 | SPARE                        |      |
| OCT        | 0        | # 77 | SPARE                        |      |
| OCT        | 0        | # 77 | SPARE                        |      |
| OCT        | 0        | # 78 | SPARE                        |      |
| OCT        | 0        | # 78 | SPARE                        |      |
| # Page 282 |          |      |                              |      |
| OCT        | 0        | # 78 | SPARE                        |      |
| OCT        | 0        | # 79 | SPARE                        |      |

July 12, 2016

1519

|       |             |      |                 |        |
|-------|-------------|------|-----------------|--------|
| OCT   | 0           | # 79 | SPARE           |        |
| OCT   | 0           | # 79 | SPARE           |        |
| ECADR | TTOGO       | # 80 | MIN/SEC         | M/S    |
| ECADR | VGDISP      | # 80 | VEL2            | DP4    |
| ECADR | DVTOTAL     | # 80 | VEL2            | DP4    |
| ECADR | DEVLVLC     | # 81 | VEL3            | DP3    |
| ECADR | DEVLVLC +2  | # 81 | VEL3            | DP3    |
| ECADR | DEVLVLC +4  | # 81 | VEL3            | DP3    |
| ECADR | DEVLVLC     | # 82 | VEL3            | DP3    |
| ECADR | DEVLVLC +2  | # 82 | VEL3            | DP3    |
| ECADR | DEVLVLC +4  | # 82 | VEL3            | DP3    |
| ECADR | DELVIMU     | # 83 | VEL3            | DP3    |
| ECADR | DELVIMU +2  | # 83 | VEL3            | DP3    |
| ECADR | DELVIMU +4  | # 83 | VEL3            | DP3    |
| ECADR | DELVOV      | # 84 | VEL3            | DP3    |
| ECADR | DELVOV +2   | # 84 | VEL3            | DP3    |
| ECADR | DELVOV +4   | # 84 | VEL3            | DP3    |
| ECADR | VGBODY      | # 85 | VEL3            | DP3    |
| ECADR | VGBODY +2   | # 85 | VEL3            | DP3    |
| ECADR | VGBODY +4   | # 85 | VEL3            | DP3    |
| ECADR | DEVLVLC     | # 86 | VEL3            | DP3    |
| ECADR | DEVLVLC +2  | # 86 | VEL3            | DP3    |
| ECADR | DEVLVLC +4  | # 86 | VEL3            | DP3    |
| ECADR | MRKBUF1 +3  | # 87 | CDU DEG         | CDU    |
| ECADR | MRKBUF1 +5  | # 87 | Y OPTICS DEG    | YOPT   |
| OCT   | 0           | # 87 | SPARE COMPONENT |        |
| ECADR | STARSAV3    | # 88 | DPFRAC          | DPFRAC |
| ECADR | STARSAV3 +2 | # 88 | DPFRAC          | DPFRAC |
| ECADR | STARSAV3 +4 | # 88 | DPFRAC          | DPFRAC |
| ECADR | LANDLAT     | # 89 | DPDEG(90)       | DP3    |
| ECADR | LANDLONG    | # 89 | DPDEG(90)       | DP3    |
| ECADR | LANDALT     | # 89 | POS5            | DP1    |
| ECADR | RANGE       | # 90 | POS5            | DP1    |
| ECADR | RRATE       | # 90 | VEL3            | DP3    |
| ECADR | RTHETA      | # 90 | DPDEG(360)      | DP4    |
| ECADR | CDUS        | # 91 | CDU DEG         | CDU    |
| ECADR | CDUT        | # 91 | Y OPTICS DEG    | YOPT   |
| OCT   | 0           | # 91 | SPARE COMPONENT |        |
| ECADR | SAC         | # 92 | CDU DEG         | CDU    |
| ECADR | PAC         | # 92 | Y OPTICS DEG    | YOPT   |
| OCT   | 0           | # 92 | SPARE COMPONENT |        |
| ECADR | OGC         | # 93 | DPDEG(90)       | DP3    |
| ECADR | OGC +2      | # 93 | DPDEG(90)       | DP3    |
| ECADR | OGC +4      | # 93 | DPDEG(90)       | DP3    |
| ECADR | MRKBUF1 +3  | # 94 | CDU DEG         | CDU    |
| ECADR | MRKBUF1 +5  | # 94 | Y OPTICS DEG    | YOPT   |

# Page 283

|       |           |      |         |      |
|-------|-----------|------|---------|------|
| OCT   | 00000     | # 94 | SPARE   |      |
| ECADR | PRAXIS    | # 95 | CDU DEG | CDU  |
| ECADR | PRAXIS +1 | # 95 | CDU DEG | CDU  |
| ECADR | PRAXIS +2 | # 95 | CDU DEG | CDU  |
| ECADR | CPHIX     | # 96 | CDU DEG | CDU  |
| ECADR | CPHIX +1  | # 96 | CDU DEG | CDU  |
| ECADR | CPHIX +2  | # 96 | CDU DEG | CDU  |
| ECADR | DSPTM1    | # 97 | WHOLE   | ARTH |
| ECADR | DSPTM1 +1 | # 97 | WHOLE   | ARTH |
| ECADR | DSPTM1 +2 | # 97 | WHOLE   | ARTH |
| ECADR | DSPTM2    | # 98 | WHOLE   | ARTH |
| ECADR | DSPTM2 +1 | # 98 | FRAC    | FRAC |
| ECADR | DSPTM2 +2 | # 98 | WHOLE   | ARTH |
| ECADR | WWPOS     | # 99 | POS9    | DP3  |
| ECADR | WWVEL     | # 99 | VEL4    | DP2  |
| ECADR | WWOPT     | # 99 | WHOLE   | ARTH |

# END OF IDADDTAB

# NN SF ROUTINES

|         |     |       |      |                |
|---------|-----|-------|------|----------------|
| RUTMTAB | OCT | 16351 | # 40 | M/S, DP3, DP3  |
|         | OCT | 00142 | # 41 | CDU, ARTH      |
|         | OCT | 16347 | # 42 | DP3, DP3, DP3  |
|         | OCT | 16512 | # 43 | DP4, DP4, DP3  |
|         | OCT | 22347 | # 44 | DP3, DP3, M/S  |
|         | OCT | 24454 | # 45 | 2INT, M/S, DP4 |
|         | OCT | 00000 | # 46 | OCT, OCT       |
|         | OCT | 00553 | # 47 | ARITH1, ARITH1 |
|         | OCT | 00143 | # 48 | ARTH, ARTH     |
|         | OCT | 06347 | # 49 | DP3, DP3, ARTH |
|         | OCT | 22347 | # 50 | DP3, DP3, M/S  |
|         | OCT | 00512 | # 51 | DP4, DP4       |
|         | OCT | 00012 | # 52 | DP4            |
|         | OCT | 24344 | # 53 | DP1, DP3, DP4  |
|         | OCT | 24344 | # 54 | DP1, DP3, DP4  |
|         | OCT | 24503 | # 55 | ARTH, DP4, DP4 |
|         | OCT | 00512 | # 56 | DP4, DP4       |
|         | OCT | 00007 | # 57 | DP3            |
|         | OCT | 16347 | # 58 | DP3, DP3, DP3  |
|         | OCT | 16347 | # 59 | DP3, DP3, DP3  |
|         | OCT | 24503 | # 60 | ARTH, DP4, DP4 |
|         | OCT | 06512 | # 61 | DP4, DP4, ARTH |
|         | OCT | 16512 | # 62 | DP4, DP4, DP3  |
|         | OCT | 22507 | # 63 | DP3, DP4, M/S  |



July 12, 2016

1521

|            |       |      |                  |
|------------|-------|------|------------------|
| OCT        | 16505 | # 64 | DP2, DP4, DP3    |
| OCT        | 20410 | # 65 | HMS, HMS, HMS    |
| OCT        | 16352 | # 66 | DP4, DP3, DP3    |
| OCT        | 24507 | # 67 | DP3, DP4, DP4    |
| OCT        | 24512 | # 68 | DP4, DP4, DP4    |
| OCT        | 24252 | # 69 | DP4, DP2, DP4    |
| OCT        | 00000 | # 70 | OCT, OCT, OCT    |
| # Page 284 |       |      |                  |
| OCT        | 0     | # 71 | OCT, OCT, OCT    |
| OCT        | 06352 | # 72 | DP4, DP3, ARTH   |
| OCT        | 24507 | # 73 | DPR, DP4, DP4    |
| OCT        | 12512 | # 74 | DP4, DP4, DP2    |
| OCT        | 22447 | # 75 | DP3, M/S, M/S    |
| OCT        | 0     | # 76 | SPARE            |
| OCT        | 0     | # 77 | SPARE            |
| OCT        | 0     | # 78 | SPARE            |
| OCT        | 0     | # 79 | SPARE            |
| OCT        | 24511 | # 80 | M/S, DP4, DP4    |
| OCT        | 16347 | # 81 | DP3, DP3, DP3    |
| OCT        | 16347 | # 82 | DP3, DP3, DP3    |
| OCT        | 16347 | # 83 | DP3, DP3, DP3    |
| OCT        | 16347 | # 84 | DP3, DP3, DP3    |
| OCT        | 16347 | # 85 | DP3, DP3, DP3    |
| OCT        | 24512 | # 86 | DP4, DP4, DP4    |
| OCT        | 00302 | # 87 | CDU, YOPT        |
| OCT        | 32655 | # 88 | DPFRAC FOR EACH  |
| OCT        | 10347 | # 89 | DP3, DP3, DP1    |
| OCT        | 24344 | # 90 | DP1, DP3, DP4    |
| OCT        | 00302 | # 91 | CDU, YOPT        |
| OCT        | 00302 | # 92 | CDU, YOPT        |
| OCT        | 16347 | # 93 | DP3, DP3, DP3    |
| OCT        | 00302 | # 94 | CDU, YOPT        |
| OCT        | 04102 | # 95 | CDU, CDU, CDU    |
| OCT        | 04102 | # 96 | CDU, CDU, CDU    |
| OCT        | 06143 | # 97 | ARTH, ARTH, ARTH |
| OCT        | 06043 | # 98 | ARTH, FRAC, ARTH |
| OCT        | 06247 | # 99 | DP3, DP2, ARTH   |

# END OF RUTMTAB

SBANK= LOWSUPER

This code is written to file src/PINBALL-NOUN-TABLES.s.

## B.83 PLANETARY INERTIAL ORIENTATION

```

1522 <src/PLANETARY-INERTIAL-ORIENTATION.s 1522>≡
 # Copyright: Public domain.
 # Filename: PLANETARY_INERTIAL_ORIENTATION.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 1243-1251
 # Mod history: 2009-05-14 RSB Adapted from the Colossus249/ file of the
 # same name, using Comanche055 page images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 1243
 # PLANETARY INERTIAL ORIENTATION
 #
 # ***** RP-TO-R SUBROUTINE *****
 # SUBROUTINE TO CONVERT RP (VECTOR IN PLANETARY COORDINATE SYSTEM, EITHER
 # EARTH-FIXED OR MOON-FIXED) TO R (SAME VECTOR IN BASIC REF. SYSTEM)
 # R = MT(T) * (RP + LP X RP) MT = M MATRIX TRANSPOSE
 #
 # CALLING SEQUENCE
 # L CALL
 # L+1 RP-TO-R
 #
 # SUBROUTINES USED
 # EARTHMX, MOONMX, EARHL

```

```

#
ITEMS AVAILABLE FROM LAUNCH DATA
504LM = THE LIBRATION VECTOR L OF THE MOON AT TIME TIMSUBL, EXPRESSED
IN THE MOON-FIXED COORD. SYSTEM RADIANS BO
#
ITEMS NECESSARY FOR SUBR. USED (SEE DESCRIPTION OF SUBR.)
#
INPUT
MPAC = 0 FOR EARTH, NON-ZERO FOR MOON
0-5D = RP VECTOR
6-7D = TIME
#
OUTPUT
MPAC = R VECTOR METERS B-29 FOR EARTH, B-27 FOR MOON

 SETLOC PLANTIN
 BANK

 COUNT* $$/LUROT

RP-TO-R STQ BHIZ
 RPREXIT
 RPTORA

 CALL # COMPUTE M MATRIX FOR MOON
 MOONMX # LP=LM FOR MOON RADIANS BO
 VLOAD

 504LM
RPTORB VXV VAD
 504RPR
 504RPR
 VXM GOTO
 MMATRIX # MPAC=R=MT(T)*(RP+LPXRP)
 RPRPXXXX # RESET PUSHLOC TO 0 BEFORE EXITING
RPTORA CALL # EARTH COMPUTATIONS
 EARTHMX # M MATRIX B-1
 CALL
 EARTHLL # L VECTOR RADIANS BO
 MXV VSL1 # LP=M(T)*L RAD B-0
 MMATRIX

Page 1244
 GOTO
 RPTORB

Page 1245
***** R-TO-RP SUBROUTINE *****
SUBROUTINE TO CONVERT R (VECTOR IN REFERENCE COORD. SYSTEM) TO RP

```

```

(VECTOR IN PLANETARY COORD SYSTEM) EITHER EARTH-FIXED OR MOON-FIXED
RP = M(T) * (R - L X R)
#
CALLING SEQUENCE
L CALL
L+1 R-TO-RP
#
SUBROUTINES USED
EARTHMX, MOONMX, EARTHLM
#
INPUT
MPAC = 0 FOR EARTH, NON-ZERO FOR MOON
0-5D = R VECTOR
6-7D = TIME
#
ITEMS AVAILABLE FROM LAUNCH DATA
504LM = THE LIBRATION VECTOR L OF THE MOON AT TIME TIMSUBL, EXPRESSED
IN THE MOON-FIXED COORD. SYSTEM RADIANS BO
#
ITEMS NECESSARY FOR SUBROUTINES USED (SEE DESCRIPTION OF SUBR.)
#
OUTPUT
MPAC = RP VECTOR METERS B-29 FOR EARTH, B-27 FOR MOON

R-TO-RP STQ BHIZ
 RPREXIT
 RTORPA
 CALL
 MOONMX
 VLOAD VXM
 504LM # LP=LM
 MMATRIX
 VSL1 # L = MT(T)*LP RADIANS BO
RTORPB VXV BVSU
 504RPR
 504RPR
 MXV # M(T)*(R-LXR) B-2
 MMATRIX
RPRPXXXX VSL1 SETPD
 OD
 GOTO
 RPREXIT
RTORPA CALL # EARTH COMPUTATIONS
 EARTHMX
 CALL
 EARTHLM

```

July 12, 2016

1525

```

 GOTO # MPAC=L=(-AX,-AY,0) RAD B-0
 RTORPB

Page 1246
***** MOONMX SUBROUTINE *****
SUBROUTINE TO COMPUTE THE TRANSFORMATION MATRIX M FOR THE MOON
#
CALLING SEQUENCE
L CALL
L+1 MOONMX
#
SUBROUTINES USED
NEWANGLE
#
INPUT
6-7D = TIME
#
ITEMS AVAILABLE FROM LAUNCH DATA
BSUB0, BDOT
TIMSUB0, NODIO, NODDOT, FSUB0, FDOT
COSI = COS(I) B-1
SINI = SIN(I) B-1
I IS THE ANGLE BETWEEN THE MEAN LUNAR EQUATORIAL PLANE AND THE
PLANE OF THE ECLIPTIC (1 DEGREE 32.1 MINUTES)
#
OUTPUT
MMATRIX = 3X3 M MATRIX B-1 (STORED IN VAC AREA)

MOONMX STQ SETPD
 EARTHMX
 8D
 AXT,1 # B REQUIRES SL 0, SL 5 IN NEWANGLE
 5
DLOAD PDDL # PD 10D 8-9D=BSUB0
 BSUB0 # 10-11D=BDOT
 BDOT
PUSH CALL # PD 12D
 NEWANGLE # EXIT WITH PD 8D AND MPAC= B REVS B0
PUSH COS # PD 10D
STODL COB # PD 8D COS(B) B-1
SIN # SIN(B) B-1
STODL SOB # SETUP INPUT FOR NEWANGLE
 FSUB0 # 8-9D=FSUB0
PDDL PUSH # PD 10D THEN 12D 10-11D=FDOT
 FDOT
AXT,1 CALL # F REQUIRES SL 1, SL 6 IN NEWANGLE.
```

# Page 1247

```

4
NEWANGLE # EXIT WITH PD 8D AND MPAC= F REVS B0
STODL AVECTR +2 # SAVE F TEMP
 NODIO # 8-9D=NODIO
PDDL PUSH # PD 10D THEN 12D 10-11D=NODDOT
 NODDOT # MPAC=5
AXT,1 CALL # NODE REQUIRES SL 0, SL 5 IN NEWANGLE
 5
NEWANGLE # EXIT WITH PD 8D AND MPAC= NODI REVS B0
PUSH COS # PD 10D 8-9D= NODI REVS B0
PUSH COS # PD 12D 10-11D= COS(NODI) B-1
STORE AVECTR
DMP SL1R
 COB # COS(NODI) B-1
STODL BVECTR +2 # PD 10D 20-25D=AVECTR=COB*SIN(NODI)
DMP SL1R # SOB*SIN(NODI)
 SOB
STODL BVECTR +4 # PD 8D
SIN PUSH # PD 10D -SIN(NODI) B-1
DCOMP # 26-31D=BVECTR=COB*COS(NODI)
STODL BVECTR # PD 8D SOB*COS(NODI)
 AVECTR +2 # MOVE F FROM TEMP LOC. TO 504F
STODL 504F
DMP SL1R
 COB
STODL AVECTR +2
SINNODI # 8-9D=SIN(NODI) B-1
DMP SL1R
 SOB
STODL AVECTR +4 # 0
 HI6ZEROS # 8-13D= CVECTR= -SOB B-1
PDDL DCOMP # PD 10D COB
 SOB
PDDL PDVL # PD 12D THEN PD 14D
 COB
 BVECTR
VXSC PDVL # PD 20D BVECTR*SINI B-2
 SINI
 CVECTR
VXSC VAD # PD 14D CVECTR*COSI B-2
 COSI
VSL1
STOVL MMATRIX +12D # PD 8D M2=BVECTR*SINI+CVECTR*COSI B-1
VXSC PDVL # PD 14D
 SINI # CVECTR*SINI B-2

```

1527

```

 BVECTR
VXSC VSU # PD 8D BVECTR*COSI B-2
 COSI
VSL1 PDDL # PD 14D
 504F # 8-13D=DVECTR=BVECTR*COSI-CVECTR*SINI B-1
COS VXSC
 DVECTR
PDDL SIN # PD 20D 14-19D= DVECTR*COSF B-2
 504F
VXSC VSU # PD 14D AVECTR*SINF B-2
 AVECTR
VSL1
STODL MMATRIX +6 # M1= AVECTR*SINF-DVECTR*COSF B-1
 504F

Page 1248

SIN VXSC # PD 8D
PDDL COS # PD 14D 8-13D=DVECTR*SINF B-2
 504F
VXSC VAD # PD 8D AVECTR*COSF B-2
 AVECTR
VSL1 VCOMP
STCALL MMATRIX # M0= -(AVECTR*COSF+DVECTR*SINF) B-1
 EARTHMX

COMPUTE X=X0+(XDOT)(T+T0)
8-9D= X0 (REVS B-0), PUSHLOC SET AT 12D
10-11D=XDOT (REVS/CSEC) SCALED B+23 FOR WEARTH,B+28 FOR NODDOT AND BDOT
AND B+27 FOR FDOT
X1=DIFFERENCE IN 23 AND SCALING OF XDOT, =0 FOR WEARTH, 5 FOR NDDOT AND
BDOT AND 4 FOR FDOT
6-7D=T (CSEC B-28), TIMSUB0= (CSEC B-42 TRIPLE PREC.)

NEWANGLE DLOAD SR # ENTER PD 12D
 6D
 14D
TAD TLOAD # CHANGE MODE TO TP
 TIMSUB0
 MPAC
STODL TIMSUBM # T+T0 CSEC B-42
 TIMSUBM +1
DMP
SL* DAD # PD 10D MULT BY XDOT IN 10-11D
 5,1 # PD 8D ADD X0 IN 8-9D AFTER SHIFTING
 # SUCH THAT SCALING IS B-0
PUSH SLOAD # PD 10D SAVE PARTIAL (X0+XDOT*T) IN 8-9D
 TIMSUBM
SL DMP

```

```

 9D
 10D # XDOT
SL* DAD # PD 8D SHIFT SUCH THAT THIS PART OF
 10D,1 # IS SCALED REVS/CSEC B-0
BOV +1 # TURN OFF OVERFLOW IF SET BY SHIFT
 +1 # INSTRUCTION BEFORE EXITING.
RVQ +1 # MPAC=X= X0+(XDOT)(T+T0) REVS B0

Page 1249
***** EARTHMX SUBROUTINE *****
SUBROUTINE TO COMPUTE THE TRANSFORMATION MATRIX M FOR THE EARTH
#
CALLING SEQUENCE
L CALL
L+1 EARTHMX
#
SUBROUTINE USED
NEWANGLE
#
INPUT
INPUT AVAILABLE FROM LAUNCH DATA AZO REVS B-0
TEPHEM CSEC B-42
6-7D= TIME CSEC B-28
#
OUTPUT
MMATRIX= 3X3 M MATRIX B-1 (STORED IN VAC AREA)

EARTHMX STQ SETPD # SET 8-9D=AZO
 EARTHMX
 8D # 10-11D=WEARTH
 AXT,1 # FOR SL 5, AND SL 10 IN NEWANGLE
 0
DLOAD PDDL # LEAVING PD SET AT 12D FOR NEWANGLE
 AZO
 WEARTH
PUSH CALL
 NEWANGLE
SETPD PUSH # 18-19D=504AZ
 18D #
 COS COS(AZ) SIN(AZ)
COS PDDL # 20-37D= MMATRIX= -SIN(AZ) COS(AZ)
 504AZ #
 0 0
SIN PDDL
 HI6ZEROS
PDDL SIN
 504AZ
DCOMP PDDL

```



```

 504AZ
COS PDVL
 HI6ZEROS
PDDL PUSH
 HIDPHALF
GOTO EARTHMX

```

```

Page 1250
***** EARTHL SUBROUTINE *****
SUBROUTINE TO COMPUTE L VECTOR FOR EARTH
#
CALLING SEQUENCE
L CALL
L+1 EARTHL
#
INPUT
AXO,AYO SET AT LAUNCH TIME WITH AYO IMMEDIATELY FOLLOWING AXO IN CORE
#
OUTPUT
-AX
MPAC= -AY RADIANS B-0
0

```

```

EARTHL DLOAD DCOMP
 AXO
 STODL 504LPL
 -AYO
 STODL 504LPL +2
 HI6ZEROS
 STOVL 504LPL +4
 504LPL
 RVQ

```

```

Page 1251
CONSTANTS AND ERASABLE ASSIGNMENTS

1B1 = DP1/2 # 1 SCALED B-1
COSI 2DEC .99964173 B-1 # COS(5521.5 SEC) B-1

SINI 2DEC .02676579 B-1 # SIN(5521.T SEC) B-1

RPREXIT = S1 # R-TO-RP AND RP-TO-R SUBR EXIT
EARTHMX = S2 # EARTHMX, MOONMX SUBR. EXITS
504RPR = 0D # 6 REGS R OR RP VECTOR
SINNODI = 8D # 2 SIN(NODI)

```

1530

July 12, 2016

|         |      |                 |                                   |                              |
|---------|------|-----------------|-----------------------------------|------------------------------|
| DVECTR  | =    | 8D              | # 6                               | D VECTOR MOON                |
| CVECTR  | =    | 8D              | # 6                               | C VECTR MOON                 |
| 504AZ   | =    | 18D             | # 2                               | AZ                           |
| TIMSUBM | =    | 14D             | # 3                               | TIME SUB M (MOON) T+10 IN GE |
| 504LPL  | =    | 14D             | # 6                               | L OR LP VECTOR               |
| AVECTR  | =    | 20D             | # 6                               | A VECTOR (MOON)              |
| BVECTR  | =    | 26D             | # 6                               | B VECTOR (MOON)              |
| MMATRIX | =    | 20D             | # 18                              | M MATRIX                     |
| COB     | =    | 32D             | # 2                               | COS(B) B-1                   |
| SOB     | =    | 34D             | # 2                               | SIN(B) B-1                   |
| 504F    | =    | 6D              | # 2                               | F(MOON)                      |
| NODDOT  | 2DEC | -.457335121 E-2 | # REVS/CSEC B+28=-1.07047011 E-8  | RAD/SEC                      |
| FDOT    | 2DEC | .570863327      | # REVS/CSEC B+27= 2.67240410 E-6  | RAD/SEC                      |
| BDOT    | 2DEC | -3.07500686 E-8 | # REVS/CSEC B+28=-7.19757301 E-14 | RAD/SEC                      |
| NODIO   | 2DEC | .986209434      | # REVS B-0 = 6.19653663041        | RAD                          |
| FSUB0   | 2DEC | .829090536      | # REVS B-0 = 5.20932947829        | RAD                          |
| BSUB0   | 2DEC | .0651201393     | # REVS B=0 = 0.40916190299        | RAD                          |
| WEARTH  | 2DEC | .973561595      | # REVS/CSEC B+23= 7.29211494 E-5  | RAD/SEC                      |

This code is written to file `src/PLANETARY-INERTIAL-ORIENTATION.s`.

## B.84 POWERED FLIGHT SUBROUTINES

```

1531 <src/POWERED-FLIGHT-SUBROUTINES.s 1531>≡
 # Copyright: Public domain.
 # Filename: POWERED_FLIGHT_SUBROUTINES.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 1365-1372
 # Mod history: 2009-05-10 SN (Sergio Navarro). Started adapting
 # from the Colossus249/ file of the same
 # name, using Comanche055 page images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 1365

 BANK 14 # SAME FBANK AS THE FINDCDUD SUB-PROGRAM
 SETLOC POWFLITE
 BANK

 EBANK= DEXDEX
 COUNT* $$/POWFL

 # CDUTRIG, CDUTRIG1, CDUTRIG2, AND CD*GR*GS ALL COMPUTE THE SINES AND
 # COSINES OF THREE 2'S COMPLEMENT ANGLES AND PLACE THE RESULT, DOUBLE
 # PRECISION, IN THE SAME ORDER AS THE INPUTS, AT SINCDU AND COSCDU. AN
 # ADDITIONAL OUTPUT IS THE 1'S COMPLEMENT ANGLES AT CDUSPOT. THESE

```

```

ROUTINES GO OUT OF THEIR WAY TO LEAVE THE MPAC AREA AS THEY FIND IT.
EXCEPT FOR THE GENERALLY UNIMPORTANT MPAC +2. THEY DIFFER ONLY IN
WHERE THEY GET THE ANGLES, AND IN METHOD OF CALLING.
#
CDUTRIG (AND CDUTRIG1, WHICH CAN BE CALLED IN BASIC) COMPUTE THE
SINES AND COSINES FROM THE CURRENT CONTENTS OF THE CDU REGISTERS.
THE CONTENTS OF CDUTEMP, ETC., ARE NOT TOUCHED SO THAT THEY MAY
CONTINUE TO FORM A CONSISTENT SET WITH THE LATEST PIPA READINGS.
#
CDUTRIG1 IS LIKE CDUTRIG EXCEPT THAT IT CAN BE CALLED IN BASIC.
#
CD*TR*GS FINDS CDU VALUES IN CDUSPOT RATHER THAN IN CDUTEMP. THIS
ALLOWS USERS TO MAKE TRANSFORMATIONS USING ARBITRARY ANGLES, OR REAL
ANGLES IN AN ORDER OTHER THAN X Y Z. A CALL TO THIS ROUTINE IS
NECESSARY IN PREPARATION FOR A CALL TO AX*SR*T IN EITHER OF ITS TWO
MODES (SMNB OR NBSM). SINCE AX*SR*T EXPECTS TO FIND THE SINES AND
COSINES IN THE ORDER Y Z X THE ANGLES MUST HAVE BEEN PLACED IN CDUSPOT
IN THIS ORDER. CD*TR*GS NEED NOT BE REPEATED WHEN AX*SR*T IS CALLED
MORE THAN ONCE, PROVIDED THE ANGLES HAVE NOT CHANGED. NOTE THAT SINCE
IT CLOBBERS BUF2 (IN THE SINE AND COSINE ROUTINES) CD*TR*GS CANNOT BE
CALLED USING BANKCALL. SORRY.
#
CD*TR*G IS LIKE CD*TR*GS EXCEPT THAT IT CAN BE CALLED IN
INTERPRETIVE.

```

|             |        |            |
|-------------|--------|------------|
| CDUTRIG     | EXIT   |            |
|             | TC     | CDUTRIGS   |
|             | TC     | INTPRET    |
|             | RVQ    |            |
| CD*TR*G     | EXIT   |            |
|             | TC     | CD*TR*GS   |
|             | TC     | INTPRET    |
|             | RVQ    |            |
| CDUTRIGS    | CA     | CDUX       |
|             | TS     | CDUSPOT +4 |
|             | CA     | CDUY       |
|             | TS     | CDUSPOT    |
| # Page 1366 |        |            |
|             | CA     | CDUZ       |
|             | TS     | CDUSPOT +2 |
| CD*TR*GS    | EXTEND |            |
|             | QXCH   | TEM2       |
|             | CAF    | FOUR       |

July 12, 2016

1533

```
TR*GL**P MASK SIX # MAKE IT EVEN AND SMALLER
 TS TEM3
 INDEX TEM3
 CA CDUSPOT
 DXCH MPAC # STORING 2'S COMP ANGLE, LOADING MPAC
 DXCH VBUF +4 # STORING MPAC FOR LATER RESTORATION
 TC USPRCADR
 CADR CDULOGIC
 EXTEND
 DCA MPAC
 INDEX TEM3
 DXCH CDUSPOT # STORING 1'S COMPLEMENT ANGLE
 TC USPRCADR
 CADR COSINE
 DXCH MPAC
 INDEX TEM3
 DXCH COSCDU # STORING COSINE
 EXTEND
 INDEX TEM3
 DCA CDUSPOT # LOADING 1'S COMPLEMENT ANGLE
 TC USPRCADR
 CADR SINE +1 # SINE +1 EXPECTS ARGUMENT IN A AND L
 DXCH VBUF +4 # BRINGING UP PRIOR MPAC TO BE RESTORED
 DXCH MPAC
 INDEX TEM3
 DXCH SINCDU
 CCS TEM3
 TCF TR*GL**P
 TC TEM2
```

# Page 1367

```

QUICTRIG, INTENDED FOR GUIDANCE CYCLE USE WHERE TIME IS CRITICAL, IS A MUCH FASTER VERSION OF
QUICTRIG COMPUTES AND STORES THE SINES AND COSINES OF THE 2'S COMPLEMENT ANGLES AT CDUSPOT, C
AND CDUSPOT +4. UNLIKE CD*TR*GS, QUICTRIG DOES NOT LEAVE THE 1'S COMPLEMENT VERSIONS OF THE
CDUSPOT. QUICTRIG'S EXECUTION TIME IS 4.1 MS; THIS IS 10 TIMES AS FAST AS CD*TR*GS. QU
CALLED FROM INTERPRETIVE AS AN RTB OP-CODE, OR FROM BASIC VIA BANKCALL OR IBNKCALL.
```

```
QUICTRIG INHINT # INHINT SINCE DAP USES THE SAME TEMPS
 EXTEND
 QXCH ITEMP1
 CAF FOUR
 +4 MASK SIX
 TS ITEMP2
 INDEX ITEMP2
 CA CDUSPOT
 TC SPSIN
```

```

EXTEND
MP BIT14 # SCALE DOWN TO MATCH INTERPRETER OUTPUTS
INDEX ITEMP2
DXCH SINCDU
INDEX ITEMP2
CA CDUSPOT
TC SPCOS
EXTEND
MP BIT14
INDEX ITEMP2
DXCH COSCDU
CCS ITEMP2
TCF QUICTRIG +4
CA ITEMP1
RELINT
TC A

Page 1368

THESE INTERFACE ROUTINES MAKE IT POSSIBLE TO CALL AX*SR*T, ETC., IN
INTERPRETIVE. LATER, WHERE POSSIBLE, THEY WILL BE ELIMINATED.
#
NBSM WILL BE THE FIRST TO GO. IT SHOULD NOT BE USED.

NBSM STQ
 X2
 LXC,1 VLOAD*
 S1 # BASE ADDRESS OF THE CDU ANGLES IS IN S1
 0,1
 STOVL CDUSPOT
 32D # VECTOR TO BE TRANSFORMED IS IN 32D
 CALL TRG*NBSM
 STCALL 32D # SINCE THERE'S NO STGOTO
 X2

THESE INTERFACE ROUTINES ARE PERMANENT. ALL RESTORE USER'S EBANK
SETTING. ALL ARE STRICT INTERPRETIVE SUBROUTINES, CALLED USING "CALL",
RETURNING VIA QPRET. ALL EXPECT AND RETURN THE VECTOR TO BE TRANSFOR-
MED INTERPRETER-STYLE IN MPAC: COMPONENTS AT MPAC, MPAC +3, AND MPAC +5.
#
TRG*SMNB AND TRG*NBSM BOTH EXPECT TO SEE THE 2'S COMPLEMENT ANGLES
AT CDUSPOT (ORDER Y Z X, AT CDUSPOT, CDUSPOT +2, AND CDUSPOT +4. ODD
LOCATIONS NEED NOT BE ZEROED). TRG*NBSM DOES THE NB TO SM TRANSFOR-
MATION: TRG*SMNB, VICE VERSA.
#
CDU*NBSM DOES ITS TRANSFORMATION USING THE PRESENT CONTENTS OF

```

July 12, 2016

1535

# THE CDL COUNTERS. OTHERWISE IT IS LIKE TRG\*NBSM.

#

# CDU\*SMNB IS THE COMPLEMENT OF CDU\*NBSM.

CDU\*SMNB EXIT

TC CDUTRIGS

TCF C\*MM\*N1

TRG\*SMNB EXIT

TC CD\*TR\*GS

C\*MM\*N1 TC MPACVBUF # AX\*SR\*T EXPECTS VECTOR IN VBUF

CS THREE # SIGNAL FOR SM TO NB TRANSFORMATION.

C\*MM\*N2 TC AX\*SR\*T

TC INTPRET

VLOAD RVQ

VBUF

CDU\*NBSM EXIT

TC CDUTRIGS

# Page 1369

TCF C\*MM\*N3

TRG\*NBSM EXIT

TC CD\*TR\*GS

C\*MM\*N3 TC MPACVBUF # FOR AX\*SR\*T

CA THREE # SIGNAL FOR NB TO SM TRANSFORMATION

TCF C\*MM\*N2

# \*NBSM\* AND \*SMNB\* EXPECT TO SEE THE SINES AND COSINES (AT SINCDU  
# AND COSCDU) RATHER THAN THE ANGLES THEMSELVES. OTHERWISE THEY ARE  
# LIKE TRG\*NBSM AND TRG\*SMNB.

#

# NOTE THAT JUST AS CD\*TR\*GS NEED BE CALLED ONLY ONCE FOR EACH SERIES  
# OF TRANSFORMATIONS USING THE SAME ANGLES, SO TOO ONLY ONE OF TRG\*NBSM  
# AND TRG\*SMNB NEED BE CALLED FOR EACH SERIES. FOR SUBSEQUENT TRANFOR-  
# MATIONS USE \*NBSM\* AND \*SMNB\*.

\*SMNB\* EXIT

TCF C\*MM\*N1

\*NBSM\* EXIT

TCF C\*MM\*N3

# AX\*SR\*T COMBINES THE OLD SMNB AND NBSM. FOR THE NB TO SM

```

TRANSFORMATION, ENTER WITH +3 IN A. FOR SM TO NB, ENTER WITH -3.
THE VECTOR TO BE TRANSFORMED ARRIVES, AND IS RETURNED, IN VBUF.
AX*SR*T EXPECTS TO FIND THE SINES AND COSINES OF THE ANGLES OF ROTATION
AT SINCDU AND COSCDU, IN THE ORDER Y Z X. A CALL TO CD*TR*GS, WITH
THE 2'S COMPLEMENT ANGLES (ORDER Y Z X) AT CDUSPOT, WILL TAKE CARE OF
THIS. HERE IS A SAMPLE CALLING SEQUENCE:--
TC CDUTRIGS
CS THREE # ("CA THREE" FOR NBSM)
TC AX*SR*T
THE CALL TO CD*TR*GS NEED NOT BE REPEATED, WHEN AX*SR*T IS CALLED MORE
THAN ONCE, UNLESS THE ANGLES HAVE CHANGED.
#
AX*SR*T IS GUARANTEED SAFE ONLY FOR VECTORS OF MAGNITUDE LESS THAN
UNITY. A LOOK AT THE CASE IN WHICH A VECTOR OF GREATER MAGNITUDE
HAPPENS TO LIE ALONG AN AXIS OF THE SYSTEM TO WHICH IT IS TO BE TRANS-
FORMED CONVINCES ONE THAT THIS IS A RESTRICTION WHICH MUST BE ACCEPTED.

AX*SR*T TS DEXDEX # WHERE IT BECOMES THE INDEX OF INDEXES.
 EXTEND
 QXCH RTNSAVER

R*TL**P CCS DEXDEX # +3 --> 0 -3 --> 2
 CS DEXDEX # THUS: +2 --> 1 -2 --> 1
 AD THREE # +1 --> 2 -1 --> 0

Page 1370

 EXTEND
 INDEX A
 DCA INDEXI
 DXCH DEXI

 CA ONE
 TS BUF
 EXTEND
 INDEX DEX1
 DCS VBUF
 TCF LOOP1 # REALLY BE A SUBTRACT, AND VICE VERSA

LOOP2 DXCH BUF # LOADING VECTOR COMPONENT, STORING INDEX

LOOP1 DXCH MPAC
 CA SINESLOC
 AD DEX1
 TS ADDRWD

 TC DMPSUB # MULTIPLY BY SIN(CDUANGLE)
 CCS DEXDEX

```



July 12, 2016

1537

```
DXCH MPAC # NBSM CASE
TCF +3
EXTEND
DCS MPAC
DXCH TERM1TMP

CA SIX # SINCDU AND COSCDU (EACH 6 WORDS) MUST
ADS ADDRWD # BE CONSECUTIVE AND IN THAT ORDER

EXTEND
INDEX BUF
INDEX DEX1
DCA VBUF
DXCH MPAC
TC DMPSUB # MULTIPLY BY COS(CDUANGLE)
DXCH MPAC
DAS TERM1TMP
DXCH TERM1TMP
DDOUBL
INDEX BUF
INDEX DEX1
DXCH VBUF
DXCH BUF # LOADING INDEX, STORING VECTOR COMPONENT
CCS A # 'CAUSE THAT'S WHERE THE INDEX NOW IS
TCF LOOP2

EXTEND
DIM DEXDEX # DECREMENT MAGNITUDE PRESERVING SIGN

Page 1371
TSTPOINT CCS DEXDEX # ONLY THE BRANCHING FUNCTION IS USED
TCF R*TL**P
TC RTNSAVER
TCF R*TL**P
TC RTNSAVER

SINESLOC ADRES SINCDU # FOR USE IN SETTING ADDRWD

INDEXI DEC 4 # ***** DON'T *****
 DEC 2 # ***** TOUCH *****
 DEC 0 # ***** THESE *****
 DEC 4 # ***** CONSTANTS *****
```

# \*\*\*\*\*

# Page 1372

# THIS SUBROUTINE COMPUTES INCREMENTAL CHANGES IN CDU(GIMBAL) ANGLES FROM INCREMENTAL CHANGES A

```
REQUIRES SM INCREMENTS AS A DP VECTOR SCALED AT ONE REVOLUTION(DTHETASM,+2,+4). S
SINCDU,+2,+4 AND COSCDU,+2,+4 RESPECTIVELY, SCALED TO ONE HALF. CDU INCREMENTS ARE
ONE REVOLUTION.
```

```
#
* COS(IGA)SEC(MGA) 0 -SIN(IGA)SEC(MGA) *
*
* -COS(IGA)TAN(MGA) 1 SIN(IGA)TAN(MGA) *
*
* SIN(IGA) 0 COS(IGA) *
```

```
BANK 14
SETLOC POWFLIT1
BANK
```

```
SMCDURES
```

```
DLOAD DMP
 DTHETASM
 COSCDUY
```

```
PDDL DMP
 DTHETASM +4
 SINCDUY
```

```
BDSU
DDV
```

```
COSCDUZ
STORE DCDU
```

```
DMP SL1 # SCALE
 SINCDUZ
```

```
BDSU
```

```
DTHETASM +2
STODL DCDU +2
 DTHETASM
```

```
DMP PDDL
 SINCDUY
 DTHETASM +4
```

```
DMP DAD
 COSCDUY
```

```
SL1
STORE DCDU +4
RVQ
```

This code is written to file `src/POWERED-FLIGHT-SUBROUTINES.s`.

## B.85 Q R-AXIS RCS AUTOPILOT

```

1539 <src/Q-R-AXIS-RCS-AUTOPILOT.s 1539>≡
 # Copyright: Public domain.
 # Filename: Q_R-AXIS_RCS_AUTOPILOT.agc
 # Purpose: Part of the source code for Luminary 1A build 099.
 # It is part of the source code for the Lunar Module's (LM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 1442-1459
 # Mod history: 2009-05-27 RSB Adapted from the corresponding
 # Luminary131 file, using page
 # images from Luminary 1A.
 # 2009-06-07 RSB Corrected "DEC 96.0" to "DEC 96", since
 # the former is not compatible with yaYUL.
 #
 # This source code has been transcribed or otherwise adapted from
 # digitized images of a hardcopy from the MIT Museum. The digitization
 # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
 # the Museum. Many thanks to both. The images (with suitable reduction
 # in storage size and consequent reduction in image quality as well) are
 # available online at www.ibiblio.org/apollo. If for some reason you
 # find that the images are illegible, contact me at info@sandroid.org
 # about getting access to the (much) higher-quality images which Paul
 # actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
 # 16:27 JULY 14, 1969

 # Page 1442

 BANK 17
 SETLOC DAPS2
 BANK

 EBANK= CDUXD

 COUNT* $$/DAPQR

 CALLQERR CA BIT13 # CALCULATE Q,R ERRORS UNLESS THESE AXES
 EXTEND # ARE IN MANUAL RATE COMMAND.
 RAND CHAN31
 CCS A

```

```

TCF +5 # IN AUTO COMPUTE Q,R ERRORS
CS DAPBOOLS # IN MANUAL RATE COMMAND?
MASK OURRCBIT
EXTEND
BZF Q,RORGTS # IF SO BYPASS CALCULATION OF ERROS.
TC QERRCALC

Q,RORGTS CCS COTROLER # CHOOSE CONTROL SYSTEM FOR THIS DAP PASS:
TCF GOTOGTS # GTS (ALTERNATES WITH RCS WHEN DOCKED)
TCF TRYGTS # GTS IF ALLOWED, OTHERWISE RCS
RCS CAF ZERO # RCS (TRYGTS MAY BRANCH TO HERE)
TS COTROLER

DXCH EDOTQ
TC ROT-TOUV
DXCH OMEGAU

X - TRANSLATION
#
INPUT: BITS 7,8 OF CH31 (TRANSLATION CONTROLLER)
ULLAGER
APSFLAG, DRIFTBIT
ACC40R2X, ACRBTRAN
#
OUTPUT: NEXTU, NEXTV CODES OF TRANSLATION FOR AFTER ROTATION
SENSETYP TELL ROTATION DIRECTION AND DESIRE
#
X-TRANS POLICIES ARE EITHER 4 JETS OR A DIAGONAL PAIR. IN 2-JET TRANSLATION THE S
WILL OVERRIDE THIS SPECIFICATION. AN ALARM RESULTS WHEN NO POLICY IS AVAILABLE BEC

SENSEGET CA BIT7 # INPUT BITS OVERRIDE THE INTERNAL BITS
EXTEND # SENSETYP WILL NOT OPPOSE ANYTRANS
RAND CHAN31
EXTEND
BZF +XORULGE

Page 1443

CA BIT8
EXTEND
RAND CHAN31
EXTEND
BZF -XTRANS

CA ULLAGER
MASK DAPBOOLS
CCS A
TCF +XORULGE

```

July 12, 2016

1541

|             |        |          |                                        |
|-------------|--------|----------|----------------------------------------|
|             | TS     | NEXTU    | # STORE NULL TRANSLATION POLICIES      |
|             | TS     | NEXTV    |                                        |
|             | CS     | DAPBOOLS | # BURNING OR DRIFTING?                 |
|             | MASK   | DRIFTBIT |                                        |
|             | EXTEND |          |                                        |
|             | BZF    | TSSENSE  |                                        |
|             | CA     | FLGWRD10 | # DPS (INCLUDING DOCKED) OR APS?       |
|             | MASK   | APSFLBIT |                                        |
|             | CCS    | A        |                                        |
|             | CAF    | TWO      | # FAVOR +X JETS DURING AN APS BURN.    |
| TSSENSE     | TS     | SENSETYP |                                        |
|             | TCF    | QRCONTRL |                                        |
| +XORULGE    | CAF    | ONE      |                                        |
| -XTRANS     | AD     | FOUR     |                                        |
|             | TS     | ROTINDEX |                                        |
|             | AD     | NEG3     |                                        |
|             | TS     | SENSETYP | # FAVOR APPROPRIATE JETS DURING TRANS. |
|             | CA     | DAPBOOLS |                                        |
|             | MASK   | ACC40R2X |                                        |
|             | CCS    | A        |                                        |
|             | TCF    | TRANS4   |                                        |
|             | CA     | DAPBOOLS |                                        |
|             | MASK   | AORBTRAN |                                        |
|             | CCS    | A        |                                        |
|             | CA     | ONE      | # THREE FOR B                          |
|             | AD     | TWO      | # TWO FOR A SYSTEM 2 JET X TRANS       |
| TSNUMBRT    | TS     | NUMBERT  |                                        |
|             | TC     | SELCTSUB |                                        |
|             | CCS    | POLYTEMP |                                        |
|             | TCF    | +3       |                                        |
|             | TC     | ALARM    |                                        |
|             | OCT    | 02002    |                                        |
|             | CA     | 00314OCT |                                        |
|             | MASK   | POLYTEMP |                                        |
| TSNEXTS     | TS     | NEXTU    |                                        |
| # Page 1444 | CS     | 00314OCT |                                        |
|             | MASK   | POLYTEMP |                                        |
|             | TS     | NEXTV    |                                        |

# Q,R-AXES RCS CONTROL MODE SELECTION

```

SWITCHES INDICATION WHEN SET
BIT13/CHAN31 AUTO, GO TO ATTSTEER
PULSES MINIMUM IMPULSE MODE
(OTHERWISE) RATE COMMAND/ATTITUDE HOLD MODE

QRCONTRL CA BIT13 # CHECK MODE SELECT SWITCH.
 EXTEND
 RAND CHAN31 # BITS INVERTED
 CCS A
 TCF ATTSTEER
CHKBIT10 CAF PULSES # PULSES = 1 FOR MIN IMP USE OF RHC
 MASK DAPBOOLS
 EXTEND
 BZF CHEKSTIK # IN ATT-HOLD/RATE-COMMAND IF BIT10=0

MINIMUM IMPULSE MODE

 INHINT
 TC IBNKCALL
 CADR ZATTEROR
 CA ZERO
 TS QERROR
 TS RERROR # FOR DISPLAYS
 RELINT

 EXTEND
 READ CHAN31
 TS TEMP31 # IS EQUAL TO DAPTEMP1
 CCS OLDQRMIN
 TCF CHECKIN

FIREQR CA TEMP31
 MASK BIT1
 EXTEND
 BZF +QMIN

 CA TEMP31
 MASK BIT2
 EXTEND
 BZF -QMIN

 CA TEMP31
 MASK BIT5

Page 1445

 EXTEND
 BZF +RMIN

```

|         |        |          |                                       |
|---------|--------|----------|---------------------------------------|
|         | CA     | TEMP31   |                                       |
|         | MASK   | BIT6     |                                       |
|         | EXTEND |          |                                       |
|         | BZF    | -RMIN    |                                       |
|         | TCF    | XTRANS   |                                       |
| CHECKIN | CS     | TEMP31   |                                       |
|         | MASK   | OCT63    |                                       |
|         | TS     | OLDQRMIN |                                       |
|         | TCF    | XTRANS   |                                       |
| +QMIN   | CA     | 14MS     |                                       |
|         | TS     | TJU      |                                       |
|         | CS     | 14MS     |                                       |
|         | TCF    | MINQR    |                                       |
| -QMIN   | CS     | 14MS     |                                       |
|         | TS     | TJU      |                                       |
|         | CA     | 14MS     |                                       |
|         | TCF    | MINQR    |                                       |
| +RMIN   | CA     | 14MS     |                                       |
|         | TCF    | +2       |                                       |
| -RMIN   | CS     | 14MS     |                                       |
|         | TS     | TJU      |                                       |
| MINQR   | TS     | TJV      |                                       |
|         | CA     | MINADR   |                                       |
|         | TS     | RETJADR  |                                       |
|         | CA     | ONE      |                                       |
|         | TS     | OLDQRMIN |                                       |
| MINRTN  | TS     | AXISCTR  |                                       |
|         | CA     | DAPBOOLS |                                       |
|         | MASK   | CSMDOCKD |                                       |
|         | EXTEND |          |                                       |
|         | BZF    | MIMRET   |                                       |
|         | INDEX  | AXISCTR  | # IF DOCKED, USE 60MS MINIMUM IMPULSE |
|         | CCS    | TJU      |                                       |
|         | CA     | 60MS     |                                       |
|         | TCF    | +2       |                                       |
|         | CS     | 60MS     |                                       |
|         | INDEX  | AXISCTR  |                                       |
|         | TS     | TJU      |                                       |
| MIMRET  | CA     | DAPBOOLS |                                       |
|         | MASK   | AORBTRAN |                                       |
|         | CCS    | A        |                                       |
|         | CA     | ONE      |                                       |

```

Page 1446
AD TWO
TS NUMBERT

TCF AFTERTJ

60MS DEC 96 # RSB 2009 -- was 96.0.
MINADR GENADR MINRTN
OCT63 OCT 63
14MS = +TJMINT6

TRANS4 CA FOUR
 TCF TSNUMBRT

RATE COMMAND MODE:
#
DESCRIPTION (SAME AS P-AXIS)

CHEKSTIK TS INGTS # NOT IN GTS WHEN IN ATT HOLD
 CS ONE # 1/ACCS WILL DO THE NULLING DRIVES
 TS COTROLER # COME BACK TO RCS NEXT TIME
 CA BIT15
 MASK CH31TEMP
 EXTEND
 BZF RHCACTIV # BRANCH IF OUT OF DETENT.
 CA OURRCBIT # *****
 MASK DAPBOOLS # *IN DETENT* CHECK FOR MANUAL CONTROL
 EXTEND # ***** LAST TIME.
 BZF STILLRCS
 CS BIT9
 MASK RCSFLAGS
 TS RCSFLAGS # BIT 9 IS 0.
 TCF DAMPING
40CYCL OCT 50
1/10S OCT 1
LINRAT DEC 46

=====

DAMPING CA ZERO
 TS SAVEHAND
 TS SAVEHAND +1
RHCACTIV CCS SAVEHAND # *****
 TCF +3 # Q,R MANUAL CONTROL WC = A*(B+|D|)*D
 TCF +2 # *****
 TCF +1
 DOUBLE # WHERE

```



July 12, 2016

1545

```
DOUBLE
AD LINRAT
EXTEND
MP SAVEHAND
CA L
EXTEND
MP STIKSENS
XCH QLAST
COM
Page 1447
AD QLAST
TS DAPTEMP3
CCS SAVEHAND +1
TCF +3
TCF +2
TCF +1
DOUBLE
DOUBLE
AD LINRAT
EXTEND
MP SAVEHAND +1
CA L
EXTEND
MP STIKSENS
XCH RLAST
COM
AD RLAST
TS DAPTEMP4
CS QLAST
AD OMEGAQ
TS QRATEDIF
CS RLAST
AD OMEGAR
TS RRATEDIF
ENTERQR DXCH QRATEDIF
TC ROT-TOUV
DXCH URATEDIF
CCS DAPTEMP3
TC +3
TC +2
TC +1
AD -RATEDB
EXTEND
BZMF +2
TCF ENTERUV -2
CCS DAPTEMP4
WC = COMMANDED ROTATIONAL RATE
A = QUADRATIC SENSITIVITY FACTOR
B = LINEAR/QUADRATIC SENSITIVITY
|D| = ABS. VALUE OF DEFLECTION
D = HAND CONTROLLER DEFLECTION
COMMAND Q RATE, SCALED 45 DEG/SEC
INTERVAL.
TRANSFORM RATES FROM Q,R TO U,V AXES
CHECK IF Q COMMAND CHANGE EXCEEDS
BREAKOUT LEVEL. IF NOT, CHECK R.
BREAKOUT LEVEL EXCEEDED. DIRECT RATE.
R COMMAND BREAKOUT CHECK.
```

```

TC +3
TC +2
TC +1
AD -RATEDB
EXTEND
BZMF +2
TCF ENTERUV -2 # BREAKOUT LEVEL EXCEEDED. DIRECT RATE.
CA RCSFLAGS # BREAKOUT LEVEL NOT EXCEEDED. CHECK FOR
MASK QRBIT # DIRECT RATE CONTROL LAST TIME.
EXTEND
BZF +2
TCF ENTERUV # CONTINUE DIRECT RATE CONTROL.
TCF STILLRCS # PSEUDO-AUTO CONTROL.
CA 40CYCL

Page 1448
ENTERUV TS TCQR
 INHINT # DIRECT RATE CONTROL
 TC IBNKCALL
 FCADR ZATTEROR
 RELINT
 CA ZERO
 TS DYERROR
 TS DYERROR +1
 TS DZERROR
 TS DZERROR +1
 CCS URATEDIF
 TCF +3
 TCF +2
 TCF +1
 AD TARGETDB # IF TARGET DB IS EXCEEDED, CONTINUE
 EXTEND # DIRECT RATE CONTROL.
 BZMF VDB
 CCS VRATEDIF
 TCF +3
 TCF +2
 TCF +1
 AD TARGETDB
 EXTEND
 BZMF +2
 TCF QRTIME
 CA ZERO
 TS VRATEDIF
 TCF QRTIME
VDB CCS VRATEDIF
 TC +3
 TC +2

```

July 12, 2016

1547

```

TC +1
AD TARGETDB # IF TARGET DB IS EXCEEDED, CONTINUE
EXTEND
BZMF TOPSEUDO # DIRECT RATE CONTROL. IF NOT, FIRE AND
CA ZERO # SWITCH TO PSEUDO-AUTO CONTROL ON NEXT
TS URATEDIF # PASS.
QRTIME CA TCQR # DIRECT RATE TIME CHECK.
EXTEND
BZMF +5 # BRANCH IF TIME EXCEEDS 4 SEC.
CS RCSFLAGS
MASK QRBIT
ADS RCSFLAGS # BIT 11 IS 1.
TC +4
TOPSEUDO CS QRBIT
MASK RCSFLAGS
TS RCSFLAGS # BIT 11 IS 0.
CA HANDADR
TS RETJADR
CA ONE

Page 1449
BACKHAND TS AXISCTR

CA FOUR
TS NUMBERT

INDEX AXISCTR
INDEX SKIPU
TCF +1
CA FOUR
INDEX AXISCTR
TS SKIPU
TCF LOOPER

INDEX AXISCTR
CCS URATEDIF # INDEX AXIS QUANTITY
CA ZERO # 0 -U 1/JETACC-AOSU
TCF +2 # 1 +U 1/JETACC+AOSU
CA ONE # 16 -V 1/JETACC-AOSV
INDEX AXISCTR # 17 +V 1/JETACC+AOSV
AD AXISDIFF # JETACC = 2 JET ACCELERATION (1 FOR FAIL)

INDEX A
CS 1/ANET2 +1
EXTEND
INDEX AXISCTR # UPRATEDIF IS SCALED AT PI/4 RAD/SEC
```

|                        |                            |          |                                             |
|------------------------|----------------------------|----------|---------------------------------------------|
|                        | MP                         | URATEDIF | # JET TIME IN A, SCALED 32 SEC              |
|                        | TS                         | Q        |                                             |
|                        | DAS                        | A        |                                             |
|                        | AD                         | Q        |                                             |
|                        | TS                         | A        | # OVERFLOW SKIP                             |
|                        | TCF                        | +2       |                                             |
|                        | CA                         | Q        | # RIGHT SIGN AND BIGGER THAN 150MS          |
| SETTIME                | INDEX                      | AXISCTR  |                                             |
|                        | TS                         | TJU      | # SCALED AT 10.67 WHICH IS CLOSE TO 10.24   |
|                        | TCF                        | AFTERTJ  |                                             |
| ZEROTJ                 | CA                         | ZERO     |                                             |
|                        | TCF                        | SETTIME  |                                             |
| HANDADR                | GENADR                     | BACKHAND |                                             |
| # GTS WILL BE TRIED IF |                            |          |                                             |
| #                      | 1. USEQRJTS = 0,           |          |                                             |
| #                      | 2. ALLOWGTS POS,           |          |                                             |
| #                      | 3. JETS ARE OFF (Q,R-AXES) |          |                                             |
| TRYGTS                 | CAF                        | USEQRJTS | # IS JET USE MANDATORY. (AS LONG AS         |
|                        | MASK                       | DAPBOOLS | # USEQRJTS BIT IS NOT BIT 15, CCS IS SAFE.) |
|                        | CCS                        | A        |                                             |
|                        | TCF                        | RCS      |                                             |
|                        | CCS                        | ALLOWGTS | # NO. DOES AOSTASK OK CONTROL FOR GTS?      |
| # Page 1450            |                            |          |                                             |
|                        | TCF                        | +2       |                                             |
|                        | TCF                        | RCS      |                                             |
|                        | EXTEND                     |          |                                             |
|                        | READ                       | CHAN5    |                                             |
|                        | CCS                        | A        |                                             |
|                        | TCF                        | CHKINGTS |                                             |
| GOTOGTS                | EXTEND                     |          |                                             |
|                        | DCA                        | GTSCADR  |                                             |
|                        | DTCB                       |          |                                             |
| CHKINGTS               | CCS                        | INGTS    | # WAS THE TRIM GIMBAL CONTROLLING           |
|                        | TCF                        | +2       | # YES. SET UP A DAMPED NULLING DRIVE.       |
|                        | TCF                        | RCS      | # NO. NULLING WAS SET UP BEFORE. DO P       |
|                        | INHINT                     |          |                                             |
|                        | TC                         | IBNKCALL |                                             |
|                        | CADR                       | TIMEGMBL |                                             |
|                        | RELINT                     |          |                                             |
|                        | CAF                        | ZERO     |                                             |
|                        | TS                         | INGTS    |                                             |

July 12, 2016

1549

TCF RCS  
EBANK= CDUXD  
GTSCADR 2CADR GTS

# Page 1451

# SUBROUTINE TO COMPUTE Q,R-AXES ATTITUDE ERRORS FOR USE IN THE RCS AND GTS CONTROL LAWS AND TH

QERRCALC CAE CDUY # Q-ERROR CALCULATION  
EXTEND  
MSU CDUYD # CDU ANGLE -- ANGLE DESIRED (Y-AXIS)  
TS DAPTEMP1 # SAVE FOR RERRCALC  
EXTEND  
MP M21 # (CDUY-CDUYD)\*M21 SCALED AT PI RADIANS  
TS E  
CAE CDUZ # SECOND TERM CALCULATION:  
EXTEND  
MSU CDUZD # CDU ANGLE -ANGLE DESIRED (Z-AXIS)  
TS DAPTEMP2 # SAVE FOR RERRCALC  
EXTEND  
MP M22 # (CDUZ-CDUZD)\*M22 SCALED AT PI RADIANS  
AD DELQEROR # KALCMANU INERFACE ERROR  
AD E  
XCH QERROR # SAVE Q-ERROR FOR EIGHT-BALL DISPLAY.

RERRCALC CAE DAPTEMP1 # R-ERROR CALCULATION:  
EXTEND # CDU ANGLE -ANGLE DESIRED (Y-AXIS)  
MP M31 # (CDUY-CDUYD)\*M31 SCALED AT PI RADIANS  
TS E  
CAE DAPTEMP2 # SECOND TERM CALCULATION:  
EXTEND # CDU ANGLE -ANGLE DESIRED (Z-AXIS)  
MP M32 # (CDUZ-CDUZD)\*M32 SCALED AT PI RADIANS  
AD DELREROR # KALCMANU INERFACE ERROR  
AD E  
XCH RERROR # SAVE R-ERROR FOR EIGHT-BALL DISPLAY.  
TC Q

# Page 1452

# "ATTSTEER" IS THE ENTRY POINT FOR Q,R-AXES (U,V-AXES) ATTITUDE CONTROL USING THE REACTION CON

ATTSTEER EQUALS STILLRCS # "STILLRCS" IS THE RCS EXIT FROM TRYGTS.

STILLRCS CA RERROR  
LXCH A  
CA QERROR  
TC ROT-TOUV

DXCH      UERROR

# PREPARES CALL TO TJETLAW (OR SPSRCS(DOCKED))  
# PREFORMS SKIP LOGIC ON U OR Y AXIS IF NEEDED.

|       |        |          |                                            |
|-------|--------|----------|--------------------------------------------|
| TJLAW | CA     | TJLAWADR |                                            |
|       | TS     | RETJADR  |                                            |
|       | CA     | ONE      |                                            |
|       | TS     | AXISCTR  |                                            |
|       | INDEX  | AXISCTR  |                                            |
|       | INDEX  | SKIPU    |                                            |
|       | TCF    | +1       |                                            |
|       | CA     | FOUR     |                                            |
|       | INDEX  | AXISCTR  |                                            |
|       | TS     | SKIPU    |                                            |
|       | TCF    | LOOPER   |                                            |
|       | INDEX  | AXISCTR  |                                            |
|       | CA     | UERROR   |                                            |
|       | TS     | E        |                                            |
|       | INDEX  | AXISCTR  |                                            |
|       | CA     | OMEGAU   |                                            |
|       | TS     | EDOT     |                                            |
|       | CA     | DAPBOOLS |                                            |
|       | MASK   | CSMDOCKD |                                            |
|       | CCS    | A        |                                            |
|       | TCF    | +3       |                                            |
|       | TC     | TJETLAW  |                                            |
|       | TCF    | AFTERTJ  |                                            |
| +3    | CS     | DAPBOOLS | # DOCKED. IF GIMBAL USABLE DO GTS CONTROL  |
|       | MASK   | USEQRJTS | # ON THE NEXT PASS.                        |
|       | CCS    | A        | # USEQRJTS BIT MUST NOT BE BIT 15.         |
|       | TS     | COTROLER | # GIMBAL USABLE. STORE POSITIVE VALUE.     |
|       | INHINT |          |                                            |
|       | TC     | IBNKCALL |                                            |
|       | CADR   | SPSRCS   | # DETERMINE RCS CONTROL                    |
|       | RELINT |          |                                            |
|       | CAF    | FOUR     | # ALWAYS CALL FOR 2-JET CONTROL ABOUT U,V. |
|       | TS     | NUMBERT  | # FALL THROUGH TO JET SLECTION, ETC.       |

# Q,R-JET-SELECTION-LOGIC

#

|             |          |                                                   |
|-------------|----------|---------------------------------------------------|
| # INPUT:    | AXISCTR  | 0,1 FOR U,V                                       |
| #           | SNUFFBIT | ZERO TJETU,V AND TRANS. ONLY IF SET IN A DPS BURN |
| # Page 1453 |          |                                                   |
| #           | TJU,TJV  | JET TIME SCALED 10.24 SEC.                        |
| #           | NUMBERT  | INDICATES NUMBER OF JETS AND TYPE OF POLICY       |

July 12, 2016

1551

```
RETJADR WHERE TO RETURN TO
#
OUTPUT: NO.U(V) JETS RATE DERIVATION FEEDBACK
CHANNEL 5
SKIPU,SKIPV FOR LESS THAN 150MS FIRING
#
NOTES: IN CASE OF FAILURE IN DESIRED ROTATION POLICY, "ALL" UNFAILED
JETS OF THE DESIRED POLICY ARE SELECTED. SINCE THERE ARE ONLY
TWO JETS, THIS MEANS THE OTHER ONE OR NONE. THE ALARM IS SENT
IF NONE CAN BE FOUND.
#
TIMES LESS THAN 14 MSEC ARE TAKEN TO CALL FOR A SINGLE-JET
MINIMUM IMPULSE, WITH THE JET CHOSEN SEMI-RANDOMLY.

AFTERTJ CA FLAGWRD5 # IF SNUFFBIT SET DURING A DPS BURN GO TO
 MASK SNUFFBIT # XTRANS; THAT IS, INHIBIT CONTROL.
 EXTEND
 BZF DOROTAT
 CS FLGWRD10
 MASK APSFLBIT
 EXTEND
 BZF DOROTAT
 CA DAPBOOLS
 MASK DRIFTBIT
 EXTEND
 BZF XTRANS

DOROTAT CAF TWO
 TS L
 INDEX AXISCTR
 CCS TJU
 TCF +5
 TCF NOROTAT
 TCF +2
 TCF NOROTAT
 ZL
 AD ONE
 TS ABSTJ

 CA AXISCTR
 AD L
 TS ROTINDEX # 0 1 2 3 = -U -V +U +V

 CA ABSTJ
 AD -150MS
 EXTEND
```

```

Page 1454
BZMF DOSKIP
TC SELCTSUB

INDEX AXISCTR
CA INDEXES
TS L

CA POLYTEMP
INHINT
INDEX L
TC WRITEP

RELINT
TCF FEEDBACK

NOROTAT INDEX AXISCTR
 CA INDEXES
 INHINT
 INDEX A
 TC WRITEP -1

 RELINT
LOOPER CCS AXISCTR
 TC RETJADR
 TCF CLOSEOUT
 CS ABSTJ
 AD +TJMINT6 # 14MS
 EXTEND
 BZMF NOTMIN

 ADS ABSTJ
 INDEX AXISCTR
 CCS TJU
 CA +TJMINT6
 TCF +2
 CS +TJMINT6
 INDEX AXISCTR
 TS TJU

 CCS SENSETYP # ENSURE MIN-IMPULSE NOT AGAINST TRANS
 TCF NOTMIN -1
 EXTEND
 READ LOSCALAR
 MASK ONE
 TS NUMBERT

```



July 12, 2016

1553

|             |        |             |                     |
|-------------|--------|-------------|---------------------|
| NOTMIN      | TC     | SELCTSUB    |                     |
|             | INDEX  | AXISCTR     |                     |
|             | CA     | INDEXES     |                     |
|             | INHINT |             |                     |
| # Page 1455 | TS     | T6FURTHA +1 |                     |
|             | CA     | POLYTEMP    |                     |
|             | INDEX  | T6FURTHA +1 |                     |
|             | TC     | WRITEP      |                     |
|             | CA     | ABSTJ       |                     |
|             | TS     | T6FURTHA    |                     |
|             | TC     | JTLST       | # IN QR BANK BY NOW |
|             | RELINT |             |                     |
|             | CA     | ZERO        |                     |
|             | INDEX  | AXISCTR     |                     |
|             | TS     | SKIPU       |                     |
| FEEDBACK    | CS     | THREE       |                     |
|             | AD     | NUMBERT     |                     |
|             | EXTEND |             |                     |
|             | BZMF   | +3          |                     |
|             | CA     | TWO         |                     |
|             | TCF    | +2          |                     |
|             | CA     | ONE         |                     |
|             | INDEX  | AXISCTR     |                     |
|             | TS     | NO.UJETS    |                     |
|             | TCF    | LOOPER      |                     |
| XTRANS      | CA     | ZERO        |                     |
|             | TS     | TJU         |                     |
|             | TS     | TJV         |                     |
|             | CA     | FOUR        |                     |
|             | INHINT |             |                     |
|             | XCH    | SKIPU       |                     |
|             | EXTEND |             |                     |
|             | BZF    | +2          |                     |
|             | TC     | WRITEU -1   |                     |
|             | CA     | FOUR        |                     |
|             | XCH    | SKIPV       |                     |
|             | RELINT |             |                     |

```

EXTEND
BZF CLOSEOUT
INHINT
TC WRITEV -1
RELINT

INDEXES TCF CLOSEOUT
 DEC 4
 DEC 13
+TJMINT6 DEC 22
Page 1456
-150MS DEC -240
BIT8,9 OCT 00600
SCLNORM OCT 266
TJLAWADR GENADR TJLAW +3 # RETURN ADDRESS FOR RCS ATTITUDE CONTROL

THE JET LIST:
THIS IS A WAITLIST FOR T6RUPTS.
#
CALLED BY:
CA TJ # TIME WHEN NEXT JETS WILL BE WRITTEN
TS T6FURTHA
CA INDEX # AXIS TO BE WRITTEN AT TJ (FROM NOW)
TS T6FURTHA +1
TC JTLST
#
EXAMPLE -- U-AXIS AUTOPILOT WILL WRITE ITS ROTATION CODE OF
JETS INTO CHANNEL 5. IF IT DESIRES TO TURN OFF THIS POLICY WITHIN
150MS AND THEN FIRE NEXTU, A CALL TO JTLST IS MADE WITH T6FURTHA
CONTAINING THE TIME TO TURN OFF THE POLICY, T6FURTHA +1 THE INDEX
OF THE U-AXIS(4), AND NEXTU WILL CONTAIN THE "U-TRANS" POLICY OR ZERO.
#
THE LIST IS EXACTLY 3 LONG. (THIS LEADS UP TO SKIP LOGIC AND 150MS LIMIT)
THE INPUT IS THE LAST MEMBER OF THE LIST.
#
RETURNS BY:
+ TC Q
#
DEFINITIONS: (OUTPUT)
TIME6 TIME OF NEXT RUPT
T6NEXT DELTA TIME TO NEXT RUPT
T6FURTHA DELTA TIME FROM 2ND TO LAST RUPT
NXT6ADR AXIS INDEX 0 -- P-AXIS
T6NEXT +1 AXIS INDEX 4 -- U-AXIS
T6FURTHA +1 AXIS INDEX 13 -- V-AXIS

```

July 12, 2016

1555

JTLST CS T6FURTHA  
AD TIME6  
EXTEND  
BZMF MIDORLST # TIME6 -- TI IS IN A

LXCH NXT6ADR  
DXCH T6NEXT  
DXCH T6FURTHA  
TS TIME6  
LXCH NXT6ADR

TURNON CA BIT15  
EXTEND  
WOR CHAN13  
TC Q

# Page 1457  
MIDORLST

AD T6NEXT  
EXTEND  
BZMF LASTCHG # TIME6 + T6NEXT - T IS IN A

LXCH T6NEXT +1  
DXCH T6FURTHA  
EXTEND  
SU TIME6  
DXCH T6NEXT

TC Q

LASTCHG CS A  
AD NEG0  
TS T6FURTHA  
TC Q

# ROT-TOUV IS ENTERED WITH THE Q-COMPONENT OF THE QUANTITY TO BE TRANSFORMED IN A AND THE R-COMPONENT  
# ROT-TOUV TRANSFORMS THE QUANTITY INTO THE NON-ORTHOGONAL U-V AXIS SYSTEM. IN THE U-V SYSTEM  
# PRODUCED FROM RCS JET FIRINGS. AT THE COMPLETION OF ROT-TOUV, THE U-COMPONENT OF THE TRANSFORMED QUANTITY  
# A AND THE V-COMPONENT IS IN L.

ROT-TOUV LXCH ROTEMP2 # (R) IS PUT INTO ROTEMP2  
EXTEND  
MP COEFFQ  
XCH ROTEMP2 # (R) GOES TO A AND COEFFQ.(Q) TO ROTEMP2  
EXTEND

|             |        |             |                                           |
|-------------|--------|-------------|-------------------------------------------|
|             | MP     | COEFFR      |                                           |
|             | TS     | L           | # COEFFR.(R) IS PUT INTO L                |
|             | AD     | ROTEMP2     |                                           |
|             | TS     | ROTEMP1     | # COEFFQ.(Q)+COEFFR.(R) IS PUT IN ROTEMP1 |
|             | TCF    | +4          |                                           |
|             | INDEX  | A           | # COEFFQ.(Q) + COEFFR.(R) HAS OVERFLOWED  |
|             | CS     | LIMITS      | # AND IS LIMITED TO POSMAX OR NEGMAX      |
|             | TS     | ROTEMP1     |                                           |
|             | CS     | ROTEMP2     |                                           |
|             | AD     | L           | # -COEFFQ.(Q) + COEFFR.(R) IS NOW IN A    |
|             | TS     | 7           |                                           |
|             | TCF    | +3          |                                           |
|             | INDEX  | A           | # -COEFFQ.(Q) + COEFFR.(R) HAS OVERFLOWED |
|             | CS     | LIMITS      | # AND IS LIMITED TO POSMAX OR NEGMAX      |
|             | LXCH   | ROTEMP1     | # COEFFQ.(Q) + COEFFR.(R) IS PUT INTO L   |
|             | TC     | Q           |                                           |
| SELCTSUB    | INDEX  | ROTINDEX    |                                           |
|             | CA     | ALLJETS     |                                           |
|             | INDEX  | NUMBERT     |                                           |
|             | MASK   | TYPEPOLY    |                                           |
|             | TS     | POLYTEMP    |                                           |
| # Page 1458 | MASK   | CH5MASK     |                                           |
|             | CCS    | A           |                                           |
|             | TCF    | +2          |                                           |
|             | TC     | Q           |                                           |
|             | CA     | THREE       |                                           |
| FAILLOOP    | TS     | NUMBERT     |                                           |
|             | INDEX  | ROTINDEX    |                                           |
|             | CA     | ALLJETS     |                                           |
|             | INDEX  | NUMBERT     |                                           |
|             | MASK   | TYPEPOLY    |                                           |
|             | TS     | POLYTEMP    |                                           |
|             | MASK   | CH5MASK     |                                           |
|             | EXTEND |             |                                           |
|             | BZF    | FAILLOOP -2 |                                           |
|             | CCS    | NUMBERT     |                                           |
|             | TCF    | FAILLOOP    |                                           |
|             | INDEX  | AXISCTR     |                                           |
|             | TS     | TJU         |                                           |
|             | TC     | ALARM       |                                           |
|             | OCT    | 02004       |                                           |
|             | TCF    | NOROTAT     |                                           |
| ALLJETS     | OCT    | 00110       | # -U 6 13                                 |

July 12, 2016

1557

|          |     |       |   |     |                    |
|----------|-----|-------|---|-----|--------------------|
|          | OCT | 00022 | # | -V  | 2 9                |
|          | OCT | 00204 | # | +U  | 5 14               |
|          | OCT | 00041 | # | +V  | 1 10               |
| TYPEPOLY | OCT | 00125 | # | -X  | 1 5 9 13           |
|          | OCT | 00252 | # | +X  | 2 6 10 14          |
|          | OCT | 00146 | # | A   | 2 5 10 13          |
|          | OCT | 00231 | # | B   | 1 6 9 14           |
|          | OCT | 00377 | # | ALL | 1 2 5 6 9 10 13 14 |

# THE FOLLOWING SETS THE INTERRUPT FLIP-FLOP AS SOON AS POSSIBLE, WHICH PERMITS A RETURN TO THE

|          |    |          |
|----------|----|----------|
| CLOSEOUT | CA | ADRRUPT  |
|          | TC | MAKERUPT |

|         |       |         |
|---------|-------|---------|
| ADRRUPT | ADRES | ENDJASK |
|---------|-------|---------|

|         |      |          |
|---------|------|----------|
| ENDJASK | DXCH | DAPARUPT |
|         | DXCH | ARUPT    |
|         | DXCH | DAPBQRPT |
|         | XCH  | BRUPT    |
|         | LXCH | Q        |
|         | CAF  | NEGMAX   |
|         | DXCH | DAPZRUPT |
|         | DXCH | ZRUPT    |
|         | TCF  | NOQRSM   |

# NEGATIVE DAPZRUPT SIGNALS JASK IS OVER.

# Page 1459

|        |        |
|--------|--------|
| BLOCK  | 3      |
| SETLOC | FFTAG6 |
| BANK   |        |

COUNT\* \$\$/DAP

|          |                 |
|----------|-----------------|
| MAKERUPT | EXTEND          |
|          | EDRUPT MAKERUPT |

This code is written to file src/Q-R-AXIS-RCS-AUTOPILOT.s.

**B.86 R30**1558 *<src/R30.s 1558>*≡

```
Copyright: Public domain.
Filename: R30.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
Website: www.ibiblio.org/apollo.
Pages: 514-524
Mod history: 2009-05-09 HG Started adapting from the Colossus249/ file
of the same name, using Comanche055 page
images 0514.jpg - 0524.jpg.
```

```
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
```

```
Notations on the hardcopy document read, in part:
```

```
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
```

```
This AGC program shall also be referred to as
Colossus 2A
```

```
Page 514
```

```
SUBROUTINE NAME: V82CALL
```

```
MOD NO: 0 DATE: 16 FEB 67
MOD BY: R. R. BAIRNSFATHER LOG SECTION: R30
MOD NO: 1 MOD BY: R. R. BAIRNSFATHER DATE: 11 APR 67 SR30.1 CHANGE
MOD NO: 2 MOD BY: ALONSO DATE: 11 DEC 67 VB82 PROGRAM
MOD NO: 3 MOD BY: ALONSO DATE: 26 MAR 68 PROG MOD TO F
```

```
NEW FUNCTIONAL DESCRIPTION: CALLED BY VERB 82 ENTER. PRIORITY 10.
USED THROUGHOUT. CALCULATE AND DISPLAY ORBITAL PARAMETERS
```

```
1. IF AVERAGE G IS OFF:
```

```
FLASH DISPLAY V04N06. R2 INDICATES WHICH SHIP'S STATE VECTOR IS
```

July 12, 2016

1559

```
TO BE UPDATED. INITIAL CHOICE IS THIS SHIP (R2=1). ASTRONAUT
CAN CHANGE TO OTHER SHIP BY V22EXE. WHERE X IS NOT EQ 1.
SELECTED STATE VECTOR UPDATED BY THISPREC (OTHPREC).
CALLS SR30.1 (WHICH CALLS TFFCONMU + TFFRP/RA) TO CALCULATE
RPER (PERIGEE RADIUS), RAPO (APOGEE RADIUS), HPER (PERIGEE
HEIGHT ABOVE LAUNCH PAD OR LAUNAR LANDING SITE), HARD (APOGEE
HEIGHT AS ABOVE), TPER (TIME TO PERIGEE), TFF (TIME TO
INTERSECT 300 KFT ABOVE PAD OR 35KFT ABOVE LANDING SITE).
FLASH MONITOR V16N44 (HAPO, HPER, TFF). TFF IS -59MS59S IF IT WAS
NOT COMPUTABLE, OTHERWISE IT INCREMENTS ONCE PER SECOND.
ASTRONAUT HAS OPTION TO MONITOR TPER BY KEYING IN N 32 E.
DISPLAY IS IN HMS, IS NEGATIVE (AS WAS TFF), AND INCREMENTS
ONCE PER SECOND ONLY IF TFF DISPLAY WAS -59M59S.
2. IF AVERAGE G IS ON:
CALLS SR30.1 APPROX EVERY TWO SECS. STATE VECTOR IS ALWAYS
FOR THIS VEHICLE. V82 DOES NOT DISTURB STATE VECTOR. RESULTS
OF SR30.1 ARE RAPO, RPER, HAPO, HPER, TPER, TFF.
FLASH MONITOR V16N44 (HAPO, HPER, TFF).
IF MODE IS P11, THEN CALL DELRSPL SO ASTRONAUT CAN MONITOR
RESULTS BY N50E. SPLASH COMPUTATION DONE ONCE PER TWO SECS.
ADDENDUM: HAPO AND HPER SHOULD BE CHANGED TO READ HAPOX AND HPERX IN THE
ABOVE REMARKS.
#
CALLING SEQUENCE: VERB 82 ENTER.
#
SUBROUTINES CALLED: SR30.1, GOXDSPF
MAYBE -- THISPREC, OTHPREC, LOADTIME, DELRSPL
#
NORMAL EXIT MODES: TC ENDEXT
#
ALARMS: NONE
#
OUTPUT: HAPOX (-29) M
HPERX (-29) M
RAPO (-29) M EARTH
(-27) M MOON
#
Page 515
RPER (-29) M EARTH
(-27) M MOON
TFF (-28) CS CONTAINS NEGATIVE QUANTITY
-TPER (-28) CS CONTAINS NEGATIVE QUANTITY
RSP-RREC(-29) M IF DELRSPL CALLED
#
ERASABLE INITIALIZATION REQUIRED: STATE VECTOR.
#
DEBRIS: QPRET, RONE, VONE, TFF/RTMU, HPERMIN, RPADTEM, V82EMFLG.
```

```

MAYBE: TSTART82, V82FLAGS, TDEC1.

EBANK= HAPOX
BANK 31
SETLOC R30LOC
BANK
COUNT* $$/R30

V82CALL TC INTPRET
 BON GOTO
 AVEGFLAG
 V82GON # IF AVERAGE G ON
 V82GOFF # IF AVERAGE G OFF

V82GOFF EXIT
 CAF TWO # ALLOW ASTRONAUT TO SELECT VEHICLE
 TS OPTIONX # DESIRED FOR ORBITAL PARAMETERS
 CAF ONE
 TS OPTIONX +1
 CAF OPTIONVN # V 04 N 06
 TC BANKCALL
 CADR GOXDSPF
 TC ENDEXT # TERMINATE
 TC +2 # PROCEED
 TC -5 # DATA IN. OPTIONX +1 = 1 FOR THIS VEHIC.
 #
 # UNEQ 1 FOR OTHER VEHICLE.
 CAF BIT4 # 80 MS
 TC WAITLIST
 EBANK= TFF
 2CADR TICKTEST

RELINT
V82GOFLP CAF TFFBANK # MAJOR RECYCLE LOOP ENTRY
 TS EBANK
 CAF ZERO
 TS V82FLAGS # ZERO FLAGS FOR TICKTEST, INHIBITS
 # DECREMENTING OF TFF AND -TPER.
 CAF PRI07
 TC FINDVAC # V82GOFF1 WILL EXECUTE STATE VECTOR

Page 516
 EBANK= TFF # UPDATE AND ORBIT CALCULATIONS FOR
 2CADR V82GOFF1 # SELECTED VEHICLE ABOUT PROPER BODY.

RELINT
V82STALL CAF THREE # STALL IN THIS LOOP AND WITHOLD V 16 N 44
 MASK V82FLAGS # UNTIL STATE VECTOR UPDATE SETS ONE OF

```



July 12, 2016

1561

|          |        |            |                                                         |
|----------|--------|------------|---------------------------------------------------------|
|          | CCS    | A          | # OUR FLAG BITS.                                        |
|          | TC     | FLAGGON    | # EXIT FROM STALL LOOP.                                 |
|          | CAF    | 1SEC       |                                                         |
|          | TC     | BANKCALL   |                                                         |
|          | CADR   | DELAYJOB   |                                                         |
|          | TC     | V82STALL   |                                                         |
| FLAGGON  | CAF    | V16N44     | # MONITOR HAPO, HPER, TFF.                              |
|          | TC     | BANKCALL   |                                                         |
|          | CADR   | GOXDSPF    |                                                         |
|          | TC     | B5OFF      | # TERM THIS TELLS TICKTEST TO KILL ITSELF               |
|          | TC     | B5OFF      | # PROCEED DITTO                                         |
|          | TC     | V82GOFLP   | # RECYCLE RECOMPUTE STATE VECT + DISPLAY                |
| OPTIONVN | VN     | 0412       |                                                         |
| V16N44   | VN     | 1644       |                                                         |
| TFFBANK  | ECADR  | TFF        |                                                         |
| V82GOFF1 | TC     | INTPRET    |                                                         |
|          | RTB    |            |                                                         |
|          |        | LOADTIME   |                                                         |
|          | STORE  | TDEC1      | # TIME FOR STATE VECTOR UPDATE.                         |
|          | STORE  | TSTART82   | # TIME FOR INTERNAL USE.                                |
|          | EXIT   |            |                                                         |
|          | CS     | OPTIONX +1 | # 1 FOR THIS VEHICLE, NOT 1 FOR OTHER                   |
|          | AD     | ONE        |                                                         |
|          | EXTEND |            |                                                         |
|          | BZF    | THISSHIP   |                                                         |
| OTHSHIP  | TC     | INTPRET    |                                                         |
|          | CALL   |            | # CALL STATE VECTOR UPDATE FOR OTHER SHIP.              |
|          |        | OTHPREC    |                                                         |
| BOTHSHIP | VLOAD  |            | # MOVE RESULTS INTO TFFCONIC STORAGE AREAS              |
|          |        | RATT       | # TO BE CALLED BY SR30.1.                               |
|          | STOVL  | RONE       | # RATT AT (-29)M FOR EARTH OR MOON                      |
|          |        | VATT       |                                                         |
|          | STORE  | VONE       | # VATT AT (-7)M/CS FOR EARTH OR MOON                    |
|          | DLOAD* |            |                                                         |
|          |        | 1/RTMUE,2  | # X2 IS 0 FOR EARTH CENTERED STATE VEC                  |
|          |        |            | # HG remark: In Comanche055 scan this line (5324) looks |
|          |        |            | # as it is in start source                              |
|          | STORE  | TFF/RTMU   | # X2 IS 2 FOR MOON                                      |
|          | DLOAD* |            | # AS LEFT BY THISPREC OR OTHPREC.                       |
|          |        | MINPERE,2  |                                                         |
|          | STORE  | HPERMIN    | # TFFRTMU, HPERMIN, AND RPADTEM ARE ALL                 |
|          | SLOAD  | BHIZ       | # EARTH/MOON PARAMETERS AS SET HERE.                    |

1562

July 12, 2016

```

 X2
 EARTHPAD
 GOTO
THISSHIP TC MOONPAD
 CALL INTPRET # CALL STATE VECTOR UPDATE FOR THIS SHIP.
 THISPREC
 GOTO
 BOTHSHIP

THE FOLLOWING CONSTANTS ARE PAIRWISE INDEXED. DO NOT SEPARATE PAIRS.

1/RTMUM 2DEC* .45162595 E-4 B14*
1/RTMUE 2DEC* .50087529 E-5 B17*

MINPERM 2DEC 10668 B-27 # 35 KFT MIN PERIGEE HEIGHT FOR MOON(-27)M
MINPERE 2DEC 91440 B-29 # 300 KFT (-29)M FOR EARTH

EARTHPAD DLOAD CLRGO # PAD 37-B RADIUS. SCALED AT (-29)M
 RPAD
 V82EMFLG # INDICATE EARTH SCALING FOR SR30.1
 BOTHPAD

MOONPAD VLOAD ABVAL # COMPUTE MOON PAD RADIUS FROM RLS VECTOR.
 RLS # SCALED AT (-27)M.
 SET

 V82EMFLG # INDICATE MOON SCALING FOR SR30.1
BOTHPAD STCALL RPADTEM
 SR30.1 # CALCULATE ORBITAL PARAMETERS
 EXIT
 CA MODREG # ARE WE IN POO
 EXTEND
 BZF CANDEL # YES, DO DELRSPL
SPLRET1 TC INTPRET
 RTB DSU
 LOADTIME
 TSTART82 # PRESENT TIME -- TIME V82GOFF1 BEGAN
 STORE TSTART82 #
 DLOAD BZE # SR30.1 SETS -TPER=0 IF HPER L/
 -TPER # HPERMIN (300 OR 35) KFT.
 TICKTFF # (-TPER = 0)
TICKTPER DLOAD DAD # (-TPER NON ZERO) TFF WAS NOT COMPUTED.
 -TPER # BUT WAS SET TO 59M59S. DON'T DICK TFF, DO
 TSTART82 # TICK -TPER. DISPLAY BOTH.
 STORE -TPER # -TPER CORRECTED FOR TIME SINCE V82GOFF1
 EXIT # BEGAN.

```

July 12, 2016

1563

# Page 518

|          |        |          |                                              |
|----------|--------|----------|----------------------------------------------|
|          | CAF    | BIT1     |                                              |
|          | TS     | V82FLAGS | # INFORMS TICKTEST TO INCREMENT ONLY -TPER   |
|          | TC     | ENDOFJOB |                                              |
| TICKTFF  | DLOAD  | DAD      | # (-TPER=0) TFF WAS COMPUTED. TICK TFF.      |
|          |        | TFF      | # DO NOT TICK -TPER. DISPLAY TFF, BUT NOT    |
|          |        | TSTART82 | # -TPER                                      |
|          | STORE  | TFF      | # TFF CORRECTED FOR TIME SINCE V82GOFF1      |
|          | EXIT   |          | # BEGAN.                                     |
|          | CAF    | BIT2     |                                              |
|          | TS     | V82FLAGS | # INFORMS TICKTEST TO INCREMENT ONLY TFF.    |
|          | TC     | ENDOFJOB |                                              |
| TICKTEST | CAF    | BIT5     | # THIS WAITLIST PROGRAM PERPETUATES ITSELF   |
|          | MASK   | EXTVBACT | # ONCE A SEC UNTIL BIT 5 OF EXTVBACT =0.     |
|          | CCS    | A        |                                              |
|          | TC     | DOTICK   |                                              |
|          | CAF    | PRI025   |                                              |
|          | TC     | NOVAC    | # TERMINATE V 82. CAN'T CALL ENDEXT IN RUPT. |
|          | EBANK= | EXTVBACT |                                              |
|          | 2CADR  | ENDEXT   |                                              |
| DOTICK   | TC     | TASKOVER |                                              |
|          | CAF    | 1SEC     | # RE-REQUEST TICKTEST.                       |
|          | TC     | WAITLIST |                                              |
|          | EBANK= | TFF      |                                              |
|          | 2CADR  | TICKTEST |                                              |
|          | CAF    | THREE    |                                              |
|          | MASK   | V82FLAGS |                                              |
|          | INDEX  | A        |                                              |
|          | TC     | +1       |                                              |
|          | TC     | TASKOVER | # IF NO FLAGBITS SET DONT' CHANGE TFF OR     |
|          |        |          | # -TPER, BUT CONTINUE LOOP.                  |
|          | TC     | TPERTICK | # ONLY BIT 1 SET. INCR -TPER BY 1 SEC.       |
| TFFTICK  | CAF    | 1SEC     | # ONLY BIT 2 SET. INCR TFF BY 1 SEC.         |
|          | TS     | L        |                                              |
|          | CAF    | ZERO     |                                              |
|          | DAS    | TFF      |                                              |
|          | TC     | TASKOVER |                                              |
| TPERTICK | CAF    | 1SEC     |                                              |
|          | TS     | L        |                                              |
|          | CAF    | ZERO     |                                              |
|          | DAS    | -TPER    |                                              |

```

Page 519
V82GON TC TASKOVER

EXIT

CAF PRI07
TC FINDVAC
EBANK= TFF
2CADR V82GON1

RELINT
CCS NEWJOB
TC CHANG1

V82REDSP CAF V16N44
 TC BANKCALL
 CADR GOXDSPF
 TC B5OFF
 TC B5OFF
 TC V82REDSP

V82GON1 TC INTERPRET

 VLOAD GOTO
 RN
 NEXTLINE
NEXTLINE STOVL RONE
 VN
 STORE VONE
 BON GOTO
 AMOONFLG
 MOONGON
 EARTHGON

MOONGON SET DLOAD
 V82EMFLG
 1/RTMUM
 STODL TFF/RTMU
 MINPERM
 STOVL HPERMIN
 RLS
 ABVAL GOTO
 V82GON2
EARTHGON CLEAR DLOAD
 V82EMFLG
 1/RTMUE

AVERAGE G ON. USE CURRENT STATE VECTOR
FOR ORBITAL PARAMETER CALCULATIONS.
LESS THAN LAMBERT
V82GON1 WILL PERFORM ORBIT CALCULATIONS
ABOUT PROPER BODY APPROX ONCE PER SEC.

WITHOLD V16 N44 UNTIL FIRST ORBIT CALC
IS DONE. NOTE: V82GON1 (PRI07, FINDVAC
JOB) IS COMPLETED BEFORE V82GON (PRI07,
NOVAC JOB).
MONITOR HAPO, HPER, TFF

TERM THIS TELLS V82GON1 TO KILL ITSELF.
PROC DITTO.
RECYCLE

THIS EXEC PROGRAM PERPETUATES ITSELF
ONCE A SEC UNTIL BIT 5 OF EXTVBACT =0.
HOLDS OFF CCS NEWJOB BETWEEN RN AND
VN FETCH SO RN, VN ARE FROM SAME
STATE VECTOR UPDATE.
RN AT (-29)M FOR EARTH OR MOON
VN AT (-7)M/CS FOR EARTH OR MOON
FLAG INDICATES BODY ABOUT WHICH ORBITAL
CALCULATIONS ARE TO BE PERFORMED.
IF SET - MOON, IF RESET - EARTH.

INDICATE MOON SCALING FOR SR30.1
LUNAR PARAMETERS LOADED HERE FOR SR30.1

SCALED AT (-27)M

INDICATE EARTH SCALING FOR SR30.1
EARTH PARAMETERS LOADED HERE FOR SR30.1

```

July 12, 2016

1565

```

 STODL TFF/RTMU
 MINPERE
 STODL HPERMIN
 RPAD
V82GON2 STCALL RPADTEM # COMMON CODE FOR EARTH & MOON.
 SR30.1
Page 520
 EXIT
 TC CHECKMM
 DEC 11
 TC V82GON3 # NOT IN MODE 11.
CANDEL TC INTPRET # IN MODE 11 OR 00
 CALL
 INTSTALL # DELRSPL DOES INTWAKE
 DLOAD CALL
 TFF
 DELRSPL # RETURN IS TO NEXT LINE (SPLRET).
SPLRET EXIT
 CA MODREG
 EXTEND
 BZF SPLRET1
V82GON3 CAF BIT5
 MASK EXTVBACT # SEE IF ASTRONAUT HAS SIGNALLED TERMINATE
 EXTEND
 BZF ENDEXT # YES, TERMINATE VB 82 LOOP
 CAF 1SEC
 TC BANKCALL # WAIT ONE SECOND BEFORE REPEATING
 CADR DELAYJOB # ORBITAL PARAMETER COMPUTATION.
 TC V82GON1
```

# Page 521

# SUBROUTINE NAME: SR30.1

# MOD NO: 0

# MOD BY: R. R. BAIRNSFATHER

# MOD NO: 1 MOD BY: R. R. BAIRNSFATHER

DATE: 11 APR 67

# MOD NO: 2 MOD BY: R. R. BAIRNSFATHER

DATE: 14 APR 67

# MOD NO: 3 MOD BY ALONSO

DATE: 11 DEC 67

# MOD NO: 4 MOD BY ALONSO

DATE: 26 MAR 68

# MOD NO: 5 MOD BY: R. R. BAIRNSFATHER

DATE: 6 AUG 68

DATE: 16 FEB 67

LOG SECTION: R32

SR30.1 CHANGED TO ALLOW

ADD OVFL CK FOR RAPO

SUBROUTINE REWRITTEN

PROG MOD TO HANDLE DIF

OVFL CK FOR HAPO & HPER

#

# NEW FUNCTIONAL DESCRIPTION: ORBITAL PARAMETERS DISPLAY FOR NOUNS 32 AND 44.

# SR30.1 CALLS TFFCONMU AND TFFRP/RA TO CALCULATE RPER (PERIGEE RADIUS),

# RAPO (APOGEE RADIUS), HPER (PERIGEE HEIGHT ABOVE LAUNCH PAD OR LUNAR

# LANDING SITE), HAPO (APOGEE HEIGHT AS ABOVE), TPER (TIME TO PERIGEE),

# TFF (TIME TO INTERSECT 300 KFT ABOVE PAD OR 35KFT ABOVE LANDING SITE).

```

IF HPER IS GREATER THAN OR EQUAL TO HPERMIN, CALCULATES TPER AND STORES
NEGATIVE. IN -TPER. OTHERWISE STORES +0 IN -TPER. WHENEVER TPER IS
CALCULATED, TFF IS NOT COMPUTABLE AND DEFAULTS TO -59MIN 59SEC. IF HAPO
WOULD EXCEED 9999.9 NM, IT IS LIMITED TO THAT VALUE FOR DISPLAY.
#
ADDENDUM: HAPO AND HPER SHOULD BE CHANGED TO READ HAPOX AND HPERX IN THE
ABOVE REMARKS.
#
CALLING SEQUENCE: CALL
SR30.1
#
SUBROUTINES CALLED: TFFCONMU, TFFRP/RA, CALCTPER, CALCTFF
#
NORMAL EXIT MODE: CALLING LINE +1 (STILL IN INTERPRETIVE MODE)
#
ALARMS: NONE
#
OUTPUT: RAPO (-29) M EARTH APOGEE RADIUS EARTH CENTERED COORD.
(-27) M MOON MOON CENTERED COORD.
RPER (-29) M EARTH PERIGEE RADIUS EARTH CENTERED COORD.
(-27) M MOON MOON CENTERED COORD.
HAPOX (-29) M APOGEE ALTITUDE ABOVE PAD OR LAND. SITE MAX V
HPERX (-29) M PERIGEE ALT. ABOVE PAD OR LAND. SITE MAX V
TFF (-28) CS TIME TO 300KFT OR 35KFT ALTITUDE
-TPER (-28) CS TIME TO PERIGEE
#
ERASABLE INITIALIZATION REQUIRED -
TFF/RTMU (+17) EARTH RECIPROCAL OF PROPER GRAV CONSTANT FOR
(+14) MOON EARTH OR MOON = 1/SQRT(MU).
RONE (-29) M STATE VECTOR
VONE (-7) M/CS STATE VECTOR
RPADTEM (-29) M EARTH RADIUS OF LAUNCH PAD OR LUNAR LANDING
(-27) M MOON SITE.
HPERMIN (-29) M EARTH (300 OR 35) KFT MINIMUM PERIGEE ALTITUDE
(-27) M MOON ABOVE LAUNCH PAD OR LUNAR LANDING SITE.
V82EMFLG (INT SW BIT) RESET FOR EARTH, SET FOR MOON.
#
DEBRIS: QPRET, PDL, S2
#
Page 522
#
COUNT* $$/SR30S
#
SR30.1 SETPD STQ # INITIALIZE PUSHDOWN LIST.
 0
 S2
SR30.1 INPUT: RONE AT (-29)M EARTH/MOON

```

July 12, 2016

1567

```
VONE AT (-7)M/CS
TFFCONMU, TFFRP/RA, CALCTPER, AND CALCTFF
CALLS REQUIRE:
EARTH CENTERED (NO RESCALING REQUIRED)
RONE SCALED TO B-29 M
VONE SCALED TO B-7 M/CS
MOON CENTERED (RESCALING REQUIRED)
RONE SCALED TO B-27 M
VONE SCALED TO B-5 M/CS

BOFF VLOAD
 V82EMFLG
 TFFCALLS
 RONE
 # OFF FOR EARTH, ON FOR MOON.

VSL2
STOVL RONE
 VONE

VSL2
STORE VONE
CALL TFFCONMU
 TFFRP/RA
 # TFFRP/RA COMPUTES RAPO, RPER.
 # RETURNS WITH RAPO IN D(MPAC).

DSU
 RPADTEM
BOFF SR2R
 # NEED HAPO AT (-29)M FOR DISPLAY.
 # IF MOON CENTERED, RESCALE FROM (-27)M.
 # IF EARTH CENTERED ALREADY AT (-29)M.
 # OFF FOR EARTH, ON FOR MOON.

 V82EMFLG
 +1
CALL # IF HAPO > MAXNM, SET HAPO =9999.9 NM.
 MAXCHK
 # OTHERWISE STORE (RAPO-RPADTEM) IN HAPO.
STODL HAPOX
 RPER

DSU
 RPADTEM
STORE MPAC +4
BOFF SR2R
 # GIVES HPER AT (-29)M EARTH, (-27)M MOON.
 # SAVE THIS FOR COMPARISON TO HPERMIN.
 # NEED HPER AT (-29)M FOR DISPLAY.
 # IF MOON CENTERED, RESCALE FROM (-27)M.
 # IF EARTH CENTERED ALREADY AT (-29)M.
 # OFF FOR EARTH, ON FOR MOON.

 V82EMFLG
 +1
CALL # IF HPER > MAXNM, SET HPER = 9999.9 NM.
 MAXCHK

Page 523
STORHPER STODL HPERX
 # STORE (RPER - RPADTEM) INTO HPERX.
```

|          |        |            |                                             |
|----------|--------|------------|---------------------------------------------|
|          |        | MPAC +4    |                                             |
|          | DSU    | BPL        | # HPERMIN AT (-29)M FOR EARTH, (-27)M MOON  |
|          |        | HPERMIN    | # IF HPER L/ HPERMIN (300 OR 35) KFT,       |
|          |        | DOTPER     | # THEN ZERO INTO -TPER.                     |
|          | DLOAD  | GOTO       | # OTHERWISE CALCULATE TPER.                 |
|          |        | HI6ZEROS   |                                             |
|          |        | SKIPTPER   |                                             |
| DOTPER   | DLOAD  | CALL       |                                             |
|          |        | RPER       |                                             |
|          |        | CALCTPER   |                                             |
|          | DCOMP  |            | # TPER IS PUT NEG INTO -TPER.               |
| SKIPTPER | STODL  | -TPER      |                                             |
|          |        | HPERMIN    | # HPERMIN AT (-29)M FOR EARTH, (-27)M MOON  |
|          | DAD    | CALL       |                                             |
|          |        | RPADTEM    | # RPADTEM AT (-29)M FOR EARTH, (-27)M MOON  |
|          |        | CALCTFF    | # GIVES 59M59S FOR TFF IF RPER G/           |
|          | DCOMP  |            | # HPERMIN + RPADTEM. (TPER WAS NON ZERO)    |
|          | STCALL | TFF        | # OTHERWISE COMPUTES TFF. (GOTO)            |
|          |        | S2         |                                             |
| MAXCHK   | DSU    | BPL        | # IF C(MPAC) > 9999.9 NM. MPAC = 9999.9 NM. |
|          |        | MAXNM      |                                             |
|          |        | +3         | # OTHERWISE C(MPAC) = B(MPAC).              |
|          | DAD    | RVQ        |                                             |
|          |        | MAXNM      |                                             |
| +3       | DLOAD  | RVQ        | # (USED BY P30 - P37 ALSO)                  |
|          |        | MAXNM      |                                             |
| MAXNM    | 2OCT   | 0106505603 |                                             |

# Page 524

# There is no source code on this page --- HG 2009

This code is written to file src/R30.s.



**B.87 R31**

```

1569 <src/R31.s 1569>≡
Copyright: Public domain.
Filename: R31.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 505-510
Contact: Onno Hommes <ohommes@cmu.edu>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-11 OH Batch 2 Assignment Comanche Transcription
2009-05-20 RSB Corrected INSTALL -> INTSTALL
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 505

BANK 34
SETLOC R31
BANK

COUNT* $$/R31

R31CALL CAF PRI03
 TC FINDVAC
 EBANK= SUBEXIT
 2CADR V83CALL

```

|          |        |          |                                          |       |
|----------|--------|----------|------------------------------------------|-------|
| DSPDELAY | CAF    | 1SEC     |                                          |       |
|          | TC     | BANKCALL |                                          |       |
|          | CADR   | DELAYJOB |                                          |       |
|          | CA     | EXTVBACT |                                          |       |
|          | MASK   | BIT12    |                                          |       |
|          | EXTEND |          |                                          |       |
|          | BZF    | DSPDELAY |                                          |       |
| DISPN5X  | CA     | FLAGWRD9 | # TEST R31FLAG (IN SUNDANCE R31FLAG WILL |       |
|          | MASK   | BIT4     | # ALWAYS BE SET AS R34 DOES NOT EXIST.   |       |
|          | EXTEND |          |                                          |       |
|          | BZF    | +3       |                                          |       |
|          | CAF    | V16N54   | # R31 USE NOUN 54                        |       |
|          | TC     | +2       |                                          |       |
|          | CAF    | V16N53   | # R34 USE NOUN 53                        |       |
|          | TC     | BANKCALL |                                          |       |
|          | CADR   | GOMARKF  |                                          |       |
|          | TC     | B5OFF    |                                          |       |
|          | TC     | B5OFF    |                                          |       |
|          | TCF    | DISPN5X  |                                          |       |
| V83      | TC     | INTPRET  |                                          |       |
|          | GOTO   |          |                                          |       |
|          |        | HAVEBASE | # INTEG STATE VECTORS                    |       |
| V83CALL  | TC     | INTPRET  |                                          |       |
|          | GOTO   |          |                                          |       |
|          |        | STATEXTP | # EXTRAPOLATE STATE VECTORS              |       |
| COMPDISP | VLOAD  | VSU      |                                          |       |
|          |        | RATT     |                                          |       |
|          |        | RONE     |                                          |       |
|          | PUSH   | ABVAL    | # RATT-RONE TO OD                        | PD= 6 |
|          | STORE  | RANGE    | # METERS B-29                            |       |
|          | NORM   | VLOAD    |                                          |       |
|          |        | X1       | # RATT-RONE                              | PD= 0 |
|          | VSR1   |          |                                          |       |
|          | VSL*   | UNIT     |                                          |       |
|          |        | 0,1      |                                          |       |
|          | PDVL   | VSU      | # UNIT(LOS) TO OD                        | PD= 6 |
|          |        | VATT     |                                          |       |
|          |        | VONE     |                                          |       |
|          | DOT    |          | # (VATT-VONE).UNIT(LOS)                  | PD= 0 |
|          | SL1    |          |                                          |       |
|          | STCALL | RRATE    | # RANGE RATE M/CS B-7                    |       |
|          |        | CDUTRIG  | # TO INITIALIZE FOR *NBSM*               |       |
|          | CALL   |          |                                          |       |

July 12, 2016

1571

```
R34ANG VLOAD R34LOS # NOTE. PDL MUST = 0.
 UNIT
 RONE

 PDVL # UR TO OD PD= 6
 THISAXIS # UNITX FOR CM, UNITZ FOR LM
 BON VLOAD # CHK R31FLAG. ON=R31 THETA, OFF=R34 PHI
 R31FLAG
 +2
 12D
 CALL
 NBSM
 VXM PUSH # UXORZ TO 6D PD=12D
 REFSMMAT
 VPROJ VSL2
 OD
 BVSU UNIT
 6D
 PDVL VXV # UP/2 TO 12D PD=18D
 RONE
 VONE
 UNIT VXV
 RONE
 DOT PDVL # SIGN TO 12D, UP/2 TO MPAC PD=18D
 12D
 VSL1 DOT # UP.UXORZ
 6D
 SIGN SL1
 12D
 ACOS
 STOVL RTHETA
 RONE
 DOT BPL
 6D
 +5
 DLOAD BDSU # IF UXORZ.R NEG, RTHETA = 1 - RTHETA
 RTHETA
 DPPOSMAX
 STORE RTHETA # RTHETA BETWEEN 0 AND 1 REV.
 EXIT
 CAF BIT5 # HAVE WE BEEN ANSWERED
 MASK EXTVBACT
 EXTEND
 BZF ENDEXT # YES, DIE

 CS EXTVBACT
 MASK BIT12
```

|        |     |          |
|--------|-----|----------|
|        | ADS | EXTVBACT |
|        | TCF | V83      |
| V16N54 | VN  | 1654     |
| V16N53 | VN  | 1653     |

# Page 508

# STATEXTP DOES AN INITIAL PRECISION EXTRAPOLATION OF THE  
 # LEM STATE VECTOR TO PRESENT TIME OR TO PIPTIME IF AV G.  
 # IS ON AND SAVES AS BASE VECTOR. IF AV G IS ON RN + VN  
 # ARE USED AS THE CM STATE VECTOR AND THE INITIAL R RDOT  
 # RTHETA ARE COMPUTED WITH NO FURTHER INTEGRATION. IF AV  
 # G IS OFF A PRECISION EXTRAPOLATION IS MADE OF THE CM  
 # STATE VECTOR TO PRESENT TIME AND.....

#  
 # THE CM + LM STATE VECTORS ARE INTEGRATED TO PRES TIME  
 # USING PRECISION OR CONIC AS SURFFLAG IS SET OR CLEAR.  
 #

# IF AV G IS ON THEN  
 # SUBSEQUENT PASSES WILL PROVIDE  
 # USE OF RN + VN AS CM STATE VECTOR AND THE LM STATE  
 # VECTOR WILL BE PRECISION INTEGRATED USING LEMPREC  
 #

# IF SURFFLAG IS SET.  
 # CM STATE VECTOR RONE VONE + LM STATE VECTOR RATT  
 # VATT ARE USED IN COMPUTING R RDOT RTHETA.  
 #

|          |        |          |                                        |
|----------|--------|----------|----------------------------------------|
| STATEXTP | RTB    | BOF      | # INITIAL INTEGRATION                  |
|          |        | LOADTIME |                                        |
|          |        | V37FLAG  |                                        |
|          |        | +3       | # AV G OFF, USE PRES TIME              |
|          | CALL   | GETRVN   | # ON, USE RN VN PIPTIME                |
|          | STORE  | BASETIME | # PRES TIME OR PIPTIME                 |
|          | STCALL | TDEC1    |                                        |
|          |        | LEMPREC  |                                        |
|          | VLOAD  |          | # BASE VECTOR, LM                      |
|          |        | RATT1    |                                        |
|          | STOVL  | BASE0TP  | # POS.                                 |
|          |        | VATT1    |                                        |
|          | STORE  | BASE0TV  | # VEL.                                 |
|          | BON    | DLOAD    |                                        |
|          |        | V37FLAG  |                                        |
|          |        | COMPDISP | # COMPUTE R RDOT RTHETA FROM           |
|          |        |          | # RONE(RN) VONE(VN) RATT+VATT(LEMPREC) |

July 12, 2016

1573

```

 TAT
STCALL TDEC1
 CSMPREC
VLOAD # BASE VECTOR, CM
 RATT1
STOVL BASETHP # POS.
 VATT1
STORE BASETHV # VEL.
HAVEBASE BON RTB # SUBSEQUENT INTEGRATIONS
 V37FLAG
 GETRVN5
 LOADTIME
STCALL TDEC1 # AV G OFF, SET INTEG, OF CM
 INTSTALL
VLOAD CLEAR
 BASETHP
Page 509
 MOONFLAG
STOVL RCV
 BASETHV
STODL VCV
 BASETIME
BOF SET # GET APPROPRIATE MOONFLAG SETTING
 MOONTHIS
 +2
 MOONFLAG
CLEAR
 INTYPFLG
BON SET
 SURFFLAG
 +2
 INTYPFLG
STCALL TET # CONIC IF LM NOT DOWN
 INTEGRVS
 # INTEGRATION --- AT LAST ---
VLOAD
 RATT
STOVL RONE
 VATT
STODL VONE # GET SET FOR CONIC EXTRAP., OTHER.
 TAT
BON CALL
 SURFFLAG
 GETRVN6
 INTSTALL
SET # LEMPREC IF LM DOWN
 # ..CONIC IF NOT DOWN
 INTYPFLG
```

|            |        |          |                                  |
|------------|--------|----------|----------------------------------|
| OTHINT     | STORE  | TDEC1    | # ENTERED IF AV G ON TO INTEG LM |
|            | VLOAD  | CLEAR    |                                  |
|            |        | BASEOTP  |                                  |
|            |        | MOONFLAG |                                  |
|            | STOVL  | RCV      |                                  |
|            |        | BASEOTV  |                                  |
|            | STODL  | VCV      |                                  |
|            |        | BASETIME |                                  |
|            | BOF    | SET      |                                  |
|            |        | MOONTHIS |                                  |
|            |        | +2       |                                  |
|            |        | MOONFLAG |                                  |
|            | STCALL | TET      |                                  |
|            |        | INTEGRVS |                                  |
|            | GOTO   |          |                                  |
|            |        | COMPDISP | # COMPUTE R RDOT RTHETA          |
| GETRVN5    | CALL   |          | # AV G ON                        |
|            |        | GETRVN   |                                  |
|            | BON    | CALL     |                                  |
|            |        | SURFFLAG |                                  |
|            |        | GETRVN6  | # LM DOWN, LMPREC                |
| # Page 510 |        |          |                                  |
|            |        | INTSTALL |                                  |
|            | CLEAR  | GOTO     |                                  |
|            |        | INTYPFLG |                                  |
|            |        | OTHINT   |                                  |
| GETRVN6    | STCALL | TDEC1    |                                  |
|            |        | LEMPREC  |                                  |
|            | GOTO   |          |                                  |
|            |        | COMPDISP | # COMPUTE R RDOT RTHETA          |
| GETRVN     | STQ    |          |                                  |
|            |        | OD       |                                  |
|            | VLOAD  | GOTO     | # AV G ON, RONE = RN VONE = VN   |
|            |        | RN       | # AND USE PIPTIME                |
|            |        | +1       |                                  |
|            | STCALL | RONE     |                                  |
|            |        | +1       |                                  |
|            | VLOAD  | GOTO     |                                  |
|            |        | VN       |                                  |
|            |        | +1       |                                  |
|            | STODL  | VONE     |                                  |
|            |        | PIPTIME  |                                  |
|            | GOTO   |          |                                  |
|            |        | OD       |                                  |
|            | SETLOC | R34      |                                  |
|            | BANK   |          |                                  |

July 12, 2016

1575

```
R34LOS EXIT
 CA CDUS
 INDEX FIXLOC
 TS 9D
 CA CDUT
 INDEX FIXLOC
 TS 11D
 CA FIXLOC
 AD SIX
 COM
 INDEX FIXLOC
 TS X1
 TC INTPRET
 CALL
 SXTNB
 STCALL 12D
 R34ANG
```

This code is written to file `src/R31.s`.

**B.88 R60 62**

```

1576 <src/R60-62.s 1576>≡
 # Copyright: Public domain.
 # Filename: R60_62.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 390-398
 # Mod history: 2009-05-09 RSB Adapted from the Colossus249/ file
 # of the same name, using Comanche055 page
 # images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 390

 BANK 34
 SETLOC MANUVER
 BANK

 EBANK= TEMPR60

 COUNT 27/R60

 # CONFORMS TO GSOP CHAPTER FOUR REVISION LOGIC 09 JAN 18, 1968

 R60CSM TC MAKECADR
 TS TEMPR60

```



July 12, 2016

1577

# INSERT PRIODSP CHECK WITH R22 (V06N49) WITH JENNINGS BRODEUR

|          |       |           |                                            |
|----------|-------|-----------|--------------------------------------------|
| REDOMANN | CAF   | BIT6      |                                            |
|          | MASK  | FLAGWRD5  | # IS 3-AXIS FLAG SET                       |
|          | CCS   | A         |                                            |
|          | TCF   | TOBALL    | # YES                                      |
|          | TC    | INTPRET   |                                            |
|          | CALL  |           |                                            |
|          |       | VECPPOINT | # TO COMPUTE FINAL ANGLES                  |
|          | STORE | CPHI      | # STORE FINAL ANGLES -- CPHI, CTHETA, CPSI |
|          | EXIT  |           |                                            |

|        |      |          |                                        |
|--------|------|----------|----------------------------------------|
| TOBALL | CAF  | V06N18   |                                        |
|        | TC   | BANKCALL |                                        |
|        | CADR | GOPERF2R | # DISPLAY PLEASE PERFORM AUTO MANEUVER |
|        | TC   | R61TEST  |                                        |
|        | TC   | REDOMANC | # PROCEED                              |
|        | TCF  | ENDMANU1 | # ENTER I.E., FINISHED WITH R60        |
|        |      |          |                                        |
|        | TC   | CHKLINUS | # TO CHECK FOR PRIORITY DISPLAYS       |
|        | TC   | ENDOFJOB |                                        |

|          |       |           |                           |
|----------|-------|-----------|---------------------------|
| REDOMANC | CAF   | BIT6      |                           |
|          | MASK  | FLAGWRD5  | # IS 3-AXIS FLAG SET      |
|          | CCS   | A         |                           |
|          | TCF   | TOBALLC   | # YES                     |
|          | TC    | INTPRET   |                           |
|          | CALL  |           |                           |
|          |       | VECPPOINT | # TO COMPUTE FINAL ANGLES |
|          | STORE | CPHI      | # STORE ANGLES            |
|          | EXIT  |           |                           |

|            |        |          |                                  |
|------------|--------|----------|----------------------------------|
| TOBALLC    | CAF    | PRI030   | # IS MODE AUTO AND CTL GNC       |
| # Page 391 |        |          |                                  |
|            | EXTEND |          |                                  |
|            | RXOR   | CHAN31   |                                  |
|            | MASK   | 13,14,15 |                                  |
|            | EXTEND |          |                                  |
|            | BZF    | +2       | # AUTO, NON-FLASH N18            |
|            | TCF    | TOBALL   | # NOT AUTO                       |
|            |        |          |                                  |
|            | CAF    | V06N18   | # STATIC UP NON-FLASHING V06 N18 |
|            | TC     | BANKCALL |                                  |
|            | CADR   | GODSPR   |                                  |
|            | TC     | CHKLINUS |                                  |

|            |        |          |                                    |
|------------|--------|----------|------------------------------------|
| STARTMNV   | TC     | BANKCALL |                                    |
|            | CADR   | GOMANUR  |                                    |
| ENDMANUV   | TCF    | TOBALL   | # FINISHED MANEUVER.               |
| ENDMANU1   | TC     | DOWNFLAG | # RESET 3-AXIS FLAG                |
|            | ADRES  | 3AXISFLG | # BIT 6 FLAG 5                     |
|            | CAE    | TEMPR60  |                                    |
|            | TC     | BANKJUMP |                                    |
| CHKLINUS   | CS     | FLAGWRD4 |                                    |
|            | MASK   | BIT12    | # IS PRIORITY DISPLAY FLAG SET?    |
|            | CCS    | A        |                                    |
|            | TC     | Q        | # NO -- EXIT                       |
|            | CA     | Q        |                                    |
|            | TS     | MPAC +2  | # SAVE RETURN                      |
|            | CS     | THREE    | # OBTAIN LOCATION FOR RESTART      |
|            | AD     | BUF2     | # HOLD Q OF LAST DISPLAY           |
|            | TS     | TBASE1   |                                    |
|            | TC     | PHASCHNG |                                    |
|            | OCT    | 71       | # 1.7SPOT FOR RELINUS              |
|            | CAF    | BIT7     |                                    |
|            | TC     | LINUS    | # GO SET BITS FOR PRIORITY DISPLAY |
|            | TC     | MPAC +2  |                                    |
| RELINUS    | CAF    | BIT5     | # IS TRACK FLAG ON                 |
|            | MASK   | FLAGWRD1 |                                    |
|            | EXTEND |          |                                    |
|            | BZF    | GOREDO20 | # NO                               |
|            | TC     | UPFLAG   |                                    |
|            | ADRES  | PDSPFLAG | # R60 PRIODSP FLAG                 |
|            | TC     | UPFLAG   |                                    |
|            | ADRES  | TARG1FLG | # FOR R52                          |
|            | CAF    | ZERO     | # RESET TO ZERO, SINCE             |
| # Page 392 | TS     | OPTIND   | # OPTIND WAS SET TO -1 BY V379     |
|            | CAF    | PRI014   | # RESTORE ORIGINAL PRIORITY        |
|            | TC     | PRI0CHNG |                                    |
|            | TC     | TBASE1   |                                    |

```

GORED020 TC PHASCHNG
 OCT 111 # 1.11 FOR PIKUP20

 TC ENDOFJOB

R61TEST CA MODREG # ARE WE IN P00. IF YES THIS MUST BE
 EXTEND # VERB49 OR VERB89 SO DO ENDEXT.
 BZF ENDMANU1 # RESET 3-AXIS & RETURN. USER DOES ENDEXT
 CA FLAGWRD4 # ARE WE IN R61 (P20)
 MASK BIT12
 EXTEND
 BZF GOTOPOOH # NO
 TC GOTOV56 # YES

BIT14+7 OCT 20100
V06N18 VN 0618

```

# Page 393

# PROGRAM DESCRIPTION -- VECPOINT

#

# THIS INTERPRETIVE SUBROUTINE MAY BE USED TO POINT A SPACECRAFT AXIS IN A DESIRED DIRECTION.  
 # TO BE POINTED MUST APPEAR AS A HALF UNIT DOUBLE PRECISION VECTOR IN SUCCESSIVE LOCATIONS OF E  
 # BEGINNING WITH THE LOCATION CALLED SCAXIS. THE COMPONENTS OF THIS VECTOR ARE GIVEN IN SPACEC  
 # THE DIRECTION IN WHICH THIS AXIS IS TO BE POINTED MUST APPEAR AS A HALF UNIT DOUBLE PRECISION  
 # SUCCESSIVE LOCATIONS OF ERASABLE MEMORY BEGINNING WITH THE ADDRESS CALLED POINTVSM. THE COMP  
 # VECTOR ARE GIVEN IN STABLE MEMBER COORDINATES. WITH THIS INFORMTION VECPOINT COMPUTES A SET  
 # ANGLES (2'S COMPLEMENT) CORRESPONDING TO THE CROSS-PRODUCT ROTATION BETWEEN SCAXIS AND POINTV  
 # IN T(MPAC) BEFORE RETURNING TO THE CALLER.

#

# THIS ROTATION, HOWEVER, MAY BRING THE S/C INTO GIMBAL LOCK. WHEN POINTING A VECTOR IN THE Y-  
 # THE TRANSPONDER AXIS, OR THE AOT FOR THE LEM, THE PROGRAM WILL CORRECT THIS PROBLEM BY ROTATI  
 # PRODUCT ATTITUDE ABOUT POINTVSM BY A FIXED AMOUNT SUFFICIENT TO ROTATE THE DESIRED S/C ATTITU  
 # LOCK. IF THE AXIS TO BE POINTED IS MORE THAN 40.6 DEGREES BUT LESS THAN 60.5 DEG FROM THE +X  
 # THE ADDITIONAL ROTATION TO AVOID GIMBAL LOCK IS 35 DEGREES. IF THE AXIS IS MORE THAN 60.5 DE  
 # THE ADDITIONAL ROTATION IS 35 DEGREES. THE GIMBAL ANGLES CORRESPONDING TO THIS ATTITUDE ARE  
 # STORED AS 2'S COMPLEMENT ANGLES N T(MPAC) BEFORE RETURNING TO THE CALLER.

#

# WHEN POINTING THE X-AXIS, OR THE THRUST VECTOR, OR ANY VECTOR WITHIN 40.6 DEG OF THE X-AXIS,  
 # CANNOT CORRECT FOR A CROSS-PRODUCT ROTATION INTO GIMBAL LOCK. IN THIS CASE A PLATFORM REALIG  
 # REQUIRED TO POINT THE VECTOR IN THE DESIRED DIRECTION. AT PRESENT NO INDICATION IS GIVEN FOR  
 # EXCEPT THAT THE FINAL MIDDLE GIMBAL ANGLE IN MPAC +2 IS GREATER THAN 59 DEGREES.

#

# CALLING SEQUENCE

#

# 1) LOAD SCAXIS, POINTVSM

```

2) CALL
VECPOINT
#
RETURNS WITH
#
1) DESIRED OUTER GIMBAL ANGLE IN MPAC
2) DESIRED INNER GIMBAL ANGLE IN MPAC +1
3) DESIRED MIDDLE GIMBAL ANGLE IN MPAC +2
#
ERASABLES USED --
#
1) SCAXIS 6
2) POINTVSM 6
3) MIS 18
4) DEL 18
5) COF 6
6) VECQTEMP 1
7) ALL OF VAC AREA 43
#
TOTAL 99
#
SETLOC VECPT
BANK
Page 394
EBANK= BCDU
COUNT 27/VECPT
#
VECPOINT STQ BOV # SAVE RETURN ADDRESS
 VECQTEMP
 VECLEAR # AND CLEAR OVFIN
VECLEAR AXC,2 RTB
 MIS # READ THE PRESENT CDU ANGLES AND
 READCDUK # STORE THEM IN PD25, 26, 27
 STCALL 25D
 CDUTODCM # S/C AXES TO STABLE MEMBER AXES (MIS)
 VLOAD VXM
 POINTVSM # RESOLVE THE POINTING DIRECTION VF INTO
 MIS # INITIAL S/C AXES (VF = POINTVSM)
 UNIT
 STORE 28D
 VXV UNIT # PD 28 29 30 31 32 33
 SCAXIS # TAKE THE CROSS PRODUCT VF X VI
 BOV VCOMP # WHERE VI = SCAXIS
 PICKAXIS

```

July 12, 2016

1581

|            |       |          |                                             |     |
|------------|-------|----------|---------------------------------------------|-----|
|            | STODL | COF      | # CHECK MAGNITUDE                           |     |
|            |       | 36D      | # OF CROSS PRODUCT                          |     |
|            | DSU   | BMN      | # VECTOR, IF LESS                           |     |
|            |       | DPB-14   | # THAN B-14 ASSUME                          |     |
|            |       | PICKAXIS | # UNIT OPERATION                            |     |
|            | VLOAD | DOT      | # INVALID.                                  |     |
|            |       | SCAXIS   |                                             |     |
|            |       | 28D      |                                             |     |
|            | SL1   | ARCCOS   |                                             |     |
| COMPMATX   | CALL  |          | # NO COMPUTE THE TRANSFORMATION FROM        |     |
|            |       | DELCOMP  | # FINAL S/C AXES TO INITIAL S/C AXES MFI    |     |
|            | AXC,1 | AXC,2    |                                             |     |
|            |       | MIS      | # COMPUTE THE TRANSFORMATION FROM FINAL     |     |
|            |       | DEL      | # S/C AXES TO STABLE MEMBER AXES            |     |
|            | CALL  |          | # MFS = MIS MFI                             |     |
|            |       | MXM3     | # (IN PD LIST)                              |     |
|            | DLOAD | ABS      |                                             |     |
|            |       | 6        | # MFS6 = SIN(CPSI)                          | \$2 |
|            | DSU   | BMN      |                                             |     |
|            |       | SINGIMLC | # = SIN(59 DEGS)                            | \$2 |
|            |       | FINDGIMB | # /CPSI/ LESS THAN 59 DEGS.                 |     |
|            |       |          | # I.E., DESIRED ATTITUDE NOT IN GIMBAL LOCK |     |
|            | DLOAD | ABS      | # CHECK TO SEE IF WE ARE POINTING           |     |
|            |       | SCAXIS   | # THE THRUST AXIS                           |     |
|            | DSU   | BPL      |                                             |     |
|            |       | SINVEC1  | # SIN 49.4 DEGS                             | \$2 |
| # Page 395 |       | FINDGIMB | # IF SO, WE ARE TRYING TO POINT IT INTO     |     |
|            | VLOAD |          | # GIMBAL LOCK, ABORT COULD GO HERE          |     |
|            | STADR |          |                                             |     |
|            | STOVL | MIS +12D |                                             |     |
|            | STADR |          | # STORE MFS (IN PD LIST) IN MIS             |     |
|            | STOVL | MIS +6   |                                             |     |
|            | STADR |          |                                             |     |
|            | STOVL | MIS      |                                             |     |
|            |       | MIS +6   | # INNER GIMBAL AXIS IN FINAL S/C AXES       |     |
|            | BPL   | VCOMP    | # LOCATE THE IG AXIS DIRECTION CLOSEST TO   |     |
|            |       | IGSAMEX  | # FINAL X S/C AXIS                          |     |
| IGSAMEX    | VXV   | BMN      | # FIND THE SHORTEST WAY OF ROTATING THE     |     |
|            |       | SCAXIS   | # S/C OUT OF GIMBAL LOCK BY A ROTATION      |     |
|            |       | U=SCAXIS | # ABOUT +- SCAXIS, I.E., IF (IG (SGN MFS3)  |     |
|            |       |          | # X SCAXIS . XF) LESS THAN Q, U = SCAXIS    |     |
|            |       |          | # OTHERWISE U = -SCAXIS.                    |     |

```

 VLOAD VCOMP
 SCAXIS
 STCALL COF # ROTATE ABOUT -SCAXIS
 CHEKAXIS
U=SCAXIS VLOAD
 SCAXIS
 STORE COF # ROTATE ABOUT + SCAXIS
CHEKAXIS DLOAD ABS
 SCAXIS # SEE IF WE ARE POINTING THE AOT
 DSU BPL
 SINVEC2 # SIN 29.5 DEGS $2
 PICKANG1 # IF SO, ROTATE 50 DEGS ABOUT +- SCAXIS
 DLOAD GOTO # IF NOT, MUST BE POINTING THE TRANSPONDER
 VECANG2 # OR SOME VECTOR IN THE Y, OR Z PLANE
 COMPMFSN # IN THIS CASE ROTATE 35 DEGS TO GET OUT
 # OF GIMBAL LOCK (VECANG2 $360)
PICKANG1 DLOAD
 VECANG1 # = 50 DEGS. $360
COMPMFSN CALL
 DELCOMP # COMPUTE THE ROTATION ABOUT SCAXIS TO
 AXC,1 AXC,2 # BRING MFS OUT OF GIMBAL LOCK
 MIS
 DEL
 CALL
 MXM3 # COMPUTE THE NEW TRANSFORMATION FROM
 # DESIRED S/C AXES TO STABLE MEMBER AXES
 # WHICH WILL ALIGN VI WITH VF AND AVOID
 # GIMBAL LOCK
FINDGIMB AXC,1 CALL
 0 # EXTRACT THE COMMANDED CDU ANGLES FROM
 DCMTOCDU # THIS MATRIX
 RTB SETPD
 V1ST02S # CONVERT TO 2'S COMPLEMENT
Page 396
 0
 GOTO
 VECQTEMP # RETURN TO CALLER
PICKAXIS VLOAD DOT # IF VF X VI = 0, FIND VF, VI
 28D
 SCAXIS
 BMN TLOAD
 ROT180
 25D
 GOTO
 VECQTEMP # IF VF = VI, CDU DESIRED = PRESENT CDU
 # PRESENT CDU ANGLES

```

July 12, 2016

1583

```

ROT180 VLOAD VXV # IF VF, VI ANTI-PARALLEL, 180 DEG ROTATION
 MIS +6 # IS REQUIRED. Y STABLE MEMBER AXIS IN
 HIUNITX # INITIAL S/C AXES.
 UNIT VXV # FIND Y(SM) X X(I)
 SCAXIS # FIND UNIT(VI X UNIT(Y(SM) X X(I)))
 UNIT BOV # I.E., PICK A VECTOR IN THE PLANE OF X(I),
 PICKX # Y(SM) PERPENDICULAR TO VI
 STODL COF
 36D # CHECK MAGNITUDE
 DSU BMN # OF THIS VECTOR.
 DPB-14 # IF LESS THAN B-14,
 PICKX # PICK X-AXIS.
 VLOAD
 COF
XROT STODL COF
 HIDPHALF
 GOTO
 COMPMATX
PICKX VLOAD GOTO # PICK THE XAXIS IN THIS CASE
 HIUNITX
 XROT
 BANK 35
 SETLOC MANUVER1
 BANK

SINGIMLC 2DEC .4285836003 # = SIN(59) $2
SINVEC1 2DEC .3796356537 # = SIN(49.4) $2
SINVEC2 2DEC .2462117800 # = SIN(29.5) $2
VECANG1 2DEC .1388888889 # = 50 DEGREES $360
VECANG2 2DEC .0972222222 # = 35 DEGREES $360

1BITDP OCT 0 # KEEP THIS BEFORE DPB(-14) *****
DPB-14 OCT 00001
Page 397 OCT 00000
 BANK 34
 SETLOC MANUVER
 BANK
```

# Page 398

# ROUTINE FOR INITIATING AUTOMATIC MANEUVER VIA KEYBOARD (V49)

```
EBANK= CPHI

COUNT 27/R62

R62DISP CAF V06N22 # DISPLAY COMMAND ICDUS CPHI, CTHETA, CPHI
 TC BANKCALL
 CADR GOFLASH
 TCF ENDEXT
 TCF GOMOVE # PROCEED
 TCF R62DISP # ENTER

 # ASTRONAUT MAY LOAD NEW ICDUS AT THIS
 # POINT.
 # SET FOR 3-AXIS MANEUVER

GOMOVE TC UPFLAG
 ADRES 3AXISFLG

 TC BANKCALL
 CADR R60CSM
 TCF ENDEXT
```

This code is written to file `src/R60-62.s`.



**B.89 R63**

1585

*<src/R63.s 1585>*≡

```

Copyright: Public domain.
Filename: R63.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 338-341
Mod history: 2009-05-16 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 338
SUBROUTINE NAME: V89CALL
MOD NO: 0 DATE: 9 JAN 1968
MOD BY: DIGITAL DEVEL GROUP LOG SECTION: R63
#
FUNCTIONAL DESCRIPTION:
#
CALLED BY VERB 89 ENTER DURING P00. PRIO 10 USED. CALCULATES AND
DISPLAYS FINAL FDAI BALL ANGLES TO POINT LM +X OR +Z AXIS AT CSM.
#
1. KEY IN V 89 E ONLY IF IN PROG 00. IF NOT IN P00, OPERATOR ERROR AND
EXIT R63, OTHERWISE CONTINUE.
#
2. IF IN P00, DO IMU STATUS CHECK ROUTINE (R02BOTH). IF IMU ON AND ITS
ORIENTATION KNOWN TO LGC, CONTINUE.

```

```

#
3. FLASH DISPLAY V 04 N 06. R2 INDICATES WHICH SPACECRAFT AXIS IS TO
BE POINTED AT CSM. INITIAL CHOICE IS PREFERRED (+Z) AXIS (R2=1).
ASTRONAUT CAN CHANGE TO (+X) AXIS (R2 NOT =1) BY V 22 E 2 E. CONTINUE
AFTER KEYING IN PROCEED.
#
4. BOTH VEHICLE STATE VECTORS UPDATED BY CONIC EQS.
#
5. HALF MAGNITUDE UNIT LOS VECTOR (IN STABLE MEMBER COORDINATES) AND
HALF MAGNITUDE UNIT SPACECRAFT AXIS VECTOR (IN BODY COORDINATES)
PREPARED FOR VECPOINT.
#
6. GIMBAL ANGLES FROM VECPOINT TRANSFORMED INTO FDAI BALL ANGLES BY
BALLANGS. FLASH DISPLAY V 06 N 18 AND AWAIT RESPONSE.
#
7 RECYCLE -- RETURN TO STEP 4.
TERMINATE -- EXIT R63
PROCEED -- RESET 3AXISFLAG AND CALL R60LEM FOR ATTITUDE MANEUVER.
#
CALLING SEQUENCE: V 89 E.
#
SUBROUTINES CALLED: CHECKPOOH, R02BOTH, GOXDSPF, CSMCONIC, LEMCONIC,
VECPOINT, BALLANGS, R60LEM.
#
NORMAL EXIT MODES: TC ENDEXT
#
ALARMS: 1. OPERATOR ERROR IF NOT IN POO.
2. PROGRAM ALARM IF IMU IS OFF.
3. PROGRAM ALARM IF IMU ORIENTATION IS UNKNOWN.
#
OUTPUT: NONE
#
ERASABLE INITIALIZATION REQUIRED: NONE
#
DEBRIS: OPTION1, +1, TDEC1, PCINTVSM, SCAXIS, CPHI, CTHETA, CPSI,
Page 339
3AXISFLAG.

EBANK= RONE
BANK 32
SETLOC BAWLANGS
BANK

COUNT* $$/R63
V89CALL TC BANKCALL # IMU STATUS CHECK. RETURNS IF ORIENTATION
CADR R02BOTH # KNOWN. ALARMS IF NOT.

```

July 12, 2016

1587

```
V89RECL CAF THREE # ALLOW ASTRONAUT TO SELECT DESIRED
 TS OPTIONX # TRACKING ATTITUDE AXIS.
 CAF ONE
 TS OPTIONX +1
 CAF VB04N12 # V 04 N 12.
 TC BANKCALL
 CADR GOFLASH
 TC ENDEXT # TERMINATE
 TC +2 # PROCEED
 TC -5 # DATA IN. OPTION1+1 = 1 FOR Z AXIS
 TC INTPRET # 2 FOR X AXIS
 RTB DAD
 LOADTIME # READ PRESENT TIME
 DP1MIN
 STORE TSTART82 # SAVE TIME FOR LEMCONIC CALL
 STCALL TDEC1 # STORE TIME FOR CSMCONIC CALL
 CSMCONIC # CSM STATE VECTOR UPDATE
 VLOAD # CSMCONIC LEFT R VECTOR IN RATT
 RATT
 STODL RONE # SAVE FOR LINE OF SIGHT (LOS) COMPUTATION
 TSTART82
 STCALL TDEC1 # STORE TIME FOR LEMCONIC CALL
 LEMCONIC # LEM STATE VECTOR UPDATE
 VLOAD VSU # CSM POSITION -- LEM POSITION -- LOS
 RONE # LOS VECTOR LEFT IN MPAC
 RATT
 MXV RTB # (REFSMAT X LOS). TRANSFORMS LOS FROM
 REFSMMAT # REFERENCE COORD TO STAB MEMB COORD.
 NORMUNIT
 STORE POINTVSM # STORE LOS FOR VECPOINT CALCULATION
 EXIT
 CS OPTIONX +1 # 1 FOR Z AXIS. 2 FOR X AXIS.
 AD ONE
 EXTEND
 BZF ALINEZ
 TC INTPRET # X AXIS ALIGNMENT
 VLOAD
 UNITX # READ (.5, 0, 0)
 # Page 340
 V89CALL1 STCALL SCAXIS # STORE SELECTED ALIGNMENT AXIS
 VECPOINT # PUTS DESIRED GIM ANG (OG,IG,MG) IN TMPAC
 STORE CPHI # STOR GIMBAL ANGLES FOR BALLANGS CALL
 EXIT
 TC BANKCALL
 CADR BALLANGS # PUTS DESIRED BALL ANGLE IN FDAIX,Y,Z
 CAF VB06N18 # V 06 N 18
```

|            |       |          |                                           |
|------------|-------|----------|-------------------------------------------|
|            | TC    | BANKCALL | # NOUN 18 REFERS TO FDAIX,Y,Z             |
|            | CADR  | GOFLASH  |                                           |
|            | TC    | ENDEXT   | # TERMINATE                               |
|            | TC    | +2       | # PROCEED                                 |
|            | TC    | V89RECL  | # RECYCLE                                 |
|            | TC    | DOWNFLAG | # RESET 3 AXIS FLAG                       |
|            | ADRES | 3AXISFLG | # RESET BIT6 FLAG WORD 5                  |
|            | TC    | BANKCALL | # PERFORMS LEM MANEUVER TO ALIGN SELECTED |
|            | CADR  | R60LEM   | # SPACECRAFT AXIS TO CSM.                 |
|            | TCF   | ENDEXT   | # TERMINATE R63                           |
| ALINEZ     | TC    | INTPRET  | # Z AXIS ALIGNMENT                        |
|            | VLOAD | GOTO     |                                           |
|            |       | UNITZ    | # READ (0, 0, .5)                         |
|            |       | V89CALL1 |                                           |
| VB04N12    | VN    | 412      |                                           |
| VB06N18    | VN    | 0618     |                                           |
| # Page 341 |       |          |                                           |
| DP1MIN     | 2DEC  | 6000     |                                           |

This code is written to file `src/R63.s`.

## B.90 RADAR LEADIN ROUTINES

1589

*<src/RADAR-LEADIN-ROUTINES.s 1589>≡*

```
Copyright: Public domain.
Filename: RADAR_LEADIN_ROUTINES.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 490-491
Mod history: 2009-05-17 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969

Page 490

 BANK 25
 SETLOC RRLEADIN
 BANK

 EBANK= RSTACK

RADAR SAMPLING LOOP.

RADAMP COUNT* $$/RLEAD
 CCS RSAMPDT # TIMES NORMAL ONCE PER SECOND SAMLING
 TCF +2
 TCF TASKOVER # +0 INSERTED MANUALLY TERMINATES TEST.

 TC WAITLIST
```

```

EBANK= RSTACK
2CADR RADSAMP

CAF PRI025
TC NOVAC
EBANK= RSTACK
2CADR DORSAMP

CAF BIT14 # FOR CYCLIC SAMPLING, RTSTDEX=
EXTEND # RTSTLOC/2 + RTSTBASE
MP RTSTLOC
AD RTSTBASE # 0 FOR RR, 2 FOR LR.
TS RTSTDEX
TCF TASKOVER

DO THE ACTUAL RADAR SAMPLE.

DORSAMP TC VARADAR # SELECTS VARIABLE RADAR CHANNEL.
 TC BANKCALL
 CADR RADSTALL

 INCR RFAILCNT # ADVANCE FAIL COUNTER BUT ACCEPT BAD DATA

DORSAMP2 INHINT
 CA FLAGWRD5 # DON'T UPDATE RSTACK IF IN R77.
 MASK R77FLBIT
 CCS A
 TCF +4

 DXCH SAMPLSUM
 INDEX RTSTLOC
 DXCH RSTACK

 CS RTSTLOC # CYCLE RTSTLOC.
 AD RTSTMAX
 EXTEND

Page 491

 BZF +3
 CA RTSTLOC
 AD TWO # STORAGE IS DP
 TS RTSTLOC
 TCF ENDOFJOB # CONTINUOUS SAMPLING AND 2N TRIES -- GONE.

VARIABLE RADAR DATA CALLER FOR ONE MEASUREMENT ONLY.

```

July 12, 2016

1591

|         |       |         |                                         |
|---------|-------|---------|-----------------------------------------|
| VARADAR | CAF   | ONE     | # WILL BE SENT TO RADAR ROUTINE IN A BY |
|         | TS    | BUF2    | # SWCALL                                |
|         | INDEX | RTSTDEX |                                         |
|         | CAF   | RDRLOCS |                                         |
|         | TCF   | SWCALL  | # NOT TOUCHING Q.                       |
| RDRLOCS | CADR  | RRRANGE | # = 0                                   |
|         | CADR  | RRRDOT  | # = 1                                   |
|         | CADR  | LRVELX  | # = 2                                   |
|         | CADR  | LRVELY  | # = 3                                   |
|         | CADR  | LRVELZ  | # = 4                                   |
|         | CADR  | LRALT   | # = 5                                   |

This code is written to file `src/RADAR-LEADIN-ROUTINES.s`.

## B.91 RCS-CSM DAP EXECUTIVE PROGRAMS

```

1592 <src/RCS-CSM-DAP-EXECUTIVE-PROGRAMS.s 1592>≡
Copyright: Public domain.
Filename: RCS-CSM_DAP_EXECUTIVE_PROGRAMS.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1037-1038
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
2009-05-20 RSB A "Page N" comment was corrected.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 1037
CALCULATION OF AMGB, AMBG ONCE EVERY SECOND
#
AMGB = 1 SIN(PSI) 0
0 COS(PSI)COS(PHI) SIN(PHI)
0 -COS(PSI)SIN(PHI) COS(PHI)
#
AMBG = 1 -TAN(PSI)COS(PHI) TAN(PSI)SIN(PHI)
0 COS(PHI)/COS(PSI) -SIN(PHI)/COS(PSI)
0 SIN(PHI) COS(PHI)
#
WHERE PHI AND PSI ARE CDU ANGLES

```



July 12, 2016

1593

```
BANK 20
SETLOC DAPS8
BANK

COUNT* $$/DAPEX
EBANK= KMPAC
AMBGUPDT CA FLAGWRD6 # CHECK FOR RCS AUTOPILOT
EXTEND
BZMF ENDOFJOB # BIT15 = 0, BIT14 = 1
MASK BIT14 # IF NOT RCS, EXIT
EXTEND
BZF ENDOFJOB # TO PROTECT TVC DAP ON SWITCHOVER

CA CDUZ
TC SPSIN2
TS AMGB1 # CALCULATE AMGB
CA CDUZ
TC SPCOS2
TS CAPSI # MUST CHECK FOR GIMBAL LOCK
CAF QUADANGL # = 7.25 DEGREES JET QUAD ANGULAR OFFSET
EXTEND
MSU CDUX
COM # CDUX - 7.25 DEG
TC SPCOS1
TS AMGB8
EXTEND
MP CAPSI
TS AMGB4
CAF QUADANGL
EXTEND
MSU CDUX
COM # CDUX - 7.25 DEG
TC SPSIN1
TS AMGB5
EXTEND
MP CAPSI
COM

Page 1038
TS AMGB7
TCF ENDOFJOB
QUADANGL DEC 660 # = 7.25 DEGREES
```

This code is written to file `src/RCS-CSM-DAP-EXECUTIVE-PROGRAMS.s`.

## B.92 RCS-CSM DIGITAL AUTOPILOT

```

1594 (src/RCS-CSM-DIGITAL-AUTOPILOT.s 1594)≡
Copyright: Public domain.
Filename: RCS-CSM_DIGITAL_AUTOPILOT.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1002-1024
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 1002
T5 INTERRUPT PROGRAM FOR THE RCS-CSM AUTOPILOT
#
START OF T5 INTERRUPT PROGRAM
#
BANK 20
SETLOC DAPS3
BANK
#
COUNT 21/DAPRC
#
EBANK= KMPAC
REDORCS LXCH BANKRUPT # RESTART OF AUTOPILOT COMES HERE
CA T5PHASE # ON A T5 RUPT

```

July 12, 2016

1595

```
EXTEND
BZMF +2 # IF T5PHASE +0, -0, OR -, RESET TO -
TCF +3 # IF T5PHASE +, LEAVE IT +. DO A FRESHDAP
CS ONE
TS T5PHASE
EXTEND
DCA RCSLOC
DXCH T5LOC # HOOK UP T5RUPT TO AUTOPILOT
TCF RCSATT +1
EBANK= KMPAC
RCSLOC 2CADR RCSATT

RCSATT LXCH BANKRUPT # SAVE BB
EXTEND # SAVE Q
QXCH QRUPT
CAF BIT15 # BIT15 CHAN31 = 0 IF IMU POWER IS ON AND
EXTEND # S/C CONT SW IS IN CMC (I.E., IF G/C AUTO
RAND CHAN31 # PILOT IS FULLY ENABLED)
EXTEND
BZF SETT5 # IF G/C AUTOPILOT IS FULLY ENABLED,
GO TO SETT5

CS RCSFLAGS # IF G/C AUTOPILOT IS NOT FULLY ENABLED,
MASK BIT14
ADS RCSFLAGS # SET NORATE FLAG,
CAF POSMAX
TS HOLDFLAG # SET HOLDFLAG +,
CAF ZERO # ZERO ERRORX, ERRORY, AND ERRORZ,
TS ERRORX
TS ERRORY
TS ERRORZ
CAF BIT14
EXTEND
RAND CHAN31 # AND CHECK FREE FUNCTION (BIT14 CHAN31).
EXTEND

Page 1003
BZF SETT5 # IF IN FREE MODE, GO TO SETT5.

TS T5PHASE # IF NOT IN FREE MODE,
CAF OCT37766 # SCHEDULE REINITIALIZATION (FRESHDAP)
TS TIME5 # IN 100 MS VIA T5RUPT

TCR ZEROJET # ZERO JET CHANNELS IN 14 MS VIA ZEROJET

DELTATT TCF KMATRIX
OCT 37770 # 80MS (TIME5)
```

```

DELTATT2 OCT 37776 # 20MS (TIME5)
ONESEK DEC 16284 # 1 SEC(TIME5)
CHAN5 EQUALS 5
CHAN6 EQUALS 6
PRI034A = PRI034

```

```

CHECK PHASE OF T5 PROGRAM
#
BECAUSE OF THE LENGTH OF THE T5 PROGRAM,IT HAS BEEN DIVIDED INTO
THREE PARTS, T5PHASE1, T5PHASE2, AND THE JET SELECTION LOGIC,
TO ALLOW FOR THE EXECUTION OF OTHER
INTERRUPTS. T5PHASE IS ALSO USED IN THE INITIALIZATION OF THE AUTOPILOT
VARIABLES AT TURN ON.
#
THE CODING OF T5PHASE IS...
#
+ = INITIALIZE T5 RCS-CSM AUTOPILOT
T5PHASE = +0 = PHASE2 OF THE T5 PROGRAM
- = RESTART DAP
-0 = PHASE1 OF THE T5 PROGRAM

```

```

SETT5 CCS T5PHASE
 TCF FRESHDAP # TURN ON AUTOPILOT
 TCF T5PHASE2 # BRANCH TO PHASE2 OF PROGRAM
 TCF REDAP # RESTART AUTOPILOT

 TS T5PHASE # PHASE 1 RESET FOR PHASE 2
 CA TIME5
 TS T5TIME # USED IN COMPENSATING FOR DELAYS IN T5
 CAF DELTATT2 # RESET FOR T5RUPT IN 20MS FOR PHASE2
 TS TIME5 # OF PROGRAM

```

```

Page 1004
IMU STATUS CHECK

```

```

 CS IMODES33 # CHECK IMU STATUS
 MASK BIT6 # BIT6 = 0 IMU OK
 CCS A # BIT6 = 1 NO IMU
 TCF RATEFILT
FREECHK CS RCSFLAGS # BIT14 INDICATES THAT RATES HAVE NOT BEEN
 MASK BIT14 # INITIALIZED
 ADS RCSFLAGS
 CAF BIT14 # NO ATTITUDE REFERENCE
 TS HOLDFLAG # STOP ANY AUTOMATIC STEERING AND PREPARE
 # TO PICK UP CDU ANGLES UPON RESUMPTION OF
 # ATTITUDE HOLD

```

July 12, 2016

1597

```

 EXTEND
 RAND CHAN31 # CHECK FOR FREE MODE
 EXTEND
 BZF KRESUME1 # IN FREE MODE PROVIDE FREE CONTROL ONLY
 TCF REINIT #TILT.....
BITS4,5 OCT 30

RATEFILT CA RCSFLAGS # SEE IF RATEFILTER HAS BEEN INITIALIZED
 MASK BIT14
 EXTEND # IF SO, PROCEED WITH RATE DERIVATION
 BZF +2
 TCF KMATRIX # IF NOT, SKIP RATE DERIVATION

RATE FILTER TIMING = 7.72 MS
#
RATE FILTER EQUATIONS
#
#
$DRHO = DELRHO - (.1)ADOT + (1 = GAIN1)DRHO$
-1
#
$ADOT = ADOT -1 + GAIN2 DRHO + KMJ DFT$
#
*
WHERE $DEL RHO = AMGB (CDU - CDU)$
-1

DRHOLOOP CAF TWO
 TS SPNDX
 DOUBLE
 TS DPNDX
 INDEX DPNDX
 CS DRHO # DRHO SCALED 180 DEGS
 EXTEND
 INDEX ATTKALMN # PICK UP DESIRED FILTER GAIN
 MP GAIN1
 INDEX DPNDX
 DAS DRHO # (1 -.064) DRHO
 EXTEND

Page 1005
 INDEX DPNDX
 DCS ADOT
 DXCH KMPAC # -(.1)ADOT
 CA QUARTER
 TC SMALLMP
 DXCH KMPAC
```

```

INDEX DPNDX
DAS DRHO
CCS SPNDX
TCF DRHOLoop

CA CDUX
XCH RHO
EXTEND
MSU RHO
COM
MEASURED BODY RATES--
_ * _ _
DELRHO = AMGB (CDU - CDU)
-1

ZL
DXCH DELTEMPX
CA CDUY
XCH RH01
EXTEND
MSU RH01
COM
TS T5TEMP
EXTEND
MP AMGB1
DAS DELTEMPX
(CDUY - RH01) SCALED 90 DEGS
DELTEMPX = (CDUX-RHO) + AMGB1(CDUY-RH01)
MUST BE DOUBLE PRECISION OR WILL LOSE
PULSES

CA AMGB4
EXTEND
MP T5TEMP
DXCH DELTEMPY
CA AMGB7
EXTEND
MP T5TEMP
DXCH DELTEMPZ
CA CDUZ
XCH RH02
EXTEND
MSU RH02
COM
TS T5TEMP
EXTEND
MP AMGB5
DAS DELTEMPY
(CDUZ - RH02) SCALED 90 DEGS
DELTEMPY = AMGB4(CDUY-RH01)
+ AMGB5(CDUZ-RH02)

CA AMGB8
EXTEND

```

July 12, 2016

1599

```
ADOTLOOP MP T5TEMP
 DAS DELTEMPZ # DELTEMPZ = AMBG7(CDUY-RH01)
 # + AMGB8(CDUZ-RH02)
 CAF TWO
 TS SPNDX
 DOUBLE
 TS DPNDX
 EXTEND
 INDEX DPNDX
 DCA DELTEMPX
 INDEX DPNDX
 DAS DRHO
 EXTEND
 INDEX DPNDX
 DCA DELTEMPX
 INDEX DPNDX
 DAS MERRORX
 INDEX DPNDX
 CA DRHO
 DOUBLE
 DOUBLE
 EXTEND
 INDEX ATTKALMN # PICK UP DESIRED FILTER GAINS
 MP GAIN2
 INDEX DPNDX # ADOT + (.16)(.1)DRHO
 DAS ADOT # -1
 INDEX SPNDX # S/C TORQUE TO INERTIA RATIO
 CA KMJ # SCALED (450)(1600)/(57.3)(16384)=1/1.3
 EXTEND
 INDEX SPNDX
 MP DFT
 INDEX DPNDX
 DAS ADOT # KMJ(DFT)
 CCS SPNDX
 TCF ADOTLOOP # END CALCULATION OF VEHICLE RATES
KMATRIX CA ATTSEC
 MASK LOW4
 CCS A
 TCF TENTHSEK
 CAF PRI034 # CALL FOR 1 SEC UPDATE OF TRANSFORMATION
 TC NOVAC # MATRIX FROM GIMBAL AXES TO BODY AXES
 EBANK= KMPAC
 2CADR AMBGUPDT
 CAF NINE
```

TENTHSEK            TS            ATTSEC

# Page 1007

# WHEN AUTOMATIC MANEUVERS ARE BEING PERFORMED, THE FOLLOWING ANGLE ADDITION MUST BE  
# SEQUENCE OF ANGULAR COMMANDS TO THE AUTOPILOT--

#

#            CDUXD = CDUXD + DELCDUX            (DOUBLE PRECISION)

#            CDUYD = CDUYD + DELCDUY            (DOUBLE PRECISION)

#            CDUZD = CDUZD + DELCDUZ            (DOUBLE PRECISION)

#

# THE STEERING PROGRAMS:

#            1) ATTITUDE MANEUVER ROUTINE

#            2) LEM TRACKING

#

# SHOULD GENERATE THE DESIRED ANGLES (CDUXD, CDUYD, CDUZD) AS WELL AS THE INCREMENTAL  
# DELCDUZ) SO THAT THE GIMBAL ANGLE COMMANDS CAN BE INTERPOLATED BETWEEN UPDATES.

#

# HOLDFLAG CODING:

#

#            + = GRAB PRESENT CDU ANGLES AND STORE IN THETADX, THETADY, THETADZ  
#            AND PERFORM ATTITUDE HOLD ABOUT THESE ANGLES  
#            ALSO IGNORE AUTOMATIC STEERING

#            SET = + BY

#            1) INITIALIZATION PHASE OF AUTOPILOT

#            2) OCCURANCE OF RHC COMMANDS

#            3) FREE MODE

#            4) SWITCH OVER TO ATTITUDE HOLD FROM AUTO

#            WHILE DOING AUTOMATIC STEERING (IN THIS CASE  
#            HOLDFLAG IS NOT ACTUALLY SET TO +, BUT THE LOGIC  
#            FUNCTIONS AS IF IT WERE.)

#            5) S/C CONTROL SWITCH IS SCS

#            6) IMU POWER OFF

#            +0 = IN ATTITUDE HOLD ABOUT A PREVIOUSLY ESTABLISHED REFERENCE

#            - = PERFORMING AUTOMATIC MANEUVER

#            -0 = NOT USED AT PRESENT

#

# NOTE THAT THIS FLAG MUST BE SET = - BY THE STEERING PROGRAM IF IT IS TO COMMAND THE

# SINCE ASTRONAUT ACTION MAY CHANGE THE HOLDFLAG SETTING, IT SHOULD BE MONITORED BY THE

# DETERMINE IF THE AUTOMATIC SEQUENCE HAS BEEN INTERRUPTED AND IF SO, TAKE THE APPROPRIATE

CS            HOLDFLAG

EXTEND

BZMF        DACNDLS

# IF HOLDFLAG +0,-0,+, BYPASS AUTOMATIC  
# COMMANDS.

DCDUINCR     CAF        TWO

DELOOP       TS        SPNDX



```

DOUBLE
TS DPNDX
EXTEND
INDEX A
DCA CDUXD

Page 1008

DXCH KMPAC
EXTEND
INDEX DPNDX
DCA DELCDUX
TC DPADD
EXTEND
DCA KMPAC
INDEX SPNDX
TS THETADX
INDEX DPNDX
DXCH CDUXD
CCS SPNDX
TCF DELOOP

```

# Page 1009

# RCS-CSM AUTOPILOT ATTITUDE ERROR DISPLAY

#

# THREE TYPES OF ATTITUDE ERRORS MAY BE DISPLAYED ON THE FDAI:

#

```

MODE 1) AUTOPILOT FOLLOWING ERRORS SELECTED BY V61E
GENERATED INTERNALLY BY THE AUTOPILOT
#

```

#

```

MODE 2) TOTAL ATTITUDE ERRORS SELECTED BY V62E
WITH RESPECT TO THE CONTENTS OF N22
#

```

#

```

MODE 3) TOTAL ASTRONAUT ATTITUDE ERRORS SELECTED BY V63E
WITH RESPECT TO THE CONTENTS OF N17
#

```

#

# MODE 1 IS PROVIDED AS A MONITOR OF THE RCS DAP AND ITS ABILITY TO TRACK AUTOMATIC STEERING CO

# MODE THE ATTITUDE ERRORS WILL BE ZEROED WHEN THE CMC MODE SWITCH IS IN FREE.

#

# MODE 2 IS PROVIDED TO ASSIST THE CREW IN MANUALLY MANEUVERING THE S/C TO THE ATTITUDE (GIMBAL  
# IN N22. THE ATTITUDE ERRORS WRT THESE ANGLES AND THE CURRENT CDU ANGLES ARE RESOLVED INTO S/  
# AS A FLY-TO INDICATOR.

#

# MODE 3 IS PROVIDED TO ASSIST THE CREW IN MANUALLY MANEUVERING THE S/C TO THE ATTITUDE (GIMBAL  
# IN N17. THE ATTITUDE ERRORS WRT THESE ANGLES AND THE CURRENT CDU ANGLES ARE RESOLVED INTO S/  
# AS A FLY-TO INDICATOR.

#

# V60 IS PROVIDED TO LOAD N17 WITH A SNAPSHOT OF THE CURRENT CDU ANGLES, THUS SYNCHRONIZING THE

# WITH THE CURRENT S/C ATTITUDE. THIS VERB MAY BE USED AT ANY TIME.

#

# THESE DISPLAYS WILL BE AVAILABLE IN ANY MODE (AUTO, HOLD, FREE, G+N, OR SCS) ONCE T

# INITIATED VIA V46E. MODE 1, HOWEVER, WILL BE MEANINGFUL ONLY IN G+N AUTO OR HOLD.

# V25N17) AN ATTITUDE REFERENCE (DESIRED GIMBAL ANGLES) INTO N17 AT ANY TIME.

|         |        |          |                                           |
|---------|--------|----------|-------------------------------------------|
| DACNDLS | CS     | RCSFLAGS | # ALTERNATE BETWEEN FDAIDSP1 AND FDAIDSP2 |
|         | MASK   | BIT4     |                                           |
|         | EXTEND |          |                                           |
|         | BZF    | FDAIDSP2 |                                           |

|          |     |          |
|----------|-----|----------|
| FDAIDSP1 | ADS | RCSFLAGS |
|          | TC  | NEEDLER  |

|          |     |        |               |
|----------|-----|--------|---------------|
| KRESUME1 | TCF | RESUME | # END PHASE 1 |
|----------|-----|--------|---------------|

# Page 1010

# FDAI ATTITUDE ERROR DISPLAY SUBROUTINE

#

# PROGRAM DESCRIPTION: D. KEENE 5/24/67

#

# THIS SUBROUTINE IS USED TO DISPLAY ATTITUDE ERRORS ON THE FDAI VIA THE DIGITAL TO A

# IN THE CDUS. CARE IS TAKEN TO METER OUT THE APPROPRIATE NUMBER OF PULSES TO THE IN

# OVERFLOW, TO CONTROL THE RELAY SEQUENCING, AND TO AVOID INTERFERENCE WITH THE COAR

# THE DACS.

#

# CALLING SEQUENCE:

#

# DURING THE INITIALIZATION SECTION OF THE USER'S PROGRAM, BIT3 OF RCSFLAGS SHOULD BE

# TURN-ON SEQUENCE WITHIN THE NEEDLES PROGRAM:

#

|   |      |          |             |
|---|------|----------|-------------|
| # | CS   | RCSFLAGS | # IN EBANK6 |
| # | MASK | BIT3     |             |
| # | ADS  | RCSFLAGS |             |

#

# THEREAFTER, THE ATTITUDE ERRORS GENERATED BY THE USER SHOULD BE TRANSFERRED TO THE P

#

|   |     |                    |                                   |
|---|-----|--------------------|-----------------------------------|
| # | AK  | SCALED 180 DEGREES | NOTE: THESE LOCATIONS ARE SUBJECT |
| # | AK1 | SCALED 180 DEGREES | TO CHANGE                         |
| # | AK2 | SCALED 180 DEGREES |                                   |

#

# FULL SCALED DEFLECTION CORRESPONDS TO 16 7/8 DEGREES OF ATTITUDE ERROR

# (= 384 BITS IN IMU ERROR COUNTER)

#

# A CALL TO NEEDLER WILL THEN UPDATE THE DISPLAY:

#

# INHINT

July 12, 2016

1603

```
TC IBNKCALL # NOTE: EBANK SHOULD BE SET TO E6
CADR NEEDLER
RELINT
#
THIS PROCESS SHOULD BE REPEATED EACH TIME THE ERRORS ARE UPDATED. AT LEAST 3 PASSES THRU THE
REQUIRED BEFORE ANYTHING IS ACTUALLY DISPLAYED ON THE ERROR METERS.
NOTE: EACH CALL TO NEEDLER MUST BE SEPARATED BY AT LEAST 50MS TO ASSURE PROPER RELAY SEQUENC
#
ERASABLE USED:
AK CDUXCMD
AK1 CDUYCMD
AK2 CDUZCMD
EDRIX A,L,Q
EDRIY T5TEMP
EDRIZ SPNDX
#
SWITCHES; RCSFLAGS BITS 3,2
#
I/O CHANNELS: CHAN12 BIT 4 (COARSE ALIGN - READ ONLY)
Page 1011
CHAN12 BIT 6 (IMU ERROR COUNTER ENABLE)
CHAN14 BIT 13,14,15 (DAC ACTIVITY)
#
SIGN CONVENTION: AK = THETAC - THETA
WHERE THETAC = COMMAND ANGLE
THETA = PRESENT ANGLE

NEEDLER CAF BIT4 # CHECK FOR COARSE ALIGN ENABLE
EXTEND
RAND CHAN12 # IF IN COARSE ALIGN DO NOT USE IMU
EXTEND
BZF NEEDLER1
CS RCSFLAGS # SET BIT3 FOR INITIALIZATION PASS
MASK BIT3
ADS RCSFLAGS
TC Q

NEEDLER1 CA RCSFLAGS
MASK SIX
EXTEND
BZF NEEDLES3
MASK BIT3
EXTEND
BZF NEEDLER2 # BIT3 = 0, BIT2 = 1

CS BIT6 # FIRST PASS BIT3 = 1
```

|             |        |            |                                          |
|-------------|--------|------------|------------------------------------------|
|             | EXTEND |            | # DISABLE IMU ERROR COUNTER TO ZERO DACS |
|             | WAND   | CHAN12     | # MUST WAIT AT LEAST 60 MS. BEFORE       |
| NEEDLE11    | CS     | ZERO       | # ENABLING COUNTERS.                     |
|             | TS     | AK         | # ZERO THE INPUTS ON FIRST PASS          |
|             | TS     | AK1        |                                          |
|             | TS     | AK2        |                                          |
|             | TS     | EDRIVEX    | # ZERO THE DISPLAY REGISTERS             |
|             | TS     | EDRIVEY    |                                          |
|             | TS     | EDRIVEZ    |                                          |
|             | TS     | CDUXCMD    | # ZERO THE OUT COUNTERS                  |
|             | TS     | CDUYCMD    |                                          |
|             | TS     | CDUZCMD    |                                          |
|             | CS     | SIX        | # RESET RCSFLAGS FOR PASS2               |
|             | MASK   | RCSFLAGS   |                                          |
|             | AD     | BIT2       |                                          |
|             | TS     | RCSFLAGS   |                                          |
|             | TC     | Q          | # END PASS1                              |
| NEEDLER2    | CAF    | BIT6       | # ENABLE IMU ERROR COUNTERS              |
|             | EXTEND |            |                                          |
|             | WOR    | CHAN12     |                                          |
|             | CS     | SIX        | # RESET RCSFLAGS TO DISPLAY ATTITUDE     |
| # Page 1012 |        |            |                                          |
|             | MASK   | RCSFLAGS   | # ERRORS WAIT AT LEAST 4 MS FOR          |
|             | TS     | RCSFLAGS   | # RELAY CLOSURE                          |
|             | TC     | Q          |                                          |
| NEEDLES3    | CAF    | BIT6       | # CHECK TO SEE IF IMU ERROR COUNTER      |
|             | EXTEND |            | # IS ENABLED                             |
|             | RAND   | CHAN12     |                                          |
|             | EXTEND |            | # IF NOT RECYCLE NEEDLES                 |
|             | BZF    | NEEDLER +5 |                                          |
| NEEDLES     | CAF    | TWO        |                                          |
| DACLOOP     | TS     | SPNDX      |                                          |
|             | CS     | QUARTER    |                                          |
|             | EXTEND |            |                                          |
|             | INDEX  | SPNDX      |                                          |
|             | MP     | AK         |                                          |
|             | TS     | L          |                                          |
|             | CCS    | A          |                                          |
|             | CA     | DACLIMIT   |                                          |
|             | TCF    | +2         |                                          |
|             | CS     | DACLIMIT   |                                          |
|             | AD     | L          |                                          |
|             | TS     | T5TEMP     | # OVFL0 CHK                              |

July 12, 2016

1605

```
TCF +4
INDEX A # ON OVERFLOW LIMIT OUTPUT TO +-384
CAF DACLIMIT
TS L
INDEX SPNDX
CS EDRIX # CURRENT VALUE OF DAC
AD L
INDEX SPNDX
ADS CDUXCMD
INDEX SPNDX
LXCH EDRIX
CCS SPNDX
TCF DACLOOP
CAF 13,14,15
EXTEND
WOR CHAN14 # SET DAC ACTIVITY BITS
TC Q
REINIT CAF DELAY200 #TILT LOGIC
 TS TIME5 # REINITIALIZE DAP IN 200MS
 TS T5PHASE
 TCF RESUME
DELAY200 DEC 16364 # 200MS
 DEC -384
```

# Page 1013

```
DACLIMIT DEC 16000
 DEC 384
```

# Page 1014

# INITIALIZATION PROGRAM FOR RCS-CSM AUTOPILOT

#

# THE FOLLOWING QUANTITIES WILL BE ZEROED AND SHOULD APPEAR IN CONSECUTIVE LOCATIONS IN MEMORY

#

```
WBODY (+1) DFT TAU2
WBODY1 (+1) DFT1 BIAS
WBODY2 (+1) DFT2 BIAS1
ADOT (+1) DRHO (+1) BIAS2
ADOT1 (+1) DRHO1 (+1) ERRORX
ADOT2 (+1) DRHO2 (+1) ERRORY
MERRORX (+1) ATTSEC ERRORZ
MERRORY (+1) TAU
MERRORZ (+1) TAU1
```

```
FRESHDAP CAF ONE # RESET HOLDFLAG TO STOP AUTOMATIC
```

|             |                                                      |                                                                           |                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | TS                                                   | HOLDFLAG                                                                  | # STEERING AND PREPARE TO PICK UP AN<br># ATTITUDE HOLD REFERENCE                                                                                                                                                        |
| REDAP       | TC<br>CADR                                           | IBNKCALL<br>S41.2                                                         | # DECODE DAPDATR1, DAPDATR2 FOR DEADBANDS<br># RATES, QUADFAILS, QUAD MANAGEMENT                                                                                                                                         |
|             | TC<br>CADR                                           | IBNKCALL<br>S40.14                                                        | # DECODE IXX, IAVG, AND CONVERT<br># TO AUTOPILOT GAINS                                                                                                                                                                  |
| ZEROT5      | CAF<br>TS<br>CAF<br>INDEX<br>TS<br>CCS<br>TCF<br>TCR | NO.T5VAR<br>SPNDX<br>ZERO<br>SPNDX<br>WBODY<br>SPNDX<br>ZEROT5<br>ZEROJET | # NO. LOCATIONS TO BE ZEROED MINUS ONE<br># ZERO ALL NECESSARY ERASABLE REGISTERS                                                                                                                                        |
|             | CS<br>TS                                             | ZERO<br>CHANTEMP                                                          | # INITIALIZE MINIMUM IMPULSE CONTROL                                                                                                                                                                                     |
|             | TS                                                   | CH31TEMP                                                                  | # INITIALIZE RHC POSITION MEMORY FOR<br># MANUAL RATE MODES                                                                                                                                                              |
|             | CAF<br>TS                                            | =.24<br>SLOPE                                                             | # INITIALIZE SWITCHING LOGIC SLOPE                                                                                                                                                                                       |
|             | CAF<br>TS                                            | FOUR<br>T5TIME                                                            | # PHASE 0 RESETS FOR PHASE 2 INTERRUPT IN<br># 60 MS. PHASE 2 RESETS FOR PHASE 1 RUPT<br># IN (80MS - T5TIME(40MS)). THEREFORE<br># PHASE 1 (RATEFILTER) BEGINS CYCLING 100<br># MS FROM NOW AND EVERY 100MS THEREAFTER. |
|             | CAF<br>TS                                            | ELEVEN<br>ATTKALMN                                                        | # RESET TO PICK UP KALMAN FILTER TAINS<br># TO INITIALIZE THE S/C ANGULAR RATES                                                                                                                                          |
| # Page 1015 | CA<br>TS<br>CA<br>TS<br>CA<br>TS<br>CAF<br>TS        | CDUX<br>RHO<br>CDUY<br>RHO1<br>CDUZ<br>RHO2<br>ZERO<br>T5PHASE            | # RESET AUTOPILOT TO BEGIN EXECUTING<br># PHASE2 OF PROGRAM                                                                                                                                                              |

```

 CS IMODES33 # CHECK IMU STATUS
 MASK BIT6 # IF BIT6 = 0 IMU IN FINE ALIGN
 CCS A # IF BIT6 = 1 IMU NOT READY
 TCF IMUAOK
 TS ATTKALMN # CANNOT USE IMU
 CAF RCSINITB # PROVIDE FREE CONTROL ONLY
 TCF RCSSWIT # DON'T START UP RATE FILTER
 # SIGNAL NO RATE FILTER

IMUAOK CAF PRI034 # START MATRIX INITIALIZATION
 TC NOVAC # BYPASS IF IMU NOT IN FINE ALIGN
 EBANK= KMPAC
 2CADR AMBGUPDT

RCSSWIT CAF RCSINIT # CLEAR BIT14 --ASSUME WE HAVE A GOOD IMU
 TS RCSFLAGS # CLEAR BIT1 --INITIALIZE T6 PROGRAM
 # SET BIT3 --INITIALIZE NEEDLES
 # CLEAR BIT4 --RESET FOR FDAIDSP1
 CAF T5WAIT60 # NEXT T5RUPT 60 MS FROM NOW TO ALLOW IMU
 # ERROR COUNTER TO ZERO.
 # (MINIMUM DELAY = 15 MS)
 TS TIME5 # SINCE ATTKALMN IS +11, PROGRAM WILL THEN
 TC RESUME # PICK UP THE KALMAN FILTER GAINS. RATE
 # FILTER WILL BEGIN OPERATING ZOOMS FROM
 # NOW

CONSTANTS USED IN INITIALIZATION PROGRAM

NO.T5VAR DEC 36
=.24 DEC .24 # = SLOPE OF 0.6/SEC
RCSINIT OCT 00004
RCSINITB OCT 20004
T5WAIT60 DEC 16378 # = 6 CS
 EBANK= KMPAC
T6ADDR 2CADR T6START

ZEROJET CAF ELEVEN # ZERO BLAST2, BLAST1, BLAST, YWORD2,
 TS SPNDX # YWORD1, PWORD2, PWORD1, RWORD2,
 CAF ZERO # AND RWORD1

Page 1016
 INDEX SPNDX
 TS RWORD1
 CCS SPNDX
 TCF ZEROJET +1

```

```

CAF FOUR
TS BLAST1 +1
CAF ELEVEN
TS BLAST2 +1

CS BIT1
MASK RCSFLAGS
TS RCSFLAGS # RESET BIT1 OF RCSFLAGS TO 0

EXTEND
DCA T6ADDR
DXCH T6LOC
CAF =+14MS # ENABLE T6RUPT TO SHUT OFF JETS IN 14 MS.
TS TIME6
CAF BIT15
EXTEND
WOR CHAN13

TC Q

T5PHASE2 CCS ATTKALMN # IF (+) INITIALIZE RATE ESTIMATE
TCF KALUPDT

TCF +2 # ONLY IF ATTKALMN POSITIVE
TCF +1
CA DELTATT2 # RESET FOR PHASE3 IN 20 MS
XCH TIME5 # (JET SELECTION LOGIC)
ADS T5TIME # TO COMPENSATE FOR DELAYS IN T5RUPT

CA RCSFLAGS # IF A HIGH RATE AUTO MANEUVER IS IN
MASK BIT15 # PROGRESS (BIT 15 OF RCSFLAGS SET), SET
EXTEND # ATTKALMN TO -1
BZF NOHIAUTO # OTHERWISE SET ATTKALMN TO 0.
CS ONE
NOHIAUTO TS ATTKALMN

Page 1017
MANUAL ROTATION COMMANDS

CS OCT01760 # RESET FORCED FIRING BITS (BITS 10 TO 5
MASK RCSFLAGS # OF RCSFLAGS) TO ZERO
TS RCSFLAGS

EXTEND
READ CHAN31

```



July 12, 2016

1609

```
TS L
CA CH31TEMP
EXTEND
RXOR LCHAN
MASK MANROT # = OCT00077
EXTEND
BZMF NOCHANGE

LXCH A
TS CH31TEMP # SAVE CONTENTS OF CHANNEL 31 IN CH31TEMP

CA L
EXTEND
MP BIT5 # PUT BITS 6-1 OF A IN BITS 10-5 OF L
CA L
ADS RCSFLAGS # SET FORCED FIRING BITS FOR AXES WITH
 # CHANGES IN COMMAND. BITS 10,9 FOR
 # ROLL, BITS 8,7 FOR YAW, BITS 6,5 FOR
 # PITCH

CS RCSFLAGS # SET RATE DAMPING FLAGS (BITS 13,12, AND
MASK OCT16000 # 11 OF RCSFLAGS)
ADS RCSFLAGS

NOCHANGE CS CH31TEMP
MASK MANROT
EXTEND
BZMF AHFNOROT # IF NO MANUAL COMMANDS, GO TO AHFNOROT

TS HOLDFLAG # SET HOLDFLAG +

TC STICKCHK # WHEN THE RHC IS OUT OF DETENT, PMANNDX,
 # YMANNDX, AND RMANNDX ARE ALL SET, BY
 # MEANS OF STICKCHK, TO 0, 1, OR 2 FOR NO,
 # +, OR - ROTATION RESPECTIVELY AS
 # COMMANDED BY THE RHC.
 #
 # HOWEVER, IT IS WELL TO NOTE THAT AFTER
 # THE RHC IS RETURNED TO DETENT, THE
 # PROGRAM BRANCHES TO AHFNOROT AND AVOIDS
 # STICKCHK SO PMANNDX, YMANNDX, AND
 # RMANNDX ARE NOT RESET TO ZERO BUT RATHER
 # LEFT SET TO THEIR LAST OUT OF DETENT
 # VALUES.
```

```

CS FLAGWRD1 # SET STIKFLAG TO INFORM STEERING
MASK BIT14 # PROGRAMS (P20) THAT ASTRONAUT HAS
ADS FLAGWRD1 # ASSUMED ROTATIONAL CONTROL OF SPACECRAFT

CAF BIT14
EXTEND
RAND CHAN31
EXTEND
BZMF FREEFUNC

CA RCSFLAGS # EXAMINE RCSFLAGS TO SEE IF RATE FILTER
MASK BIT14 # HAS BEEN INITIALIZED
CCS A # IF SO, PROCEED WITH MANUAL RATE COMMANDS
TCF REINIT #TILT, RECYCLE TO INITIALIZE FILTER

CS FIVE # IF MANUAL MANEUVER IS AT HIGH RTE, SET
AD RATEINDX # ATTKALMN TO -1.
EXTEND
BZMF +3 # OTHERWISE, LEAVE ATTKALMN ALONE.
CS ONE
TS ATTKALMN

CAF TWO # AUTO-HOLD MANUAL ROTATION
TS SPNDX
DOUBLE
TS DPNDX
INDEX SPNDX # RMANNDX = 0 NO ROTATION
CA RMANNDX # = 1 + ROTATION
EXTEND
BZF NORATE # = 2 - ROTATION
 # IF NO ROTATION COMMAND ON THIS AXIS,
 # GO TO NORATE.

AD RATEINDX # RATEINDX = 0 0.05 DEG/SEC
TS Q # = 2 0.2 DEG/SEC
INDEX Q # = 4 0.5 DEG/SEC
CA MANTABLE -1 # = 6 2.0 DEG/SEC
EXTEND
MP BIT9 # MULTIPLY MANTABLE BY 2 TO THE -6
INDEX DPNDX # TO GET COMMANDED RATE.
DXCH WBODY # SET WBODY TO COMMANDED RATE.

CA RCSFLAGS
MASK OCT16000 # IS RATE DAMPING COMPLETED (BITS 13,12 AND
EXTEND # 11 OF RCSFLAGS ALL ZERO.) IF SO, GO TO
BZF MERUPDAT # MERUPDAT TO UPDATE CUMULATIVE ATTITUDE
 # ERROR.

```

July 12, 2016

1611

# Page 1019

|          |        |             |                                            |              |
|----------|--------|-------------|--------------------------------------------|--------------|
| ZEROER   | CA     | ZERO        | # ZEROER ZEROS MERRORS                     |              |
|          | ZL     |             |                                            |              |
|          | INDEX  | DPNDX       |                                            |              |
|          | DXCH   | MERRORX     |                                            |              |
|          | TCF    | SPNDXCHK    |                                            |              |
| NORATE   | ZL     |             |                                            |              |
|          | INDEX  | DPNDX       |                                            |              |
|          | DXCH   | WBODY       | # ZERO WBODY FOR THIS AXIS                 |              |
|          | CA     | RCSFLAGS    |                                            |              |
|          | MASK   | OCT16000    |                                            |              |
|          | EXTEND |             | # IS RATE DAMPING COMPLETED                |              |
|          | BZF    | SPNDXCHK    | # YES, KEEP CURRENT MERRORX GO TO SPNDXCHK |              |
|          | TCF    | ZEROER      | # NO, GO TO ZEROER                         |              |
| MERUPDAT | INDEX  | Q           | # MERRORX=MERRORX+MEASURED CHANGE IN ANGLE |              |
|          | CS     | MANTABLE -1 | # -COMMANDED CHANGE IN ANGLE               |              |
|          | EXTEND |             | # THE ADDITION OF MEASURED CHANGE IN ANGLE |              |
|          | MP     | BIT7        | # HAS ALREADY BEEN DONE IN THE RATE FILTER |              |
|          | INDEX  | DPNDX       | # COMMANDED CHANGE IN ANGLE = WBODY TIMES  |              |
|          | DAS    | MERRORX     | # .1SEC = MANTABLE ENTRY TIMES 2 TO THE -8 |              |
| SPNDXCHK | INDEX  | DPNDX       |                                            |              |
|          | CA     | MERRORX     |                                            |              |
|          | INDEX  | SPNDX       |                                            |              |
|          | TS     | ERRORX      | # ERRORX = HIGH ORDER WORD OF MERRORX      |              |
|          | CCS    | SPNDX       |                                            |              |
|          | TCF    | SETWBODY    |                                            |              |
|          | TCF    | JETS        |                                            |              |
| OCT01760 | OCT    | 01760       | # FORCED FIRING BITS MASK                  |              |
| OCT01400 | OCT    | 01400       | # ROLL FORCED FIRING MASK                  | ORDER OF     |
| OCT00060 | OCT    | 00060       | # PITCH FORCED FIRING MASK                 | DEFINITION   |
| OCT00300 | OCT    | 00300       | # YAW FORCED FIRING MASK                   | MUST BE      |
|          |        |             | #                                          | PRESERVED    |
|          |        |             | #                                          | FOR INDEXING |
| MANROT   | OCT    | 77          |                                            |              |
| OCT16000 | OCT    | 16000       | # RATE DAMPING FLAGS MASK                  |              |
| MANTABLE | DEC    | .0071111    |                                            |              |
|          | DEC    | -.0071111   |                                            |              |
|          | DEC    | .028444     |                                            |              |
|          | DEC    | -0.028444   |                                            |              |
|          | DEC    | .071111     |                                            |              |

1612

July 12, 2016

|             |       |          |                                            |
|-------------|-------|----------|--------------------------------------------|
|             | DEC   | -.071111 |                                            |
|             | DEC   | .284444  |                                            |
|             | DEC   | -.284444 |                                            |
| =+14MS      | DEC   | 23       |                                            |
| FREEFUNC    | INDEX | RMANNDX  | # ACCELERATION                             |
| # Page 1020 |       |          |                                            |
|             | CA    | FREETAU  | # COMMANDS                                 |
|             | TS    | TAU      |                                            |
|             | INDEX | PMANNDX  |                                            |
|             | CA    | FREETAU  | # FREETAU 0 SEC                            |
|             | TS    | TAU1     | # +1 +0.10 SEC                             |
|             | INDEX | YMANNDX  | # +2 -0.10 SEC                             |
|             | CA    | FREETAU  | # (+3) 0 SEC                               |
|             | TS    | TAU2     |                                            |
|             | TCF   | T6PROGM  |                                            |
|             |       |          |                                            |
| FREETAU     | DEC   | 0        |                                            |
|             | DEC   | 480      |                                            |
|             | DEC   | -480     |                                            |
|             | DEC   | 0        |                                            |
|             |       |          |                                            |
| T6PROGM     | CAF   | ZERO     | # FOR MANUAL ROTATIONS                     |
|             | TS    | ERRORX   |                                            |
|             | TS    | ERRORY   |                                            |
|             | TS    | ERRORZ   |                                            |
|             | TCF   | T6PROG   |                                            |
|             |       |          |                                            |
| # Page 1021 |       |          |                                            |
|             | DEC   | .2112    | # FILTER GAIN FOR TRANSLATION, LEM ON      |
|             | DEC   | .8400    | # FILTER GAIN FOR TRANSLATION 2(ZETA)WN DT |
|             | DEC   | .2112    | # FILTER GAIN FOR 4 DEGREE/SEC MANEUVERS   |
| GAIN1       | DEC   | .0640    | # KALMAN FILTER GAINS FOR INITIALIZATION   |
|             | DEC   | .3180    | # OF ATTITUDE RATES                        |
|             | DEC   | .3452    |                                            |
|             | DEC   | .3774    |                                            |
|             | DEC   | .4161    |                                            |
|             | DEC   | .4634    |                                            |
|             | DEC   | .5223    |                                            |
|             | DEC   | .5970    |                                            |
|             | DEC   | .6933    |                                            |
|             | DEC   | .8151    |                                            |
|             | DEC   | .9342    |                                            |
|             |       |          |                                            |
|             | DEC   | .0174    | # FILTER GAIN FOR TRANSLATION, LEM ON      |
|             | DEC   | .3600    | # FILTER GAIN FOR TRANSLATION (WN)(WN)DT   |
|             | DEC   | .0174    | # FILTER GAIN FOR 4 DEGREE/SEC MANEUVERS   |

July 12, 2016

1613

|             |        |          |                                          |
|-------------|--------|----------|------------------------------------------|
| GAIN2       | DEC    | .0016    | # SCALED 10                              |
|             | DEC    | .0454    |                                          |
|             | DEC    | .0545    |                                          |
|             | DEC    | .0666    |                                          |
|             | DEC    | .0832    |                                          |
|             | DEC    | .1069    |                                          |
|             | DEC    | .1422    |                                          |
|             | DEC    | .1985    |                                          |
|             | DEC    | .2955    |                                          |
|             | DEC    | .4817    |                                          |
|             | DEC    | .8683    |                                          |
| STICKCHK    | TS     | T5TEMP   |                                          |
|             | MASK   | THREE    | # INDICES FOR MANUAL ROTATION            |
|             | TS     | PMANNDX  |                                          |
|             | CA     | T5TEMP   |                                          |
|             | EXTEND |          | # MAN RATE 0 0 RATE (DP)                 |
|             | MP     | QUARTER  | # +1 + RATE (DP)                         |
|             | TS     | T5TEMP   | # +2 - RATE (DP)                         |
|             | MASK   | THREE    | # (+3) 0 RATE (DP)                       |
|             | TS     | YMANNDX  |                                          |
|             | CA     | T5TEMP   |                                          |
|             | EXTEND |          |                                          |
|             | MP     | QUARTER  |                                          |
|             | TS     | RMANNDX  |                                          |
|             | TC     | Q        |                                          |
| KALUPDT     | TS     | ATTKALMN | # INITIALIZATION OF ATTITUDE RATES USING |
|             |        |          | # KALMAN FILTER TAKES 1.1 SEC            |
|             | CA     | DELTATT  | # =1SEC - 80MS                           |
|             | AD     | T5TIME   | # + DELAYS                               |
| # Page 1022 | TS     | TIME5    |                                          |
|             | TCF    | +3       |                                          |
|             | CAF    | DELTATT2 | # SAFETY PLAY TO ASSURE                  |
|             | TS     | TIME5    | # A T5RUPT                               |
| KRESUME2    | CS     | ZERO     | # RESET FOR PHASE1                       |
|             | TS     | T5PHASE  | # RESUME INTERRUPTED PROGRAM             |
|             | TCF    | RESUME   |                                          |
| FDAIDSP2    | CS     | BIT4     | # RESET FOR FDAIDSP1                     |
|             | MASK   | RCSFLAGS |                                          |
|             | TS     | RCSFLAGS |                                          |
|             | CS     | FLAGWRD0 | # ON - DISPLAY ONE OF THE TOTAL ATTITUDE |
|             | MASK   | BIT9     | # ERRORS                                 |

```

EXTEND
BZF FDAITOTL
EXTEND
DCS ERRORX # OFF - DISPLAY AUTOPILOT FOLLOWING ERROR
DXCH AK
CS ERRORZ
TS AK2
TCF RESUME # END PHASE 1

FDAITOTL CA FLAGWRD9
 MASK BIT6
 EXTEND
 BZF WRTN17 # IS N22ORN17 (BIT6 OF FLAGWRD9) = 0
 # IF SO, GO TO WRTN17
WRTN22 EXTEND # OTHERWISE, CONTINUE ON TO WRTN22 AND
 DCA CTHETA # GET SET TO COMPUTE TOTAL ATTITUDE
 DXCH WTEMP # ERROR WRT N22 BY PICKING UP THE THREE
 CA CPHI # COMPONENTS OF N22

GETAKS EXTEND # COMPUTE TOTAL ATTITUDE ERROR FOR
 MSU CDUX # DISPLAY ON FDAI ERROR NEEDLES
 TS AK
 CA WTEMP
 EXTEND
 MSU CDUY
 TS T5TEMP
 EXTEND
 MP AMGB1
 ADS AK
 CA T5TEMP
 EXTEND
 MP AMGB4
 TS AK1
 CA T5TEMP
 EXTEND
 MP AMGB7
 TS AK2
 CA WTEMP +1
 EXTEND
 MSU CDUZ
 TS T5TEMP
 EXTEND
 MP AMGB5
 ADS AK1
 CA T5TEMP

```

July 12, 2016

1615

```
EXTEND
MP AMGB8
ADS AK2
TCF RESUME # END PHASE1 OF RCS DAP

WRTN17 EXTEND # GET SET TO COMPUTE TOTAL ASTRONAUT
DCA CPHIX +1 # ATTITUDE ERROR WRT N17 BY PICKING UP
DXCH WTEMP # THE THREE COMPONENTS OF N17
CA CPHIX
TCF GETAKS
```

# Page 1024 (empty page)

This code is written to file `src/RCS-CSM-DIGITAL-AUTOPILOT.s`.

## B.93 RCS FAILURE MONITOR

```

1616 <src/RCS-FAILURE-MONITOR.s 1616>≡
 # Copyright: Public domain.
 # Filename: RCS_FAILURE_MONITOR.agc
 # Purpose: Part of the source code for Luminary 1A build 099.
 # It is part of the source code for the Lunar Module's (LM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 190-192
 # Mod history: 2009-05-19 HG Transcribed from page images.
 #
 # This source code has been transcribed or otherwise adapted from
 # digitized images of a hardcopy from the MIT Museum. The digitization
 # was performed by Paul Fjeld, and arranged for by Deborah Douglas of
 # the Museum. Many thanks to both. The images (with suitable reduction
 # in storage size and consequent reduction in image quality as well) are
 # available online at www.ibiblio.org/apollo. If for some reason you
 # find that the images are illegible, contact me at info@sandroid.org
 # about getting access to the (much) higher-quality images which Paul
 # actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
 # 16:27 JULY 14, 1969
 #
 # Page 190
 # PROGRAM DESCRIPTION:
 #
 # AUTHOR: J. S. MILLER
 #
 # MODIFIED 6 MARCH 1968 BY P. S. WEISSMAN TO SET UP JOB FOR 1/ACCS WHEN THE MASKS ARE
 #
 # THIS ROUTINE IS ATTACHED TO T4RUPT, AND IS ENTERED EVERY 480 MS. ITS FUNCTION IS TO
 # OF CHANNEL 32 TO SEE IF ANY ISOLATION-VALVE CLOSURE BITS HAVE APPEARED OR DISAPPEARED
 # FAILURES BY LAMPS LIT BY THE GRUMMAN FAILURE-DETECTION CIRCUITRY; THEY MAY RESPOND
 # ISOLATE PAIRS OF JETS FROM THE PROPELLANT TANKS AND SET BITS IN CHANNEL 32). IN THE
 # DIFFER FROM 'PVALVEST', THE RECORD OF ACTIONS TAKEN BY THIS ROUTINE, THE APPROPRIATE
 # 'CH6MASK', USED BY THE DAP JET-SELECTION LOGIC, ARE UPDATED, AS IS 'PVALVEST'. TO
 # ROUTINE, NO MORE THAN ONE CHANGE IS ACCEPTED PER ENTRY. THE HIGHEST-NUMBERED BIT IN
 # ACTION IS THE ONE PROCESSED.
 #
 # THE CODING IN THE FAILURE MONITOR HAS BEEN WRITTEN SO AS TO HAVE ALMOST COMPLETE RE

```



July 12, 2016

1617

```
EXAMPLE, NO ASSUMPTION IS MADE WHEN SETTING A 'CH5MASK' BIT TO 1 THAT THE PREVIOUS STATE IS C
COURSE SHOULD BE. ONE CASE WHICH MAY BE SEEN TO EVADE PROTECTION IS THE OCCURRENCE OF A REST
ONE OR BOTH DAP MASK-WORDS BUT BEFORE UPDATING 'PVALVEST', COUPLED WITH A CHANGE IN THE VALVE
FORMER STATE. THE CONSEQUENCE OF THIS IS THAT THE NEXT ENTRY WOULD NOT SEE THE CHANGE INCOM
ORATED BY THE LAST PASS (BECAUSE IT WENT AWAY AT JUST THE RIGHT TIME), BUT THE DAP MASK-WORDS
THIS COMBINATION OF EVENTS SEEMS QUITE REMOTE, BUT NOT IMPOSSIBLE UNLESS THE CREW OPERATES TH
SECOND INTERVALS OR LONGER. IN ANY EVENT, A DISAGREEMENT BETWEEN REALITY AND THE DAP MASKS W
THE MISINTERPRETED SWITCH IS REVERSED AND THEN RESTORED TO ITS CORRECT POSITION (SLOWLY).
#
CALLING SEQUENCE:
#
TCF RCSMONIT # (IN INTERRUPT MODE, EVERY 480 MS.)
#
EXIT: TCF RCSMONEX (ALL PATHS EXIT VIA SUCH AN INSTRUCTION)

RCSMONEX EQUALS RESUME

ERASABLE INITIALIZATION REQUIRED:
#
VIA FRESH START: PVALVEST = +0 (ALL JETS ENABLED)
CH5MASK,CH6MASK = +0 (ALL JETS OK)
#
OUTPUT: CH5MASK & CH6MASK UPDATED (1'S WHERE JETS NOT TO BE USED, IN CHANNEL 5 & 6 FORM
PVALTEST UPDATED (1'S WHEN VALVE CLOSURES HAVE BEEN TRANSLATED INTO CH5MASK & C
JOB TO DO 1/ACCS.
#
DEBRIS: A, L, AND Q AND DEBRIS OF NOVAC.
#
SUBROUTINE CALLED: NOVAC.

 EBANK= CH5MASK

 BANK 23
 SETLOC RCSMONT
 BANK

Page 191

 COUNT* $$/T4RCS

RCSMONIT EQUALS RCSMON

RCSMON CS ZERO
 EXTEND
 RXOR CHAN32 # PICK UP + INVERT INVERTED CHANNEL 32.
 MASK LOW8 # KEEP JET-FAIL BITS ONLY.
 TS Q
```

```

CS PVALVEST
MASK Q
TS L
CS Q
MASK PVALVEST
ADS L

EXTEND
BZF RCSMONEX

EXTEND
MP BIT7
XCH L

-3 INCR L
 DOUBLE
 OVSK
 TCF -3

 INDEX L
 CA BIT8 -1
 TS Q
 MASK PVALVEST
 CCS A
 TCF VOPENED

 CS CH5MASK
 INDEX L
 MASK 5FAILTAB
 ADS CH5MASK

 CS CH6MASK
 INDEX L
 MASK 6FAILTAB
 ADS CH6MASK

 CA Q
 ADS PVALVEST

 TCF 1/ACCFIX

Page 192
VOPENED INDEX L
 CS 5FAILTAB
 MASK CH5MASK
 TS CH5MASK

```

# - -  
# FORM PC + PC.  
# (P = PREVIOUS ISOLATION VALVE STATE (CH3)  
# C = CURRENT VALVE STATE (CH3)  
# RESULT NZ INDICATES ACTION REQUIRED.  
# QUIT IF NO ACTION REQUIRED.  
# MOVE BITS 8-1 OF A TO 14-7 OF L.  
# ZERO TO L IN THE PROCESS.  
# BOUND TO GET OVERFLOW IN THIS LOOP.  
# SINCE WE ASSURED INITIAL NZ IN A.  
# SAVE THE RELEVANT BIT (8-1).  
# LOOK AT PREVIOUS VALVE STATE BIT.  
# THE VALVE HAS JUST BEEN OPENED.  
# THE VALVE HAS JUST BEEN CLOSED.  
# SET INHIBIT BIT FOR CHANNEL 5 JET.  
# SET INHIBIT BIT FOR CHANNEL 6 JET.  
# RECORD ACTION TAKEN.  
# SET UP 1/ACCJOB AND EXIT.  
# A VALVE HAS JUST BEEN OPENED.  
# REMOVE INHIBIT BIT FOR CHANNEL 5 JET.

July 12, 2016

1619

```

 INDEX L
 CS 6FAILTAB
 MASK CH6MASK
 TS CH6MASK
 # REMOVE INHIBIT BIT FOR CHANNEL 6 JET.

 CS Q
 MASK PVALVEST
 TS PVALVEST
 # RECORD ACTION TAKEN.

1/ACCFIX CAF PRI027 # SET UP 1/ACCS SO THAT THE SWITCH CURVES
 TC NOVAC # FOR TJETLAW CAN BE MODIFIED IF CH5MASK
 EBANK= AOSQ # HAS BEEN ALTERED.
 2CADR 1/ACCJOB

 TCF RCSMONEX # EXIT.

5FAILTAB EQUALS -1 # CH 5 JET BIT CORRESPONDING TO CH 32 BIT:
 OCT 00040 # 8
 OCT 00020 # 7
 OCT 00100 # 6
 OCT 00200 # 5
 OCT 00010 # 4
 OCT 00001 # 3
 OCT 00004 # 2
 OCT 00002 # 1

6FAILTAB EQUALS -1 # CH 6 JET BIT CORRESPONDING TO CH 32 BIT:
 OCT 00010 # 8
 OCT 00020 # 7
 OCT 00004 # 6
 OCT 00200 # 5
 OCT 00001 # 4
 OCT 00002 # 3
 OCT 00040 # 2
 OCT 00100 # 1
```

This code is written to file `src/RCS-FAILURE-MONITOR.s`.

## B.94 REENTRY CONTROL

```

1620 <src/REENTRY-CONTROL.s 1620>≡
 # Copyright: Public domain.
 # Filename: REENTRY_CONTROL.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 844-882
 # Mod history: 2009-05-08 RSB Adapted from the Colossus249/ file of the
 # same name, using Comanche055 page images.
 # 2009-05-23 RSB In a couple of 2OCT statements, removed the
 # space between the first and second octal words.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A
 #
 # Page 844
 # ENTRY INITIALIZATION ROUTINE
 # -----
 #
 # BANK 25
 # SETLOC REENTRY
 # BANK
 #
 # COUNT* $$/ENTRY
 # EBANK= RTINIT
 #
 # EBENTRY = EBANK7

```

July 12, 2016

1621

```
EBAOG EQUALS EBANK6
NTRYPRIO EQUALS PRIO20 # (SERVICER)
CM/FLAGS EQUALS STATE +6

STARTENT EXIT # MM = 63

 # COME HERE FROM CM/POSE. RESTARTED IN CM/POSE.
 # INITIALIZE ALL SWITCHES TO ZERO
 # EXCEPT LATSW, ENTRYDSP, AND GONEPAST.
 # GONEBY 112D BIT8 FLAG7, SELF-INITIALIZING.

 # ENTRYDSP = 92D B13
 # GONEPAST=95D B10 RELVELSW=96D B9
 # EGSW = 97D B8
 # HIND=99D B6 INRLSW=100D B5
 # LATSW=101D B4 .05GSW=102D B3

 # L/DCMINR = LAD COS(15)

 # KLAT = LAD/24

 # Q7 = Q7F
 # 1.0 -1BIT

 # 1.0 -1BIT
```

AD ENTRYSW  
TS CM/FLAGS

RELINT

TC INTERPRET

SLOAD

LODPAD

STORE LOD

SLOAD

LADPAD

STORE LAD

DMP

STODL L/DCMINR

LATSLOPE

DMP

SR1

LAD

STODL KLAT

Q7F

STODL Q7

NEARONE

STODL FACTOR

LAD

```

SIGN DCOMP
HEADSUP
STCALL L/D # MAY BE NOISE FOR DISPLAY P61
 # L/D = - LAD SGN(HEADSUP)

 STARTEN1 # RETURN VIA GOTOADDR
VLOAD VXV
 VN # (-7) M/CS
 UNITR # .5 UNIT REF COORDS
UNIT DOT
 RT # RT/2 TARGET VECTOR REF COORDS
STORE LATANG # LATANG = UNI.RT /4
DCOMP RTB
 SIGNMPAC
STODL K2ROLL # K2ROLL = -SGN(LATANG)

 LAD
DMP DAD
 Q21
 Q22
STORE Q2 # Q2 = -1152 + 500 LAD

SSP SSP
 GOTOADDR # SET SELECTOR FOR INITIAL PASS
 INITROLL
 POSEXIT
 SCALEPOP # SET CM/POSE TO CONTINUE AT SCALEPOP

RTB
 SERVNOOUT # OMIT INITIAL DISPLAY, SINCE 1ST GUESSBAD

CALCULATE THE INITIAL TARGET VECTOR: RTINIT, ALSO RTEAST, RTNORM, AND RT. ALL ARE
REFERENCE COORDINATES.

STARTEN1 STQ VLOAD
 GOTOADDR
 LAT(SPL) # TARGET COORDINATES
 CLEAR # DO CALL USING PAD RADIUS. WILL UNIT IT.
 ERADFLAG # ANYWAY.
 LUNAFLAG
STODL LAT
 3ZEROS

STODL LAT +4 # SET ALT=0.
 PIPTIME # ESTABLISH RTINIT AT TIME OF PRESENT

RN AND VN.

```

July 12, 2016

1623

```

 STCALL TIME/RTO # SAVE TIME BASE OF RTINIT.
 LALOTORV # C(MPAC) =TIME (PIPTIME)
 UNIT # ANSWER IN ALPHAV ALSO
 STODL RTINIT # .5 UNIT TARGET REF COORDS
 500SEC # NOMINAL ENTRY TIME FOR P63
 # TIME/RTO = PIPTIME, STILL.
 STCALL DTEAROT # INITIALIZE EARROT
 EARROT1 # GET R5
 DOT SL1
 UNITR # RT/2 IN MPAC
 ACOS
 STCALL THETAH # RANGE ANGLE /360
 GOTOADDR # RETURN TO CALLER

500SEC 2DEC 50000 B-28 # CS.

ENTMASK OCT 11774
ENTRYSW OCT 11010 # ENTRYDSP B13. GONEPAST B10. LATSW B4
Page 847

SCALEPOP CALL
 TARGETNG

 EXIT

REFAZE10 TC PHASCHNG
 OCT 10035 # SERVICER 5.3 RESTART AT REFAZE10
 TC INTPRET

JUMP TO PARTICULAR RE-ENTRY PHASE:
SEQUENCE
 GOTO
 GOTOADDR

GOTOADDR CONTAINS THE ADDRESS OF THE ROLL COMMAND EQUATIONS TO THE CURRENT PHASE OF
RE-ENTRY. SEQUENCING IS AS FOLLOWS:
#
INITROLL ADDRESS IS SET HERE INITIALLY. HOLDS INITIAL ROLL ATTITUDE UNTIL KAT IS EXCE
ATTITUDE UNTIL VRTHRESH IS EXCEEDED. THEN BRANCHES TO
#
HUNTEST THIS SECTION CHECKS TO SEE IF THE PREDICTED RANGE AT NOMINAL L/D FROM PRESENT
THAN THE DESIRED RANGE.
IF NOT --- A ROLL COMMAND IS GENERATED BY THE CONSTANT DRAG CONTROLLER.
IF SO --- CONTROL AND GOTOADDR ARE SET TO UPCONTRL.
USUALLY NO ITERATION IS INVOLVED EXCEPT IF THE RANGE DESIRED IS TOO LONG ON THE
```

```

HUNTEST.
#
UPCONTRL CONTROLS ROLL DURING THE SUPER-CIRCULAR PHASE. UPCONTRL IS TERMINATED
(A) WHEN THE DRAG (AS MEASURED BY THE PIPAS) FALLS BELOW Q7,
(B) IF RDOT IS NEGATIVE AND REFERENCE VL EXCEEDS V.
IN CASE (A), GOTOADDR IS SET TO KEP2 AND IN CASE (B), TO PREDICT3.
ENTRY.
#
KEP2 GOTOADDR IS SET HERE DURING THE KEPLER PHASE TO MONITOR DRAG. THE SPIN
TRIMMED IN PITCH AND YAW TO THE COMPUTED RELATIVE VELOCITY. THE LAST
WHEN THE MEASURED DRAG EXCEEDS Q7 +0.5, GOTOADDR IS SET TO
#
PREDICT3 THIS CONTROLS THE FINAL SUB-ORBITAL PHASE. ROLL COMMANDS CEASE
WHEN V IS LESS THAN VQUIT . AN EXIT IS MADE TO
#
P67.1 THE LAST COMPUTED ROLL ANGLE IS MAINTAINED. RATE DAMPING IS DONE IN
AND LONGITUDE ARE COMPUTED FOR DISPLAY.
ENTRY IS TERMINATED WHEN DISKY RESPONSE IS MADE TO THIS FINAL FLASHING

```

# Page 848

# PROCESS AVERAGE G OUTPUT...SCALE IT AND GET INPUT DATA

# \* START TARGETING ...

EBANK= RTINIT

# TARGETNG IS CALLED BY P61, FROM GROUP 4.  
 # TARGETNG IS CALLED BY ENTRY, FROM GROUP 5.

|          |       |                                      |                                                                                                                                                               |
|----------|-------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TARGETNG | BOFF  | VLOAD<br>RELVELSW<br>GETVEL<br>-VREL | # ALL MM COME HERE.<br># ENTER WITH PROPER EB FROM CM/POSE(TEST)<br># RELVELSW = 96D BIT9<br># WANT INERTIAL VEL. GO GET IT.<br># NEW V IS RELATIVE, CONTINUE |
|          | VCOMP | GOTO<br>GETUNITV -1                  | # (VREL) = (V) + KWE UNITR*UNITW<br># - VREL WAS LEFT BY CM/POSE                                                                                              |
| GETVEL   | VLOAD | VXSC<br>VN<br>KVSCALE                | # INERTIAL V WANTED<br># KVSCALE = (12800 / .3048) / 2VS<br># KVSCALE = .81491944                                                                             |
|          | STORE | VEL                                  | # V/2 VS                                                                                                                                                      |
| GETUNITV | UNIT  | STQ<br>60GENRET                      |                                                                                                                                                               |
|          | STODL | UNITV                                |                                                                                                                                                               |



July 12, 2016

1625

```

 34D
STORE VSQUARE # VSQ/4

 36D
DSU V # LEQ = VSQUARE - 1
 FOURTH
STODL LEQ # 4 G'S FULL SCALE
 LEQ
 # LEQ/4

 36D
STOVL V # V/2 VS = VEL/2 VS

 VEL
DOT SL1 # RDOT= V.UNITR
 UNITR
STOVL RDOT # RDOT /2 VS

 DELV
ABVAL DMP # PIPA COUNTS IN PLATFORM COORDS.
 KASCALE
SL1 BZE
 SETMIND
DSTORE STOVL D # ACCELERATION USED TO APPROX DRAG
 VEL
 VXV UNIT # UNI = UNIT(V*R)

Page 849
 UNITR
STORE UNI # .5 UNI REF COORDS.

 DLOAD
BOFF RELVELSW
 GETETA
 3ZEROS
UPDATERT DSU DAD # PIPTIME-TIME/RT0 =ELAPSED TIME SINCE
 # RTINIT WAS ESTABLISHED.
 TIME/RT0
 PIPTIME
STCALL DTEAROT # GET PREDICTED TARGET VECTOR RT

 EARROT2
DOT SETPD # SINCE (RT) UNIT VECT, THIS IS 1/4 MAX
 UNI
 0
 # LATANG = RT.UNI
STOVL LATANG # LATANG = MAC LATANG / 4

 RT
CLEAR GONEBY # SHOW HAVE NOT GONE PAST TARGET.
```

```

 VXV DOT # IF RT*UNITR.UNI NEG, GONEBY=1
 UNITR # GONEPAST IS CONDITIONAL SW SET IN
 UNI # FINAL PHASE.
 BPL SET
 +2
 GONEBY # SHOW HAVE GONE PAST TARGET.

 VLOAD
 RT
GETANGLE DOT DSU # THETA = ARCCOS(RT.UNITR)
 UNITR
 NEAR1/4 # TO IMPROVE ACCURACY, CALC RANGE BY
 BPL DAD # TINYTHET IF HIGH ORDER PART OF
 TINYTHET # ARCCOS ARGUMENT IS ZERO
 NEAR1/4
 SL1 ACOS
THETDONE STORE THETAH # THETAH/360
 # HI WORD, LO BIT =1.32 NM=360 60/16384

 BON DCOMP
 GONEBY # =1 IF HAVE GONE PAST TARGET.
 # (SIGN MAY BECOME ERRATIC VERY NEAR
 # TARGET DUE TO LOSS OF PRECISION.)
 +1
 STODL RTGON67 # RANGE ERROR: NEG IF WILL FALL SHORT.

 D
 DSU BMN

 .05G
 NO.05G
 SET VLOAD
 .05GSW
 DELVREF
 PUSH DOT
 UXA/2
 SL1 DSQ
 PDVL VSQ # EXCHANGE WITH PDL.
 DSU DDV
 0
 BOV SQRT
 NOLDCALC # OVFL LAST CLEARED IN EARROT2 ABOVE.
 STORE L/DCALC

NOLDCALC GOTO
 60GENRET

```

July 12, 2016

1627

```
NO.05G CLEAR GOTO # THIS WAY FOR DAP. (MAY INTERRUPT)
 .05GSW # .05GSW = 102D B3
 NOLDCALC # KEEP SINGLE EXIT FOR TARGETING
```

# Page 851

# SUBROUTINES CALLED BY SCALEPOP (TARGETING):

```
 BANK 26
 SETLOC REENTRY1
 BANK
 COUNT* $$/ENTRY

GETETA DLOAD DDV # D = D +D(-RDOT/HS -2D/V) DT/2
 # DT/2 = 2/2 =1
 RDOT
 -HSCALED
PDDL DMP
 D
 -KSCALE
DDV DAD
 V
 # -RDOT/HS FROM PDL.
DMP DAD
 D
 D
STORE D

BON DLOAD # EGSW INDICATES FINAL PHASE
 EGSW
 SUBETA
 THETAH
DMP GOTO
 KTETA # = 1000x2PI/(2)E14 163.84
 UPDATERT

SUBETA DLOAD DSU # SWITCH FROM INERTIAL TO RELATIVE VEL.
 V
 VMIN
BPL SET
 SUBETA2
 RELVELSW

SUBETA2 DLOAD DMP
```

|            |        |                  |                                           |
|------------|--------|------------------|-------------------------------------------|
|            |        | THETAH           |                                           |
|            |        | KT1              | # KT1 = KT                                |
|            | DDV    | GOTO             |                                           |
|            |        | V                | # KT = RE(2 PI)/2 VS 16384 163.84/ 2 VSAT |
|            |        | UPDATERT         |                                           |
| SETMIND    | DLOAD  | GOTO             |                                           |
|            |        | 1BITDP           |                                           |
|            |        | DSTORE           |                                           |
| # Page 852 |        |                  |                                           |
| TINYTHET   | DSU    | ABS              | # ENTER WITH X-.249                       |
|            |        | 1BITDP +1        | # GET 1/4 - MPAC                          |
|            | SL     | SQRT             | # SCALE UP BEFORE SQRT                    |
|            |        | 13D              | # HAS FACTOR FOR UP SCALING               |
|            | DMP    | GOTO             |                                           |
|            |        | KACOS            |                                           |
|            |        | THETDONE         |                                           |
| # Page 853 |        |                  |                                           |
| # * START  |        | INITIAL ROLL ... |                                           |
|            | BANK   | 25               |                                           |
|            | SETLOC | REENTRY          |                                           |
|            | BANK   |                  |                                           |
|            | COUNT* | \$\$/ENTRY       |                                           |
| INITROLL   | BON    | BOFF             | # MM = 63, 64, ...                        |
|            |        | INRLSW           | # IF D- .05G NEG, GO TO LIMITL/D          |
|            |        | INITRL1          |                                           |
|            |        | .05GSW           |                                           |
|            |        | LIMITL/D         |                                           |
|            |        |                  | # MM = 64, NOW                            |
|            |        |                  | # 3                                       |
|            |        |                  | # KA = KA1 LEG + KA2                      |
|            | DLOAD  | DSQ              |                                           |
|            |        | LEQ              |                                           |
|            | DMP    | DDV              |                                           |
|            |        | LEQ              |                                           |
|            |        | 1/KA1            | # = 25 /(64 1.8)                          |
|            | DAD    | RTB              |                                           |
|            |        | KA2              | # = .2                                    |
|            |        | P64              | # ROLLC VI RDOT                           |

July 12, 2016

1629

```

XXX.XX DEG XXXXX. FPS XXXXX. FPS

STORE KAT

DSU BMN
 KALIM
 +4

DLOAD

 KALIM
STORE KAT
DLOAD DSU # IF V-VFINAL1 NEG, GO TO FINAL PHASE.
 V
 VFINAL1
CLEAR BPL # (CAN'T CLEAR INRLSW AFTER HERE: RESTARTS)
 GONEPAST # GONEPAST WAS INITIALLY SET=1 TO FORCE
 # ROLLC TO REMAIN AS DEFINED BY HEADSUP
 # UNTIL START OF P64. (UNTIL D > .05G)

 DOEQ
SSP GOTO
 GOTOADDR
 KEP2 # AND IDLE UNTIL D > 0.2 G. (NO P66 HERE)
 INROLOUT # GO TO LIMITL/D AFTER SETTING INRLSW.

DOEQ DLOAD DMP # D0 = KA3 LEQ + KA4
Page 854 LEQ
 KA3
DAD KA4
STORE D0 # D0/805
BDDV BOV
 C001 # (-4/25 G) B-8
 +1 # CLEAR OVFLND, IF ON.
STODL C/D0 # (-4/D0) B-8
 LAD # IF V-VFINAL +K(RDOT/V)CUBED POS,L/D=-LAD
STODL L/D
 RDOT
DDV PUSH
 V
DSQ DMP
DDV DSU
 1/K44
 VFINAL
#
V-VFINAL +(RDOT/V) / K44 OVFL $
#
DAD BOV
```

```

V
INROLOUT # GO TO LIMITL/D AFTER SETTING INRLSW.
BMN DLOAD
INROLOUT # GO TO LIMITL/D AFTER SETTING INRLSW.
LAD
DCOMP
STORE L/D

INROLOUT BOFSET # SET INRLSW AT END FOR RESTART PROTECTION
 # END OF PRE .05G PATH OF INITROLL.
 INRLSW # SWITCH IS ZERO INITIALLY.
 LIMITL/D # (GO TO)

KATEST DLOAD DSU # IF KAT - D POS, GO TO CONSTD
 KAT
 D # IF POS, OUT WITH COMMAND VIA LIMITL/D
BPL GOTO
 LIMITL/D
 CONSTD

INITRL1 DLOAD DAD # IF RDOT + VRCONT POS, GO TO HUNTEST
 RDOT
 VRCONT
BMN CALL # IF POSITIVE, FALL INTO HUNTEST.
 KATEST

 FOREHUNT # INITIALIZE HUNTEST.

Page 855
* START HUNT TEST ...

 # MM = 64
SSP # INITIALIZE HUNTEST ON FIRST PASS
 GOTOADDR
 HUNTEST # MUST GO AFTER FOREHUNT FOR RESTARTS.

HUNTEST DLOAD
 D
STODL A1 # A1/805 = A1/25G

 LAD
STODL TEM1B
 RDOT
BMN DLOAD # IF RDOT NEG, TEM1B=LAD, OTHERWISE = LEWD
 AOCALC
 LEWD
STODL TEM1B

```

July 12, 2016

1631

```
AOCALC DDV RDOT
 DAD # V1 = V + RDOT/TEM1B
 TEM1B
 V
STODL V1 # V1/2 VS

 RDOT
DSQ DDV # A0=(V1/V)SQ(D+RDOT SQ/(TEM1B 2 C1 HS)
 TEM1B
DDV DAD
 2C1HS
 D
DMP DMP
 V1
 V1
DDV
 VSQUARE
STODL A0 # A0/805 = A0/25G

 RDOT
BPL DLOAD
 V1LEAD
 A0
STORE A1 # A1/25G

V1LEAD DLOAD BPL # IF L/D NEG, V1=V1 - 1000
 L/D
 HUNTEST1

 DLOAD DSU
 V1

Page 856

 VQUIT
STORE V1

HUNTEST1 DLOAD DMP # ALP = 2 C1 HS A0/LEWD V1 V1
 A0
 2C1HS
DDV SETPD
 V1
 0
DDV DDV
 V1
 LEWD
STORE ALP
```

```

BDSU BDDV # FACT1 = V1 / (1 - ALP)
 BARELY1
 V1
STODL FACT1 # FACT1 / 2VS

 ALP
DSU DMP # FACT2 = ALP(ALP - 1) / A0
 BARELY1
 ALP
DDV
 A0
STORE FACT2 # FACT2 (25G)

DMP DAD
 Q7 # Q7 / 805 = Q7 / 25G
 ALP # VL=FACT1 (1-SQRT(Q7 FACT2 +ALP))
SQRT BDSU
 BARELY1
DMP
 FACT1
STORE VL # VL / 2 VS

BDSU DMP # GAMMAL1 = LEWD (V1-VL)/VL
 V1
 LEWD
DDV
 VL
STODL GAMMAL1 # GAMMAL1 USED IN UPCONTROL.

 # GAMMAL1 = PDL 22D.
 VL
DSU BMN # IF VL-VLMIN NEG, GO TO PREFINAL
 VLMIN
 PREFINAL

DLOAD DSQ

 VL
STODL VBARS # VBARS / 4 VS VS

 HALVE
DSU BMN # IF VSAT-VL NEG, GO TO CONSTD
 VL
 BECONSTD
STODL DVL # SET MODE=HUNTEST, CONTINUE IN CONSTD
 # DVL / 2VS

```



July 12, 2016

1633

```

 HALVE
STORE VS1 # VS1 = VSAT

DSU BMN # IF V1 GREATER THAN VSAT, GO ON
 V1
 GETDHOOK

BDSU DVL

STODL DVL # DVL = DVL - (VSAT-V1) = V1 - VL
 V1

STORE VS1 # VS1 = V1, IN THIS CASE

GETDHOOK DLOAD CALL # DHOOK=((1-VS1/FACT1) SQ -ALP)/FACT2
 VS1 # VS1 / 2 VS
 DHOOKYQ7 # GO CALC DHOOK
STORE DHOOK # DHOOK / 25G

SR DDV
 6 # CHOOK
 Q7

DSU CHOOK # = .25/16 = (-6)
STORE AHOOKDV

DAD DMP # GAMMAL= GAMMAL1-CH1 DVL SQ(1+AHOOK DVL)
 1/16TH
 CH1

DMP DMP
 DVL
 DVL

DDV DDV
 DHOOK
 VBARS

BDSU BMN
 GAMMAL1
 NEGAMA
HUNTEST3 STORE GAMMAL

DSU GAMMAL1 # GAMMAL1=GAMMAL1 +Q19 (GAMMAL-GAMMAL1)

DMP DAD

 Q19
 GAMMAL1
STODL GAMMAL1
```

# Page 858

## GAMMAL

# Page 859

# \*START

RANGE PREDICTION ...

# C(MPAC) = GAMMAL.

# COSG = 1-GAMMAL SQ/2, TRUNCATED SERIES

RANGER

DSQ

SR2

BDSU

HALVE

STODL

COSG/2

VBARS

# E=SQRT(1+VBARS.....

DSU

DMP

HALVE

VBARS

DMP

DMP

COSG/2

COSG/2

SL2

DAD

C1/16

# C1/16 = 1/16

SQRT

PDDL

# E/4 INTO PDL

VBARS

DMP

DMP

# ASKEP/2 = ARCSIN(VBARS COSG SING/E)

COSG/2

GAMMAL

DDV

ASIN

SL1

PUSH

# ASKEP TO PDL 0.

STODL

ASKEP

# BALLISTIC RANGE ASKEP/2PI

# FOR TM, STORE RANGE COMPONENTS OVERLAPPING

VL

DMP

DAD

# ASP1 = Q2 + Q3 VL

Q3

Q2

STORE

ASP1

# FINAL PHASE RANGE ASP1/2 PI

PDDL

DSQ

# ASP1 TO PDL 2.

V1

#

2

# ASPUP= -C12 LOG(V1 Q7/VBARS A0)/GAMMAL1

DMP

DDV

Q7

VBARS

DDV

CALL

A0

LOG

# RETURN WITH -LOG IN MPAC

July 12, 2016

1635

# Page 860

```
DMP DDV
 C12
 GAMMAL1
STORE ASPUP # UP PHASE RANGE ASPUP / 2 PI

PDDL DMP # ASPUP TO PDL 4.
 KC3 # KC3 = -4 VS VS / 2 PI 805 RE
 # ASPDWN = KC3 RDOT V / A0
 RDOT
DMP DDV
 V
 A0
DDV PUSH # ASPDWN TO PDL 6.
 LAD
STODL ASPDWN # RANGE TO PULL OUT ASPDWN /2 PI

 Q6
DSU DMP # ASP3 = Q5(Q6-GAMMAL)
 GAMMAL
 Q5
STOVL ASP3 # GAMMA CORRECTION ASP3/2PI

 ASKEP # GET HI-WD AND
STODL ASP3(TM) # SAVE HI-WORD OF ASP'S FOR TM.

 ASP3
DAD DAD

 # ASPDWN FROM PDL 6.
 # ASPUP FROM PDL 4.

DAD DAD

 # ASP1 FROM PDL 2.
 # ASKEP FROM PDL 0.
 # CLEAR OVFFIND.
DSU BOVB
 THETAH
 TCDANZIG
STORE DIFF # DIFF = (ASP-THETAH) / 2 PI
 # ASP=ASKEP+ASP1+ASPUP+ASP3+ASPDWN = TOTAL RANGE

ABS DSU # IF ABS(THETAH-ASP) -25NM NEG, GO TO UPSY
 25NM
BMN BON
 GOTOUPSY
 HIND
 GETLEWD
```

July 12, 2016

```

 DLOAD BPL
 DIFF
 DCONSTD # EVENTUALLY SETS MODE = HUNTEST.
GETLEWD DLOAD DMP
 # DLEWD = DLEWD (DIFF/(DIFFOLD-DIFF))
 DLEWD
 DIFF
 PDDL DSU
 DIFFOLD
 DIFF
Page 861
 BDDV
LWDSTORE STADR
 STORE DLEWD
 DAD BMN # IF LEWD+DLEWD NEG, DLEWD=-LEWD/2
 LEWD
 LEWDPTR
 BOV
 LEWDOVFL
 STORE LEWD

SIDETRAK EXIT

 CA EBENTRY
 TS EBANK

 CA PRI016 # DROP GRP 5 RESTART PRIO TO 1 LESS THAN
 TS PHSPRDT5 # GRP 4.

 TC PHASCHNG
 OCT 00474 # RESTART GRP 4 AT PRE-HUN.
 # FORCE RESTART TO PICK UP IN GRP 4:
 # USE PRIO 17 FOR GRP 4 (< SERVICER PRIO)
 CA PRI016 # CONTINUE GRP 5 AT LOWER PRIO THAN EITHER
 # GRP 4 OR SERVICER.

 TC PRIOCHNG

 CAF ADENDEXT # SIDETRACK NEXT PASS UNTIL THIS ONE DONE.
 TS GOTOADDR # ONLY AFTER RESTART IS LEFT AFTER DETOUR.

 TC INTPRET

 DLOAD SET
 DIFF
 HIND
 STODL DIFFOLD # DIFFOLD / 2 PI

```

July 12, 2016

1637

```

 Q7F
STCALL Q7 # Q7 / 805 FPSS
 HUNTEST # (GO TO)

LEWDOVFL DLOAD
 NEARONE
STCALL LEWD
 DCONSTD # (GO TO) ALSO WILL SET MODE = HUNTEST

LEWDPTR DLOAD SR1
 LEWD
DCOMP GOTO
 LWDSTORE

Page 862
NEGAMA DMP DMP # ENTER WITH GAMMAL IN MPAC

 VL
 1/3RD
PDDL DMP # PUSH GAMMAL VL/3
 LEWD
 1/3RD
PDDL DAD # PUSH LEWD/3
 AHOOKDV
 1/24TH
DMP DMP # DEL VL = (GAMMAL VL/3)/(LEWD/3-DVL
 DVL # (2/3 + AHOOKDV)(CH1 GS/DHOOK VL))
 CH1
DDV DDV
 DHOOK
 VL
BDSU BDDV
 # LEWD/3
 # GAMMAL VL /3

DAD
 VL
STCALL VL # VL/2 VS

 DHOOKYQ7 # GO CALC Q7
 # Q7=((1-VL/FACT1)SQ - ALP)/FACT2
STODL Q7 # Q7 / 25G

 VL
DSQ
STODL VBARS # VBARS / 4 VS VS
```

[illegible]

July 12, 2016

1639

|            |       |          |                                          |
|------------|-------|----------|------------------------------------------|
| UPCONTRL   | DLOAD | DSU      | # IF D-140 POS, NOSWITCH =1              |
|            |       | D        | # (SUPPRESS LATERAL SWITCH)              |
|            |       | C21      |                                          |
|            | BMN   | SET      |                                          |
|            |       | +2       |                                          |
|            |       | NOSWITCH |                                          |
|            | DLOAD | DSU      | # IF V-V1 POS, GO TO DOWN CONTROL.       |
|            |       | V        |                                          |
|            |       | V1       |                                          |
|            | BPL   | DLOAD    |                                          |
|            |       | DOWNCNTL |                                          |
|            |       | D        |                                          |
|            | DSU   | BMN      | # IF D- Q7 NEG, GO TO KEP                |
|            |       | Q7       |                                          |
|            |       | KEP      |                                          |
|            | DLOAD | BPL      | # IF RDOT NEG, DO VLTEST                 |
|            |       | RDOT     |                                          |
|            |       | CONT1    |                                          |
| VLTEST     | DLOAD | DSU      | # IF V-VL-C18 NEG,EGSW=1,MODE=PREDICT3   |
|            |       | V        |                                          |
|            |       | VL       |                                          |
|            | DSU   | BMN      |                                          |
|            |       | C18      |                                          |
|            |       | PREFINAL |                                          |
| CONT1      | DLOAD |          | # IF D-A0 POS, L/D = LAD, GO TO LIMITL/D |
|            |       | D        |                                          |
|            | DSU   | BMN      |                                          |
|            |       | A0       |                                          |
|            |       | CONT3    |                                          |
|            | DLOAD | GOTO     |                                          |
|            |       | LAD      |                                          |
|            |       | STOREL/D |                                          |
| CONT3      | DLOAD | DMP      | # VREF=FACT1(1-SQRT(FACT2 D + ALP))      |
|            |       | D        |                                          |
|            |       | FACT2    |                                          |
| # Page 865 | DAD   | SQRT     |                                          |
|            |       | ALP      |                                          |
|            | BDSU  | DMP      |                                          |
|            |       | BARELY1  |                                          |
|            |       | FACT1    |                                          |

```

STORE VREF # VREF / 2VS

BDSU DMP # RDOTREF = LEWD(V1-VREF)
 V1
 LEWD
STODL RDOTREF # RDOTREF / 2VS

 VS1
DSU BMN # IF VSAT-VREF NEG, GO TO CONTINU2
 VREF
 CONTINU2

PUSH PUSH # VS1-VREF TO PDL TWICE
DMP DDV # RDHOOK=CHI1(1+DV AHOOKDV/DVL) DV DV
 AHOOKDV # /DHOOK VREF
 DVL # WHERE DV = (VS1-VREF)
DAD DMP
 1/16TH
 CH1
DMP DMP
 # VS1-VREF FROM PDL TWICE.
DDV
 DHOOK
DDV BDSU
 VREF
 RDOTREF # C(RDOTREF)= LEWD (V1-VREF)
STORE RDOTREF # RDOTREF = RDOTREF - RDHOOK

CONTINU2 DLOAD DSU
 D
 Q7MIN
BOVB BMN
 TCDANZIG # CLEAR OVFL IND, IF ON.
 UPCNTRL3
DLOAD DSU
 A1
 Q7
PDDL DSU
 D
 Q7
DDV STADR
STORE FACTOR # FACTOR / 25G

Page 866
SKIPPER
 # DELTA L/D=--((RDOT-RDOTREF)F1 KB1+V-VREF)F1

```



July 12, 2016

1641

```
WHERE F1 = FACTOR

UPCNTRL3 DLOAD RDOT
 DSU DMP # L/D = LEWD
 RDOTREF # -((RDOT-RDOTREF)F1/KB1+V-VREF)F1/KB2
 FACTOR
 DDV DAD
 1/KB1
 V
 DSU DMP
 VREF
 FACTOR
 DDV PUSH

 -1/KB2 # DELTA L/D INTO PDL
 BOV ABS # NONLINEAR CIRCUIT FOR REDUCING HIGH GAIN
 GOMAXL/D
 DSU BMN
 PT1/16
 NEXT1
 DMP DAD
 POINT1
 PT1/16
 SIGN PUSH # ATTACH SIGN OF PUSH TO MPAC THEN PUSH

NEXT1 DLOAD SL4
 DAD
 LEWD
NEGTESTS BOV PUSH # L/D TO PDL FOR USE IN NEGTESTS.
 GOMAXL/D
 STODL L/D
 D
 DSU BMN
 C20
 LIMITL/D
 CLEAR DLOAD
 LATSW # =21D. ROLL OVER TOP, REGARDLESS.
 BPL DLOAD # L/D FROM PDL.
 LIMITL/D
 3ZEROS
```

```

 STCALL L/D
 LIMITL/D # (GO TO)

Page 867
DCONSTD DLOAD # TWO RANGER ENTRIES TO CONSTD HERE
 DIFF
 # SAVE OLD VALUE OF DIFF FOR NEXT PASS.
 STODL DIFFOLD # DIFFOLD / 2 PI
 Q7F
 STORE Q7

BECONSTD SSP RTB # A HUNTEST ENTRY INTO CONSTD.
 GOTOADDR # RESET MODE TO HUNTEST
 HUNTEST
 KILLGRP4 # DEACTIVATE GRP4 FROM HUNTEST.

CONSTD BOVB
 TCDANZIG # CLEAR OVF IND IF ON.

 DLOAD DMP
 LEQ
 C/DO # C/DO = -4/DO B-8
 PDDL DMP # LEQ C/DO INTO PDL
 2HS # 2HS / 4 VS VS
 DO
 DDV DAD # RDOTREF = -2 HS DO/V
 V
 RDOT
 DMP DAD
 K2D # C/DO LEQ + K2D(RDOT-RDOTREF) INTO PD
 PDDL
 DO # DO /805

CONSTD1 BDSU # ENTER WITH DREF IN MPAC
 D
 DMP DAD
 K1D # K2D TERM FROM PUSH
 SL GOTO
 8D
 NEGTESTS # (GO TO)

DOWNCNTL BOVB
 TCDANZIG # INITIAL PART OF UPCONTROL.
 # CLEAR OVFind, IF ON.

 DLOAD SR

```

July 12, 2016

1643

# Page 868

```

 LAD
 8D
PDDL DSU # RDTR = LAD(V1-V)
 V
 V1
DMP DAD
 LAD

 RDOT
DMP DAD
 K2D

 # PUSH UP LAD.
PDDL DSU # LAD + K2D(RDOT-RDTR) INTO PD
 V1
 V
DSQ DMP
 LAD
DDV PDDL # (V1-V)SQ LAD/(2 C1 HS) INTO PD
 2C1HS
 V1
DSQ DDV
 VSQUARE
BDDV DSU # DREF = (V/V1)SQ A0 - PD
 A0

 # PUSH UP HERE
GOTO # C(MPAC) = DREF
 CONSTD1
```

# Page 869

# \* START

```

 #
 2 2
 # DREF = (V/V1) A0 -(V-V1) LAD/2 C1 HS

BALLISTIC PHASE ...

 # MM = 66 UPCONTRL ENTRY INTO KEP2
```

KEP

```

RTB SSP
 P66
 GOTOADDR
 KEP2

 # DISPLAY TRIM GIMBAL ANGLE VALUES.
 # SET GOTOADDR TO KEPLER PHASE.
```

```

 # KEP2 CAN ALSO BE STARTED UP DIRECTLY FROM INITROLL
 # IN P64. PROGRAM WILL IDLE IN P64 UNTIL D EXCEEDS
 # .2 G BEFORE GOING ON TO P67.
```

KEP2

```

DLOAD DSU # IF Q7F+KDMIN -D NEG, GO TO FINAL PHASE.
 Q7FKDMIN
 D
BMN TLOAD # (Q7F + KDMIN)/805
```

July 12, 2016

```

Page 870
START FINAL PHASE ...

PREFINAL SSP RTB
 GOTOADDR
 PREFINAL
 P67
 # MM = 67
 # RESTART PROTECT: RESET GOTOADDR IF CAME
 # FROM HUNTEST.
 # DISABLES GRP4. FINE IF FROM HUNTEST, BUT
 # MAY ALSO REMOVE RESTART PROTECTION OF
 # N69 (P65).
 # ROLLC XRUNGERR DNRNGERR
 # XXX.XX DEG XXXX.X NM XXXX.XX NM

 SET SSP
 EGSW
 GOTOADDR
 PREDICT3

PREDICT3 DLOAD DSU
 V
 VQUIT
 BMN EXIT
 STEEROFF

 CA EBENTRY
 TS EBANK
 # PRECAUTIONARY.

 CA TWELVE
BACK TS JJ

 CS V
 INDEX JJ
 AD VREFER
 CCS A
 CCS JJ
 TCF BACK
 # VREF - V, HIGHEST VREF AT END OF TABLE.
 # IF VREF-V POS LOOP BACK
 # DECREMENT JJ, JJ CANNOT BE ZERO

```

July 12, 2016

1645

```
AD ONE
TS TEM1B # V-VREF IN TEM1B (MUST BE POSITIVE NUM)

INDEX JJ
CS VREFER
INDEX JJ
AD VREFER +1 # V(K+1) - V(K) (POS NUM)
XCH TEM1B
ZL
EXTEND
DV TEM1B
TS GRAD # GRAD = (V-VREF)/(VK+1 - VK) (POS NUM)

CAF FIVE
```

# Page 871  
BACK2

```
TS MM
CAF THIRTEEN
ADS JJ
INDEX A
CS VREFER
INDEX JJ
AD VREFER +1 # X(K+1) - X(K)
EXTEND
MP GRAD
INDEX JJ
AD VREFER
INDEX MM
TS FX # FX = AK + GRAD (AK+1 - AK)
CCS MM
TCF BACK2
XCH FX +1 # ZERO FX +1 AND GET DREFR
AD D
EXTEND
MP FX +5 # F1
DXCH MPAC # MPAC = F1(D-DREF)

EXTEND
DCS RDOT # FORM RDOTREF - RDOT
DDOUBL
DDOUBL
DDOUBL # SCALE UP BY 8 FOR THIS PHASE.
AD FX +3 # RDOTREF
EXTEND
MP FX +4 # F2
AD FX +2 # RTOGO
```

```

DAS MPAC # ADD F2(DADV1-DADV2)
CA MPAC
TS PREDANG

TC INTERPRET # L/D = LOD + (THETA- PREDANG)/ Y

SR3 DSU
 THETAH
BON BOFF
 GONEPAST
 GONEGLAD
 GONEBY
 HAVDNRNG
DLOAD SET # SET GONEPAST IF GONEBY SET & LATCH IN-PLACE
 MAXRNG # DISPLAY = 9999.9 IF GONEBY
 GONEPAST
STCALL DNRNGERR
 GONEGLAD

HAVDNRNG STORE DNRNGERR # = (PREDANG - THETA) /360
Page 872

DCOMP BOVB DDV # FALLS SHORT IF NEG, OVERSHOOT IF POS
 TCDANZIG
 FX # CLEAR OVFLND IF ON.
 BOV # FX= DRANGE/D L/D = Y
 5
 GOMAXL/D
DAD BOV
 LOD
 GOMAXL/D
STCALL L/D
 GLIMITER # (GO TO)

GONEGLAD AND GOPOSMAX ENTRY POINTS FOR GLIMITER ...

GONEGLAD DLOAD GONEGLAD # SET L/D = -LAD
 GONEGLAD # (ANY NEGATIVE NUMBER WILL DO)

GOMAXL/D RTB DMP L/D = LAD SIGN(MPAC)
 SIGNMPAC
 LAD
 STORE L/D # AND FALL INTO GLIMITER SECTION

GLIMITER DLOAD DSU # IF GMAX/2-D POS, GO TO LIMITL/D
 GMAX/2

```

July 12, 2016

1647

```

 D
BPL DAD # IF GMAX -D NEG, GO TO GOPOSLAD
 LIMITL/D
 GMAX/2
BMN DMP
 GOPOSLAD
 2HS
PDDL DMP # 2HS(GMAX-D) INTO PD
 LEQ
 1/GMAX
DAD DMP
 LAD
PDDL DDV # 2HS(GMAX-D) (LEQ/GMAX+LAD) INTO PD
 2HSGMXSQ
 VSQUARE
DAD SQRT # XLIM = SQRT(PD+(2HSGMAX/V)SQ)
DAD BPL # IF RDOT+XLIM POS, GO TO LIMITL/D
 RDOT
 LIMITL/D

GOPOSLAD DLOAD
 LAD
STOREL/D STORE L/D

Page 873
LIMITL/D DLOAD
 L/D
 STODL L/D1
 VSQUARE

BON # NO LATERAL CONTROL IF PAST TARGET
 GONEPAST
 L355
DMP DAD # Y= KLAT VSQUARE + LATBIAS
 KLAT
 LATBIAS # Y INTO PD
L350 PDDL ABS # IF ABS(L/D)-L/DCMINR NEG, GO TO L353
 L/D
 DSU BMN
 L/DCMINR
 L353
 DLOAD SIGN # IF K2ROLL LATANG NEG, GO TO L357
 LATANG
 K2ROLL
 BMN DLOAD
 L357
```

```

L353 SR1 PUSH # Y = Y/2
 DLOAD SIGN # IF LATANG SIGN(K2ROLL)-Y POS, SWITCH
 LATANG
 K2ROLL

 DSU
 BMN DLOAD
 L355
 K2ROLL

 BONCLR DCOMP # IF NOSWITCH =1, K2ROLL= K2ROLL
 NOSWITCH
 L355
 STORE K2ROLL # K2ROLL = -K2ROLL

L355 DLOAD DDV # ROLL C = ACOS((L/D1) / LAD)
 L/D1
 LAD # MPAC SET TO +-1 IF OVERFLOW***
 SR1 ACOS
 SIGN CLEAR
 K2ROLL
 NOSWITCH
 STORE ROLL C

ENDEXIT EXIT

OVERNOUT CA BIT13 # ENTRYDSP =92D B13
 MASK CM/FLAGS
 EXTEND
 BZF NODISKY # OMIT DISPLAY.

Page 874
 CA ENTRYVN # ALL ENTRY DISPLAYS ARE DONE HERE.
 TC BANKCALL
 CADR REGODSPR # NO ABORT IF DISKY IN USE

NODISKY INHINT

 CCS NEWJOB # PROTECT READACCS GRP 5, IF SIDETRACKED.
 TC CHANG1
SERVNOUT TC POSTJUMP # (COME HERE FROM P67.3)
 CADR SERVEXIT # AND END AVERAGEG JOB VIA ENDOJOB.

Page 875
DISPLAY WHEN V IS LESS THAN VQUIT.

STEEROFF EXIT
 CA EBENTRY # PRECAUTIONARY.
 TS EBANK

```



July 12, 2016

1649

```
CA PRI016 # 2 LESS THAN NTRYPRIO.
TC NOVAC
EBANK= AOG # ANY EB HERE
2CADR P67.1 # START UP REMAINDER OF P67

 # RTOGO LAT LONG
 # XXXX.X NM XXX.XX DEG XXX.XX DEG

TC 2PHSCHNG # INHINT/RELINT DONE.
OCT 00414 # 4.41 RESTART FOR P67.1 DISPLAY JOB.
OCT 10035 # SERVICER 5.3 RESTART.

CA P67.2CAD # HEREAFTER, DO LAT, LONG.
TS GOTOADDR

TC INTPRET
GOTO

P67.2CAD P67.2 # CONTINUE FOR LAT, LONG THIS TIME.

L357 DLOAD SIGN # L/D = L/DCMINR SIGN(L/D)
 L/DCMINR
 L/D
STCALL L/D1
 L355 # (GO TO)

Page 876
TABLE USED FOR SUB-ORBITAL REFERENCE TRAJECTORY CONTROL.

VREFER DEC .019288 # REFERENCE VELOCITY SCALED V/51532.3946
 DEC .040809 # 13 POINTS ARE STORED AS THE INDEPENDENT
 DEC .076107 # VARIABLE AND THEN SIX 13-POINT FUNCTIONS
 DEC .122156 # OF V ARE STORED CONSECUTIVELY
 DEC .165546
 DEC .196012
 DEC .271945
 DEC .309533
 DEC .356222
 DEC .404192
 DEC .448067
 DEC .456023
 DEC .67918 # HIGHVELOCITY FOR SAFETY.

 DEC -.010337 # DRANGE/DA SCALED DRDA/(2700/805)
 DEC -.016550
 DEC -.026935
 DEC -.042039
```

DEC -.058974  
 DEC -.070721  
 DEC -.098538  
 DEC -.107482  
 DEC -.147762  
 DEC -.193289  
 DEC -.602557  
 DEC -.99999  
 DEC -.99999

DEC -.0478599 B-3 # -DRANGE/DRDOT  
 DEC -.0683663 B-3 # SCALED ((2VS/8 2700) DR/DRDOT)  
 DEC -.1343468 B-3  
 DEC -.2759846 B-3  
 DEC -.4731437 B-3  
 DEC -.6472087 B-3  
 DEC -1.171693 B-3  
 DEC -1.466382 B-3  
 DEC -1.905171 B-3  
 DEC -2.547990 B-3  
 DEC -4.151220 B-3  
 DEC -5.813617 B-3  
 DEC -5.813617 B-3

# Page 877

DEC -.0134001 B3 # RDOTREF SCALED (8 RDT/2VS)  
 DEC -.013947 B3  
 DEC -.013462 B3  
 DEC -.011813 B3  
 DEC -.0095631 B3  
 DEC -.00806946 B3  
 DEC -.006828 B3  
 DEC -.00806946 B3  
 DEC -.0109791 B3  
 DEC -.0151498 B3  
 DEC -.0179817 B3  
 DEC -.0159061 B3  
 DEC -.0159061 B3

DEC .0008067 # RANGE TO GO SCALED RTGO/2700  
 DEC .0032963 # 8.9  
 DEC .0081852 # 22.1  
 DEC .017148  
 DEC .027926  
 DEC .037  
 DEC .063298

July 12, 2016

1651

DEC .077889  
DEC .098815  
DEC .127519  
DEC .186963  
DEC .238148  
DEC .294185185

DEC -.051099 # -AREF/805  
DEC -.074534  
DEC -.101242  
DEC -.116646  
DEC -.122360  
DEC -.127081  
DEC -.147453  
DEC -.155528  
DEC -.149565  
DEC -.118509  
DEC -.034907  
DEC -.007950  
DEC -.007950

# Page 878

DEC .004491 # DRANGE/D L/D SCALED Y/2700  
DEC .008081  
DEC .016030  
DEC .035815  
DEC .069422  
DEC .104519  
DEC .122  
DEC .172407  
DEC .252852  
DEC .363148  
DEC .512963  
DEC .558519  
DEC .558519 # END OF STORED REFERENCE

# Page 879

# REENTRY CONSTANTS.

# DEFINED BY EQUALS

DEC15 = LOW4  
#GAMMAL1 = 22D

MAXRNG 20CT 1663106755 # DNRNGERR = 9999.9 IF GONEPAST=1

1652

July 12, 2016

|                                        |        |             |                                              |
|----------------------------------------|--------|-------------|----------------------------------------------|
|                                        | BANK   | 26          |                                              |
|                                        | SETLOC | REENTRY1    |                                              |
|                                        | BANK   |             |                                              |
|                                        | COUNT* | \$\$/ENTRY  |                                              |
| BARELY1                                | =      | NEARONE     | # COMMON TO BOTH DISK,DANCE, DEFND IN TFF    |
| #1BITDP                                |        |             | # COMMON TO BOTH DISK AND DANCE. DEFND IN VE |
| 1/12TH                                 | DEC    | .083333     | # DP 1/12 USES HI WORD IN 1/3 BELOW          |
| 1/3RD                                  | 2DEC   | .3333333333 | # DP 1/3                                     |
| 1/16TH                                 | =      | DP2(-4)     |                                              |
| # BELOW: VS = VSAT = 25766.1973 FT/SEC |        |             |                                              |
| # RE = 21,202,900 FEET                 |        |             |                                              |
| LEWD1                                  | 2DEC   | .15         |                                              |
| POINT1                                 | 2DEC   | .1          |                                              |
| POINT2                                 | 2DEC   | .2          | # .2                                         |
| DLEWD0                                 | 2DEC   | -.05        | # -.05                                       |
| GMAX/2                                 | 2DEC   | .16         | # 8 GS / 2                                   |
| 3ZEROS                                 | EQUALS | HI6ZEROS    |                                              |
| NEAR1/4                                | 2OCT   | 0777700000  | # 1/4 LESS 1 BIT IN UPPER PART.              |
| C18                                    | 2DEC   | .0097026346 | # 500/2VS                                    |
| Q7FKDMIN                               | 2DEC   | .0080745342 | # 6.5/805 (Q7F +KDMIN) = 6 + .5)             |
| C1/16                                  | =      | DP2(-4)     |                                              |
| Q3                                     | 2DEC   | .167003132  | # .07 2VS/21600                              |
| # Page 880                             |        |             |                                              |
| Q5                                     | 2DEC   | .326388889  | # .3 23500/21600                             |
| Q6                                     | 2DEC   | .0349       | # 2 DEG. APPROX 820/23500                    |
| Q7F                                    | 2DEC   | .0074534161 | # 6/805 (VALUE OF Q7 IN FIXED MEM.)          |
| Q19                                    | =      | HALVE       | # Q19 = .5                                   |

July 12, 2016

1653

|                      |       |                     |                                   |
|----------------------|-------|---------------------|-----------------------------------|
| Q21                  | 2DEC  | .0231481481         | # 500/21600                       |
| Q22                  | 2DEC  | -.0533333333        | # -1152/21600                     |
| VLMIN                | 2DEC  | .34929485           | # 18000/2 VS                      |
| VMIN                 | =     | FOURTH              | # (VS/2) / 2VS                    |
| C12                  | 2DEC  | .00684572901        | # 32 28500/(21202900 2 PI)        |
| 1/KB1                | 2DEC  | .29411765           | # 1 / 3.4                         |
| -1/KB2               | 2DEC  | -.0057074322 B4     | # - 1/(.0034 2 VS) EXP +4         |
| VQUIT                | 2DEC  | .019405269          | # 1000 /2VS                       |
| C20                  | 2DEC  | .21739130           | # (175 FPSS) LIFT UP IF ABOVE C20 |
| C21                  | 2DEC  | .17391304           | # 140/805                         |
| 25NM                 | 2DEC  | .0011574074         | # 25/21600 (25 NAUT MILES)        |
| K1D                  | 2DEC  | .0314453125         | # =C16 805/256 = .01 805/256      |
| K2D                  | 2DEC  | -.201298418         | # -C17 2VS/256 = -.001 2VS/256    |
| KVSCALE              | 2DEC  | .81491944           | # 12800/(2 VS .3048)              |
| KASCALE              | 2DEC  | .97657358           | # 5.85 16384/(4 .3048 100 805)    |
| KTETA                | 2DEC* | .383495203 E2 B-14* | # 1000 2PI/16384(163.84)          |
| KT1                  | 2DEC* | .157788327 E2 B-14* | # RE(2PI)/2 VS(16384) 163.84      |
| .05G                 | 2DEC  | .002                | # .05/25                          |
| LATBIAS              | 2DEC  | .00003              | # APPRX .5 NM/ 4(21600/2 PI)      |
| KWE                  | 2DEC  | .120056652 B-1      |                                   |
| KACOS                | 2DEC  | .004973592          | # 1/32(2PI)                       |
| CHOOK                | 2DEC  | 1 B-6               | # .25/16                          |
| # Page 881<br>1/24TH | 2DEC  | .0833333333 B-1     |                                   |

1654

July 12, 2016

|          |        |              |                               |
|----------|--------|--------------|-------------------------------|
| CH1      | 2DEC   | .32 B1       | # 16 CH1/25 = 16 (1) /25      |
| KC3      | 2DEC   | -.0247622232 | # -(4 VS VS/ 2 PI 805 RE)     |
| VRCONT   | 2DEC   | .0135836886  | # 700/2 VSAT                  |
| HALVE    | EQUALS | HIDPHALF     |                               |
| FOURTH   | EQUALS | HIDP1/4      |                               |
| 1/GMAX   | EQUALS | HALVE        | # 4/GMAX = 4 / 8              |
| 2HS      | 2DEC   | .0172786611  | # 2 28500 25 32.2/(4 VS VS)   |
| 2HSGMXSQ | 2DEC   | .0000305717  | # (2 28500 8 32.2/ 4 VS VS)SQ |
| C001     | 2DEC   | -.000625     | # -(4/25)/256 LEQ/DO CONST    |
| POINT8   | 2DEC   | .8           |                               |
| 2C1HS    | 2DEC   | .0215983264  | # 2 1.25 28500 805/(2 VS)SQ   |
| PT1/16   | 2DEC   | .1 B-4       |                               |
| 1/K44    | 2DEC   | .00260929464 | # 2 VS/19749550               |
| VFINAL   | 2DEC   | .51618016    | # 26600/2 VS                  |
| VFINAL1  | 2DEC   | .523942273   | # = 27000 / 2 VS              |
| 1/KA1    | 2DEC   | .30048077    | # 25/(1.3 64)                 |
| KA2      | 2DEC   | .008         | # .2/25                       |
| KA3      | 2DEC   | .44720497    | # = 90 4/805                  |
| KA4      | 2DEC   | .049689441   | # 40/805                      |
| KALIM    | 2DEC   | .06          | # 1.5/25                      |
| Q7MIN    | =      | KA4          | # = 40/805 = .049689441       |
| -HSCALED | 2DEC   | -.55305018   | # -28500/2 VS                 |
| -KSCALE  | 2DEC   | -.0312424837 | # -805/VS                     |
| COS15    | 2DEC   | .965         |                               |
| LATSLOPE | EQUALS | 1/12TH       |                               |

July 12, 2016

1655

```
... END OF RE-ENTRY CONSTANTS ...
```

This code is written to file `src/REENTRY-CONTROL.s`.

## B.95 RESTARTS ROUTINE

```

1656 <src/RESTARTS-ROUTINE.s 1656>≡
 # Copyright: Public domain.
 # Filename: RESTARTS_ROUTINE.agc
 # Purpose: Part of the source code for Comanche, build 055. It
 # is part of the source code for the Command Module's
 # (CM) Apollo Guidance Computer (AGC), Apollo 11.
 # Assembler: yaYUL
 # Reference: pp. 1414-1419
 # Contact: Ron Burkey <info@sandroid.org>
 # Website: http://www.ibiblio.org/apollo.
 # Mod history: 2009-05-07 RSB Adapted from Colossus249 file of the same
 # name, and page images. Corrected various
 # typos in the transcription of program
 # comments, and these should be back-ported
 # to Colossus249.
 #
 # The contents of the "Comanche055" files, in general, are transcribed
 # from scanned documents.
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. April 1, 1969.
 #
 # This AGC program shall also be referred to as Colossus 2A
 #
 # Prepared by
 #
 # Massachusetts Institute of Technology
 # 75 Cambridge Parkway
 # Cambridge, Massachusetts
 #
 # under NASA contract NAS 9-4065.
 #
 # Refer directly to the online document mentioned above for further
 # information. Please report any errors to info@sandroid.org.

 # Page 1414

 BANK 01
 SETLOC RESTART
 BANK

 EBANK= PHSNAME1 # GOPROG MUST SWITCH TO THIS EBANK

 COUNT 01/RSROU

 RESTARTS CA MPAC +5 # GET GROUP NUMBER -1

```



July 12, 2016

1657

```
DOUBLE # SAVE FOR INDEXING
TS TEMP2G

CA PHS2CADR # SET UP EXIT IN CASE IT IS AN EVEN
TS TEMPSWCH # TABLE PHASE

CA RTRNCADR # TO SAVE TIME ASSUME IT WILL GET NEXT
TS GOLOC +2 # GROUP AFTER THIS

CA TEMPPHS
MASK OCT1400
CCS A # IS IT A VARIABLE OR TABLE RESTART
TCF ITSAVAR # IT'S A VARIABLE RESTART

GETPART2 CCS TEMPPHS # IS IT AN X.1 RESTART
 CCS A
 TCF ITSATBL # NO, IT'S A TABLE RESTART

CA PRI014 # IT IS AN X.1 RESTART, THEREFORE START
TC FINDVAC # THE DISPLAY RESTART JOB
EBANK= LST1
2CADR INITDSP

TC RTRNCADR # FINISHED WITH THIS GROUP, GET NEXT ONE

ITSAVAR MASK OCT1400 # IS IT TYPE B ?
 CCS A
 TCF ITS LIKE B # YES, IT IS TYPE B

 EXTEND # STORES THE JOB (OR TASK) 2CADR FOR EXIT
 NDX TEMP2G
 DCA PHSNAME1
 DXCH GOLOC

CA TEMPPHS # SEE IF THIS IS A JOB, TASK, OR A LONGCALL
MASK OCT7
AD MINUS2
CCS A
TCF ITSLNGCL # IT'S A LONGCALL

Page 1415
RTRNCADR TC SWRETURN # CAN'T GET HERE.
 TCF ITSAWAIT

TCF ITSAJOB # IT'S A JOB
```

1658

July 12, 2016

|          |      |          |                                 |
|----------|------|----------|---------------------------------|
| ITSAWAIT | CA   | WILTCADR | # SET UP WAITLIST CALL          |
|          | TS   | GOLOC -1 |                                 |
|          | NDX  | TEMP2G   | # DIRECTLY STORED               |
|          | CA   | PHSPRDT1 |                                 |
| TIMETEST | CCS  | A        | # IS IT AN IMMEDIATE RESTART    |
|          | INCR | A        | # NO.                           |
|          | TCF  | FINDTIME | # FIND OUT WHEN IT SHOULD BEGIN |
|          | TCF  | ITSINDIR | # STORED INDIRECTLY             |
|          | TCF  | IMEDIATE | # IT WANTS AN IMMEDIATE RESTART |

# \*\*\*\*\* THIS MUST BE IN FIXED FIXED \*\*\*\*\*

|          |        |          |                                          |
|----------|--------|----------|------------------------------------------|
|          | BLOCK  | 02       |                                          |
|          | SETLOC | FFTAG2   |                                          |
|          | BANK   |          |                                          |
|          | COUNT  | 02/RSROU |                                          |
| ITSINDIR | LXCH   | GOLOC +1 | # GET THE CORRECT E BANK IN CASE THIS IS |
|          | LXCH   | BB       | # SWITCHED ERASABLE                      |
|          | NDX    | A        | # GET THE TIME INDIRECTLY                |
|          | CA     | 1        |                                          |
|          | LXCH   | BB       | # RESTORE THE BB AND GOLOC               |
|          | LXCH   | GOLOC +1 |                                          |
|          | TCF    | FINDTIME | # FIND OUT WHEN IT SHOULD BEGIN          |

# \*\*\*\*\* YOU MAY RETURN TO SWITCHED FIXED \*\*\*\*\*

|          |        |          |                                             |
|----------|--------|----------|---------------------------------------------|
|          | BANK   | 01       |                                             |
|          | SETLOC | RESTART  |                                             |
|          | BANK   |          |                                             |
|          | COUNT  | 01/RSROU |                                             |
| FINDTIME | COM    |          | # MAKE NEGATIVE SINCE IT WILL BE SUBTRACTED |
|          | TS     | L        | # AND SAVE                                  |
|          | NDX    | TEMP2G   |                                             |
|          | CS     | TBASE1   |                                             |
|          | EXTEND |          |                                             |

July 12, 2016

1659

|          |        |          |                                           |
|----------|--------|----------|-------------------------------------------|
|          | SU     | TIME1    |                                           |
|          | CCS    | A        |                                           |
|          | COM    |          |                                           |
|          | AD     | OCT37776 |                                           |
|          | AD     | ONE      |                                           |
|          | AD     | L        |                                           |
|          | CCS    | A        |                                           |
|          | CA     | ZERO     |                                           |
|          | TCF    | +2       |                                           |
|          | TCF    | +1       |                                           |
| IMEDIATE | AD     | ONE      |                                           |
|          | TC     | GOLOC -1 |                                           |
| ITSLIKEB | CA     | RTRNCADR | # TYPE B, SO STORE RETURN IN              |
|          | TS     | TEMPSWCH | # TEMPSWCH IN CASE OF AN EVEN PHASE       |
|          | CA     | PRT2CADR | # SET UP EXIT TO GET TABLE PART OF THIS   |
|          | TS     | GOLOC +2 | # VARIABLE TYPE OF PHASE                  |
|          | CA     | TEMPPHS  | # MAKE THE PHASE LOOK RIGHT FOR THE TABLE |
|          | MASK   | OCT177   | # PART OF THIS VARIABLE PHASE             |
|          | TS     | TEMPPHS  |                                           |
|          | EXTEND |          |                                           |
|          | NDX    | TEMP2G   | # OBTAIN THE JOB'S 2CADR                  |
|          | DCA    | PHSNAME1 |                                           |
|          | DXCH   | GOLOC    |                                           |
| ITSAJOB  | NDX    | TEMP2G   | # NOW ADD THE PRIORITY AND LET'S GO       |
|          | CA     | PHSPRDT1 |                                           |
| CHKNOVAC | TS     | GOLOC -1 | # SAVE PRIO UNTIL WE SEE IF IT'S          |
|          | EXTEND |          | # A FINDVAC OR A NOVAC                    |
|          | BZMF   | ITSNOVAC |                                           |
|          | CAF    | FVACCADR | # POSITIVE, SET UP FINDVAC CALL.          |
|          | XCH    | GOLOC -1 | # PICK UP PRIO                            |
|          | TC     | GOLOC -1 | AND GO                                    |
| ITSNOVAC | CAF    | NOVACADR | # NEGATIVE,                               |
|          | XCH    | GOLOC -1 | # SET UP NOVAC CALL,                      |
|          | COM    |          | # CORRECT PRIO,                           |
|          | TC     | GOLOC -1 | # AND GO                                  |
| ITSATBL  | TS     | CYR      | # FIND OUT IF THE PHASE IS ODD OR EVEN    |
|          | CCS    | CYR      |                                           |
|          | TCF    | +1       | # IT'S EVEN                               |
|          | TCF    | ITSEVEN  |                                           |

```

Page 1417
CA RTRNCADR # IN CASE THIS IS THE SECOND PART OF A
TS GOLOC +2 # TYPE B RESTART, WE NEED PROPER EXIT

CA TEMPPHS # SET UP POINTER FOR FINDING OUR PLACE IN
TS SR # THE RESTART TABLES
AD SR
NDX TEMP2G
AD SIZETAB +1
TS POINTER

CONTBL2 EXTEND # FIND OUT WHAT'S IN THE TABLE
NDX POINTER
DCA CADRTAB # GET THE 2CADR

LXCH GOLOC +1 # STORE THE BB INFORMATION

CCS A # IS IT A JOB OR IT IT TIMED
INCR A # POSITIVE, MUST BE A JOB
TCF ITSAJOB2

INCR A # MUST BE EITHER A WAITLIST OR LONGCALL
TS GOLOC # LET'S STORE THE CORRECT CADR

CA WTLTCADR # SET UP OUR EXIT TO WAITLIST
TS GOLOC -1

CA GOLOC +1 # NOW FIND OUT IF IT IS A WAITLIST CALL
MASK BIT10 # THIS SHOULD BE ONE IF WE HAVE -BB
CCS A # FOR THAT MATTER SO SHOULD BE BITS 9,8,7,
 # 6,5, AND LAST BUT NOT LEAST (PERHAPS NOT
 # IN IMPORTANCE ANYWAY. BUT 4
TCF ITSWTLST # IT IS A WAITLIST CALL

NDX POINTER # OBTAIN THE ORIGINAL DELTA T
CA PRDTTAB # ADDRESS FOR THIS LONGCALL

TCF ITSLGCL1 # NOW GO GET THE DELTA TIME

***** THIS MUST BE IN FIXED FIXED *****

BLOCK 02
SETLOC FFTAG2
BANK

COUNT 02/RSROU

```

July 12, 2016

1661

```
ITSLGCL1 LXCH GOLOC +1 # OBTAIN THE CORRECT E BANK
 LXCH BB
 LXCH GOLOC +1 # AND PRESERVE OUR E AND F BANKS

 EXTEND
 NDX A # GET THE DELTA TIME
 DCA 0

Page 1418
 LXCH GOLOC +1 # RESTORE OUR E AND F BANK
 LXCH BB # RESTORE THE TASKS E AND F BANKS
 LXCH GOLOC +1 # AND PRESERVE OUR L
 TCF ITSLGCL2 # NOT GET'S PROCESS THIS LONGCALL

***** YOU MAY RETURN TO SWITCHED FIXED *****

 BANK 01
 SETLOC RESTART
 BANK

ITSLGCL2 COUNT 01/RSROU
 DXCH LONGTIME

 EXTEND
 DCS TIME2 # CALCULATE TIME LEFT
 DAS LONGTIME
 EXTEND
 DCA LONGBASE
 DAS LONGTIME

 CCS LONGTIME # FIND OUT HOW THIS SHOULD BE RESTARTED
 TCF LONGCLCL
 TCF +2
 TCF IMEDIATE -3
 CCS LONGTIME +1
 TCF LONGCLCL

 NOOP
 TCF IMEDIATE -3 # CAN'T GET HERE *****
 TCF IMEDIATE

LONGCLCL CA LGCLCADR # WE WILL GO TO LONGCALL
 TS GOLOC -1

 EXTEND
 DCA LONGTIME # PREPARE OUR ENTRY TO LONGCALL
 TC GOLOC -1
```

|             |        |          |                                            |
|-------------|--------|----------|--------------------------------------------|
| ITSLNGCL    | CA     | WILTCADR | # ASSUME IT WILL GO TO WAITLIST            |
|             | TS     | GOLOC -1 |                                            |
|             | NDX    | TEMP2G   |                                            |
|             | CS     | PHSPRDT1 | # GET THE DELTA T ADDRESS                  |
|             | TCF    | ITSLGCL1 | # NOW GET THE DELTA TIME                   |
| ITSWTLST    | CS     | GOLOC +1 | # CORRECT THE BBCON INFORMATION            |
|             | TS     | GOLOC +1 |                                            |
| # Page 1419 | NDX    | POINTER  | # GET THE DT AND FIND OUT IF IT WAS STORED |
|             | CA     | PRDTTAB  | # DIRECTLY OR INDIRECTLY                   |
|             | TCF    | TIMETEST | # FIND OUT HOW THE TIME IS STORED          |
| ITSAJOB2    | XCH    | GOLOC    | # STORE THE CADR                           |
|             | NDX    | POINTER  | # ADD THE PRIORITY AND LET'S GO            |
|             | CA     | PRDTTAB  |                                            |
|             | TCF    | CHKNOVAC |                                            |
| ITSEVEN     | CA     | TEMPSWCH | # SET FOR EITHER THE SECOND PART OF THE    |
|             | TS     | GOLOC +2 | # TABLE, OR A RETURN FOR THE NEXT GROUP    |
|             | NDX    | TEMP2G   | # SET UP POINTER FOR OUR LOCATION WITHIN   |
|             | CA     | SIZETAB  | # THE TABLE                                |
|             | AD     | TEMPPHS  | # THIS MAY LOOK BAD BUT LET'S SEE YOU DO   |
|             | AD     | TEMPPHS  | # BETTER IN TIME OR NUMBER OF LOCATIONS    |
|             | AD     | TEMPPHS  |                                            |
|             | TS     | POINTER  |                                            |
|             | TCF    | CONTBL2  | # NO PROCESS WHAT IS IN THE TABLE          |
| PHSPART2    | CA     | THREE    | # SET THE POINTER FOR THE SECOND HALF OF   |
|             | ADS    | POINTER  | # THE TABLE                                |
|             | CA     | RTRNCADR | # THIS WILL BE OUR LAST TIME THROUGH THE   |
|             | TS     | GOLOC +2 | # EVEN TABLE, SO AFTER IT GET THE NEXT     |
|             |        |          | # GROUP                                    |
|             | TCF    | CONTBL2  | # SO LET'S GET THE SECOND ENTRY IN THE TBL |
| TEMPPHS     | EQUALS | MPAC     |                                            |
| TEMP2G      | EQUALS | MPAC +1  |                                            |

July 12, 2016

1663

|          |        |           |
|----------|--------|-----------|
| POINTER  | EQUALS | MPAC +2   |
| TEMPSWCH | EQUALS | MPAC +3   |
| GOLOC    | EQUALS | VAC5 +20D |
| MINUS2   | EQUALS | NEG2      |
| OCT177   | EQUALS | LOW7      |

|          |        |          |
|----------|--------|----------|
| PHS2CADR | GENADR | PHSPART2 |
| PRT2CADR | GENADR | GETPART2 |
| LGCLCADR | GENADR | LONGCALL |
| FVACCADR | GENADR | FINDVAC  |
| WTLTCADR | GENADR | WAITLIST |
| NOVACADR | GENADR | NOVAC    |

This code is written to file `src/RESTARTS-ROUTINE.s`.

## B.96 RESTART TABLES

```

1664 <src/RESTART-TABLES.s 1664>≡
 # Copyright: Public domain.
 # Filename: RESTART_TABLES.agc
 # Purpose: Part of the source code for Comanche, build 055. It
 # is part of the source code for the Command Module's
 # (CM) Apollo Guidance Computer (AGC), Apollo 11.
 # Assembler: yaYUL
 # Reference: pp. 211-221
 # Contact: Ron Burkey <info@sandroid.org>,
 # Fabrizio Bernardini <fabrizio@spacecraft.it>
 # Website: http://www.ibiblio.org/apollo.
 # Mod history: 2009-05-16 FB Transcription Batch 2 Assignment.
 # 2009-05-20 RSB Added a missing comment mark. Corrected mismarked
 # Page 217 -> 220.
 # 2009-05-21 RSB Fixed value of 5.21SPOT.
 #
 # The contents of the "Comanche055" files, in general, are transcribed
 # from scanned documents.
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. April 1, 1969.
 #
 # This AGC program shall also be referred to as Colossus 2A
 #
 # Prepared by
 #
 # Massachussets Institute of Technology
 # 75 Cambridge Parkway
 # Cambridge, Massachusetts
 #
 # under NASA contract NAS 9-4065.
 #
 # Refer directly to the online document mentioned above for further
 # information. Please report any errors to info@sandroid.org.
 #
 # Page 211
 # RESTART TABLES
 # -----
 #
 # THERE ARE TWO FORMS OF RESTART TABLES FOR EACH GROUP. THEY ARE KNOWN AS THE EVEN P
 # RESTART TABLES. THE ODD TABLES HAVE ONLY ONE ENTRY OF THREE LOCATIONS WHILE THE EV
 # EACH USING THREE LOCATIONS. THE INFORMATION AS TO WHETHER IT IS A JOB, WAITLIST, C
 # WAY THINGS ARE PUT IN TO THE TABLES.
 #
 # A JOB HAS ITS PRIORITY STORED IN A PRDTTAB OF THE CORRECT PHASE SPOT -- A POSITIVE

```



July 12, 2016

1665

```
FINDVAC JOB, A NEGATIVE PRIORITY A NOVAC. THE 2CADR OF THE JOB IS STORED IN THE CADRTAB.
FOR EXAMPLE,
#
5.7SPOT OCT 23000
2CADR SOMEJOB
#
A RESTART OF GROUP 5 WITH PHASE SEVEN WOULD THEN CAUSE SOMEJOB TO BE RESTARTED AS A FINDVAC W
#
5.5SPOT OCT -23000
2CADR ANYJOB
#
HERE A RESTART OF GROUP 5 WITH PHASE 7 WOULD CAUSE ANYJOB TO BE RESTARTED AS A NOVAC WITH PRI
A LONGCALL HAS ITS GENADR OF ITS 2CADR STORED NEGATIVELY AND ITS BBCON STORED POSITIVELY. IN
PLACED THE LOCATION OF A DP REGISTER THAT CONTAINS THE DELTA TIME THAT LONGCALL HAD BEEN ORIG
WITH. EXAMPLE,
#
3.6SPOT GENADR DELTAT
-GENADR LONGTASK
BBCON LONGTASK
#
OCT 31000
2CADR JOBAGAIN
#
THIS WOULD START UP LONGTASK AT THE APPROPRIATE TIME, OR IMMEDIATELY IF THE TIME HAD ALREADY
BE NOTED THAT IF DELTAT IS IN A SWITCHED E BANK, THIS INFORMATION SHOULD BE IN THE BBCON OF T
TASK. FROM ABOVE, WE SEE THAT THE SECOND PART OF THIS PHASE WOULD BE STARTED AS A JOB WITH A
#
WAITLIST CALLS ARE IDENTIFIED BY THE FACT THAT THEIR 2CADR IS STORED NEGATIVELY. IF PRDTTAB
IS POSITIVE, THEN IT CONTAINS THE DELTA TIME, IF PRDTTAB IS NEGATIVE THEN IT IS THE -GENADR C
LOCATION CONTAINING THE DELTA TIME, THAT IS, THE TIME IS STORED INDIRECTLY. IT SHOULD BE NOTE
IF THE TIME IS STORED INDIRECTLY, THE BBCON MUST CONTAIN THE NECESSARY E BANK INFORMATION IF
WAITLIST WE HAVE ONE FURTHER OPTION, IF -0 IS STORED IN PRDTTAB, IT WILL CAUSE AN IMMEDIATE R
TASK. EXAMPLES,
#
OCT 77777 # THIS WILL CAUSE AN IMMEDIATE RESTART
-2CADR ATASK # OF THE TASK "ATASK"
#
DEC 200 # IF THE TIME OF THE 2 SECONDS SINCE DU
-2CADR DUMMY # WAS PUT ON THE WAITLIST IS UP, IT WILL
IN 10 MS, OTHERWISE IT WILL BEGIN WHE
IT NORMALLY WOULD HAVE BEGUN.
#
Page 212
-GENADR DTIME # WHERE DTIME CONTAINS THE DELTA TIME
-2CADR TASKTASK # OTHERWISE THIS IS AS ABOVE
#
***** NOW THE TABLES THEMSELVES *****
```

|                                                        |        |                |                                       |
|--------------------------------------------------------|--------|----------------|---------------------------------------|
|                                                        | BANK   | 01             |                                       |
|                                                        | SETLOC | RESTART        |                                       |
|                                                        | BANK   |                |                                       |
|                                                        | COUNT  | 01/RSTAB       |                                       |
| PRDTTAB                                                | EQUALS | 12000          | # USED TO FIND THE PRIORITY OR DELTA  |
| CADRTAB                                                | EQUALS | 12001          | # THIS AND THE NEXT RELATIVE LOC CONT |
|                                                        |        |                | # RESTART 2CADR                       |
| SIZETAB                                                | TC     | 1.2SPOT -12006 |                                       |
|                                                        | TC     | 1.3SPOT -12004 |                                       |
|                                                        | TC     | 2.2SPOT -12006 |                                       |
|                                                        | TC     | 2.3SPOT -12004 |                                       |
|                                                        | TC     | 3.2SPOT -12006 |                                       |
|                                                        | TC     | 3.3SPOT -12004 |                                       |
|                                                        | TC     | 4.2SPOT -12006 |                                       |
|                                                        | TC     | 4.3SPOT -12004 |                                       |
|                                                        | TC     | 5.2SPOT -12006 |                                       |
|                                                        | TC     | 5.3SPOT -12004 |                                       |
|                                                        | TC     | 6.2SPOT -12006 |                                       |
|                                                        | TC     | 6.3SPOT -12004 |                                       |
| 1.2SPOT                                                | EQUALS | 3.2SPOT        |                                       |
| # ANY MORE GROUP 1.EVEN RESTART VALUES SHOULD GO HERE. |        |                |                                       |
| 1.3SPOT                                                | DEC    | 120            | # THIS NUMBER MUST BE EQUAL C(JTAGTIN |
|                                                        | EBANK= | AOG            |                                       |
|                                                        | -2CADR | SETJTAG        |                                       |
| 1.5SPOT                                                | OCT    | 10000          |                                       |
|                                                        | EBANK= | DAPDATR1       |                                       |
|                                                        | 2CADR  | RED040.9       |                                       |
| 1.7SPOT                                                | OCT    | 10000          |                                       |
|                                                        | EBANK= | ESTROKER       |                                       |
|                                                        | 2CADR  | RELINUS        |                                       |
| 1.11SPOT                                               | OCT    | 10000          |                                       |
|                                                        | EBANK= | ESTROKER       |                                       |
|                                                        | 2CADR  | PIKUP20        |                                       |
| # ANY MORE GROUP 1.ODD RESTART VALUES SHOULD GO HERE.  |        |                |                                       |
| 2.2SPOT                                                | EQUALS | 1.2SPOT        |                                       |

# ANY MORE GROUP 2.EVEN RESTART VALUES SHOULD GO HERE

# Page 213

2.3SPOT            GENADR   600SECS  
                  -GENADR STATEINT  
                  EBANK=   RRECTCSM  
                  BBCON   STATEINT

2.5SPOT           OCT       05000  
                  EBANK=   RRECTCSM  
                  2CADR   STATINT1

2.7SPOT           OCT       10000  
                  EBANK=   MRKBUF2  
                  2CADR   R22

2.11SPOT          OCT       14000  
                  EBANK=   LANDMARK  
                  2CADR   V94ENTER

2.13SPOT          OCT       10000  
                  EBANK=   MRKBUF2  
                  2CADR   REDOR22

# ANY MORE GROUP 2.ODD RESTART VALUES SHOULD GO HERE.

3.2SPOT           EQUALS   4.2SPOT

# ANY MORE GROUP 3.EVEN RESTART VALUES SHOULD GO HERE

3.3SPOT           OCT       20000  
                  EBANK=   TGO  
                  2CADR   S40.13

3.5SPOT           DEC       0  
                  DEC       0  
                  DEC       0

3.7SPOT           OCT       22000  
                  EBANK=   TEPHEM  
                  2CADR   MATRXJOB

3.11SPOT          OCT       22000  
                  EBANK=   TEPHEM  
                  2CADR   REP11

3.13SPOT        OCT       22000  
                 EBANK=   TEPHEM  
                 2CADR    REP11A

3.15SPOT       -GENADR TGO       +1  
                 EBANK=   TGO  
                 -2CADR   ENGINOFF

# Page 214

# ANY MORE GROUP 3.ODD RESTART VALUES SHOULD GO HERE

4.2SPOT        OCT       77777  
                 EBANK=   TIG  
                 -2CADR   PRECHECK

OCT       30000  
EBANK=   DELVIMU  
2CADR    P47BODY

4.4SPOT        OCT       77777  
                 EBANK=   TIG  
                 -2CADR   PRECHECK

DEC       2996  
EBANK=   DAPDATR1  
-2CADR   TTG/0

4.6SPOT        OCT       77777  
                 EBANK=   TIG  
                 -2CADR   PRECHECK

DEC       2496  
EBANK=   TIG  
-2CADR   TIG-5

# ANY MORE GROUP 4.EVEN RESTART VALUES SHOULD GO HERE

4.3SPOT        DEC       40  
                 EBANK=   PACTOFF  
                 -2CADR   DOTVCON

4.5SPOT        DEC       160  
                 EBANK=   PACTOFF  
                 -2CADR   DOSTRULL

July 12, 2016

1669

|            |         |          |                                            |
|------------|---------|----------|--------------------------------------------|
| 4.7SPOT    | DEC     | 500      |                                            |
|            | EBANK=  | PACTOFF  |                                            |
|            | -2CADR  | TIG-0    |                                            |
| 4.11SPOT   | DEC     | 250      |                                            |
|            | EBANK=  | DAPDATR1 |                                            |
|            | -2CADR  | V97E40.6 |                                            |
| 4.13SPOT   | DEC     | 200      |                                            |
|            | EBANK=  | WHOCARES |                                            |
|            | -2CADR  | R40ENABL |                                            |
| 4.15SPOT   | OCT     | 16000    | # PRELAUNCH OPTICAL VERIFICATION           |
|            | EBANK=  | OGC      |                                            |
| # Page 215 |         |          |                                            |
|            | 2CADR   | COMPVER  | # CALLS FOR OPTICS DATA AGAIN (STD LEADIN) |
| 4.17SPOT   | OCT     | 16000    |                                            |
|            | EBANK=  | XSM      |                                            |
|            | 2CADR   | AZMTHCG1 |                                            |
| 4.21SPOT   | GENADR  | P40TMP   | # DELTA TIME USED IN SETTING UP            |
|            | -GENADR | TIGBLNK  | # LONG CALL OF TIGBLNK BY P40,P41          |
|            | EBANK=  | P40TMP   |                                            |
|            | BBCON   | TIGBLNK  |                                            |
| 4.23SPOT   | OCT     | 12000    | # PROTECT P40S/SV BY P40 P41               |
|            | EBANK=  | TIG      |                                            |
|            | 2CADR   | P40S/SV  |                                            |
| 4.25SPOT   | OCT     | 24000    |                                            |
|            | EBANK=  | BESTI    |                                            |
|            | 2CADR   | PROG52   |                                            |
| 4.27SPOT   | DEC     | 250      |                                            |
|            | EBANK=  | PACTOFF  |                                            |
|            | -2CADR  | DOTVCRCS |                                            |
| 4.31SPOT   | OCT     | 13000    |                                            |
|            | EBANK=  | STAR     |                                            |
|            | 2CADR   | R51      | +1                                         |
| 4.33SPOT   | DEC     | 2100     | # PROTECT CONTINUING JOB TO START P63      |
|            | EBANK=  | AOG      |                                            |
|            | -2CADR  | WAKEP62  |                                            |

1670

July 12, 2016

|            |         |           |                                        |
|------------|---------|-----------|----------------------------------------|
| 4.35SPOT   | OCT     | 12000     |                                        |
|            | EBANK=  | DAPDATR1  |                                        |
|            | 2CADR   | POSTBURN  |                                        |
| 4.37SPOT   | DEC     | 500       |                                        |
|            | EBANK=  | TIG       |                                        |
|            | -2CADR  | TIGAVEG   |                                        |
| 4.41SPOT   | OCT     | 17000     | # PROTECT DISPLAY JOB IN P67           |
|            | EBANK=  | AOG       |                                        |
|            | 2CADR   | P67.1     |                                        |
| 4.43SPOT   | -GENADR | S61DT     | # PROTECT TASK TO START PREREAD,ENTRY  |
|            | EBANK=  | S61DT     | # S61.1C WILL CHANGE EBANK=EB7 FOR P67 |
|            | -2CADR  | S61.1C    |                                        |
| 4.45SPOT   | OCT     | 13000     | # PROTECT CONTINUING JOB S61.1         |
|            | EBANK=  | AOG       | # (ENTRY IMU ALIGNMENT)                |
| # Page 216 |         |           |                                        |
|            | 2CADR   | S61.1A -1 |                                        |
| 4.47SPOT   | OCT     | 17000     | # PROTECT HUNTEST ITERATION            |
|            | EBANK=  | AOG       |                                        |
|            | 2CADR   | PRE-HUNT  |                                        |
| 4.51SPOT   | OCT     | 77777     | # PROTECT FDAI ATTITUDE                |
|            | EBANK=  | BODY3     | # ERROR DISPLAY IN P11                 |
|            | -2CADR  | ATERTASK  |                                        |
| 4.53SPOT   | DEC     | -0        |                                        |
|            | EBANK=  | END-E7    | # EBANK7 FOR TIG                       |
|            | -2CADR  | V97ETASK  |                                        |
| 4.55SPOT   | OCT     | 13000     | # PROTECT P65 RESPONSIVE DISPLAY.      |
|            | EBANK=  | RTINIT    |                                        |
|            | 2CADR   | P65.1     |                                        |
| 4.57SPOT   | -GENADR | P40TMP    |                                        |
|            | EBANK=  | P40TMP    |                                        |
|            | -2CADR  | TIGON     |                                        |
| 4.61SPOT   | OCT     | 77777     |                                        |
|            | EBANK=  | PACTOFF   |                                        |
|            | -2CADR  | IGNITION  |                                        |
| 4.63SPOT   | OCT     | 77777     |                                        |

July 12, 2016

1671

EBANK= PACTOFF  
-2CADR DOSPSOFF

4.65SPOT DEC 10  
EBANK= TIG  
-2CADR TIG-5

4.67SPOT DEC -0  
EBANK= CSMMASS  
-2CADR V97TTASK

4.71SPOT DEC 250  
EBANK= DAPDATR1 # (FOR RCSDAPON)  
-2CADR V97TRCS

4.73SPOT DEC -0  
EBANK= V97VCNTR  
-2CADR V97PTASK

4.75SPOT DEC -0  
EBANK= DAPDATR1  
-2CADR SPSOFF97

# Page 217

4.77SPOT DEC -0  
EBANK= PACTOFF  
-2CADR TIG-0

# ANY MORE GROUP 4.ODD RESTART VALUES SHOULD GO HERE

5.2SPOT OCT 32000  
EBANK= DVCNTR  
2CADR NORMLIZE

DEC 200  
EBANK= AOG  
-2CADR REREADAC

5.4SPOT OCT 20000  
EBANK= DVCNTR  
2CADR SERVICER

DEC 200  
EBANK= AOG  
-2CADR REREADAC

# ANY MORE GROUP 5.EVEN RESTART VALUES SHOULD GO HERE

5.3SPOT            DEC        200  
                     EBANK=    AOG  
                     -2CADR   REREADAC

5.5SPOT            OCT        77777  
                     EBANK=    AOG  
                     -2CADR   RED05.5

5.7SPOT            OCT        20000                            # USED BY PRELAUNCH  
                     EBANK=    XSM  
                     2CADR    RSTGTS1

5.11SPOT           OCT        77777  
                     EBANK=    XSM  
                     -2CADR   ALLOOP1

5.13SPOT           OCT        20000  
                     EBANK=    XSM  
                     2CADR    WTLISTNT

5.15SPOT           OCT        20000  
                     EBANK=    XSM  
                     2CADR    RESTEST1

5.17SPOT           OCT        20000  
                     EBANK=    XSM

# Page 218  
                     2CADR    GEOSTRT4

5.21SPOT           OCT        22000  
                     EBANK=    XSM  
                     2CADR    ALFLT1

5.23SPOT           OCT        77777  
                     EBANK=    XSM  
                     -2CADR   SPECSTS

5.25SPOT           OCT        20000  
                     EBANK=    XSM  
                     2CADR    RESTEST3

5.27SPOT           OCT        20000  
                     EBANK=    XSM  
                     2CADR    RESTAIER



July 12, 2016

1673

5.31SPOT        DEC       0  
                 DEC       0  
                 DEC       0

5.33SPOT        OCT       20000  
                 EBANK=   XSM  
                 2CADR   RESCHNG

5.35SPOT        DEC       0  
                 2DEC       0

5.37SPOT        OCT       77777  
                 EBANK=   AOG  
                 -2CADR   CHEKAVEG

5.41SPOT        OCT       77777                    # TO PROTECT PREREAD AT TIG-30A  
                 EBANK=   DVCNTR                    #                    TIG-15 T+60  
                 -2CADR   PREREAD

# ANY MORE GROUP 5.ODD RESTART VALUES SHOULD GO HERE

6.2SPOT        OCT       77777                    # USED BY P40 AFTER GIMB DR TST TO REPOS'N  
                 EBANK=   AK                        # ENGINE UNTIL TVCDAPON  
                 -2CADR   PRE40.6  
  
                 DEC       100  
                 EBANK=   TTOGO  
                 -2CADR   CLOKTASK

# ANY MORE 6.ODD RESTART VALUES SHOULD GO HERE

# Page 219

6.3SPOT        DEC       100  
                 EBANK=   TIG  
                 -2CADR   CLOKTASK

6.5SPOT        OCT       30000                    # PROTECT INCREMENTING OF TIME2,TIME1 BY  
                 EBANK=   TEPHEM                    # P27(UPDATE PROGRAM)  
                 2CADR   TIMEDIDR

6.7SPOT        OCT       0  
                 OCT       0  
                 OCT       0

6.11SPOT        -GENADR CM/GYMDT                    # PROTECT TASK TO READ CDUS.

EBANK= CM/GYMDT  
-2CADR READGYMB

# FOR ENTRY DAP

6.13SPOT DEC 0  
DEC 0  
DEC 0

# Page 220

# PROGRAM DESCRIPTION: NEWPHASE

# MOD: 1

# MOD BY: COPPS

#

# FUNCTIONAL DESCRIPTION:

#

# NEWPHASE IS THE QUICK WAY TO MAKE A NON VARIABLE PHASE CHANGE. IT INCLUDES T  
# TBASE OF THE GROUP. IF TBASE IS TO BE SET, -C(TIME1) IS STORED IN THE TBASE

#

# (L-1) TBASE0  
# (L) TBASE1 (IF GROUP=1)  
# (L+1)  
# (L+2) TBASE2 (IF GROUP=2)  
# -----  
# (L+6) TBASE4 (IF GROUP=4)  
# (L+7)  
# (L+8) TBASE5 (IF GROUP=5)  
#

# IN ANY CASE, THE NEGATIVE OF THE PHASE, FOLLOWED (IN THE NEXT REGISTER) BY T  
# PHASE TABLE AS FOLLOWS:

#

# (L) -PHASE1 (IF GROUP=1)  
# (L+1) PHASE1  
# (L+2) -PHASE2 (IF GROUP=2)  
# (L+3) PHASE2  
# -----  
# (L+7) PHASE4  
# (L+8) -PHASE5 (IF GROUP=5)  
# (L+9) PHASE5  
#

# CALLING SEQUENCE:

#

# EXAMPLE IS FOR PLACING A PHASE OF FIVE INTO GROUP THREE:

#

# 1) IF TBASE IS NOT TO BE SET:  
# L-1 CA FIVE  
# L TC NEWPHASE  
# L+1 OCT 00003

DATE: 11 NOV

ASSEMBLY: SU

LOG SECTION:

July 12, 2016

1675

```
#
2) IF TBASE IS TO BE SET:
L-1 CS FIVE
L TC NEWPHASE
L+1 OCT 00003
#
SUBROUTINES CALLED: NONE
#
NORMAL EXIT MODE: AT L+2 OF CALLING SEQUENCE
#
ALARM OR ABORT EXITS: NONE
#
OUTPUT: PHASE TABLE AND TBASE TABLE UPDATED
#
ERASABLE INITIALIZATION REQ'D: NONE
Page 221
DEBRIS: A,L,TEMPG

*** WARNING *** THIS PROGRAM IS TO BE PLACED IN FIXED-FIXED AND UNSWITCHED ERASABLE.

 BLOCK 02
 SETLOC FFTAG1
 BANK

 COUNT* $$/PHASE

NEWPHASE INHINT

 TS L # SAVE FOR FURTHER USE

 NDX Q # OBTAIN THE GROUP NUMBER
 CA 0
 INCR Q # OBTAIN THE RETURN ADDRESS
 DOUBLE # SAVE THE GROUP IN A FORM USED FOR
 TS TEMPG # INDEXING

 CCS L # SEE IF WE ARE TO SET TBASE
 TCF +7 # NO, THE DELTA T WAS POSITIVE
 TCF +6

NUFAZ+10 INCR A # SET TBASE AND STORE PHASE CORRECTLY
 TS L

 CS TIME1 # SET TBASE
 NDX TEMPG
 TS TBASE1 -2
```

1676

July 12, 2016

```
CS L
NDX TEMPG
DXCH -PHASE1 -2
```

```
NOW PUT THE PHASE IN THE RIGHT TAB
```

```
RELINT
```

```
TC Q
```

```
NOW RETURN TO CALLER
```

This code is written to file `src/RESTART-TABLES.s`.

## B.97 RT8 OP CODES

```

1677 <src/RT8-OP-CODES.s 1677>≡
Copyright: Public domain.
Filename: RT8_OP_CODES.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 1508-1516
Contact: Ron Burkey <info@sandroid.org>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-07 RSB Adapted from Colossus249/RT8_OP_CODES.agc
and page images.
2009-05-07 RSB Oops! Left out the entire last page before.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachusetts Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 1508

 BANK 22
 SETLOC RTBCODES
 BANK

 EBANK= XNB
 COUNT* $$/RTB

LOAD TIME2, TIME1 INTO MPAC:

LOADTIME EXTEND
 DCA TIME2

```

TCF SLOAD2

# CONVERT THE SINGLE PRECISION 2'S COMPLEMENT NUMBER ARRIVING IN MPAC (SCALED IN HALF-REVOLUTIONS)  
# DP 1'S COMPLEMENT NUMBER SCALED IN REVOLUTIONS.

CDULOGIC

CCS MPAC

CAF ZERO

TCF +3

NOOP

CS HALF

TS MPAC +1

CAF ZERO

XCH MPAC

EXTEND

MP HALF

DAS MPAC

TCF DANZIG # MODE IS ALREADY AT DOUBLE-PRECISION

# READ THE PIPS INTO MPAC WITHOUT CHANGING THEM:

READPIPS

INHINT

CA PIPAX

TS MPAC

CA PIPAY

TS MPAC +3

CA PIPAZ

RELINT

TS MPAC +5

CAF ZERO

TS MPAC +1

TS MPAC +4

TS MPAC +6

VECMODE

TCF VMODE

# FORCE TP SIGN AGREEMENT IN MPAC:

SGNAGREE

TC

TPAGREE

# Page 1509

TCF DANZIG

# CONVERT THE DP 1'S COMPLEMENT ANGLE SCALED IN REVOLUTIONS TO A SINGLE PRECISION 2'S  
# SCALED IN HALF-REVOLUTIONS.

July 12, 2016

1679

|        |     |         |
|--------|-----|---------|
| 1ST02S | TC  | 1T02SUB |
|        | CAF | ZERO    |
|        | TS  | MPAC +1 |
|        | TCF | NEWMODE |

# DO 1ST02S ON A VECTOR OF ANGLES:

|         |      |         |                                 |
|---------|------|---------|---------------------------------|
| V1ST02S | TC   | 1T02SUB | # ANSWER ARRIVES IN A AND MPAC. |
|         | DXCH | MPAC +5 |                                 |
|         | DXCH | MPAC    |                                 |
|         | TC   | 1T02SUB |                                 |
|         | TS   | MPAC +2 |                                 |
|         | DXCH | MPAC +3 |                                 |
|         | DXCH | MPAC    |                                 |
|         | TC   | 1T02SUB |                                 |
|         | TS   | MPAC +1 |                                 |
|         | CA   | MPAC +5 |                                 |
|         | TS   | MPAC    |                                 |

|        |     |         |               |
|--------|-----|---------|---------------|
| TPMODE | CAF | ONE     | # MODE IS TP. |
|        | TCF | NEWMODE |               |

# V1ST02S FOR 2 COMPONENT VECTOR. USED BY RR.

|          |      |         |
|----------|------|---------|
| 2V1ST02S | TC   | 1T02SUB |
|          | DXCH | MPAC +3 |
|          | DXCH | MPAC    |
|          | TC   | 1T02SUB |
|          | TS   | L       |
|          | CA   | MPAC +3 |
|          | TCF  | SLOAD2  |

# SUBROUTINE TO DO DOUBLING AND 1'S TO 2'S CONVERSION:

|         |        |      |                              |
|---------|--------|------|------------------------------|
| 1T02SUB | DXCH   | MPAC | # FINAL MPAC +1 UNSPECIFIED. |
|         | DDOUBL |      |                              |
|         | CCS    | A    |                              |
|         | AD     | ONE  |                              |
|         | TCF    | +2   |                              |
|         | COM    |      | # THIS WAS REVERSE OF MSU.   |
|         | TS     | MPAC | # AND SKIP ON OVERFLOW.      |

# Page 1510

TC Q

INDEX A  
 CAF LIMITS  
 ADS MPAC  
 TC Q

# OVERFLOW UNCORRECT AND IN MSU.

# Page 1511

# SUBROUTINE TO INCREMENT CDUS

INCRCDUS CAF LOCTHETA  
 TS BUF  
 CAE MPAC  
 TC CDUINC

# PLACE ADRES(THETA) IN BUF.

# INCREMENT IN 1'S COMPL.

INCR BUF  
 CAE MPAC +3  
 TC CDUINC

INCR BUF  
 CAE MPAC +5  
 TC CDUINC

TCF VECMODE

LOCTHETA ADRES THETAD

# THE FOLLOWING ROUTINE INCREMENTS IN 2'S COMPLEMENT THE REGISTER WHOSE ADDRESS IS IN  
 # QUANTITY FOUND IN TEM2. THIS MAY BE USED TO INCRMENT DESIRED IMU AND OPTICS CDU AD  
 # (+0 UNEQUAL TO -0) QUANTITY. MAY BE CALLED BY BANKCALL/SWCALL.

CDUINC TS TEM2  
 INDEX BUF  
 CCS 0  
 AD ONE  
 TCF +4  
 AD ONE  
 AD ONE  
 COM

# 1'S COMPL. QUANT. ARRIVES IN ACC. STORE IT

# CHANGE 2'S COMPLE. ANGEL (IN BUF) INTO 1'S

# OVEFLOW HERE IF 2'S COMPL. IS 180 DEG.

AD TEM2  
 CCS A  
 AD ONE  
 TCF +2  
 COM

# SULT MOVES FROM 2ND TO 3D QUAD. (OR BACK)

# BACK TO 2'S COMPL.



July 12, 2016

1681

|       |        |                                           |
|-------|--------|-------------------------------------------|
| TS    | TEM2   | # STORE 14-BIT QUANTITY WITH PRESENT SIGN |
| TCF   | +4     |                                           |
| INDEX | A      | # SIGN.                                   |
| CAF   | LIMITS | # FIX IT, BY ADDING IN 37777 OR 40000     |
| AD    | TEM2   |                                           |
| INDEX | BUF    |                                           |
| TS    | 0      | # STORE NEW ANGLE IN 2'S COMPLEMENT.      |
| TC    | Q      |                                           |

# Page 1512

# RTB TO TORQUE GYROS, EXCEPT FOR THE CALL TO IMUSTALL. ECADR OF COMMANDS ARRIVES IN X1.

|          |       |          |                                            |
|----------|-------|----------|--------------------------------------------|
| PULSEIMU | INDEX | FIXLOC   | # ADDRESS OF GYRO COMMANDS SHOULD BE IN X1 |
|          | CA    | X1       |                                            |
|          | TC    | BANKCALL |                                            |
|          | CADR  | IMUPULSE |                                            |
|          | TCF   | DANZIG   |                                            |

# Page 1513

# EACH ROUTINE TAKES A 3X3 MATRIX STORED IN DOUBLE PRECISION IN A FIXED AREA OF ERASABLE MEMORY  
# WITH THE TRANSPOSE MATRIX. TRANSP1 USES LOCATIONS XNB+0,+1 THROUGH XNB+16D,+17D AND TRANSP2  
# XNB1+0,+1 THROUGH XNB1+16D,+17D. EACH MATRIX IS STORED BY ROWS.

|         |        |          |
|---------|--------|----------|
| XNBEB   | ECADR  | XNB      |
| XNB1EB  | ECADR  | XNB1     |
|         | EBANK= | XNB      |
| TRANSP1 | CAF    | XNBEB    |
|         | TS     | EBANK    |
|         | DXCH   | XNB +2   |
|         | DXCH   | XNB +6   |
|         | DXCH   | XNB +2   |
|         | DXCH   | XNB +4   |
|         | DXCH   | XNB +12D |
|         | DXCH   | XNB +4   |
|         | DXCH   | XNB +10D |
|         | DXCH   | XNB +14D |
|         | DXCH   | XNB +10D |
|         | TCF    | DANZIG   |
|         | EBANK= | XNB1     |
| TRANSP2 | CAF    | XNB1EB   |

```

TS EBANK
DXCH XNB1 +2
DXCH XNB1 +6
DXCH XNB1 +2

DXCH XNB1 +4
DXCH XNB1 +12D
DXCH XNB1 +4

DXCH XNB1 +10D
DXCH XNB1 +14D
DXCH XNB1 +10D
TCF DANZIG

```

# Page 1514

# THE SUBROUTINE SIGNMPAC SETS C(MPAC, MPAC +1) TO SIGN(MPAC).

# FOR THIS, ONLY THE CONTENTS OF MPAC ARE EXAMINED. ALSO +0 YIELDS POSMAX AND -0 YI

#

# ENTRY MAY BE BY EITHER OF THE FOLLOWING:

# 1. LIMIT THE SIZE OF MPAC ON INTERPRETIVE OVERFLOW:

# ENTRY: BOVB

#

SIGNMPAC

# 2. GENERATE IN MPAC THE SIGNUM FUNCTION OF MPAC:

# ENTRY: RTB

#

SIGNMPAC

# IN EITHER CASE, RETURN IS TO THE NEXT INTERPRETIVE INSTRUCTION IN THE CALLING SEQU

SIGNMPAC EXTEND

DCA DPOSMAX

DXCH MPAC

CCS A

DPMODE CAF ZERO # SETS MPAC +2 TO ZERO IN THE PROCESS

TCF SLOAD2 +2

TCF +1

EXTEND

DCS DPOSMAX

TCF SLOAD2

# RTB OP CODE NORMUNIT IS LIKE INTERPRETIVE INSTRUCTION UNIT, EXCEPT THAT IT CAN BE I

# UP WHEN THE VECTOR BEING UNITIZED IS VERY SMALL -- IT WILL BLOW UP WHEN ALL COMPON

# IS USED AND THE UPPER ORDER HALVES OF ALL COMPONENTS ARE ZERO, THE MAGNITUDE RETUR

# BY A FACTOR OF 2(13) AND THE SQUARED MAGNITUDE RETURNED AT 34D WILL BE TOO BIG BY A

NORMUNX1 CAF ONE

TCF NORMUNIT +1

NORMUNIT CAF ZERO

July 12, 2016

1683

```
AD FIXLOC
TS MPAC +2
TC BANKCALL # GET SIGN AGREEMENT IN ALL COMPONENTS
CADR VECAGREE
CCS MPAC
TCF NOSHIFT
TCF +2
TCF NOSHIFT
CCS MPAC +3
TCF NOSHIFT
TCF +2
TCF NOSHIFT
CCS MPAC +5
TCF NOSHIFT
TCF +2
TCF NOSHIFT

Page 1515
CA MPAC +1 # SHIFT ALL COMPONENTS LEFT 13
EXTEND
MP BIT14
DAS MPAC # DAS GAINS A LITTLE ACCURACY
CA MPAC +4
EXTEND
MP BIT14
DAS MPAC +3
CA MPAC +6
EXTEND
MP BIT14
DAS MPAC +5
CAF THIRTEEN
INDEX MPAC +2
TS 37D
OFFTUNIT TC POSTJUMP
CADR UNIT +1 # SKIP THE "TC VECAGREE" DONE AT UNIT

NOSHIFT CAF ZERO
TCF OFFTUNIT -2

RTB VECSGNAG ... FORCES SIGN AGREEMENT OF VECTOR IN MPAC.

VECSGNAG TC BANKCALL
 CADR VECAGREE
 TC DANZIG
```

# Page 1516

# MODULE CHANGE FOR NEW LUNAR GRAVITY MODEL

|          |        |          |    |                                           |
|----------|--------|----------|----|-------------------------------------------|
|          | SETLOC | MODCHG3  |    |                                           |
|          | BANK   |          |    |                                           |
| QUALITY1 | BOF    | DLOAD    |    |                                           |
|          |        | MOONFLAG |    |                                           |
|          |        | NBRANCH  |    |                                           |
|          |        | URPV     |    |                                           |
|          | DSQ    | GOTO     |    |                                           |
|          |        | QUALITY2 |    |                                           |
|          | SETLOC | MODCHG2  |    |                                           |
|          | BANK   |          |    |                                           |
| QUALITY2 | PDDL   | DSQ      |    | # SQUARE INTO 2D, B2                      |
|          |        | URPV     | +2 | # Y COMPONENT, B1                         |
|          | DSU    |          |    |                                           |
|          | DMP    | VXSC     |    | # 5(Y**2-X**2)UR                          |
|          |        | 5/8      |    | # CONSTANT, 5B3                           |
|          |        | URPV     |    | # VECTOR, RESULT MAXIMUM IS 5, SCALING    |
|          |        |          |    | # HERE B6                                 |
|          | VSL3   | PDDL     |    | # STORE SCALED B3 IN 2D, 4D, 6D FOR XYZ   |
|          |        | URPV     |    | # X COMPONENT, B1                         |
|          | SR1    | DAD      |    | # 2 X X COMPONENT FOR B3 SCALING          |
|          |        | 2D       |    | # ADD TO VECTOR X COMPONENT OF ANSWER.    |
|          |        |          |    | # SAME AS MULTIPLYING BY UNITX. MAX IS 7. |
|          | STODL  | 2D       |    |                                           |
|          |        | URPV     | +2 | # Y COMPONENT, B1                         |
|          | SR1    | BDSU     |    | # 2 X Y COMPONENT FOR B3 SCALING          |
|          |        | 4D       |    | # SUBTRACT FROM VECTOR Y COMPONENT OF     |
|          |        |          |    | # ANSWER, SAME AS MULTIPLYING BY UNITY.   |
|          |        |          |    | # MAX IS 7.                               |
|          | STORE  | 4D       |    | # 2D HAS VECTOR, B3.                      |
|          | SLOAD  | VXSC     |    | # MULTIPLY COEFFICIENT TIMES VECTOR IN 2D |
|          |        | E3J22R2M |    |                                           |
|          | PDDL   | RVQ      |    | # J22 TERM X R**4 IN 2D, SCALED B61       |
|          |        | COSPHI/2 |    | # SAME AS URPV +4, Z COMPONENT            |

# \*\*\* END OF CHIEFTAN.028 \*\*\*

This code is written to file src/RT8-OP-CODES.s.

## B.98 RTB OP CODES

1685

*<src/RTB-OP-CODES.s 1685>*≡

```
Copyright: Public domain.
Filename: RTB_OP_CODES.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1397-1401
Mod history: 2009-05-10 SN (Sergio Navarro). Started adapting
from the Luminary131/ file of the same
name, using Luminary099 page images.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
```

# Page 1397

```
BANK 22
SETLOC RTBCODES
BANK
```

```
EBANK= XNB
COUNT* $$/RTB
```

# LOAD TIME2, TIME1 INTO MPAC:

```
LOADTIME EXTEND
 DCA TIME2
 TCF SLOAD2
```

# CONVERT THE SINGLE PRECISION 2'S COMPLEMENT NUMBER ARRIVING IN MPAC (SCALED IN HALF-REVOLUTION

# DP 1'S COMPLEMENT NUMBER SCALED IN REVOLUTIONS.

|          |        |         |                                       |
|----------|--------|---------|---------------------------------------|
| CDULOGIC | CCS    | MPAC    |                                       |
|          | CAF    | ZERO    |                                       |
|          | TCF    | +3      |                                       |
|          | NOOP   |         |                                       |
|          | CS     | HALF    |                                       |
|          | TS     | MPAC +1 |                                       |
|          | CAF    | ZERO    |                                       |
|          | XCH    | MPAC    |                                       |
|          | EXTEND |         |                                       |
|          | MP     | HALF    |                                       |
|          | DAS    | MPAC    |                                       |
|          | TCF    | DANZIG  | # MODE IS ALREADY AT DOUBLE-PRECISION |

# FORCE TP SIGN AGREEMENT IN MPAC:

|          |     |         |
|----------|-----|---------|
| SGNAGREE | TC  | TPAGREE |
|          | TCF | DANZIG  |

# CONVERT THE DP 1'S COMPLEMENT ANGLE SCALED IN REVOLUTIONS TO A SINGLE PRECISION 2'S  
# SCALED IN HALF-REVOLUTIONS.

|        |     |         |
|--------|-----|---------|
| 1ST02S | TC  | 1T02SUB |
|        | CAF | ZERO    |
|        | TS  | MPAC +1 |
|        | TCF | NEWMODE |

# DO 1ST02S ON A VECTOR OF ANGLES:

|             |      |         |                                 |
|-------------|------|---------|---------------------------------|
| V1ST02S     | TC   | 1T02SUB | # ANSWER ARRIVES IN A AND MPAC. |
|             | DXCH | MPAC +5 |                                 |
|             | DXCH | MPAC    |                                 |
|             | TC   | 1T02SUB |                                 |
| # Page 1398 | TS   | MPAC +2 |                                 |
|             | DXCH | MPAC +3 |                                 |
|             | DXCH | MPAC    |                                 |
|             | TC   | 1T02SUB |                                 |
|             | TS   | MPAC +1 |                                 |
|             | CA   | MPAC +5 |                                 |
|             | TS   | MPAC    |                                 |

July 12, 2016

1687

```
TPMODE CAF ONE # MODE IS TP.
 TCF NEWMODE
```

# V1STO2S FOR 2 COMPONENT VECTOR. USED BY RR.

```
2V1STO2S TC 1TO2SUB
 DXCH MPAC +3
 DXCH MPAC
 TC 1TO2SUB
 TS L
 CA MPAC +3
 TCF SLOAD2
```

# SUBROUTINE TO DO DOUBLING AND 1'S TO 2'S CONVERSION:

```
1TO2SUB DXCH MPAC # FINAL MPAC +1 UNSPECIFIED.
 DDOUBL
 CCS A
 AD ONE
 TCF +2
 COM # THIS WAS REVERSE OF MSU.

 TS MPAC # AND SKIP ON OVERFLOW.
 TC Q

 INDEX A # OVERFLOW UNCORRECT AND IN MSU.
 CAF LIMITS
 ADS MPAC
 TC Q
```

# THE FOLLOWING ROUTINE INCREMENTS IN 2S COMPLEMENT THE REGISTER WHOSE ADDRESS IS IN BUF BY THE  
# QUANTITY FOUND IN TEM2. THIS MAY BE USED TO INCRMENT DESIRED IMU AND OPTICS CDU ANGLES OR AN  
# (+0 UNEQUAL TO -0) QUANTITY. MAY BE CALLED BY BANKCALL/SWCALL.

```
CDUINC TS TEM2 # 1S COMPL.QUANT. ARRIVES IN ACC. STORE IT
 INDEX BUF
 CCS 0 # CHANGE 2S COMPL. ANGLE(IN BUF)INTO 1S
 AD ONE
 TCF +4
 AD ONE

Page 1399

 AD ONE # OVEFLOW HERE IF 2S COMPL. IS 180 DEG.
 COM

 AD TEM2 # SULT MOVES FROM 2ND TO 3D QUAD. (OR BACK)
```

```

CCS A # BACK TO 2S COMPL.
AD ONE
TCF +2
COM
TS TEM2 # STORE 14BIT QUANTITY WITH PRESENT SIGN
TCF +4
INDEX A # SIGN.
CAF LIMITS # FIX IT, BY ADDING IN 37777 OR 40000
AD TEM2

INDEX BUF
TS 0 # STORE NEW ANGLE IN 2S COMPLEMENT.
TC Q

```

# Page 1400

# RTB TO TORQUE GYROS, EXCEPT FOR THE CALL TO IMUSTALL. ECADR OF COMMANDS ARRIVES IN

```

PULSEIMU INDEX FIXLOC # ADDRESS OF GYRO COMMANDS SHOULD BE IN X1
CA X1
TC BANKCALL
CADR IMUPULSE
TCF DANZIG

```

# Page 1401

# THE SUBROUTINE SIGNMPAC SETS C(MPAC, MPAC +1) TO SIGN(MPAC).

# FOR THIS, ONLY THE CONTENTS OF MPAC ARE EXAMINED. ALSO +0 YIELDS POSMAX AND -0 YIELD

#

# ENTRY MAY BE BY EITHER OF THE FOLLOWING:

# 1. LIMIT THE SIZE OF MPAC ON INTERPRETIVE OVERFLOW:

# ENTRY: BOVB

# SIGNMPAC

# 2. GENERATE IN MPAC THE SIGNUM FUNCTION OF MPAC:

# ENTRY: RTB

# SIGNMPAC

# IN EITHER CASE, RETURN IS TO THE NEXT INTERPRETIVE INSTRUCTION IN THE CALLING SEQU

```

SIGNMPAC EXTEND
DCA DPOSMAX
DXCH MPAC
CCS A
DPMODE CAF ZERO # SETS MPAC +2 TO ZERO IN THE PROCESS
TCF SLOAD2 +2
TCF +1
EXTEND
DCS DPOSMAX
TCF SLOAD2

```



# RTB OP CODE NORMUNIT IS LIKE INTERPRETIVE INSTRUCTION UNIT, EXCEPT THAT IT CAN BE DEPENDED ON  
 # UP WHEN THE VECTOR BEING UNITIZED IS VERY SMALL -- IT WILL BLOW UP WHEN ALL COMPONENT ARE ZERO  
 # IS USED AND THE UPPER ORDER HALVES OF ALL COMPONENTS ARE ZERO, THE MAGNITUDE RETURNED IN 36D  
 # BY A FACTOR OF 2(13) AND THE SQUARED MAGNITUDE RETURNED ATE 34D WILL BE TOO BIG BY A FACTOR C

|             |        |             |                                        |
|-------------|--------|-------------|----------------------------------------|
| NORMUNX1    | CAF    | ONE         |                                        |
|             | TCF    | NORMUNIT +1 |                                        |
| NORMUNIT    | CAF    | ZERO        |                                        |
|             | AD     | FIXLOC      |                                        |
|             | TS     | MPAC +2     |                                        |
|             | TC     | BANKCALL    | # GET SIGN AGREEMENT IN ALL COMPONENTS |
|             | CADR   | VECAGREE    |                                        |
|             | CCS    | MPAC        |                                        |
|             | TCF    | NOSHIFT     |                                        |
|             | TCF    | +2          |                                        |
|             | TCF    | NOSHIFT     |                                        |
|             | CCS    | MPAC +3     |                                        |
|             | TCF    | NOSHIFT     |                                        |
|             | TCF    | +2          |                                        |
|             | TCF    | NOSHIFT     |                                        |
|             | CCS    | MPAC +5     |                                        |
|             | TCF    | NOSHIFT     |                                        |
|             | TCF    | +2          |                                        |
|             | TCF    | NOSHIFT     |                                        |
| # Page 1402 |        |             |                                        |
|             | CA     | MPAC +1     | # SHIFT ALL COMPONENTS LEFT 13         |
|             | EXTEND |             |                                        |
|             | MP     | BIT14       |                                        |
|             | DAS    | MPAC        | # DAS GAINS A LITTLE ACCURACY          |
|             | CA     | MPAC +4     |                                        |
|             | EXTEND |             |                                        |
|             | MP     | BIT14       |                                        |
|             | DAS    | MPAC +3     |                                        |
|             | CA     | MPAC +6     |                                        |
|             | EXTEND |             |                                        |
|             | MP     | BIT14       |                                        |
|             | DAS    | MPAC +5     |                                        |
|             | CAF    | THIRTEEN    |                                        |
|             | INDEX  | MPAC +2     |                                        |
|             | TS     | 37D         |                                        |
| OFFTUNIT    | TC     | POSTJUMP    |                                        |
|             | CADR   | UNIT +1     | # SKIP THE "TC VECAGREE" DONE AT UNIT  |
| NOSHIFT     | CAF    | ZERO        |                                        |
|             | TCF    | OFFTUNIT -2 |                                        |

```
RTB VECSGNAG ... FORCES SIGN AGREEMENT OF VECTOR IN MPAC.
```

```
VECSGNAG TC BANKCALL
 CADR VECAGREE
 TC DANZIG
```

```
*** END OF SKIPPER .087 ***
```

This code is written to file `src/RTB-OP-CODES.s`.

## B.99 S-BAND ANTENNA FOR CM

```

1691 <src/S-BAND-ANTENNA-FOR-CM.s 1691>≡
Copyright: Public domain.
Filename: S-BAND_ANTENNA_FOR_CM.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Jim Lawton <jim.lawton@gmail.com>.
Website: www.ibiblio.org/apollo.
Pages: 934-935
Mod history: 2009-05-11 JVL Adapted from the Colossus249/ file
of the same name, using Comanche055 page
images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 934
S-BAND ANTENNA FOR CM

BANK 23
SETLOC SBAND
BANK

COUNT* $$/R05
EBANK= EMSALT

SBANDANT TC BANKCALL # V 64 E GETS US HERE
 CADR R02BOTH # CHECK IF IMU IS ON AND ALIGNED
 TC INTPRET

```

|            |        |          |                                           |
|------------|--------|----------|-------------------------------------------|
|            | RTB    | CALL     |                                           |
|            |        | LOADTIME | # PICKUP CURRENT TIME SCALED B-28         |
|            |        | CDUTRIG  | # COMPUTE SINES AND COSINES OF CDU ANGLES |
|            | STCALL | TDEC1    | # ADVANCE INTEGRATION TO TIME IN TDEC1    |
|            |        | CSMCONIC | # USING CONIC INTEGRATION                 |
|            | SLOAD  | BHIZ     | # ORIGIN OF REFERENCE INERTIAL SYSTEM IS  |
|            |        | X2       | # EARTH = 0, MOON = 2                     |
|            |        | EISOI    |                                           |
|            | VLOAD  |          |                                           |
|            |        | RATT     |                                           |
|            | STORE  | RCM      | # MOVE RATT TO PREVENT WIPEOUT            |
|            | DLOAD  | CALL     | # MOON, PUSH ON                           |
|            |        | TAT      | # GET ORIGINAL TIME                       |
|            |        | LUNPOS   | # COMPUTE POSITION VECTOR OF MOON         |
|            | VAD    | VCOMP    | # R= -(REM+RCM) = NEG. OF S/C POS. VEC    |
|            |        | RCM      |                                           |
|            | GOTO   |          |                                           |
|            |        | EISOI +2 |                                           |
| EISOI      | VLOAD  | VCOMP    | # EARTH, R= -RCM                          |
|            |        | RATT     |                                           |
|            | SETPD  | MXV      | # RCS TO STABLE MEMBER: B-1X B-29X B+1    |
|            |        | 2D       | # 2D                                      |
|            |        | REFSMMAT | # STABLE MEMBER. B-1X B-29X B+1= B-29     |
|            | VSL1   | PDDL     | # 8D                                      |
|            |        | HI6ZEROS |                                           |
|            | STOVL  | YAWANG   | # ZERO OUT YAWANG, SET UP FOR SMNB        |
|            |        | RCM      | # TRANSFORMATION. SM COORD. SCALED B-29   |
|            | CALL   |          |                                           |
|            |        | *SMNB*   |                                           |
|            | STORE  | R        | # SAVE NAV. BASE COORDINATES              |
|            | UNIT   | PDVL     | # 14D                                     |
|            |        | R        |                                           |
|            | VPROJ  | VSL2     | # COMPUTE PROJECTION OF VECTOR INTO CM    |
|            |        | HIUNITZ  | # XY-PLANE, R-(R.UZ)UZ                    |
|            | BVSU   | BOV      | # CLEAR OVERFLOW INDICATOR IF SET         |
|            |        | R        |                                           |
|            |        | COVCNV   |                                           |
| COVCNV     | UNIT   | BOV      | # TEST OVERFLOW FOR INDICATION OF NULL    |
|            |        | NOADJUST | # VECTOR                                  |
|            | PUSH   | DOT      | # 20D                                     |
| # Page 935 |        |          |                                           |
|            |        | HIUNITX  | # COMPUTE YAW ANGLE = ACOS (URP.UX)       |
|            | SL1    | ACOS     | # REVOLUTIONS SCALED BO                   |
|            | PDVL   | DOT      | # 22D YAWANG                              |
|            |        | URP      |                                           |
|            |        | HIUNITY  | # COMPUTE FOLLOWING: URP.UY               |

|          |        |             |                                          |
|----------|--------|-------------|------------------------------------------|
|          | SL1    | BPL         | # POSITIVE                               |
|          |        | NOADJUST    | # YES, 0-180 DEGREES                     |
|          | DLOAD  | DSU         | # NO, 181-360 DEGREES 20D                |
|          |        | DPPOSMAX    | # COMPUTE 2 PI MINUS YAW ANGLE           |
|          | PUSH   |             | # 22D YAWANG                             |
| NOADJUST | VLOAD  | DOT         | # COMPUTE PITCH ANGLE                    |
|          |        | UR          | # ACOS (UR.UZ) - PI/2                    |
|          |        | HIUNITZ     |                                          |
|          | SL1    | ACOS        | # REVOLUTIONS BO                         |
|          | DSU    |             |                                          |
|          |        | HIDP1/4     |                                          |
|          | STODL  | RHOSB       |                                          |
|          |        | YAWANG      |                                          |
|          | STORE  | GAMMASB     | # PATCH FOR CHECKOUT                     |
|          | EXIT   |             |                                          |
|          | CA     | EXTVBACT    | # IS BIT 5 STILL ON                      |
|          | MASK   | BIT5        |                                          |
|          | EXTEND |             |                                          |
|          | BZF    | ENDEXT      | # NO, WE HAVE BEEN ANSWERED              |
|          | CAF    | V06N51      | # DISPLAY ANGLES                         |
|          | TC     | BANKCALL    |                                          |
|          | CADR   | GOMARKFR    |                                          |
|          | TC     | B5OFF       | # TERMINATE                              |
|          | TC     | B5OFF       |                                          |
|          | TC     | ENDOFJOB    | # RECYCLE                                |
|          | CAF    | BIT3        | # IMMEDIATE RETURN                       |
|          | TC     | BLANKET     | # BLANK R3                               |
|          | CAF    | BIT1        | # DELAY MINIMUM TIME TO ALLOW DISPLAY IN |
|          | TC     | BANKCALL    |                                          |
|          | CADR   | DELAYJOB    |                                          |
|          | TCF    | SBANDANT +2 |                                          |
| V06N51   | VN     | 0651        |                                          |
| RCM      | EQUALS | 2D          |                                          |
| UR       | EQUALS | 8D          |                                          |
| URP      | EQUALS | 14D         |                                          |
| YAWANG   | EQUALS | 20D         |                                          |
| PITCHANG | EQUALS | 22D         |                                          |
| R        | EQUALS | RCM         |                                          |
|          | SBANK= | LOWSUPER    |                                          |

This code is written to file `src/S-BAND-ANTENNA-FOR-CM.s`.

## B.100 S-BAND ANTENNA FOR LM

```

1694 <src/S-BAND-ANTENNA-FOR-LM.s 1694>≡
Copyright: Public domain.
Filename: S-BAND_ANTENNA_FOR_LM.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 486-489
Mod history: 2009-05-17 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
2009-06-07 RSB Corrected a misprint.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 486
SUBROUTINE NAME: R05 -- S-BAND ANTENNA FOR LM
#
MOD0 BY T. JAMES
MOD1 BY P. SHAKIR
#
FUNCTIONAL DESCRIPTION
#
THE S-BAND ANTENNA ROUTINE, R05, COMPUTES AND DISPLAYS THE PITCH AND
YAW ANTENNA GIMBAL ANGLES REQUIRED TO POINT THE LM STEERABLE ANTENNA
TOWARD THE CENTER OF THE EARTH. THIS ROUTINE IS SELECTED BY THE ASTRO-
NAUT VIA DSKY ENTRY DURING COASTING FLIGHT OR WHEN THE LM IS ON THE MOON
SURFACE. THE EARTH OR MOON REFERENCE COORDINATE SYSTEM IS USED DEPENDING
ON WHETHER THE LM IS ABOUT TO ENTER OR HAS ALREADY ENTERED THE MOON

```

July 12, 2016

1695

```
SPHERE OF INFLUENCE, RESPECTIVELY.
#
TO CALL SUBROUTINE, ASTRONAUT KEYS IN V 64 E
#
SUBROUTINES CALLED ---
R02BOTH
INTPRET
LOADTIME
LEMCONIC
LUNPOS
CDUTRIG
SMNB
BANKCALL
B500FF
ENDOFJOB
BLANKRET
#
RETURNS WITH
PITCH ANGLE IN PITCHANG REV. BO
YAW ANGLE IN YAWANG REV. BO
#
ERASABLES USED
PITCHANG
YAWANG
RLM
VAC AREA

 BANK 41
 SETLOC SBAND
 BANK

 EBANK= WHOCARES
 COUNT* $$/R05
SBANDANT TC BANKCALL
Page 487

 CADR R02BOTH # CHECK IF IMU IS ON AND ALIGNED
 TC INTPRET
 SETPD RTB
 OD
 LOADTIME # PICK UP CURRENT TIME
 STCALL TDEC1 # ADVANCE INTEGRATION TO TIME IN TDEC1
 LEMCONIC # USING CONIC INTEGRATION
 SLOAD BHIZ
 X2 # X2 =0 EARTH SPHERE, X2 =2 MOON SPHERE
 CONV4
 VLOAD
```

```

 RATT
 RLM
 TAT
CONV3 CALL LUNPOS # UNIT POSITION VECTOR FROM EARTH TO MOON
 VXSC
 VMOON
 REMDIST # MEAN DISTANCE FROM EARTH TO MOON
 VAD
 RLM
 GOTO
CONV4 VLOAD CONV5
CONV5 SETPD UNIT # UE = -UNIT(RATT) EARTH SPHERE
 OD # UE = -UNIT((REM)(UEM) + RL) MOON SPHERE
 VCOMP # SET PL POINTER TO 0
 CDUTRIG # COMPUTE SINES AND COSINES OF CDU ANGLES
 MXV # TRANSFORM REF. COORDINATE SYSTEM TO
 REFMMAT # STABLE MEMBER B-1 X B-1 X B+1 = B-1
 PUSH # 8D
 HI6ZEROS
 STORE PITCHANG
 STOVL YAWANG # ZERO OUT ANGLES
 CALL
 SMNB
 STODL RLM # PRE-MULTIPLY RLM BY (NBSA) MATRIX(BO)
 RLM +2
 PUSH DSU
 RLM
 DMP
 10VSQRT2
 STODL RLM +2
 DAD DMP
 RLM
 10VSQRT2
 STOVL RLM # R B-1
 RLM
 UNIT PDVL
 RLM
 VPROJ VSL2 # PROJECTION OF R ONTO LM XZ PLANE.
 HIUNITY
 BVSU BOV # CLEAR OVERFLOW INDICATOR IF ON
 RLM
 COVCNV

```



July 12, 2016

1697

|            |        |          |                                   |        |
|------------|--------|----------|-----------------------------------|--------|
| COVCNV     | UNIT   | BOV      | # EXIT ON OVERFLOW                |        |
|            |        | SBANDEX  |                                   |        |
|            | PUSH   | VXV      | # URP VECTOR B-1                  |        |
|            |        | HIUNITZ  |                                   |        |
|            | VSL1   | VCOMP    | # UZ X URP = -(URP X UZ)          |        |
|            | STORE  | RLM      | # X VEC B-1                       |        |
|            | DOT    | PDVL     | # SGN(X.UY) UNSCALED              |        |
|            |        | HIUNITY  |                                   |        |
|            |        | RLM      |                                   |        |
|            | ABVAL  | SIGN     |                                   |        |
|            | ASIN   |          | # ASIN((SGN(X.UY))ABV(X))         | REV BO |
|            | STOVL  | PITCHANG |                                   |        |
|            |        | URP      |                                   |        |
|            | DOT    | BPL      |                                   |        |
|            |        | HIUNITZ  |                                   |        |
|            |        | NOADJUST | # YES, -90 TO +90                 |        |
|            | DLOAD  | DSU      |                                   |        |
|            |        | HIDPHALF |                                   |        |
|            |        | PITCHANG |                                   |        |
|            | STORE  | PITCHANG |                                   |        |
| NOADJUST   | VLOAD  | VXV      |                                   |        |
|            |        | UR       | # Z = (UR X URP)                  |        |
|            |        | URP      |                                   |        |
|            | VSL1   |          |                                   |        |
|            | STODL  | RLM      | # Z VEC B-1                       |        |
|            |        | PITCHANG |                                   |        |
|            | SIN    | VXSC     |                                   |        |
|            |        | HIUNITZ  |                                   |        |
|            | PDDL   | COS      |                                   |        |
|            |        | PITCHANG |                                   |        |
|            | VXSC   | VSU      |                                   |        |
|            |        | HIUNITX  | # (UX COS ALPHA) - (UZ SIN ALPHA) |        |
|            | DOT    | PDVL     | # YAW.Z                           |        |
|            |        | RLM      |                                   |        |
|            |        | RLM      |                                   |        |
|            | ABVAL  | SIGN     |                                   |        |
|            | ASIN   |          |                                   |        |
|            | STORE  | YAWANG   |                                   |        |
| SBANDEX    | EXIT   |          |                                   |        |
|            | CA     | EXTVBACT |                                   |        |
|            | MASK   | BIT5     | # IS BIT5 STILL ON                |        |
|            | EXTEND |          |                                   |        |
|            | BZF    | ENDEXT   | # NO                              |        |
|            | CAF    | PRI05    |                                   |        |
| # Page 489 | TC     | PRI0CHNG |                                   |        |

```
CAF V06N51 # DISPLAY ANGLES
TC BANKCALL
CADR GOMARKFR
TC B5OFF # TERMINATE
TC B5OFF # PROCEED
TC ENDOFJOB # RECYCLE
CAF BIT3 # IMMEDIATE RETURN
TC BLANKET # BLANK R3
CAF PRIO4
TC PRIOCHNG
TC SBANDANT +2 # YES, CONTINUE DISPLAYING ANGLES.
V06N51 VN 0651
10VSQRT2 2DEC .7071067815 # 1/SQRT(2)

UR EQUALS 0D
URP EQUALS 6D
SBANK= LOWSUPER
```

```
END OF LNYAIDE .001 ***
```

This code is written to file `src/S-BAND-ANTENNA-FOR-LM.s`.

**B.101    SERVICER207**

1699

*<src/SERVICER207.s 1699>*≡

```

Copyright: Public domain.
Filename: SERVICER207.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 819-836
Contact: Ron Burkey <info@sandroid.org>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-05-12 RSB Adapted from Colossus249 file of the same
name and Comanche 055 page images.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
#
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.
#
Page 819
SERVICER207
#
PROGRAM NAME: PREREAD, READACCS, SERVICER, AVERAGE G.
MOD NO. 00 BY M. HAMILTON DEC. 12, 1966
#
FUNCTIONAL DESCRIPTION
#
THE ROUTINES DESCRIBED BELOW ARE USED TO CALCULATE VALUES OF RN, VN, AND GDT/2 DURING ACCELER
THE SEVERAL ROUTINES COMPRISE A PACKAGE AND ARE NOT MEANT TO BE USED AS SEPARATE SUBROUTINES.
#
GENERAL REFERENCES TO SERVICER OR AVERAGE G ARE UNDERSTOOD TO REFER TO THE ENTIRE SET OF
RRADACCS, SERVICER, AVERAGE G, INTEREAD, SMOOTHER, AND ANY ADDITIONAL ROUTINES ATTACHED AT AV

```

```

#
PROGRAMS INITIATING SERVICER ARE REQUIRED TO MAKE A WAITLIST CALL FOR PREREAD (OR,
AT 2 SECONDS BEFORE THE FIRST AVERAGE G UPDATE IN ORDER TO INITIALIZE THE SEQUENCE.
2 SECONDS FROM THAT TIME ON AS LONG AS AVEGFLAG REMAINS SET.
#
THE USE OF ERASABLE AVGEXIT ALLOWS VARIOUS ROUTINES TO BE PERFORMED AS PART OF THE
EXPLANATION OF AVGEXIT BELOW).
#
DESCRIPTIONS OF INDIVIDUAL ROUTINES FOLLOW.
#
PREREAD
#
PREVIOUSLY EXTRAPOLATED VALUES COPIED FROM RN1, VN1, AND PIPTIME1 INTO
LASTBIAS JOB SCHEDULED.
PIPS READ AND CLEARED VIA PIPASR SUBROUTINE.
AVERAGE G FLAG SET ON.
DRIFT FLAG SET OFF.
V37 FLAG SET ON.
INITIALIZATION OF 1) THRUST MONITOR (DVMON) -- DVCNTR SET TO ON
2) TOTAL ACCUMULATED DELV VALUE (DVTOTAL) --
3) AXIS VECTOR (AXIS) -- SET TO (.5,0,0).
NORMLIZE JOB SCHEDULED.
READACCS TASH CALLED IN 2 SECONDS.
#
NORMLIZE
#
GDT/2 INITIALIZED VIA CALCGRAV ROUTINE.
#
READACCS
#
IF ONMON FLAG SET QUICKREAD ROUTINE IS PERFORMED BEFORE PIPASR ZEROS 7
ONMONITOR LOOP IS INITIATED TO PROVIDE DOWNLINK INFORMATION DURING E
PIPS READ AND CLEARED BY PIPASR SUBROUTINE.
IF CM/DSTBY IS ON, ENTRY VARIABLES INITIALIZED AND SETJTAG TASK CALLED
Page 820
#
SERVICER207
#
IF AVERAGEG FLAG ON READACCS CALLED TO RECYCLE IN 2 SECONDS.
IF AVERAGEG FLAG OFF AVERAGE G EXIT (AVGEXIT) SET TO 2CADR AVGEND
SERVICER JOB SCHEDULED.
TEST CONNECTOR OUTBIT TURNED ON.
#
ONMNITOR
#
A SEQUENCE OF THREE PASSES THROUGH QUICKREAD FOLLOWING A CALL TO READ

```

July 12, 2016

1701

```
SEC INTERVALS. INTERVALS ARE COUNTED OUT BY PIPCTR, INITIALIZED AT 3 BY READAC
#
QUIKREAD
#
READS CURRENT PIPS INTO X,Y,ZPIPBUFF. READS OLD X,Y,ZPIPBUFF INTO X,Y,ZOLDBUFF.
DOWNLIST DURING ENTRY.
#
SERVICER
#
DELV VALUES CHECKED TO DETECT RUNAWAY PIP:
IF BAD PIP 1) ALARM SENT.
2) COMPENSATION, DVTOTAL ACCUMULATION, AND DVMON BYPASS
TRANSFERRED TO AVERAGE G.
PIPS COMPENSATED VIA 1/PIPA SUBROUTINE.
DVTOTAL INCREMENTED BY ABSOLUTE VALUE OF DELV.
THRUST MONITOR (DVMON) PERFORMED UNLESS IDLE FLAG IS ON.
CONTROL TRANSFERRED TO AVERAGE Q.
#
DVMON
#
THRESHOLD VALUD (PLACED IN DVTHRUSH BY USER) CHECKED AGAINST ABSOLUTE VALUE OF
THRUST LEVEL.
#
IF THRUST 1) ULLAGE OFF ROUTINE PERFORMED.
2) STEERING FLAG TURNED ON AT FIRST DETECTION OF THRUST
3) CONTROL TRANSFERRED TO AVERAGE G.
IF NO THRUST 1) ON FIRST PASS THROUGH MONITOR, CONTROL TRANSFERRED T
2) ON SUBSEQUENT PASSES, CONTROL TRANSFERRED TO ENGINE
HAS FAILED FOR 3 CONSECUTIVE PASSES.
#
ENGINE FAIL
#
ENGFALL1 TASK CALLED IN 2.5 SECONDS. THIS WILL RETURN CONTROL TO TIG-5 SO THAT
SEQUENCE MAY BE REPEATED.
ENGINOF3 PERFORMED.
DAP SET UP FOR RCS.
#
AVERAGE G
Page 821
RN1, VN1, GDT1/2 CALCULATED VIA CALCRVG ROUTINE BY UPDATING RN, VN WITH DELV AN
OF GDT/2
RN1, VN1, GDT1/2, PIPTIME1 COPIED INTO RN, VN, GDT/2, PIPTIME FOR RESTART PROTE
CONTROL TRANSFERRED TO ADDRESS SPECIFIED BY USER (OR BY READACCS FOR LAST PASS)
LAST PASS (AVGEND) 1) FREE FALL GYRO COMPENSATION SET UP.
2) DRIFT FLAG TURNED ON.
3) STATE VECTOR TRANSFERRED VIA AVETOMID ROUTINE.
```

```

4) ONMONITOR FLAG RESET.
5) V37 FLAG RESET.
6) TEST CONNECTOR OUTBIT RESET.
7) CONTROL TRANSFERRED TO CANV37 TO CONTINUE
#
CALLING SEQUENCE
#
PREREAD ENTERED DIRECTLY FROM TIG-30 VIA POSTJUMP.
READACCS CALLED AS WAITLIST TASK.
#
SUBROUTINES CALLED
#
UTILITY ROUTINES: PHASCHNG FLAGUP FLAGDOWN NOVAC FINDVAC WAITLIST ALARM NEW
#
OTHER: PIPASR 1/PIPA CALCGRAV CALCRVG AVETOMID
#
NORMAL EXIT MODES
#
ENDOFJOB TASKOVER CANV37
#
AVGEXIT: THIS IS A DOUBLE PRECISION ERASABLE LOCATION BY WHICH CONTROL
OF EACH CYCLE OF AVERAGE G.
THE 2CADR OF A ROUTINE TO BE PERFORMED AT THAT TIME (E.G., ST
AT 2 SECOND INTERVALS) MAY BE SET BY THE USER INTO AV
ALL SUCH ROUTINES SHOULD RETURN TO SERVEXIT, WHICH IS THE NO
#
SERVEXIT: DOES A PHASE CHANGE FOR RESTART PROTECTION AND GOES TO ENDOF
THE 2CADR OF SERVEXIT IS SET INTO AVGEXIT BY THE USERI F NO C
#
AVGEND: LAST PASS OF AVERAGE G EXITS HERE, BYPASSING SPECIAL ROUTINE
FINAL EXIT IS TO CANV37. F AVI
#
OUTPUT
#
DVTOTAL(2) PIPTIME(2) XPIPBUF(2) YPIPBUF(2) ZPIPBUF(2)
RN(6) REFERENCE COORD. SCALED AT 2(+29) M/CS
VN(6) REFERENCE COORD. SCALED AT 2(+7) M/CS
GDT/2(6) REFERENCE COORD. SCALED AT 2(+7) M/CS
DELV(6) STABLE MEMB. COORD. SCALED AT 2(+14)*5.85*10(-4) M/CS (KI
Page 822
DELVREF(6) REFERENCE COORD. SCALED AT 2(+7) M/CS
#
INITIALIZATION
#
ONMONITOR FLAG SET BY ENTRY TO SHOW PIPBUF VALUES REQUIRED.
IDLE FLAG ON IF DVMON TO BE BYPASSED.

```

July 12, 2016

1703

```
DVTHRUSH SET TO APPROPRIATE VALUE FOR DVMON.
AVGEXIT SET TO 2CADR OF ROUTINE, IF ANY, TO BE PERFORMED AFTER EACH CYCLE OF AVERAGE G.
TO BE DONE, AVGEXIT SET TO SERVEXIT.
VALUES NEEDED
REFSMMAT
UNITW: FULL UNIT VECTOR, IN REFERENCE COORD., OF EARTH'S ROTATIONAL VECTOR
RN1, VN1, PIPTIME1: IN REFERENCE COORD., CONSISTENT WITH TIME OF EXECUTION OF
#
DEBRIS
#
CENTRALS A, L, Q
OTHER INTERNAL: DVCNTR(1) PIPAGE(1) PIPCTR(1) AVGEXIT(2)
EXTERNAL: ITEMP1(1) ITEMP2(1) RUPTREG1(1) TEMX(1) TEMY(1) TEMZ(1)
USEFUL DEBRIS
RN1(0) VN1(0) GDT1/0 PIPTIME1(2?)
THESE LOCATIONS USED AS BUFFER STORAGE FOR NEWLY CALCUL
AND PIPTIME DURING PERFORMANCE OF SERVICER ROUTINES.
UNITR: HALF UNIT VECTOR OF RN, REFERENCE COORD.
RMAG SCALED AT 2(+58) IN 36D.
RMAGSQ SCALED AT 2(+58) IN 34D.
(RE/RMAG)SQ IN 32D
#
BANK 27
SETLOC SERVICES
BANK
#
EBANK= DVCNTR
***** PREREAD *****
#
COUNT 37/SERV
#
PREREAD CAF PRI021 # CALLER MUST PROTECT PREREAD
TC NOVAC
EBANK= NBDX
2CADR LASTBIAS # DO LAST GYRO COMPENSATION IN FREE FALL
#
CALL-TO AND LASTBIAS ITSELF ARE NOT
PROTECTED. REREADAC SETS 1/PIPADT
TO 2.0 SECS IN CASE LASTBIAS LOST.
(REDUNDANT IF LASTBIAS IS AOK.)
#
Page 823
RED05.31 TC PREREAD1
#
CAF PRI032
TC FINDVAC # SET UP NORMLIZE JOB REQUIRED PRIOR TO
EBANK= DVCNTR # FIRST AVERAGE G PASS
```

```

2CADR NORMLIZE

CAF 2SECS
TC WAITLIST
EBANK= AOG
2CADR READACCS

CS TWO
TC NEWPHASE
OCT 5

TCF TASKOVER

PREREAD1 EXTEND
 QXCH RUPTREG1

TC PIPASR # CLEAR + READ PIPS LAST TIME IN FREE FALL

CAF ONE # SET UP PIPAGE FOR REREADAC IN CASE A
TS PIPAGE # RESTART OCCURS BEFORE READACCS

CS FLAGWRD1 # SET AVEG FLAG
MASK BIT1
ADS FLAGWRD1

CA POSMAX
MASK FLAGWRD2
TS FLAGWRD2 # KNOCK DOWN DRIFT FLAG

CS FLAGWRD7 # SET V37 FLAG
MASK BIT6
ADS FLAGWRD7

CAF ZERO
TS DVTOTAL # CLEAR DVTOTAL
TS DVTOTAL +1

TC RUPTREG1

Page 824
***** READACCS *****
 EBANK= AOG
READACCS TC PIPASR

PIPSDONE CAF FIVE
 TS L

```



July 12, 2016

1705

```
COM
DXCH -PHASE5

RED05.5 CAF ONE # SHOW PIPS HAVE BEEN READ
 TS PIPAGE

 CA TWO # SET PIPCTR FOR ONMINTOR
 TS PIPCTR # AFTER ABOVE PHASCHNG

 CS CM/FLAGS
 MASK BIT2 # CM/DSTBY
 CCS A
 TC CHEKAVEG

 CS PIPTIME1 +1
 TS TBASE6 # FOR RESTARTS
 EXTEND # CONTINUE FOR ENTRY DAP
 DCA AOG
 DXCH AOG/PIP
 CA AMG
 XCH AMG/PIP
 EXTEND
 DCA ROLL/180
 DXCH ROLL/PIP
 CA BETA/180
 XCH BETA/PIP
 CA CM/FLAGS
 MASK BIT12 # CM/DAPARM 93D BIT12
 EXTEND # DURING ENTRY, WHEN RCS DAP IS INACTIVE,
 BZF NOSAVPIP # SAVE PIPAS EACH 0.5 SEC FOR TM.

 CA 0.5SEC
 TC WAITLIST
 EBANK= XPIPBUF
 2CADR QUIKREAD

 CA DELVX # NO NEED TO RESTART PROTECT THIS.
 XCH XPIPBUF # SAVE PIPAS AS READ (BUT NOT COMPENSATED)
 TS XOLDBUF

 CA DELVY
 XCH YPIPBUF
 TS YOLDBUF

 CA DELVZ
```

1706

July 12, 2016

|          |        |          |                                         |
|----------|--------|----------|-----------------------------------------|
|          | XCH    | ZPIPBUF  |                                         |
|          | TS     | ZOLDBUF  |                                         |
| NOSAVPIP | CA     | FIVE     |                                         |
|          | TS     | CM/GYMDT |                                         |
|          | CA     | JTAGTIME | # ACTIVATE CM/RCS AFTER PIPUP TO GO     |
|          |        |          | # IN JTAGTIME +5 CS.                    |
|          | TC     | WAITLIST |                                         |
|          | EBANK= | AOG      |                                         |
|          | 2CADR  | SETJTAG  |                                         |
|          | CS     | THREE    | # 1.3SPOT FOR SETJTAG                   |
|          | TC     | NEWPHASE |                                         |
|          | OCT    | 1        |                                         |
|          | CAF    | OCT37    |                                         |
|          | TS     | L        |                                         |
|          | COM    |          |                                         |
|          | DXCH   | -PHASE5  |                                         |
| CHEKAVEG | CS     | FLAGWRD1 |                                         |
|          | MASK   | BIT1     |                                         |
|          | CCS    | A        | # IF AVEG FLAG DOWN SET FINAL EXIT AVEG |
|          | TC     | AVEGOUT  |                                         |
|          | CAF    | 2SECS    |                                         |
|          | TC     | WAITLIST |                                         |
|          | EBANK= | AOG      |                                         |
|          | 2CADR  | READACCS |                                         |
| MAKESERV | CAF    | PRI020   | # ESTABLISH SERVICER ROUTINE            |
|          | TC     | FINDVAC  |                                         |
|          | EBANK= | DVCNTR   |                                         |
|          | 2CADR  | SERVICER |                                         |
|          | CS     | FOUR     | # RESTART SERVICER AND READACCS         |
|          | TC     | NEWPHASE |                                         |
|          | OCT    | 5        |                                         |
|          | CAF    | BIT9     |                                         |
|          | EXTEND |          |                                         |
|          | WOR    | DSALMOUT | # TURN TEST CONNECTOR OUTBIT ON         |
|          | TCF    | TASKOVER | # END PREVIOUS READACCS WAITLIST TASK   |

July 12, 2016

1707

# Page 826

```
AVEGOUT EXTEND
 DCA AVOUTCAD
 DXCH AVGEXIT
 TCF MAKESERV

 EBANK= DVCNTR
AVOUTCAD 2CADR AVGEND
```

# Page 827

```
ROUTINE NAME: ONMNITOR
MOD 04 BY BAIRNSFATHER 30 APR 1968 REDO ONMNITOR TO SAVE PIPS EACH 0.5 SEC FOR TM,ENTRY.
MOD 03 BY FISHER DECEMBER 1967
MOD 02 BY RYE SEPT 1967
MOD 01 BY KOSMALA 23 MAR 1967
MOD 00 BY KOSMALA 27 FEB 1967
#
FUNCTIONAL DESCRIPTION
#
THE PURPOSE OF ONMONITOR IS TO PROVIDE 1/2 SEC. READING OF PIPAS FOR DOWNLIST DURING EN
X,Y,XPIPBUF CONTAIN PRESET VALUES X,Y,ZOLDBUF CONTAIN VALUES FROM PREVIOUS READING.
#
CALLING SEQUENCE
#
CALL AS WAITLIST TASK. TERMINATES ITSELF IN TASKOVER
#
INITIALIZATION
#
PIPCTR = 2 (FOR DT = 0.5 SEC)
X,Y,ZPIPBUF SET TO PREVIOUS PIPAX,Y,Z
#
OUTPUT
#
X,Y,ZPIPBUF, X,Y,ZOLDBUF
#
DEBRIS
#
X,Y,ZPIPBUF CONTAIN LAST PIPAX,Y,Z VALUES
X,Y,ZOLDBUF CONTAIN LAST-BUT-ONE PIPAX,Y,Z VALUES
RUPTREG1
PIPCTR

ONMNITOR TS PIPCTR

 TC FIXDELAY # WAIT
0.5SEC DEC 50
```

```

QUIKREAD CAF TWO
 TS RUPTREG1
 INDEX A
 CA PIPAX # SAVE ACTUAL PIPAS FOR TM.
 INDEX RUPTREG1
 XCH XPIPBUFF # UPDATE X,Y,ZPIPBUFF
 INDEX RUPTREG1
 TS XOLDBUFF # AND X,Y,ZOLDBUFF
CHKCTR CCS RUPTREG1
 TCF QUIKREAD +1 # LOOP AGAIN
 CCS PIPCTR
 TCF ONMNITOR
 TC TASKOVER

```

# Page 828

# \*\*\*\*\* SERVICER \*\*\*\*\*

```

 EBANK= DVCNTR

SERVICER CAF TWO
 INHINT
PIPCHECK TS RUPTREG1

 DOUBLE
 INDEX A
 CCS DELVX
 TC +2
 TC PIPLOOP

 AD -MAXDELV # DO PIPA-SATURATION TEST BEFORE
 EXTEND
 BZMF PIPLOOP # COMPENSATION.

 TC ALARM
 OCT 00205 # SATURATED-PIPA ALARM ***CHANGE LATER
 TC AVERAGEG

PIPLOOP CCS RUPTREG1
 TCF PIPCHECK

 TC PHASCHNG # RESTART REREADAC + SERVICER
 OCT 16035
 OCT 20000
 EBANK= DVCNTR
 2CADR DVTOTUP

```

July 12, 2016

1709

|            |        |            |                                        |
|------------|--------|------------|----------------------------------------|
|            | TC     | BANKCALL   | # PIPA COMPENSATION CALL               |
|            | CADR   | 1/PIPA     |                                        |
| DVTOTUP    | TC     | INTPRET    |                                        |
|            | VLOAD  | ABVAL      | # GET ABS VALUE OF DELV                |
|            |        | DELV       |                                        |
|            | DMP    | EXIT       |                                        |
|            |        | KPIP1      | # SCALE AT 2(+7)                       |
|            | EXTEND |            |                                        |
|            | DCA    | MPAC       |                                        |
|            | DAS    | DVTOTAL    | # ACCUMULATE DVTOTAL                   |
| AVERAGEG   | TC     | PHASCHNG   |                                        |
|            | OCT    | 10035      |                                        |
|            | TC     | INTPRET    |                                        |
|            | CALL   |            |                                        |
| # Page 829 |        | CALCRVG    |                                        |
|            | EXIT   |            |                                        |
|            | TC     | PHASCHNG   |                                        |
|            | OCT    | 10035      |                                        |
|            | CAF    | OCT31      | # COPY RN1,VN1,GOT102,GOBL1/2,PIPTIME1 |
|            | TC     | GENTRAN    | # INTO RN, VN, GDT/12, GOBL/2,PIPTIME  |
|            | ADRES  | RN1        |                                        |
|            | ADRES  | RN         |                                        |
|            | RELINT |            | # GENTRAN DOES AN INHINT               |
|            | TC     | PHASCHNG   |                                        |
|            | OCT    | 10035      |                                        |
|            | EXTEND |            |                                        |
|            | DCA    | AVGEXIT    |                                        |
|            | DXCH   | Z          | # AVERAGEG EXIT                        |
| AVGEND     | CA     | PIPTIME +1 | # FINAL AVERAGE G EXIT                 |
|            | TS     | OLDBT1     | # SET UP FREE FALL GYRO COMPENSATION   |
|            | TC     | UPFLAG     | # SET DRIFTFLG                         |
|            | ADRES  | DRIFTFLG   | # BIT 15 FLAG 2                        |
|            | TC     | 2PHSCHNG   |                                        |
|            | OCT    | 5          | # GROUP 5 OFF                          |
|            | OCT    | 05022      | # GROUP 2 ON FOR AVETOMID              |
|            | OCT    | 20000      |                                        |

|            |        |          |                                                                                                   |
|------------|--------|----------|---------------------------------------------------------------------------------------------------|
|            | TC     | INTPRET  |                                                                                                   |
|            | CALL   |          |                                                                                                   |
|            |        | AVETOMID | # CONVERT STATE VECTOR TO REFERENCE SCALE.                                                        |
|            | EXIT   |          |                                                                                                   |
|            | CAF    | ZERO     | # ZERO MARK COUNTERS.                                                                             |
|            | TS     | VHFCNT   |                                                                                                   |
|            | TS     | TRKMKCNT |                                                                                                   |
|            | TC     | BANKCALL |                                                                                                   |
|            | CADR   | PIPFREE  |                                                                                                   |
|            | CS     | BIT9     |                                                                                                   |
|            | TS     | MRKBUF2  | # INVALIDATE MARK BUFFER                                                                          |
|            | EXTEND |          |                                                                                                   |
|            | WAND   | DSALMOUT |                                                                                                   |
|            | TC     | DOWNFLAG |                                                                                                   |
|            | ADRES  | CM/DSTBY |                                                                                                   |
|            | TC     | DOWNFLAG |                                                                                                   |
|            | ADRES  | V37FLAG  |                                                                                                   |
| # Page 830 |        |          |                                                                                                   |
|            | CAF    | BIT7     | # RESTORE GROUP 1 + 2 IF P20 IS RUNNING.                                                          |
|            | MASK   | FLAGWRDO |                                                                                                   |
|            | EXTEND |          |                                                                                                   |
|            | BZF    | +4       |                                                                                                   |
|            | TC     | 2PHSCHNG |                                                                                                   |
|            | OCT    | 111      | # 1.11SPOT                                                                                        |
|            | OCT    | 132      | # 2.13SPOT                                                                                        |
|            | TC     | POSTJUMP |                                                                                                   |
|            | CADR   | CANV37   |                                                                                                   |
| SERVEXIT   | TC     | PHASCHNG |                                                                                                   |
|            | OCT    | 00035    | # A, 5.3 = REREADAC (ONLY)                                                                        |
|            | TCF    | ENDOFJOB |                                                                                                   |
| DVTHRUSH   | EQUALS | ELEVEN   | # 15 PERCENT OF 2SEC PIPA ACCUMULATION,<br># FOR 503-FULL CSM/LEM....DELV SC.AT<br># 5.85 CM/SEC. |

July 12, 2016

1711

```
-MAXDELV DEC -6398 # 3200 PPS FOR 2 SEC CCS TAKES 1

JTAGTIME DEC 120 # = 1 SEC + T CDU, T CDU = .1 SEC

2.5SEC DEC 250
MDOTFAIL DEC 144.0 B-16 # 5 SEC MASS LOSS AT 28.8 KG/SEC
 # SHOULD BE 2-4 SECS FOR NO START
 # 6-8 SECS FOR FAILURE
```

# Page 831

```
NORMLIZE PERFORMS THE INITIALIZATION REQUIRED PRIOR TO THE FIRST ENTRY TO AVERAGE, AND SCALE
HAS 1 LEADING BINARY ZERO. IN MOST MISSIONS, RN WILL BE SCALED AT 2(+29), BUT IN THE 206 MIS
SCALED AT 2(+24) M.
```

```
NORMLIZE CAF THIRTEEN # SET UP TO COPY 14 REGS: RN1,VN1,PIPTIME1
 TC GENTRAN # INTO RN,VN,PIPTIME
 ADRES RN1 # FROM HERE
 ADRES RN # TO HERE

 RELINT
 TC INTPRET
 VLOAD CALL # LOAD RN FOR CALCGRAV
 RN
 CALCGRAV # INITIALIZE UNITR RMAG GDT1

 STOVL GDT/2
 GOBL1/2
 STORE GOBL/2
 EXIT
 TCF ENDOFJOB
```

# Page 832

```
***** PIPA READER *****
```

```
MOD NO. 00 BY D. LICKLY DEC. 9 1966
```

```
#
```

```
FUNCTIONAL DESCRIPTION
```

```
#
```

```
SUBROUTINE TO READ PIPA COUNTERS, TRYING TO BE VERY CAREFUL SO THAT IT WILL BE RESTARTABLE.
PIPA READINGS ARE STORED IN THE VECTOR DELV. THE HIGH ORDER PART OF EACH COMPONENT CONTAINS
RESTARTS BEGIN AT REREADAC.
```

```
#
```

```
AT THE END OF THE PIPA READER THE CDUS ARE READ AND STORED AS A
VECTOR IN CDUTEMP. THE HIGH ORDER PART OF EACH COMPONENT CONTAINS
THE CDU READING IN 2'S COMP IN THE ORDER CDUX,Y,Z. THE THRUST
VECTOR ESTIMATOR IN FINDCDUD REQUIRES THE CDUS BE READ AT PIPTIME.
```

```
#
```

```

CALLING SEQUENCE AND EXIT
#
CALL VIA TC, ISWCALL, ETC.
#
EXIT IS VIA Q.
#
INPUT
#
INPUT IS THROUGH THE COUNTERS PIPAX, PIPAY, PIPAZ, AND TIME2.
#
OUTPUT
#
HIGH ORDER COMPONENTS OF THE VECTOR DELV CONTAIN THE PIPA READINGS.
#
PIPTIME CONTAINS TIME OF PIPA READING.
#
DEBRIS (ERASABLE LOCATIONS DESTROYED BY THE PROGRAM)
#
LOW ORDER DELV'S ARE ZEROED FOR TM INDICATION.
TEMX TEMY TEMZ PIPAGE

PIPASR EXTEND
 DCA TIME2
 DXCH PIPTIME1 # CURRENT TIME POSITIVE VALUE
 CS ZERO # INITIALIZE THESE AT NEG ZERO.
 TS TEMX
 TS TEMY
 TS TEMZ

Page 833
 CA ZERO
 TS DELVZ # OTHER DELVS OK INCLUDING LOW ORDER
 TS DELVY
 TS DELVX +1 # LOW ORDER DELV'S ARE ZEROED FOR TM: THIS
 TS DELVY +1 # IF DNLNK=D LOW ORDER DELVS ARE NZ, THEY
 TS DELVZ +1 # CONTAIN PROPER COMPENSATION. IF=0, THEN
 # THE TM VALUES ARE BEFORE COMPENSATION.

 TS PIPAGE # SHOW PIPA READING IN PROGRESS

REPIP1 EXTEND
 DCS PIPAX # X AND Y PIPS READ
 DXCH TEMX
 DXCH PIPAX # PIPAS SET TO NEG ZERO AS READ.
 TS DELVX
 LXCH DELVY

```



July 12, 2016

1713

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
| REPIP3     | CS     | PIPAZ    | # REPEAT PROCESS FOR Z PIP                 |
|            | XCH    | TEMZ     |                                            |
|            | XCH    | PIPAZ    |                                            |
| DODELVZ    | TS     | DELVZ    |                                            |
|            | TC     | Q        |                                            |
|            | EBANK= | AOG      |                                            |
| REREADAC   | CCS    | PHASE5   | # LAST PASS CHECK                          |
|            | TCF    | +2       |                                            |
|            | TCF    | TASKOVER |                                            |
|            | CAF    | PRI031   | # RESTART MAY HAVE WIPE OUT LASTBIAS, AN   |
|            | TS     | 1/PIPADT | # UNPROTECTED NOVAC FROM PREREAD,          |
|            |        |          | # WHICH SET(S) UP 1/PIPADT (THUSLY)        |
|            |        |          | # FOR NON-COASTING COMPENSATION...BE       |
|            |        |          | # SURE 1/PIPADT IS AOK. (PRI031 IS         |
|            |        |          | # 2.0SEC SC.AT B+8(CS)                     |
|            | CCS    | PIPAGE   |                                            |
|            | TCF    | READACCS | # PIP READING NOT STARTED. GO TO BEGINNING |
|            | CAF    | DONEADR  | # SET UP RETURN FROM PIPASR                |
|            | TS     | Q        |                                            |
|            | CCS    | DELVZ    |                                            |
|            | TC     | Q        | # Z DONE, GO DO CDUS                       |
|            | TCF    | +3       | # Z NOT DONE, CHECK Y.                     |
|            | TC     | Q        |                                            |
|            | TC     | Q        |                                            |
| # Page 834 | ZL     |          |                                            |
|            | CCS    | DELVY    |                                            |
|            | TCF    | +3       |                                            |
|            | TCF    | CHKTEMX  | # Y NOT DONE, CHECK X.                     |
|            | TCF    | +1       |                                            |
|            | LXCH   | PIPAZ    | # Y DONE, ZERO Z PIP.                      |
|            | CCS    | TEMZ     |                                            |
|            | CS     | TEMZ     | # TEMZ NOT = -0, CONTAINS -PIPAZ VALUE.    |
|            | TCF    | DODELVZ  |                                            |
|            | TCF    | -2       |                                            |
|            | LXCH   | DELVZ    | # TEMZ = -0, L HAS ZPIP VALUE.             |
|            | TC     | Q        |                                            |

# Page 835

```

#*****
#
ROUTINE CALCRVG INTEGRATES THE EQUATIONS OF MOTION BY AVERAGING THE THRUST AND GRAV
ACCELERATIONS OVER A TIME INTERVAL OF 2 SECONDS.
#
FOR THE EARTH-CENTERED GRAVITATIONAL FIELD, THE PERTURBATION DUE TO OBLATENESS IS C
HARMONIC COEFFICIENT J.
#
ROUTINE CALCRVG REQUIRES...
#
1) THRUST ACCELERATION INCREMENTS IN DELV SCALED SAME AS PIPAX,Y,Z IN STABLE
2) VN SCALED 2(+7) M/CS IN REFERENCE COORDS.
3) RN SCALED AT 2(+29) METERS IN REFERENCE COORDS.
4) UNITW THE EARTH'S UNIT ROTATIONAL VECTOR (SCALED AS A FULL UNIT VECTOR) IN
#
IT LEAVES RN1 UPDATED (SCALED AT 2(+29)M, VN1 (SCALED AT 2(+7)M/CS), AND GDT1/2 (SC
UNIT VECTOR UNITR, RMAG IN 36D SCALED AT 2(+29)M, R MAG SQ. IN 34D SCALED AT 2(+58)

```

| CALCGRV | UNIT  | PUSH     | # ENTER WITH RN IN MPAC |
|---------|-------|----------|-------------------------|
|         | STORE | UNITR    |                         |
|         | LXC,1 | SLOAD    |                         |
|         |       | RTX2     |                         |
|         |       | X1       |                         |
|         | BMN   | VLOAD    |                         |
|         |       | ITISMOON |                         |
|         | DOT   | PUSH     |                         |
|         |       | UNITW    |                         |
|         | DSQ   | BDSU     |                         |

July 12, 2016

1715

|            |        |            |                                           |
|------------|--------|------------|-------------------------------------------|
|            |        | DP1/20     |                                           |
|            | PDDL   | DDV        |                                           |
|            |        | RESQ       |                                           |
|            |        | 34D        | # (RN)SQ                                  |
|            | STORE  | 32D        | # TEMP FOR (RE/RN)SQ                      |
|            | DMP    | DMP        |                                           |
|            |        | 20J        |                                           |
|            | VXSC   | PDDL       |                                           |
|            |        | UNITR      |                                           |
|            | DMP    | DMP        |                                           |
|            |        | 2J         |                                           |
|            |        | 32D        |                                           |
|            | VXSC   | VAD        |                                           |
|            |        | UNITW      |                                           |
|            | STADR  |            |                                           |
|            | STORE  | GOBL1/2    |                                           |
|            | VAD    | PUSH       |                                           |
|            |        | UNITR      |                                           |
| ITISMOON   | DLOAD  | NORM       |                                           |
|            |        | 34D        |                                           |
|            |        | X2         |                                           |
|            | BDDV*  | SLR*       |                                           |
| # Page 836 |        |            |                                           |
|            |        | -MUDT(E),1 |                                           |
|            |        | 0 -21D,2   |                                           |
|            | VXSC   | STADR      |                                           |
|            | STORE  | GDT1/2     | # SCALED AT 2(+7) M/CS                    |
|            | RVQ    |            |                                           |
| CALCRVG    | VLOAD  | VXSC       |                                           |
|            |        | DELV       |                                           |
|            |        | KPIP1      |                                           |
|            | VXM    | VSL1       |                                           |
|            |        | REFSMMAT   |                                           |
|            | STORE  | DELVREF    | # DELV IN REF COORDS AT 2(+7)             |
|            | VSR1   | PUSH       |                                           |
|            | VAD    | PUSH       | # (DV-OLDGDT)/2 TO PD SCALED AT 2(+7)M/CS |
|            |        | GDT/2      |                                           |
|            | VAD    | VXSC       |                                           |
|            |        | VN         |                                           |
|            |        | 2SEC(22)   |                                           |
|            | VAD    | STQ        |                                           |
|            |        | RN         |                                           |
|            |        | 31D        |                                           |
|            | STCALL | RN1        | # TEMP STORAGE OF RN SCALED 2(+29)M       |
|            |        | CALCGRAV   |                                           |

|          |        |                       |                                            |
|----------|--------|-----------------------|--------------------------------------------|
|          | VAD    | VAD                   |                                            |
|          | VAD    |                       |                                            |
|          |        | VN                    |                                            |
|          | STCALL | VN1                   | # TEMP STORAGE OF VN SCALED 2(+7) M/CS.    |
|          |        | 31D                   |                                            |
| KPIP     | 2DEC   | .1024                 | # SCALES DELV TO 2(+4)                     |
| KPIP1    | 2DEC   | 0.074880              | # 207 DELV SCALING. 1 PULSE = 5.85 CM/SEC. |
| -MUDT(E) | 2DEC*  | -7.9720645 E+12 B-44* |                                            |
| -MUDT(M) | 2DEC*  | -9.805556 E+10 B-44*  |                                            |
| 2SEC(22) | 2DEC   | 200 B-22              |                                            |
| DP1/20   | 2DEC   | 0.05                  |                                            |
| RESQ     | 2DEC*  | 40.6809913 E12 B-59*  |                                            |
| 20J      | 2DEC*  | 3.24692010 E-2 B1*    |                                            |
| 2J       | 2DEC*  | 3.24692010 E-3 B1*    |                                            |

This code is written to file `src/SERVICER207.s`.

## B.102 SERVICE ROUTINES

```

1717 <src/SERVICE-ROUTINES.s 1717>≡
 # Copyright: Public domain.
 # Filename: SERVICE_ROUTINES.agc
 # Purpose: Part of the source code for Comanche, build 055. It
 # is part of the source code for the Command Module's
 # (CM) Apollo Guidance Computer (AGC), Apollo 11.
 # Assembler: yaYUL
 # Reference: pp. 1485-1492
 # Contact: Ron Burkey <info@sandroid.org>
 # Website: http://www.ibiblio.org/apollo.
 # Mod history: 2009-05-07 RSB Adapted from Colossus249 file of the same
 # name, and page images. Corrected various
 # typos in the transcription of program
 # comments, and these should be back-ported
 # to Colossus249.
 #
 # The contents of the "Comanche055" files, in general, are transcribed
 # from scanned documents.
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. April 1, 1969.
 #
 # This AGC program shall also be referred to as Colossus 2A
 #
 # Prepared by
 #
 # Massachussets Institute of Technology
 # 75 Cambridge Parkway
 # Cambridge, Massachusetts
 #
 # under NASA contract NAS 9-4065.
 #
 # Refer directly to the online document mentioned above for further
 # information. Please report any errors to info@sandroid.org.
 #
 # Page 1485
 BLOCK 3
 SETLOC FFTAG6
 BANK
 COUNT 03/FLAG

 UPENT2 TS L # WHICH FLAGWORD IS IT
 MASK OCT7
 XCH L # SAVE IN L FOR INDEXING

```

```

 MASK OCT77770 # OBTAIN THE BIT INFORMATION
 INHINT # PREVENT INTERRUPTS
 TS ITEMP1 # STORE THE BIT INFORMATION TEMPORARILY

 NDX L
 CS FLAGWRDO
 MASK ITEMP1
 NDX L
 ADS FLAGWRDO
 RELINT # RELEASE INTERRUPT INHIBIT

 INCR Q # OBTAIN THE CORRECT RETURN ADDRESS
 TC Q # RETURN

DOWNENT2 TS L # WHICH FLAGWORD IS IT
 MASK OCT7
 XCH L # SAVE IN L FOR INDEXING

 MASK OCT77770 # OBTAIN THE BIT INFORMATION
 COM # START TO PROCESS THE INFORMATION

 INHINT # PREVENT INTERRUPTS
 NDX L
 MASK FLAGWRDO
 NDX L
 TS FLAGWRDO
 RELINT # RELEASE INTERRUPT INHIBIT

 INCR Q # OBTAIN THE CORRECT RETURN ADDRESS
 TC Q

OCT7 EQUALS SEVEN
 BANK 10

```

# Page 1486

# UPFLAG AND DOWNFLAG ARE ENTIRELY GENERAL FLAG SETTING AND CLEARING SUBROUTINES. US  
 # NOT IN INTERRUPT, ONE MAY SET OR CLEAR ANY SINGLE, NAMED BIT IN ANY ERASABLE REGISTER.  
 # EBANK SETTING. A "NAMED" BIT, AS THE WORD IS USED HERE, IS ANY BIT WITH A NAME FOR  
 # ASSEMBLER.

#

# AT PRESENT THE ONLY NAMED BITS ARE THOSE IN THE FLAGWORDS. ASSEMBLER CHANGES WILL  
 # NAME ANY BIT IN ERASABLE MEMORY.

#

# CALLING SEQUENCES ARE AS FOLLOWS --

```

TC UPFLAG TC DOWNFLAG
ADRES NAME OF FLAG ADRES NAME OF FLAG

```

July 12, 2016

1719

#  
# RETURN IS TO THE LOCATION FOLLOWING THE "ADRES" ABOUT .58 MS AFTER THE "TC".  
# UPON RETURN A CONTAINS THE CURRENT FLAGWRD SETTING.

|             |        |           |                               |
|-------------|--------|-----------|-------------------------------|
|             | BLOCK  | 02        |                               |
|             | SETLOC | FFTAG1    |                               |
|             | BANK   |           |                               |
|             | COUNT* | \$\$/FLAG |                               |
| UPFLAG      | CA     | Q         |                               |
|             | TC     | DEBIT     |                               |
|             | COM    |           | # +(15 - BIT)                 |
|             | EXTEND |           |                               |
|             | ROR    | LCHAN     | # SET BIT                     |
| COMFLAG     | INDEX  | ITEMP1    |                               |
|             | TS     | FLAGWRD0  |                               |
|             | LXCH   | ITEMP3    |                               |
|             | RELINT |           |                               |
|             | TC     | L         |                               |
| DOWNFLAG    | CA     | Q         |                               |
|             | TC     | DEBIT     |                               |
|             | MASK   | L         | # RESET BIT                   |
|             | TCF    | COMFLAG   |                               |
| DEBIT       | AD     | ONE       | # CET DE BITS                 |
|             | INHINT |           |                               |
|             | TS     | ITEMP3    |                               |
|             | CA     | LOW4      | # DEC15                       |
|             | TS     | ITEMP1    |                               |
|             | INDEX  | ITEMP3    |                               |
|             | CA     | 0 -1      | # ADRES                       |
|             | TS     | L         |                               |
|             | CA     | ZERO      |                               |
| # Page 1487 | EXTEND |           |                               |
|             | DV     | ITEMP1    | # A = FLAGWRD, L = (15 - BIT) |
|             | DXCH   | ITEMP1    |                               |
|             | INDEX  | ITEMP1    |                               |
|             | CA     | FLAGWRD0  |                               |
|             | TS     | L         | # CURRENT STATE               |
|             | INDEX  | ITEMP2    |                               |
|             | CS     | BIT15     | # -(15 - BIT)                 |
|             | TC     | Q         |                               |

# Page 1488

```

DELAYJOB -- A GENERAL ROUTINE TO DELAY A JOB A SPECIFIC AMOUNT OF TIME BEFORE PICK
#
ENTRANCE REQUIREMENTS ...
CAF DT # DELAY JOB FOR DT CENTISECS
TC BANKCALL
CADR DELAYJOB

 BANK 06
 SETLOC DLAYJOB
 BANK

THIS MUST REMAIN IN BANK 0 *****

 COUNT 00/DELAY

DELAYJOB INHINT
 TS Q # STORE DELAY DT IN Q FOR DLY -1 IN
 CAF DELAYNUM # WAITLIST
DELOOP TS RUPTREG1
 INDEX A
 CA DELAYLOC # IS THIS DELAYLOC AVAILABLE
 EXTEND
 BZF OK2DELAY # YES
 CCS RUPTREG1 # NO, TRY NEXT DELAYLOC
 TCF DELOOP
 TC BAILOUT # NO AVAILABLE LOCS.
 OCT 1104
OK2DELAY CA TCSLEEP # SET WAITLIST IMMEDIATE RETURN
 TS WAITEXIT
 CA FBANK
 AD RUPTREG1 # STORE BBANK FOR TASK CALL
 TS L
 CAF WAKECAD # STORE CADR FOR TASK CALL
 TCF DLY2 -1 # DLY IS IN WAITLIST ROUTINE
TCGETCAD TC MAKECADR # GET CALLER'S FCADR
 INDEX RUPTREG1
 TS DELAYLOC # SAVE DELAY CADRS
 TC JOBSLEEP

```



July 12, 2016

1721

```
WAKER CAF ZERO
 INDEX BBANK
 XCH DELAYLOC # MAKE DELAYLOC AVAILABLE
```

# Page 1489

```
 TC JOBWAKE
 TC TASKOVER
```

```
TCSLEEP GENADR TCGETCAD -2
WAKECAD GENADR WAKER
```

# Page 1490

# GENTRAN, A BLOCK TRANSFER ROUTINE

# WRITTEN BY D. EYLES

# MOD 1 BY KERNAN

UTILITYM REV 17 11/18/67

# MOD 2 BY SCHULENBERG -- (REMOVE RELINT) -- SKIPPER REV 4 2/28/68

#

# THIS ROUTINE IS USEFUL FOR TRANSFERRING N CONSECUTIVE ERASABLE OR FIXED QUANTITIES TO SOME OTHER  
# CONSECUTIVE ERASABLE LOCATIONS. IF BOTH BLOCKS OF DATA ARE IN SWITCHABLE EBANKS, THEY MUST BE

#

# GENTRAN IS CALLABLE IN A JOB AS WELL AS A RUPT. THE CALLING SEQUENCE IS:

```
I CA N-1 # NO. NUMBER OF QUANTITIES MINUS ONE.
I +1 TC GENTRAN # IN FIXED-FIXED.
I +2 ADRES L # STARTING ADRES OF DATA TO BE MOVED.
I +3 ADRES M # STARTING ADRES OF DUPLICATION BLOCK.
I +4 # RETURNS HERE.
```

#

# GENTRAN TAKES 25 MCT'S (300 MICROSECONDS) PER ITEM + 5 MCT'S (60 MICS) FOR ENTERING AND EXITING

# A, L, AND ITEMP1 ARE NOT PRESERVED.

```
BLOCK 02
SETLOC FFTAG4
BANK
```

EBANK= ITEMP1

COUNT\* \$\$/TRAN

```
GENTRAN INHINT
 TS ITEMP1 # SAVE N-1
 INDEX Q # C(Q) = ADRES L.
 AD 0 # ADRES (L + N - 1).
 INDEX A
 CA 0 # C(ABOVE).
 TS L # SAVE DATA.
```

```

CA ITEMP1
INDEX Q
AD 1 # ADRES (M + N - 1).
INDEX A
LXCH 0 # STUFF IT.
CCS ITEMP1 # LOOP UNTIL N-1 = 0.
TCF GENTRAN +1
TCF Q+2 # RETURN TO CALLER.

```

# Page 1491

```

B50FF ZERO BIT 5 OF EXTVBACK, WHICH IS SET BY TESTXACT.
MAY BE USED AS NEEDED BY ANY EXTENDED VERB WHICH HAS DONE TESTXACT

```

```

COUNT* $$/EXTVB

```

```

B50FF CS BIT5
 MASK EXTVBACT
 TS EXTVBACT
 TC ENDOFJOB

```

# Page 1492

```

SUBROUTINES TO TURN OFF AND TURN ON TRACKER FAIL LIGHT.

```

```

TRFAILOF INHINT
 CS OCT40200 # TURN OFF TRACKER LIGHT
 MASK DSPTAB +11D
 AD BIT15
 TS DSPTAB +11D
 CS OPTMODES # TO INSURE THAT OCDU FAIL WILL GO ON
 MASK BIT7 # AGAIN IF IT WAS ON IN ADDITION TO
 ADS OPTMODES # TRACKER FAIL.

```

```

REQ RELINT
 TC Q

```

```

TRFAILON INHINT
 CS DSPTAB +11D # TURN ON
 MASK OCT40200
 ADS DSPTAB +11D
 TCF REQ

```

## B.103 SERVICER

```

1723 <src/SERVICER.s 1723>≡
Copyright: Public domain.
Filename: SERVICER.agc
Purpose: Part of the source code for Luminary, build 099. It
is part of the source code for the Lunar Module's
(LM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 857-897
Contact: Ron Burkey <info@sandroid.org>,
Fabrizio Bernardini <fabrizio@spacecraft.it>
Website: http://www.ibiblio.org/apollo.
Mod history: 2009-06-01 FB Transcription Batch 4 Assignment.
2009-06-05 RSB Fixed a couple of typos, plus a goofy relative
label reference from the original source.
#
The contents of the "Luminary099" files, in general, are transcribed
from scanned documents.
#
Assemble revision 001 of AGC program Luminary099 by NASA
2021112-061. July 14, 1969.
#
Prepared by
#
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 857

BANK 37
SETLOC SERV1
BANK

EBANK= DVCNTR

***** PREREAD *****

COUNT* $$/SERV

PREREAD CAF SEVEN # 5.7 SPOT TO SKIP LASTBIAS AFTER
 TC GNUFAZE5 # RESTART.

```

1724

July 12, 2016

|          |        |           |                                                                                  |
|----------|--------|-----------|----------------------------------------------------------------------------------|
|          | CAF    | PRI021    |                                                                                  |
|          | TC     | NOVAC     |                                                                                  |
|          | EBANK= | NBDX      |                                                                                  |
|          | 2CADR  | LASTBIAS  | # DO LAST GYRO COMPENSATION IN FREE FALL                                         |
| BIBIBIAS | TC     | PIPASR +3 | # CLEAR + READ PIPS LAST TIME IN FRE5+F133<br># DO NOT DESTROY VALUE OF PIPTIME1 |
|          | CS     | FLAGWRD7  |                                                                                  |
|          | MASK   | SUPER011  | # SET V37FLAG AND AVEGFLAG (BITS 5 AND 6                                         |
|          | ADS    | FLAGWRD7  | # OF FLAGWRD7)                                                                   |
|          | CS     | DRFTBIT   |                                                                                  |
|          | MASK   | FLAGWRD2  | # RESET DRIFTFLAG                                                                |
|          | TS     | FLAGWRD2  |                                                                                  |
|          | CAF    | FOUR      | # INITIALIZE DV MONITOR                                                          |
|          | TS     | PIPAGE    |                                                                                  |
|          | CAF    | ENDJBCAD  | # POINT OUTROUTE TO END-OF-JOB.                                                  |
|          | TS     | OUTROUTE  |                                                                                  |
|          | CAF    | PRI022    |                                                                                  |
|          | TC     | FINDVAC   | # TO FIRST ENTRY TO AVERAGE G                                                    |
|          | EBANK= | DVCNTR    |                                                                                  |
|          | 2CADR  | NORMLIZE  |                                                                                  |
| GOREADAX | CA     | TWO       | # 5.2SPOT FOR REREADAC AND NORMLIZE                                              |
|          | TC     | GNUTFAZ5  |                                                                                  |
|          | CA     | 2SECS     | # WAIT TWO SECONDS FOR READACCS                                                  |
|          | TC     | VARDELAY  |                                                                                  |

# Page 858

# \*\*\*\*\* READACCS \*\*\*\*\*

|          |     |          |                                           |
|----------|-----|----------|-------------------------------------------|
| READACCS | CS  | OCT37771 | # THIS PIECE OF CODING ATTEMPTS TO        |
|          | AD  | TIME5    | # SYNCHRONIZE READACCS WITH THE DIGITAL   |
|          | CCS | A        | # AUTOPILOT SO THAT A PAXIS RUPT WILL     |
|          | CS  | ONE      | # OCCUR APPROXIMATELY 70 MILLISECONDS     |
|          | TCF | +2       | # FOLLOWING THE READACCS RUPT. THE 70 MS  |
|          | CA  | ONE      | # OFFSET WAS CHOSEN SO THAT THE PAXIS     |
| +2       | ADS | TIME5    | # RUPT WOULD NOT OCCUR SIMULTANEOUSLY     |
|          |     |          | # WITH ANY OF THE 8 SUBSEQUENT R10,R11    |
|          |     |          | # INTERRUPTS -- THUS MINIMIZING THE POSS- |
|          |     |          | # IBILITY OF LOSING DOWNRUPTS.            |

July 12, 2016

1725

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | TC     | PIPASR   | # READ THE PIPAS.                          |
| PIPSDONE   | CA     | FIVE     |                                            |
|            | TC     | GNUFAZE5 |                                            |
| RED05.5    | CAF    | ONE      |                                            |
|            | TS     | PIPAGE   |                                            |
|            | CA     | PRI020   |                                            |
|            | TC     | FINDVAC  |                                            |
|            | EBANK= | DVCNTR   |                                            |
|            | 2CADR  | SERVICER | # SET UP SERVICER JOB                      |
|            | CA     | BIT9     |                                            |
|            | EXTEND |          |                                            |
|            | WOR    | DSALMOUT | # TURN ON TEST CONNECTOR OUTBIT            |
|            | CA     | FLAGWRD7 |                                            |
|            | MASK   | AVEGFBIT |                                            |
|            | EXTEND |          |                                            |
|            | BZF    | AVEGOUT  | # AVEGFLAG DOWN -- SET UP FINAL EXIT       |
|            | CA     | FLAGWRD6 |                                            |
|            | MASK   | MUNFLBIT |                                            |
|            | EXTEND |          |                                            |
|            | BZF    | MAKEACCS | # MUNFLAG CLEAR -- BYPASS LR AND DISP.     |
|            | CCS    | PHASE2   |                                            |
|            | TCF    | MAKEACCS | # PHASE 2 ACTIVATED -- AVOID MULTIPLE R10. |
|            | CAF    | SEVEN    | # SET PIPCTR FOR 4X/SEC RATE.              |
|            | TS     | PIPCTR   |                                            |
|            | CS     | TIME1    | # SET TBASE2 .05 SECONDS IN THE PAST.      |
|            | AD     | FIVE     |                                            |
|            | AD     | NEG1/2   |                                            |
|            | AD     | NEG1/2   |                                            |
|            | XCH    | TBASE2   |                                            |
| # Page 859 | CAF    | DEC17    | # 2.21SPOT FOR R10,R11                     |
|            | TS     | L        |                                            |
|            | COM    |          |                                            |
|            | DXCH   | -PHASE2  |                                            |
|            | CAF    | OCT24    | # FIRST R10,R11 IN .200 SECONDS            |
|            | TC     | WAITLIST |                                            |
|            | EBANK= | UNIT/R/  |                                            |

```

 2CADR R10,R11

MAKEACCS CA FOUR
 TCF GOREADAX # DO PHASE CHANGE AND RECALL READACCS

AVEGOUT EXTEND
 DCA AVOUTCAD # SET UP FINAL SERVICER EXIT
 DXCH AVGEXIT

 CA FOUR # SET 5.4 SPOT FOR REREADAC AND SERVICER
 TC GNUTFAZ5 # IF REREADAC IS CALLED, IT WILL EXIT
 TC TASKOVER # END TASK WITHOUT CALLING READACCS

GNUTFAZ5 TS L # SAVE INPUT IN L
 CS TIME1
 TS TBASE5 # SET TBASE5
 TCF +2

GNUFAZE5 TS L # SAVE INPUT IN L
 CS L # -PHASE IN A, PHASE IN L.
 DXCH -PHASE5 # SET -PHASE5,PHASE5
 TC Q

 EBANK= DVCNTR
AVOUTCAD 2CADR AVGEND

ENDJBCAD CADR SERVEXIT +2

OCT37771 OCT 37771

 BANK 33
 SETLOC SERVICES
 BANK

 COUNT* $$/SERV

Page 860
***** SERVICER *****

SERVICER TC PHASCHNG # RESTART REREADAC + SERVICER
 OCT 16035
 OCT 20000
 EBANK= DVCNTR
 2CADR GETABVAL

 CAF PRI031 # INITIALIZE 1/PIPADT IN CASE RESTART HAS

```

July 12, 2016

1727

```

TS 1/PIPADT # CAUSED LASTBIAS TO BE SKIPPED.

TC BANKCALL
CADR 1/PIPA # PIPA COMPENSATION CALL

GETABVAL TC INTPRET
VLOAD ABVAL
 DELV

EXIT
CA MPAC
TS ABDELV # ABDELV = CM/SEC*2(-14).
EXTEND
MP KPIP
DXCH ABDVCONV # ABDVCONV = M/CS * 2(-5).
EXTEND
DCA MASS
DXCH MASS1 # INITIALIZE MASS1 IN CASE WE SKIP MASSMON
CS FLAGWRD8 # ARE WE ON THE SURFACE?
MASK SURFFBIT
EXTEND
BZF MOONSPOT # YES: BYPASS MASS MESS

CA FLGWRD10 # NO: WHICH VEX SHOULD BE USED?
MASK APSFLBIT
CCS A
EXTEND # IF EXTEND IS EXECUTED, APSVEX --> A,
DCA APSVEX # OTHERWISE DPSVEX --> A
TS Q

EXTEND
DCA ABDVCONV
EXTEND
OCT10002 DV Q # WHERE APPROPRIATE VEX RESIDES
EXTEND
MP MASS
DAS MASS1

MOONSPOT CA KPIP1 # TP MPAC = ABDELV AT 2(14) CM/SEC
TC SHORTMP # MULTIPLY BY KPIP1 TO GET

Page 861
DXCH MPAC # ABDELV AT 2(7) M/CS
DAS DVTOTAL # UPDATE DVTOTAL FOR DISPLAY

TC TMPTOSPT
TC BANKCALL
```

|            |        |          |                                         |
|------------|--------|----------|-----------------------------------------|
|            | CADR   | QUICTRIG |                                         |
|            | CAF    | XNBPIPAD |                                         |
|            | TC     | BANKCALL |                                         |
|            | CADR   | FLESHPOT |                                         |
|            | TC     | INTPRET  |                                         |
| AVERAGEG   | BON    | CALL     |                                         |
|            |        | MUNFLAG  |                                         |
|            |        | RVBOTH   |                                         |
|            |        | CALCRVG  |                                         |
|            | EXIT   |          |                                         |
| GOSERV     | TC     | QUIKFAZ5 |                                         |
| COPYCYCL   | TC     | COPYCYC  |                                         |
| #          | CA     | ZERO     | # A IS ZERO ON RETURN FROM COPYCYC      |
|            | TS     | PIPATMPX |                                         |
|            | TS     | PIPATMPY |                                         |
|            | TS     | PIPATMPZ |                                         |
|            | CS     | STEERBIT | # CLEAR STEERSW PRIOR TO DVMON.         |
|            | MASK   | FLAGWRD2 |                                         |
|            | TS     | FLAGWRD2 |                                         |
|            | CAF    | IDLEFBIT | # IS THE IDLE FLAG SET?                 |
|            | MASK   | FLAGWRD7 |                                         |
|            | CCS    | A        |                                         |
|            | TCF    | NODVMON1 | # IDLEFLAG = 1, HENCE SET AUXFLAG TO 0. |
|            | CS     | FLAGWRD6 |                                         |
|            | MASK   | AUXFLBIT |                                         |
|            | CCS    | A        |                                         |
|            | TCF    | NODVMON2 | # AUXFLAG = 0, HENCE SET AUXFLAG TO 1.  |
| DVMON      | CS     | DVTHRUSH |                                         |
|            | AD     | ABDELV   |                                         |
|            | EXTEND |          |                                         |
|            | BZMF   | LOTHRUST |                                         |
|            | CS     | FLAGWRD2 | # SET STEERSW.                          |
|            | MASK   | STEERBIT |                                         |
|            | ADS    | FLAGWRD2 |                                         |
| DVCNTSET   | CAF    | ONE      | # ALLOW TWO PASSES MAXIMUM NOW THAT     |
| # Page 862 | TS     | DVCNTR   | # THRUST HAS BEEN DETECTED.             |



|          |        |          |                             |
|----------|--------|----------|-----------------------------|
|          | CA     | FLGWRD10 | # BRANCH IF APSFLAG IS SET. |
|          | MASK   | APSFLBIT |                             |
|          | CCS    | A        |                             |
|          | TCF    | USEJETS  |                             |
|          | CA     | BIT9     | # CHECK GIMBAL FAIL BIT     |
|          | EXTEND |          |                             |
|          | RAND   | CHAN32   |                             |
|          | EXTEND |          |                             |
|          | BZF    | USEJETS  |                             |
| USEGTS   | CS     | USEQRJTS |                             |
|          | MASK   | DAPBOOLS |                             |
|          | TS     | DAPBOOLS |                             |
|          | TCF    | SERVOUT  |                             |
| NODVMON1 | CS     | AUXFLBIT | # SET AUXFLAG TO 0.         |
|          | MASK   | FLAGWRD6 |                             |
|          | TS     | FLAGWRD6 |                             |
|          | TCF    | USEJETS  |                             |
| NODVMON2 | CS     | FLAGWRD6 | # SET AUXFLAG TO 1.         |
|          | MASK   | AUXFLBIT |                             |
|          | ADS    | FLAGWRD6 |                             |
|          | TCF    | USEJETS  |                             |
| LOTHRUST | TC     | QUIKFAZ5 |                             |
|          | CCS    | DVCNTR   |                             |
|          | TCF    | DECCNTR  |                             |
|          | CCS    | PHASE4   | # COMFAIL JOB ACTIVE?       |
|          | TCF    | SERVOUT  | # YES: WON'T NEED ANOTHER.  |
|          | TC     | PHASCHNG | # 4.37SPOT FOR COMFAIL.     |
|          | OCT    | 00374    |                             |
|          | CAF    | PRI025   |                             |
|          | TC     | NOVAC    |                             |
|          | EBANK= | WHICH    |                             |
|          | 2CADR  | COMFAIL  |                             |
|          | TCF    | SERVOUT  |                             |
| DECCNTR  | TS     | DVCNTR1  |                             |
|          | TC     | QUIKFAZ5 |                             |
|          | CA     | DVCNTR1  |                             |

|            |        |            |                                          |
|------------|--------|------------|------------------------------------------|
|            | TS     | DVCNTR     |                                          |
|            | INHINT |            |                                          |
| # Page 863 | TC     | IBNKCALL   | # IF THRUST IS LOW, NO STEERING IS DONE  |
|            | CADR   | STOPRATE   |                                          |
| USEJETS    | CS     | DAPBOOLS   | # AND THE DESIRED RATES ARE SET TO ZERO. |
|            | MASK   | USEQRJTS   |                                          |
|            | ADS    | DAPBOOLS   |                                          |
| SERVOUT    | RELINT |            |                                          |
|            | TC     | BANKCALL   |                                          |
|            | CADR   | 1/ACCS     |                                          |
|            | CA     | PRIORITY   |                                          |
|            | MASK   | LOW9       |                                          |
|            | TS     | PUSHLOC    |                                          |
|            | ZL     |            |                                          |
|            | DXCH   | FIXLOC     | # FIXLOC AND DVFIND                      |
|            | TC     | QUIKFAZ5   |                                          |
|            | EXTEND |            | # EXIT TO SELECTED ROUTINE WHETHER THERE |
|            | DCA    | AVGEXIT    | # IS THRUST OR NOT. THE STATE OF STEERSW |
|            | DXCH   | Z          | # WILL CONVEY THIS INFORMATION.          |
| XNBPIPAD   | ECADR  | XNBPIP     |                                          |
|            | BANK   | 32         |                                          |
|            | SETLOC | SERV2      |                                          |
|            | BANK   |            |                                          |
|            | COUNT* | \$\$/SERV  |                                          |
| AVGEND     | CA     | PIPTIME +1 | # FINAL AVERAGE G EXIT                   |
|            | TS     | 1/PIPADT   | # SET UP FREE FALL GYRO COMPENSATION.    |
|            | TC     | UPFLAG     | # SET DRIFT FLAG.                        |
|            | ADRES  | DRIFTFLG   |                                          |
|            | TC     | BANKCALL   |                                          |
|            | CADR   | PIPFREE    |                                          |
|            | CS     | BIT9       |                                          |
|            | EXTEND |            |                                          |
|            | WAND   | DSALMOUT   |                                          |
|            | TC     | 2PHSCHNG   |                                          |
|            | OCT    | 5          | # GROUP 5 OFF                            |
|            | OCT    | 05022      | # GROUP 2 ON                             |

July 12, 2016

1731

```
OCT 20000

TC INTPRET
SET CLEAR
 NOR29FLG # SHUT OFF R29 WHEN SERVICER ENDS.
 SWANDISP # SHUT OFF R10 WHEN SERVICER ENDS.
CLEAR CALL # RESET MUNFLAG.
 MUNFLAG

Page 864

 AVETOMID
CLEAR EXIT
 V37FLAG

AVERTRN CA OUTROUTE # RETURN TO DESIRED POINT.
 TC BANKJUMP

OUTGOAVE = AVERTRN
DVCNTR1 = MASS1

Page 865

SETLOC SERV3
BANK
COUNT* $$/SERV

SERVIDLE EXTEND
 DCA SVEXTADR
 DXCH AVGEXIT

 CS FLAGWRD7 # DISCONNECT THE DELTA-V MONITOR
 MASK IDLEFBIT
 ADS FLAGWRD7

 CAF LRBYBIT # TERMINATE R12 IS RUNNING.
 TS FLGWRD11

 EXTEND
 DCA NEG0
 DXCH -PHASE1

 CA FLAGWRD6 # DO NOT TURN OFF PHASE 2 IF MUNFLAG SET.
 MASK MUNFLBIT
 CCS A
 TCF +4

 EXTEND
 DCA NEG0
 DXCH -PHASE2
```

July 12, 2016

```

+4 EXTEND
 DCA NEG0
 DXCH -PHASE3

 EXTEND
 DCA NEG0
 DXCH -PHASE6

 CAF OCT33 # 4.33SPOT FOR GOP00FIX
 TS L
 COM
 DXCH -PHASE4

 TCF WHIMPER # PERFORM A SOFTWARE RESTART AND PROCEED
 # TO GOTOP00H WHILE SERVICER CONTINUES TO
 # RUN, ALBEIT IN A GROUND STATE WHERE
 # ONLY STATE-VECTOR DEPENDENT FUNCTIONS
 # ARE MAINTAINED.

 EBANK= DVCNTR

Page 866
SVEXTADR 2CADR SERVEXIT

 BANK 32
 SETLOC SERV
 BANK
 COUNT* $$ /SERV

SERVEXIT TC PHASCHNG
 OCT 00035

+2 TCF ENDOFJOB

 BANK 23
 SETLOC NORMLIZ
 BANK

 COUNT* $$ /SERV

Page 867
NORMLIZE TC INTPRET
 VLOAD BOFF
 RN1
 MUNFLAG
 NORMLIZ1

```

July 12, 2016

1733

|            |        |            |                                |
|------------|--------|------------|--------------------------------|
|            | VSL6   | MXV        |                                |
|            |        | REFSMMAT   |                                |
|            | STCALL | R          |                                |
|            |        | MUNGRAV    |                                |
|            | VLOAD  | VSL1       |                                |
|            |        | VN1        |                                |
|            | MXV    |            |                                |
|            |        | REFSMMAT   |                                |
|            | STOVL  | V          |                                |
|            |        | V(CSM)     |                                |
|            | VXV    | UNIT       |                                |
|            |        | R(CSM)     |                                |
|            | STORE  | UHYP       |                                |
| ASCSPOT    | EXIT   |            |                                |
|            | EXTEND |            | # MAKE SURE GROUP 2 IS OFF     |
|            | DCA    | NEGO       |                                |
|            | DXCH   | -PHASE2    |                                |
|            | TC     | POSTJUMP   |                                |
|            | CADR   | NORMLIZ2   |                                |
|            | BANK   | 33         |                                |
|            | SETLOC | SERVICES   |                                |
|            | BANK   |            |                                |
|            | COUNT* | \$\$/SERV  |                                |
| NORMLIZ1   | CALL   |            |                                |
|            |        | CALCGRAV   |                                |
|            | EXIT   |            |                                |
| NORMLIZ2   | CA     | EIGHTEEN   |                                |
|            | TC     | COPYCYC +1 | # DO NOT COPY MASS IN NORMLIZE |
|            | TC     | ENDOFJOB   |                                |
| COPYCYC    | CA     | OCT24      | # DEC 20                       |
| +1         | INHINT |            |                                |
| +2         | MASK   | NEG1       | # REDUCE BY 1 IF ODD           |
|            | TS     | ITEMP1     |                                |
|            | EXTEND |            |                                |
|            | INDEX  | ITEMP1     |                                |
|            | DCA    | RN1        |                                |
|            | INDEX  | ITEMP1     |                                |
| # Page 868 | DXCH   | RN         |                                |
|            | CCS    | ITEMP1     |                                |
|            | TCF    | COPYCYC +2 |                                |

```

 TC Q # RETURN UNDER INHINT

EIGHTEEN DEC 18

Page 869
***** PIPA READER *****
MOD NO. 00 BY D. LICKLY, DEC. 9 1966
#
FUNCTIONAL DESCRIPTION
SUBROUTINE TO READ PIPA COUNTERS, TRYING TO BE VERY CAREFUL SO THAT WILL BE P
PIPA READINGS ARE STORED IN THE VECTOR DELV. THE HIGH ORDER PART OF EACH COM
RESTARTS BEGIN AT REREADAC.
#
AT THE END OF THE PIPA READER THE CDUS ARE READ AND STORED AS A
VECTOR IN CDUTEMP. THE HIGH ORDER PART OF EACH COMPONENT CONTAINS
THE CDU READING IN 25 COMP IN THE ORDER CDUX,Y,Z. THE THRUST
VECTOR ESTIMATOR IN FINDCDUD REQUIRES THE CDUS BE READ AT PIPTIME.
#
CALLING SEQUENCE AND EXIT
CALL VIA TC, ISWCALL, ETC.
EXIT IS VIA Q.
#
INPUT
INPUT IS THROUGH THE COUNTERS PIPAX, PIPAY, PIPAZ, AND TIME2.
#
OUTPUT
HIGH ORDER COMPONENTS OF THE VECTOR DELV CONTAIN THE PIPA READINGS.
PIPTIME CONTAINS TIME OF PIPA READING.
#
DEBRIS (ERASABLE LOCATIONS DESTROYED BY PROGRAM)
TEMX, TEMY, TEMZ, PIPAGE

 BANK 37
 SETLOC SERV1
 BANK

 COUNT* $$/SERV

PIPASR EXTEND
Page 870
 DCA TIME2
 DXCH PIPTIME1 # CURRENT TIME POSITIVE VALUE
 +3 CS ZERO # INITIALIZE THESE AT NEG. ZERO.
 TS TEMX
 TS TEMY
 TS TEMZ

```

July 12, 2016

1735

```

CA ZERO
TS DELVZ
TS DELVZ +1
TS DELVY
TS DELVY +1
TS DELVX +1
TS PIPAGE # SHOW PIPA READING IN PROGRESS

REPIP1 EXTEND
DCS PIPAX # X AND Y PIPS READ
DXCH TEMX
DXCH PIPAX # PIPAS SET TO NEG ZERO AS READ.
TS DELVX
LXCH DELVY

REPIP3 CS PIPAZ # REPEAT PROCESS FOR Z PIP
XCH TEMZ
XCH PIPAZ
DODELVZ TS DELVZ

REPIP4 EXTEND # COMPUTE GUIDANCE PERIOD
DCA PIPTIME1
DXCH PGUIDE
EXTEND
DCS PIPTIME
DAS PGUIDE

CA CDUX # READ CDUS INTO HIGH ORDER CDUTEMPS
TS CDUTEMPX
CA CDUY
TS CDUTEMPY
CA CDUZ
TS CDUTEMPZ
CA DELVX
TS PIPATMPX
CA DELVY
TS PIPATMPY
CA DELVZ
TS PIPATMPZ

TC Q

Page 871
REREADAC CCS PIPAGE
TCF READACCS # PIP READING NOT STARTED. GO TO BEGINNING
```

|            |        |          |                                         |
|------------|--------|----------|-----------------------------------------|
|            | CAF    | DONEADR  | # SET UP RETURN FROM PIPASR             |
|            | TS     | Q        |                                         |
|            | CCS    | DELVZ    |                                         |
|            | TCF    | REPIP4   | # Z DONE, GO DO CDUS                    |
|            | TCF    | +3       | # Z NOT DONE, CHECK Y.                  |
|            | TCF    | REPIP4   |                                         |
|            | TCF    | REPIP4   |                                         |
|            | ZL     |          |                                         |
|            | CCS    | DELVY    |                                         |
|            | TCF    | +3       |                                         |
|            | TCF    | CHKTEMX  | # Y NOT DONE, CHECK X.                  |
|            | TCF    | +1       |                                         |
|            | LXCH   | PIPAZ    | # Y DONE, ZERO Z PIP.                   |
|            | CCS    | TEMZ     |                                         |
|            | CS     | TEMZ     | # TEMZ NOT = -0, CONTAINS -PIPAZ VALUE. |
|            | TCF    | DODELVZ  |                                         |
|            | TCF    | -2       |                                         |
|            | LXCH   | DELVZ    | # TEMZ = -0, L HAS ZPIP VALUE.          |
|            | TCF    | REPIP4   |                                         |
| CHKTEMX    | CCS    | TEMX     | # HAS THIS CHANGED                      |
|            | CS     | TEMX     | # YES                                   |
|            | TCF    | +3       | # YES                                   |
|            | TCF    | -2       | # YES                                   |
|            | TCF    | REPIP1   | # NO                                    |
|            | TS     | DELVX    |                                         |
|            | CS     | TEMY     |                                         |
|            | TS     | DELVY    |                                         |
|            | CS     | ZERO     | # ZERO X AND Y PIPS                     |
|            | DXCH   | PIPAX    | # L STILL ZERO FROM ABOVE               |
|            | TCF    | REPIP3   |                                         |
| DONEADR    | GENADR | PIPSDONE |                                         |
| # Page 872 |        |          |                                         |
|            | BANK   | 33       |                                         |
|            | SETLOC | SERVICES |                                         |
|            | BANK   |          |                                         |



July 12, 2016

1737

```

COUNT* $$/SERV

TMPTOSPT CA CDUTEMPY # THIS SUBROUTINE, CALLED BY AN RTB FROM
 TS CDUSPOTY # INTERPRETIVE, LOADS THE CDUS CORRESPON-
 CA CDUTEMPZ # DING TO PIPTIME INTO THE CDUSPOT VECTOR.
 TS CDUSPOTZ
 CA CDUTEMPX
 TS CDUSPOTX
 TC Q

LRHTASK IS A WAITLIST TASK SET BY READACCS DURING THE DESCENT BRAKING
PHASE WHEN THE ALT TO THE LUNAR SURFACE IS LESS THAN 25,000 FT. THIS
TASK CLEARS THE ALTITUDE MEASUREMENT MADE DISCRETE AND INITIATES THE
LANDING RADAR MEASUREMENT JOB (LRHJOB) TO TAKE A ALTITUDE MEASUREMENT
50 MS PRIOR TO THE NEXT READACCS TASK.

 BANK 21
 SETLOC R10
 BANK

COUNT* $$/SERV

LRHTASK CS FLGWRD11
 MASK LRBYBIT
 EXTEND
 BZF GRP2OFF # LR BYPASS SET -- BYPASS ALL LR READING.

 CA READLBIT
 MASK FLGWRD11 # IS READLR FLAG SET?
 EXTEND
 BZF GRP2OFF # NO. BYPASS LR READ.

 CS FLGWRD11
 MASK NOLRRBIT # IS LR READ INHIBITED?
 EXTEND
 BZF GRP2OFF # YES. BYPASS LR READ.

 CA PRI032 # LR READ OK. SET JOB TO DO IT
 TC NOVAC # ABOUT 50 MS. PRIOR TO PIPA READ.
 EBANK= HMEAS
 2CADR LRHJOB

GRP2OFF EXTEND
 DCA NEG0
 DXCH -PHASE2
 TCF R10,R11A
```

BANK 33  
 SETLOC SERVICES  
 BANK

# Page 873

COUNT\* \$\$/SERV

# HIGATASK IS ENTERED APPROXIMATELY 6 SECS PRIOR TO HIGATE DURING THE  
 # DESCENT PHASE. HIGATASK SETS THE HIGATE FLAG (BIT11) AND THE LR INHIBIT  
 # FLAG (BIT10) IN LRSTAT. THE HIGATJOB IS SET UP TO REPOSITION THE LR  
 # ANTENNA FROM POSITION 1 TO POSITION 2. IF THE REPOSITIONING IS  
 # SUCCESSFUL THE ALT BEAM AND VELOCITY BEAMS ARE TRANSFORMED TO THE NEW  
 # ORIENTATION IN NB COORDINATES AND STORED IN ERASABLE.

|          |          |                                   |
|----------|----------|-----------------------------------|
| HIGATASK | INHINT   |                                   |
| CS       | PRI03    | # SET HIGATE AND LR INHIBIT FLAGS |
| MASK     | FLGWRD11 |                                   |
| AD       | PRI03    |                                   |
| TS       | FLGWRD11 |                                   |
| CAF      | PRI032   |                                   |
| TC       | FINDVAC  | # SET LR POSITIONING JOB (POS2)   |
| EBANK=   | HMEAS    |                                   |
| 2CADR    | HIGATJOB |                                   |
| TCF      | CONTSERV | # CONTINUE SERVICER               |

# Page 874

# MUNRETRN IS THE RETURN LOC FROM SPECIAL AVE G ROUTINE (MUNRVG)

|          |          |                                            |
|----------|----------|--------------------------------------------|
| MUNRETRN | EXIT     |                                            |
| CS       | FLGWRD11 |                                            |
| MASK     | LRBYBIT  |                                            |
| EXTEND   |          |                                            |
| BZF      | COPYCYC1 | # BYPASS LR LOGIC IF BIT15 IS SET.         |
| CA       | READLBIT | # SEE IF ALT < 35000 FT LAST CYCLE         |
| MASK     | FLGWRD11 |                                            |
| EXTEND   |          |                                            |
| BZF      | 35KCHK   | # ALT WAS > 35000 FT LAST CYCLE    CHK NOW |
| CAF      | XORFLBIT | # WERE WE BELOW 30000 FT LAST PASS?        |
| MASK     | FLGWRD11 |                                            |
| EXTEND   |          |                                            |
| BZF      | XORCHK   | # NO -- TEST THIS PASS                     |
| HITEST   | CAF      | PSTHIBIT    # CHECK FOR HIGATE             |

July 12, 2016

1739

```

 MASK FLGWRD11
 EXTEND
 BZF HIGATCHK # NOT AT HIGATE LAST CYCLE -- CHK THIS CYCLE

POS2CHK CAF BIT7 # VERIFY LR IN POS2
 EXTEND
 RAND CHAN33
 EXTEND
 BZF UPDATCHK # IT IS -- CHECK FOR LR UPDATE
 CAF BIT13
 EXTEND
 RAND CHAN12
 EXTEND
 BZF LRPOSALM # LR NOT IN POS2 OR REPOSITIONING -- BAD
 TCF CONTSERV # LR BEING REPOSITIONED -- CONTINUE SERV

HIGATCHK CA TTF/8 # IS TTF > CRITERION? (TTF IS NEGATIVE)
 AD RPCRTIME
 EXTEND
 BZMF POS1CHK # NO

 CA EBANK4 # MUST SWITCH EBANKS
 XCH EBANK
 TS L # SAVE IN L

 EBANK= XNBPIP
 CS XNBPIP # UXBXP IN GSOP CH5
 EBANK= DVCNTR
 LXCH EBANK # RESTORE EBANK
 AD RPCRTQSW # QSW - UXBXP

Page 875

 EXTEND
 BZMF HIGATASK # IF UXBXP > QSW, THEN REPOSITION

POS1CHK CAF BIT6 # HIGATE NOT IN SIGHT -- DO POS1 CHK
 EXTEND
 RAND 33
 EXTEND
 BZF UPDATCHK # LR IN POS1 -- CHECK FOR LR UPDATE

LRPOSALM TC ALARM # LR NOT IN PROPER POS-ALARM-BYPASS UPDATE
 OCT 511 # AND CONTINUE SERVICER

CONTSERV INHINT
 CS BITS4-7
 MASK FLGWRD11 # CLEAR LR MEASUREMENT MADE DISCRETES.
```

|                        |        |          |                                            |
|------------------------|--------|----------|--------------------------------------------|
|                        | TS     | FLGWRD11 |                                            |
|                        | TC     | IBNKCALL | # SET LR LITES PROPERLY                    |
|                        | CADR   | R12LITES |                                            |
| # Page 876<br>COPYCYC1 | TC     | QUIKFAZ5 |                                            |
| R29?                   | CA     | FLAGWRD3 |                                            |
|                        | MASK   | NR29&RDR |                                            |
|                        | CCS    | A        | # IS NOR29FLG OR READRFLG SET?             |
|                        | TCF    | R29NODES | # YES, SO DON'T DESIGNATE.                 |
|                        | CA     | RADMODES | # NO, SO R29 IS CALLED FOR.                |
|                        | MASK   | OCT10002 | # IS THE RR NOT ZEROING ITS CDUS, AND      |
|                        | CCS    | A        | # IS THE RENDEZVOUS RADAR IN AUTO MODE?    |
|                        | TCF    | R29NODES | # NO, SO DON'T DESIGNATE.                  |
|                        | CA     | RADMODES |                                            |
|                        | MASK   | PRI022   |                                            |
|                        | CCS    | A        | # IS RR REPOSITIONING OR REMODING?         |
|                        | TCF    | NOR29NOW | # YES: COME BACK IN 2 SECONDS & TRY AGAIN. |
|                        | TCF    | R29      |                                            |
| R29NODES               | INHINT |          | # R29 NOT ALLOWED THIS CYCLE.              |
|                        | CS     | DESIGBIT | # SHOW THAT DESIGNATION IS OFF.            |
|                        | MASK   | RADMODES |                                            |
|                        | TS     | RADMODES |                                            |
| NOR29NOW               | TC     | INTPRET  | # INTERPRET DOES A RELINT.                 |
|                        | VLOAD  | ABVAL    | # MPAC = ABVAL( NEW SM. POSITION VECTOR )  |
|                        |        | R1S      |                                            |
|                        | PUSH   | DSU      | # (2)                                      |
|                        |        | /LAND/   |                                            |
|                        | STORE  | HCALC    | # NEW HCALC*2(24)M.                        |
|                        | STORE  | HCALC1   |                                            |
|                        | DMPR   | RTB      |                                            |
|                        |        | ALTCONV  |                                            |
|                        |        | SGNAGREE |                                            |
|                        | STOVL  | ALTBITS  | # ALTITUDE FOR R10 IN BIT UNITS.           |
|                        |        | UNIT/R/  |                                            |
|                        | VXV    | UNIT     |                                            |
|                        |        | UHYP     |                                            |
|                        | STOVL  | UHYP     | # DOWNRANGE HALF-UNIT VECTOR FOR R10.      |
|                        |        | R1S      |                                            |

July 12, 2016

1741

```
Page 877

VXM VSR4
 REFSMMAT
STOVL RN1 # TEMP. REF. POSITION VECTOR*2(29)M.
 V1S
VXM VSL1
 REFSMMAT
STOVL VN1 # TEMP. REF. VELOCITY VECTOR 2(7) M/CS.
 UNIT/R/
VXV ABVAL

 V1S
SL1 DSQ
DDV
DMPR RTB
 ARCONV1
 SGNAGREE

COPYCYC2 EXIT # LEAVE ALTITUDE RATE COMPENSATION IN MPAC
 INHINT
CA UNIT/R/ # UPDATE RUNIT FOR R10.
TS RUNIT
CA UNIT/R/ +2
TS RUNIT +1
CA UNIT/R/ +4
TS RUNIT +2
CA MPAC # LOAD NEW DALTRATE FOR R10.
TS DALTRATE

EXTEND
DCA R1S
DXCH R
EXTEND
DCA R1S +2
DXCH R +2
EXTEND
DCA R1S +4
DXCH R +4
EXTEND
DCA V1S
DXCH V
EXTEND
DCA V1S +2
DXCH V +2
EXTEND
DCA V1S +4
DXCH V +4
```

TCF COPYCYCL # COMPLETE THE COPYCYCL.

# Page 878

# ALTCHK COMPARES CURRENT ALTITUDE (IN HCALC) WITH A SPECIFIED ALTITUDE FROM A TABLE  
# ITS CALLING SEQUENCE IS AS FOLLOWS:-

#

|   |     |                                                   |          |                  |
|---|-----|---------------------------------------------------|----------|------------------|
| # | L   | CAF                                               | N        |                  |
| # | L+1 | TC                                                | BANKCALL |                  |
| # | L+2 | CADR                                              | ALTCHK   |                  |
| # | L+3 | RETURN HERE IF HCALC STILL > SPECIFIED CRITERION. |          | C(L) = +0.       |
| # | L+4 | RETURN HERE IF HCALC < OR = SPECIFIED CRITERION.  |          | C(A) = C(L) = +0 |

#

# ALTCHK MUST BE BANKCALLED EVEN FROM ITS OWN BANK. N IS THE LOCATION, RELATIVE TO  
# OF THE BEGINNING OF THE DP CONSTANT TO BE USED AS A CRITERION.

|        |        |           |                               |
|--------|--------|-----------|-------------------------------|
| ALTCHK | EXTEND |           |                               |
|        | INDEX  | A         |                               |
|        | DCA    | ALTCRIT   |                               |
|        | DXCH   | MPAC +1   |                               |
|        | EXTEND |           |                               |
|        | DCS    | HCALC     |                               |
|        | DAS    | MPAC +1   |                               |
|        | TC     | BRANCH +4 |                               |
|        | CAF    | ZERO      | # BETTER THAN A NOOP, PERHAPS |
|        | INCR   | BUF2      |                               |
|        | TCF    | SWRETURN  |                               |

|         |   |       |
|---------|---|-------|
| ALTCRIT | = | 25KFT |
|---------|---|-------|

|       |      |           |       |
|-------|------|-----------|-------|
| 25KFT | 2DEC | 7620 B-24 | # (0) |
|-------|------|-----------|-------|

|       |      |            |       |
|-------|------|------------|-------|
| 50KFT | 2DEC | 15240 B-24 | # (2) |
|-------|------|------------|-------|

|      |      |            |       |
|------|------|------------|-------|
| 50FT | 2DEC | 15.24 B-24 | # (4) |
|------|------|------------|-------|

|       |      |           |       |
|-------|------|-----------|-------|
| 30KFT | 2DEC | 9144 B-24 | # (6) |
|-------|------|-----------|-------|

|          |     |           |                            |
|----------|-----|-----------|----------------------------|
| 2KFT/SEC | DEC | 6.096 B-7 | # 2000 FT/SEC AT 2(7) M/CS |
|----------|-----|-----------|----------------------------|

# (A remark was likely to be needed here to explain XORCHK) 4/Jun/09,FB

|        |      |          |                          |
|--------|------|----------|--------------------------|
| XORCHK | CAF  | SIX      | # ARE WE BELOW 30000 FT? |
|        | TC   | BANKCALL |                          |
|        | CADR | ALTCHK   |                          |
|        | TCF  | HITEST   | # CONTINUE LR UPDATE     |

July 12, 2016

1743

|       |          |                                |
|-------|----------|--------------------------------|
| TC    | UPFLAG   | # YES: INHIBIT X-AXIS OVERRIDE |
| ADRES | XOVINFLG |                                |
| TC    | UPFLAG   |                                |
| ADRES | XORFLG   |                                |
| TCF   | HITEST   | # CONTINUE LR UPDATE           |

|        |     |     |                          |
|--------|-----|-----|--------------------------|
| 35KCHK | CAF | TWO | # ARE WE BELOW 35000 FT? |
|--------|-----|-----|--------------------------|

# Page 879

|       |          |                                         |
|-------|----------|-----------------------------------------|
| TC    | BANKCALL |                                         |
| CADR  | ALTCHK   |                                         |
| TCF   | CONTSERV |                                         |
| TC    | UPFLAG   |                                         |
| ADRES | READLR   | # SET READLR FLAG TO ENABLE LR READING. |
| TCF   | CONTSERV |                                         |

# Page 880

# \*\*\*\*\*

|          |       |          |                                         |      |
|----------|-------|----------|-----------------------------------------|------|
| CALCGRAV | UNIT  | PUSH     | # SAVE UNIT/R/ IN PUSHLIST              | (18) |
|          | STORE | UNIT/R/  |                                         |      |
|          | LXC,1 | SLOAD    | # RTX2 = 0 IF EARTH ORBIT, =2 IF LUNAR. |      |
|          |       | RTX2     |                                         |      |
|          |       | RTX2     |                                         |      |
|          | DCOMP | BMN      |                                         |      |
|          |       | CALCGRV1 |                                         |      |
|          | VLOAD | DOT      | #                                       | (12) |
|          |       | UNITZ    |                                         |      |
|          |       | UNIT/R/  |                                         |      |
|          | SL1   | PUSH     | #                                       | (14) |
|          | DSQ   | BDSU     |                                         |      |
|          |       | DP1/20   |                                         |      |
|          | PDDL  | DDV      |                                         |      |
|          |       | RESQ     |                                         |      |
|          |       | 34D      | # (RN)SQ                                |      |
|          | STORE | 32D      | # TEMP FOR (RE/RN)SQ                    |      |
|          | DMP   | DMP      |                                         |      |
|          |       | 20J      |                                         |      |
|          | VXSC  | PDDL     |                                         |      |
|          |       | UNIT/R/  |                                         |      |
|          | DMP   | DMP      |                                         |      |
|          |       | 2J       |                                         |      |
|          |       | 32D      |                                         |      |
|          | VXSC  | VSL1     |                                         |      |
|          |       | UNITZ    |                                         |      |
|          | VAD   | STADR    |                                         |      |

|            |        |          |                                             |
|------------|--------|----------|---------------------------------------------|
|            | STORE  | UNITGOBL |                                             |
|            | VAD    | PUSH     | # MPAC = UNIT GRAVITY VECTOR. (18)          |
| CALCGRV1   | DLOAD  | NORM     | # PERFORM A NORMALIZATION ON RMAGSQ IN      |
|            |        | 34D      | # ORDER TO BE ABLE TO SCALE THE MU FOR      |
|            |        | X2       | # MAXIMUM PRECISION.                        |
|            | BDDV*  | SLR*     |                                             |
|            |        | -MUDT,1  |                                             |
|            |        | 0 -21D,2 |                                             |
|            | VXSC   | STADR    |                                             |
|            | STORE  | GDT1/2   | # SCALED AT 2(+7) M/CS                      |
|            | RVQ    |          |                                             |
| CALCRVG    | VLOAD  | VXM      |                                             |
|            |        | DELV     |                                             |
|            |        | REFSMMAT |                                             |
|            | VXSC   | VSL1     |                                             |
|            |        | KPIP1    |                                             |
|            | STORE  | DELVREF  |                                             |
|            | VSR1   | PUSH     |                                             |
|            | VAD    | PUSH     | # (DV-OLDGDT)/2 TO PD SCALED AT 2(+7) M/CS. |
| # Page 881 |        | GDT/2    |                                             |
|            | VAD    | PDDL     |                                             |
|            |        | VN       |                                             |
|            |        | PGUIDE   |                                             |
|            | SL     | VXSC     |                                             |
|            |        | 6D       |                                             |
|            | VAD    | STQ      |                                             |
|            |        | RN       |                                             |
|            |        | 31D      |                                             |
|            | STCALL | RN1      | # TEMP STORAGE OF RN SCALED 2(+29) M        |
|            |        | CALCGRAV |                                             |
|            | VAD    | VAD      |                                             |
|            | VAD    |          |                                             |
|            |        | VN       |                                             |
|            | STCALL | VN1      | # TEMP STORAGE OF VN SCALED 2(+7) M/CS      |
|            |        | 31D      |                                             |
| DP1/20     | 2DEC   | 0.05     |                                             |
| SHIFT11    | 2DEC   | 1 B-11   |                                             |

# Page 882

\*\*\*\*\*  
# MUNRVG IS A SPECIAL AVERAGE G INTEGRATION ROUTINE USED BY THRUSTING  
# PROGRAMS WHICH FUNCTION IN THE VICINITY OF AN ASSUMED SPHERICAL MOON.



July 12, 2016

1745

# THE INPUT AND OUTPUT QUANTITIES ARE REFERENCED TO THE STABLE MEMBER  
# COORDINATE SYSTEM.

|        |        |          |                                           |
|--------|--------|----------|-------------------------------------------|
| RVBOTH | VLOAD  | PUSH     |                                           |
|        |        | G(CSM)   |                                           |
|        | VAD    | PDDL     |                                           |
|        |        | V(CSM)   |                                           |
|        |        | PGUIDE   |                                           |
|        | DDV    | VXSC     |                                           |
|        |        | SHIFT11  |                                           |
|        | VAD    |          |                                           |
|        |        | R(CSM)   |                                           |
|        | STCALL | R1S      |                                           |
|        |        | MUNGRAV  |                                           |
|        | VAD    | VAD      |                                           |
|        |        | V(CSM)   |                                           |
|        | STADR  |          |                                           |
|        | STORE  | V1S      |                                           |
|        | EXIT   |          |                                           |
|        | TC     | QUIKFAZ5 |                                           |
|        | TC     | INTPRET  |                                           |
|        | VLOAD  |          |                                           |
|        |        | GDT1/2   |                                           |
|        | STOVL  | G(CSM)   |                                           |
|        |        | R1S      |                                           |
|        | STOVL  | R(CSM)   |                                           |
|        |        | V1S      |                                           |
|        | STORE  | V(CSM)   |                                           |
|        | EXIT   |          |                                           |
|        | TC     | QUIKFAZ5 |                                           |
|        | TC     | INTPRET  |                                           |
| MUNRVG | VLOAD  | VXSC     |                                           |
|        |        | DELV     |                                           |
|        |        | KPIP2    |                                           |
|        | PUSH   | VAD      | # 1ST PUSH: DELV IN UNITS OF 2(8) M/CS    |
|        |        | GDT/2    |                                           |
|        | PUSH   | VAD      | # 2ND PUSH: (DELV + GDT)/2, UNITS OF 2(7) |
|        |        | V        | # (12)                                    |
|        | PDDL   | DDV      |                                           |
|        |        | PGUIDE   |                                           |
|        |        | SHIFT11  |                                           |
|        | VXSC   |          |                                           |
|        | VAD    |          |                                           |
|        |        | R        |                                           |
|        | STCALL | R1S      | # STORE R SCALED AT 2(+24) M              |
|        |        | MUNGRAV  |                                           |

# Page 883

|          |        |            |                                            |
|----------|--------|------------|--------------------------------------------|
|          | VAD    | VAD        |                                            |
|          | VAD    |            |                                            |
|          |        | V          |                                            |
|          | STORE  | V1S        | # STORE V SCALED AT 2(+7) M/CS.            |
|          | ABVAL  |            |                                            |
|          | STOVL  | ABVEL      | # STORE SPEED FOR LR AND DISPLAYS.         |
|          |        | UNIT/R/    |                                            |
|          | DOT    | SL1        |                                            |
|          |        | V1S        |                                            |
|          | STOVL  | HDOTDISP   | # HDOT = V. UNIT(R)*2(7) M/CS.             |
|          |        | R1S        |                                            |
|          | VXV    | VSL2       |                                            |
|          |        | WM         |                                            |
|          | STODL  | DELVS      | # LUNAR ROTATION CORRECTON TERM*2(5) M/CS. |
|          |        | 36D        |                                            |
|          | DSU    |            |                                            |
|          |        | /LAND/     |                                            |
|          | STCALL | HCALC      | # FOR NOW, DISPLAY WHETHER POS OR NEG      |
|          |        | MUNRETRN   |                                            |
| MUNGRAV  | UNIT   |            | # AT 36D HAVE ABVAL(R), AT 34D R.R         |
|          | STODL  | UNIT/R/    |                                            |
|          |        | 34D        |                                            |
|          | SL     | BDDV       |                                            |
|          |        | 6D         |                                            |
|          |        | -MUDTMUN   |                                            |
|          | DMP    | VXSC       |                                            |
|          |        | SHIFT11    |                                            |
|          |        | UNIT/R/    |                                            |
|          | STORE  | GDT1/2     | # 1/2GDT SCALED AT 2(7) M/CS.              |
|          | RVQ    |            |                                            |
| 1.95SECS | DEC    | 195        |                                            |
| 7.5      | 2DEC   | .02286 B-6 | # 7.5 FT/SEC AT 2(6) M/CS                  |
| 2SEC(18) | 2DEC   | 200 B-18   |                                            |
| 2SEC(28) | 2OCT   | 0000000310 | # 2SEC AT 2(28)                            |
| 4SEC(28) | 2DEC   | 400 B-28   |                                            |
| BITS4-7  | OCT    | 110        |                                            |

# Page 884

|          |     |          |                               |
|----------|-----|----------|-------------------------------|
| UPDATCHK | CAF | NOLRRBIT | # SEE IF LR UPDATE INHIBITED. |
|----------|-----|----------|-------------------------------|

July 12, 2016

1747

```

 MASK FLGWRD11
 CCS A
 TCF CONTSERV # IT IS -- NO LR UPDATE
 CAF RANGEDBIT # NO INHIBIT -- SEE ALT MEAS. THIS CYCLE.
 MASK FLGWRD11
 EXTEND
 BZF VMEASCHK # NO ALT MEAS THIS CYCLE -- CHECK FOR VEL

POSUPDAT CA FIXLOC # SET PUSHLIST TO ZERO
 TS PUSHLOC

 TC INTPRET
 VLOAD VXM
 HBEAMNB
 XNBPIP # HBEAM SM AT 2(2)
 PDVL VSL2 # STORE HBEAM IN PD 0-5
 V1S # SCALE V AT 2(5) M/CS
 VAD DOT
 DELVS # V RELATIVE TO SURFACE AT 2(5) M/CS
 OD # V ALONG HBEAM AT 2(7) M/CS.
 DMP EXIT
 RADSKAL # SCALE TO RADAR COUNTS X 5

 CS FLGWRD12 # TEST LR ALTITUDE SCALE FACTOR
 MASK ALTSCBIT
 EXTEND
 BZF +3 # BRANCH IF HIGH SCALE

 CA SKALSKAL # RESCALE IF LOW SCALE
 TC SHORTMP

+3 TC INTPRET
 DAD SL
 HMEAS
 7D
 DMP VXSC # SLANT RANGE AT 2(21), PUSH UP FOR HBEAM
 HSCAL # SLANT RANGE VECTOR AT 2(23) M
 DOT DSU
 UNIT/R/ # ALTITUDE AT 2(24) M
 HCALC # DELTA H AT 2(24) M
 STORE DELTAH
 EXIT

 CA FLGWRD11
 MASK PSTHIBIT
 EXTEND # DO NOT PERFORM DATA REASONABLENESS TEST
```

```

Page 885
BZF NOREASON # UNTIL AFTER HIGATE

TC INTERPRET
ABS DSU
 DELQFIX # ABS(DELTAH) - DQFIX 50 FT NOM
SL3 DSU # SCALE TO 2(21)
 HCALC # ABS(DELTAH) - (50 + HCALC/8) AT 2(21)
EXIT

INCR LRLCTR
TC BRANCH
TCF HFAIL # DELTA H TOO LARGE
TCF HFAIL # DELTA H TOO LARGE
TC DOWNFLAG # TURN OFF ALT FAIL LAMP
ADRES HFLSHFLG

NOREASON CS FLGWRD11
 MASK LRINHBIT
 CCS A
 TCF VMEASCHK # UPDATE INHIBITED -- TEST VELOCITY ANYWAY

 TC INTERPRET # DO POSITION UPDATE
 DLOAD SR4
 HCALC # RESCALE H TO 2(28)M
EXIT
EXTEND
DCA DELTAH # STORE DELTAH IN MPAC AND
DXCH MPAC # BRING HCALC INTO A,L
TC ALSIGNAG
EXTEND
BZF +2 # IF HIGH PART OF HCALC IS NON-ZERO, THEN
 TCF VMEASCHK # HCALC > HMAX,
 TS MPAC +2 # SO UPDATE IS BYPASSED
 # FOR LATER SHORTMP

 CS L # -H AT 2(14) M
 AD LRHMAX # HMAX - H
EXTEND
BZMF VMEASCHK # IF H >HMAX, BYPASS UPDATE
EXTEND
MP LRWH # WH(HMAX - H)
EXTEND
DV LRHMAX # WH(1 - H/HMAX)
TS MPTMP
TC SHORTMP2 # DELTAH (WH)(1 - H/HMAX) IN MPAC
TC INTERPRET # MODE IS DP FROM ABOVE
SL1

```

July 12, 2016

1749

```
Page 886

VXSC VAD
 UNIT/R/
 R1S # DELTAR = DH(WH) (1 - H/HMAX) UNIT/R/
STCALL GNUR
 MUNGRAV
EXIT

TC QUIKFAZ5

CA ZERO
RUPDATED TC GNURVST

VMEASCHK TC QUIKFAZ5 # RESTART AT NEXT LOCATION
 CS FLGWRD11
 MASK VELDABIT # IS V READING AVAILABLE?
 CCS A
 TCF VALTCHK # NO: SEE IF V READING TO BE TAKEN

VELUPDAT CS VSELECT # PROCESS VELOCITY DATA
 TS L
 ADS L # -2 VSELECT IN L
 AD L
 AD L # -6 VSELECT IN A
 INDEX FIXLOC
 DXCH X1 # X1 = -6 VSELECT, X2 = -2 VSELECT

 CA EBANK4
 TS EBANK
 EBANK= LRXCDU

 CA LRYCDU # STORE LRCDUS IN CDUSPOTS
 TS CDUSPOT
 CA LRZCDU
 TS CDUSPOT +2
 CA LRXCDU
 TS CDUSPOT +4

 TC BANKCALL
 CADR QUICTRIG # GET SINES AND COSINES FOR NBSM

 CA FIXLOC
 TS PUSHLOC # SET PD TO ZERO

 TC INTERPRET
 VLOAD* CALL
 VZBEAMNB,1 # CONVERT VBEAM FROM NB TO SM
```

# Page 887

```

 NBSM
PDDL SL # STORE IN PD 0-5
 VMEAS # LOAD VELOCITY MEASUREMENT
 12D
DMP* PUSH # SCALE TO M/CS AT 2(6)
 VZSCAL,2 # AND STORE IN PD 6-7

EXIT
CS ONE
TS MODE # CHANGE STORE MODE TO VECTOR

CA PIPTEM # STORE DELV IN MPAC

ZL
DXCH MPAC

CA PIPTEM +1
ZL
DXCH MPAC +3

CA PIPTEM +2
ZL
DXCH MPAC +5

CA EBANK7
TS EBANK # RESTORE EBANK 7
EBANK= DVCNTR
TC INTPRET
VXSC PDDL
 KPIP1 # SCALE DELV TO 2(7) M/CS AND PUSH
 LRVTIME # TIME OF DELV AT 2(28) CS
DSU DDV
 PIPTIME # TU - T(N-1)
 2SEC(28)
VXSC VSL1 # G(N-1)(TU - T(N-1))
 GDT/2 # SCALED AT 2(7) M/CS
VAD VAD # PUSH UP FOR DELV
 V # VU = V(N-1) + DELVU + G(N-1) DTU
VSL2 VAD # SCALE TO 2(5) M/CS AND SUBTRACT
 DELVS # MOON ROTATION.
PUSH ABVAL # STORE IN PD
SR4 DAD # ABS(VM)/8 + 7.5 AT 2(6)
 7.5
STOVL 20D # STORE IN 20D AND PICK UP VM
DOT BDSU # V(EST) AT 2(6)
 0 # DELTAV = VMEAS - V(EST)
PUSH ABS

```

July 12, 2016

1751

```

DSU EXIT # ABS(DV) - (7.5 + ABS(VM)/8))
 20D

INCR LRMCTR
TC BRANCH
TCF VFAIL # DELTA V TOO LARGE. ALARM
TCF VFAIL # DELTA V TOO LARGE. ALARM

TC DOWNFLAG # TURN OFF VEL FAIL LAMP
ADRES VFLSHFLG

CA FLGWRD11
MASK VXINHBIT
EXTEND
BZF VUPDAT # IF VX INHIBIT RESET, INCORPORATE DATA.

Page 888
TC DOWNFLAG
ADRES VXINH # RESET VX INHIBIT

CA VSELECT
AD NEG2 # IF VSELECT = 2 (X AXIS).
EXTEND # BYPASS UPDATE
BZF ENDVDAT

VUPDAT
CS FLGWRD11
MASK LRINHBIT
CCS A
TCF VALTCHK # UPDATE INHIBITED

TS MPAC +1

CA ABVEL # STORE E7 ERASABLES NEEDED IN TEMPS
TS ABVEL*
CA VSELECT
TS VSELECT*
CA EBANK5
TS EBANK # CHANGE EBANKS

EBANK= LRVF
CS LRVF
AD ABVEL* # IF V < VF, USE WVF
EXTEND
BZMF USEVF

CS ABVEL*
AD LRVMAX # VMAX - V
```

```

EXTEND
BZMF WSTOR -1 # IF V > VMAX, W = 0

EXTEND
INDEX VSELECT*
MP LRWVZ # WV(VMAX - V)

EXTEND
DV LRVMAX # WV(1 - V/VMAX)
TCF WSTOR

USEVF INDEX VSELECT*
 CA LRWVFZ # USE APPROPRIATE CONSTANT WEIGHT
 TCF WSTOR

 -1 CA ZERO
WSTOR TS MPAC
 CS BIT7 # (=64D)
 AD MODREG
 EXTEND

Page 889
 BZMF +3 # IF IN P65,P66,P67, USE ANOTHER CONSTANT

 CA LRWVFF
 TS MPAC

 +3 CA EBANK7
 TS EBANK # CHANGE EBANKS

 EBANK= ABVEL
 TC INTPRET
 DMP VXSC # W(Delta V)(VBEAMSM) UP 6-7, 0-5
 VAD

 V1S # ADD WEIGHTED Delta V TO VELOCITY
 STORE GNUV
 EXIT

 TC QUIKFAZ5 # DO NOT RE-UPDATE

 CA SIX
VUPDATED TC GNURVST # STORE NEW VELOCITY VECTOR
ENDVDAT = VALTCHK

VALTCHK TC QUIKFAZ5 # DO NOT REPEAT ABOVE

 CAF READVBIT # TEST READVEL TO SEE IF VELOCITY READING

```



July 12, 2016

1753

|            |        |          |                                         |
|------------|--------|----------|-----------------------------------------|
|            | MASK   | FLGWRD11 | # IS DESIRED.                           |
|            | CCS    | A        |                                         |
|            | TCF    | READV    | # YES -- READ VELOCITY                  |
|            | CS     | ABVEL    | # NO -- SEE IF VELOCITY < 2000 FT/SEC   |
|            | AD     | 2KFT/SEC |                                         |
|            | EXTEND |          |                                         |
|            | BZMF   | CONTSERV | # V > 2000 FT/SEC DO NOT READ VEL       |
|            | TC     | UPFLAG   | # V < 2000 FT/SEC SET READVEL AND READ. |
|            | ADRES  | READVEL  |                                         |
| READV      | CAF    | PRI032   | # SET UP JOB TO READ VELOCITY BEAMS.    |
|            | TC     | NOVAC    |                                         |
|            | EBANK= | HMEAS    |                                         |
|            | 2CADR  | LRVJOB   |                                         |
|            | TCF    | CONTSERV | # CONTINUE WITH SERVICER                |
| GNURVST    | TS     | BUF      | # STORE GNUR (=GNUV) IN R1S OR V1S      |
|            | EXTEND |          | # A = 0 FOR R, A = 6 FOR V              |
|            | DCA    | GNUR     |                                         |
|            | INDEX  | BUF      |                                         |
|            | DXCH   | R1S      |                                         |
|            | EXTEND |          |                                         |
| # Page 890 | DCA    | GNUR +2  |                                         |
|            | INDEX  | BUF      |                                         |
|            | DXCH   | R1S +2   |                                         |
|            | EXTEND |          |                                         |
|            | DCA    | GNUR +4  |                                         |
|            | INDEX  | BUF      |                                         |
|            | DXCH   | R1S +4   |                                         |
|            | TC     | Q        |                                         |
| QUIKFAZ5   | CA     | EBANK3   |                                         |
|            | XCH    | EBANK    | # SET EBANK 3                           |
|            | DXCH   | L        | # Q TO A, A TO L                        |
|            | EBANK= | PHSNAME5 |                                         |
|            | TS     | PHSNAME5 |                                         |
|            | LXCH   | EBANK    |                                         |
|            | EBANK= | DVCNTR   |                                         |
|            | TC     | A        |                                         |
| HFAIL      | CS     | LRRCTR   |                                         |
|            | EXTEND |          |                                         |
|            | BZF    | NORLITE  | # IF R = 0, DO NOT TURN ON TRK FAIL     |

```

 AD LRLCTR
 MASK NEG3
 EXTEND
 BZF +2
 TCF NORLITE

 TC UPFLAG
 ADRES HFLSHFLG
 # AND SET BIT TO TURN ON TRACKER FAIL LITE

NORLITE CA LRLCTR
 TS LRRCTR
 # SET R = L

 TCF VMEASCHK

VFAIL CS LRSCTR
 EXTEND
 BZF NOLITE
 AD LRMCTR
 MASK NEG3
 EXTEND
 BZF +2
 TCF NOLITE
 # DELTA Q LARGE
 # IF S = 0, DO NOT TURN ON TRACKER FAIL
 # M-S
 # TEST FOR M-S > 3
 # IF M-S > 3, THEN TWO OR MORE OF THE
 # LAST FOUR V READINGS WERE BAD,
 # SO TURN ON VELOCITY FAIL LIGHT

 TC UPFLAG
 ADRES VFLSHFLG
 # AND SET BIT TO TURN ON TRACKER FAIL LITE

Page 891
NOLITE CA LRMCTR
 TS LRSCTR
 # SET S = M

 CCS VSELECT
 TCF ENDVDAT
 # TEST FOR Z COMPONENT
 # NOT Z, DO NOT SET VX INHIBIT

 TC UPFLAG
 ADRES VXINH
 TCF ENDVDAT
 # Z COMPONENT - SET FLAG TO SKIP X
 # COMPONENT, AS ERROR MAY BE DUE TO CROSS
 # LOBE LOCK UP NOT DETECTED ON X AXIS.

Page 892

LRVJOB IS SET WHEN THE LEM IS BELOW 15000 FT DURING THE LANDING PHASE
THIS JOB INITIALIZES THE LANDING RADAR READ ROUTINE FOR 5 VELOCITY
SAMPLES AND GOES TO SLEEP WHILE THE SAMPLING IS DONE -- ABOUT 500 MS.
WITH A GOODEND RETURN THE DATA IS STORED IN VMEAS AND BIT7 OF LRSTAT
IS SET. THE GIMBAL ANGLES ARE READ ABOUT MIDWAY IN THE SAMPLINGS.

170MS EQUALS ND1

```

July 12, 2016

1755

```
LRVJOB CA 170MS # SET TASK TO READ CDUS + PIPAS
 TC WAITLIST
 EBANK= LRVTIME
 2CADR RDGIMS

 CCS VSELECT # SEQUENCE LR VEL BEAM SELECTOR
 TCF +2
 CAF TWO # IF ZERO, RESET TO TWO
 DOUBLE # 2XVSELECT USED FOR VBEAM INDEX IN LRVEL
 TC BANKCALL # GO INITIALIZE LR VEL READ ROUTINE
 CADR LRVEL
 TC BANKCALL # PUT LRVJOB TO SLEEP ABOUT 500 MS
 CADR RADSTALL
 TCF VBAD
 CCS STILBADV # IS DATA GOOD JUST PRESENT?
 TCF VSTILBAD # JUST GOOD -- MUST WAIT 4 SECONDS.

 INHINT
 EXTEND
 DCA SAMPLSUM # GOOD RETURN -- STOW AWAY VMEAS
 DXCH VMEAS
 CA EBANK4 # FOR DOWNLINK
 TS EBANK
 EBANK= LRVTIME

 EXTEND
 DCA LRVTIME
 DXCH LRVTIMDL
 EXTEND
 DCA LRXCDU
 DXCH LRXCDUDL
 CA LRZCDU
 TS LRZCDUDL
 CA EBANK7
 TS EBANK
 EBANK= VSELECT

 CS FLGWRD11 # SET BIT TO INDICATE VELOCITY
 MASK VELDABIT # MEASUREMENT MADE

Page 893

 ADS FLGWRD11
 CCS VSELECT # UPDATE VSELECT
 TCF +2
 CA TWO
 TS VSELECT
```

```

 TCF ENDOFJOB

VBAD CAF TWO # SET STILBAD TO WAIT 4 SECONDS
VSTILBAD TS STILBADV
 TCF ENDLRV

LRHJOB IS SET BY LRHTASK WHEN LEM IS BELOW 25000 FT. THIS JOB
INITIALIZES THE LR READ ROUTINE FOR AN ALT MEASUREMENT AND GOES TO
SLEEP WHILE THE SAMPLING IS DONE -- ABOUT 95 MS. WITH A GOODEND RETURN
THE ALT DATA IS STORED IN HMEAS AND BIT7 OF LRSTAT IS SET.

 BANK 34
 SETLOC R12STUFF
 BANK

 COUNT* $$/SERV

LRHJOB TC BANKCALL # INITIATE LR ALT MEASUREMENT
 CADR LRALT
 TC BANKCALL # LRHJOB TO SLEEP ABOUT 95MS
 CADR RADSTALL
 TCF HBAD
 CCS STILBADH # IS DATA GOOD JUST PRESENT?
 TCF HSTILBAD # JUST GOOD -- MUST WAIT 4 SECONDS.

 INHINT
 EXTEND
 DCA SAMPLSUM # GOOD RETURN -- STORE AWAY LRH DATA
 DXCH HMEAS # LRH DATA 1.079 FT/BIT
 EXTEND # FOR DOWNLINK
 DCA PIPTIME1
 DXCH MKTIME

 EXTEND
 DCA CDUTEMPY # CDUY,Z = AIG,AMG
 DXCH AIG

 CA CDUTEMPX # CDUX = AOG
 TS AOG

 CS FLGWRD11 # SET BIT TO INDICATE RANGE
 MASK RNGEDBIT # MEASUREMENT MADE.
 ADS FLGWRD11
 ENDLRH TC ENDOFJOB # TERMATE LRHJOB

Page 894

```

July 12, 2016

1757

```
HBAD CA FLAGWRD5
 MASK RNGSCBIT # IS BAD RETURN DUE TO SCALE CHANGE?
 EXTEND
 BZF HSTILBAD -1 # NO RESET HSTILBAD
 TC DOWNFLAG # YES RESET SCALE CHANGE BIT AND IGNORE
 ADRES RNGSCFLG
 TC ENDOFJOB

HSTILBAD CAF TWO # SET STILBAD TO WAIT 4 SECONDS
 TS STILBADH
 TC ENDOFJOB

 BANK 34
 SETLOC SERV4
 BANK

 COUNT* $$/SERV

RDGIMS IS A TASK SET UP BY LRVJOB TO PICK UP THE IMU CDUS AND TIME
AT ABOUT THE MIDPOINT OF THE LR VEL READ ROUTINE WHEN 5 VEL SAMPLES
ARE SPECIFIED.

RDGIMS EBANK= LRVTIME
 EXTEND
 DCA TIME2 # PICK UP TIME2, TIME1
 DXCH LRVTIME # AND SAVE IN LRVTIME

 EXTEND
 DCA CDUX # PICK UP CDUX AND CDUY
 DXCH LRXCDU # AND SAVE IN LRXCDU AND LRYCDU

 CA CDUZ
 TS LRZCDU # SAVE CDUZ IN LRZCDU

 CA PIPAX
 TS PIPTM # SAVE PIPAX IN PIPTM

 EXTEND
 DCA PIPAY # PICK UP PIPAY AND PIPAZ
 DXCH PIPTM +1 # AND SAVE IN PIPTM +1 AND PIPTM +2
 TC TASKOVER

 BANK 33
 SETLOC SERVICES
 BANK
```

COUNT\* \$\$/SERV

EBANK= DVCNTR

# Page 895

# HIGATJOB IS SET APPROXIMATELY 6 SECONDS PRIOR TO HIGH GATE DURING  
 # THE DESCENT BURN PHASE OF LUNAR LANDING. THIS JOB INITIATES THE  
 # LANDING RADAR REPOSITIONING ROUTINE AND GOES TO SLEEP UNTIL THE  
 # LR ANTENNA MOVES FROM POSITION 1 TO POSITION 2. IF THE LR ANTENNA  
 # ACHIEVES POSITION 2 WITHIN 22 SECONDS THE ALTITUDE AND VELOCITY  
 # BEAM VECTORS ARE RECOMPUTED TO REFLECT THE NEW ORIENTATION WITH  
 # RESPECT TO THE NB. BIT10 OF LRSTAT IS CLEARED TO ALLOW LR  
 # MEASUREMENTS AND THE JOB TERMINATES.

|          |       |          |                                            |
|----------|-------|----------|--------------------------------------------|
| HIGATJOB | TC    | BANKCALL | # START LRPOS2 JOB                         |
|          | CADR  | LRPOS2   |                                            |
|          | TC    | BANKCALL | # PUT HIGATJOB TO SLEEP UNTIL JOB IS DONE  |
|          | CADR  | RADSTALL |                                            |
|          | TCF   | POSALARM | # BAD END      ALARM                       |
| POSGOOD  | CA    | PRI023   | # REDUCE PRIORITY FOR INTERPRETIVE COMPS.  |
|          | TC    | PRI0CHNG |                                            |
|          | TC    | SETPOS2  | # LR IN POS2 -- SET UP TRANSFORMATIONS     |
|          | TC    | DOWNFLAG |                                            |
|          | ADRES | NOLRREAD | # RESET NOLRREAD FLAG TO ENABLE LR READING |
|          | TC    | ENDOFJOB |                                            |
| POSALARM | CA    | OCT523   |                                            |
|          | TC    | BANKCALL |                                            |
|          | CADR  | PRIOLARM | # FLASH ALARM CODE                         |
|          | TCF   | GOTPOOH  | # TERMINATE                                |
|          | TCF   | +3       | # PROCEED -- TRY AGAIN                     |
|          | TCF   | ENDOFJOB | # V 32 E      TERMINATE R12                |
|          | TC    | ENDOFJOB |                                            |
|          | +3    | CA       | BIT7                                       |
|          |       | EXTEND   | # SEE IF IN POS2 YET                       |
|          |       | RAND     | CHAN33                                     |
|          |       | EXTEND   |                                            |
|          | BZF   | POSGOOD  | # POS2 ACHIEVED    SET UP ANTENNA BEAMS    |
|          | TCF   | POSALARM | # STILL DIDN'T MAKE IT    REALARM          |
| OCT523   | OCT   | 00523    |                                            |

```

SETPOS1 TC MAKECADR # MUST BE CALLED BY BANKCALL
 TS LRADRET1 # SAVE RETURN CADR. SINCE BUP2 CLOBBBERED

 CAF TWO
 TS STILBADH # INITIALIZE STILBAD
 TS STILBADV # INITIALIZE STILBAD

 CA ZERO # INDEX FOR LRALPHA, LRBETA IN POS 1.

Page 896 TS LRLCTR # SET L,M,R, ANS S TO ZERO
 TS LRMCTR
 TS LRRCTR
 TS LRSCTR
 TS VSELECT # INITIALIZE VSELECT

 TC SETPOS # CONTINUE WITH COMPUTATIONS.

 CA LRADRET1
 TC BANKJUMP # RETURN TO CALLER

SETPOS2 CA TWO # INDEX FOR POS2
SETPOS XCH Q # SAVE INDEX IN Q
 TS LRADRET # SAVE RETURN

 CA EBANK5
 TS EBANK
EBANK= LRALPHA

EXTEND
INDEX Q
DCA LRALPHA # LRALPHA IN A, LRBETA IN L
TS CDUSPOT +4 # ROTATION ABOUT X
LXCH CDUSPOT # ROTATION ABOUT Y
CA ZERO
TS CDUSPOT +2 # ZERO ROTATION ABOUT Z.

 CA EBANK7
 TS EBANK
EBANK= LRADRET

 TC INTPRET
VLOAD CALL
 UNITY # CONVERT UNITY(ANTENNA) TO NB
 TRG*SMNB
STOVL VYBEAMNB

```

1760

July 12, 2016

```
UNITX # CONVERT UNITX(ANTENNA) TO NB
CALL
SMNB
STORE VXBEAMNB
VXV VSL1
VYBEAMNB
STOVL VZBEAMNB # Z = X * Y
HBEAMANT
CALL
SMNB # CONVERT TO NB
STORE HBEAMNB
EXIT
```

# Page 897

```
TC LRADRET
```

This code is written to file `src/SERVICER.s`.



## B.104 SINGLE PRECISION SUBROUTINES

```

1761 <src/SINGLE-PRECISION-SUBROUTINES.s 1761>≡
Copyright: Public domain.
Filename: SINGLE_PRECISION_SUBROUTINES.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1207
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 1207

BLOCK 02

SINGLE PRECISION SINE AND COSINE

COUNT 02/INTER

SPCOS AD HALF # ARGUMENTS SCALED AT PI
SPSIN TS TEMK
 TCF SPT
 CS TEMK
SPT DOUBLE
 TS TEMK
 TCF POLLEY

```

|        |        |        |                       |
|--------|--------|--------|-----------------------|
|        | XCH    | TEMK   |                       |
|        | INDEX  | TEMK   |                       |
|        | AD     | LIMITS |                       |
|        | COM    |        |                       |
|        | AD     | TEMK   |                       |
|        | TS     | TEMK   |                       |
|        | TCF    | POLLEY |                       |
|        | TCF    | ARG90  |                       |
| POLLEY | EXTEND |        |                       |
|        | MP     | TEMK   |                       |
|        | TS     | SQ     |                       |
|        | EXTEND |        |                       |
|        | MP     | C5/2   |                       |
|        | AD     | C3/2   |                       |
|        | EXTEND |        |                       |
|        | MP     | SQ     |                       |
|        | AD     | C1/2   |                       |
|        | EXTEND |        |                       |
|        | MP     | TEMK   |                       |
|        | DDOUBL |        |                       |
|        | TS     | TEMK   |                       |
|        | TC     | Q      |                       |
| ARG90  | INDEX  | A      |                       |
|        | CS     | LIMITS |                       |
|        | TC     | Q      | # RESULT SCALED AT 1. |

# SPROOT WAS DELETED IN REV 51 OF MASTER. ASS. CONT. HAS CARDS.

This code is written to file `src/SINGLE-PRECISION-SUBROUTINES.s`.

**B.105 SPS BACK-UP RCS CONTROL**

1763

*<src/SPS-BACK-UP-RCS-CONTROL.s 1763>≡*

```

Copyright: Public domain.
Filename: SPS_BACK-UP_RCS_CONTROL.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1507-1510
Mod history: 2009-05-27 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 1507
PROGRAM NAME: SPSRCS
AUTHOR: EDGAR M. OSHIKA (AC ELECTRONICS)
MODIFIED: TO RETURN TO ALL AXES VIA Q BY P. S. WEISSMAN, OCT 7, 1968
MODIFIED TO IMPROVE BENDING STABILITY BY G. KALAN, FEB. 14, 1969
#
FUNCTIONAL DESCRIPTION:
THE PROGRAM CONTROLS THE FIRING OF ALL RCS JETS IN THE DOCKED CONFIGURATION ACCORDING TO
PLANE LOGIC.
#
1. JET SENSE TEST (SPSRCS)
IF JETS ARE FIRING NEGATIVELY, SET OLDSENSE NEGATIVE AND CONTINUE
IF JETS ARE FIRING POSITIVELY, SET OLDSENSE POSITIVE AND CONTINUE
IF JETS ARE NOT FIRING, SET OLDSENSE TO ZERO AND GO TO OUTER RATE LIMIT TEST
#

```

```

2. RATE DEAD BAND TEST
IF JETS ARE FIRING NEGATIVELY AND RATE IS GREATER THAN TARGET RATE, I
JETS ON AND GO TO INHIBITION LOGIC. OTHERWISE, CONTINUE.
IF JETS ARE FIRING POSITIVELY AND RATE IS LESS THAN TARGET RATE, I
JETS ON AND GO TO INHIBITION LOGIC. OTHERWISE, CONTINUE.
#
3. OUTER RATE LIMIT TEST (SPSSTART)
IF MAGNITUDE OF EDOT IS GREATER THAN 1.73 DEG/SEC SET JET FIRING TIME
TO REDUCE RATE AND GO TO INHIBITION LOGIC. OTHERWISE, CONTINUE.
#
4. COAST ZONE TEST
IF STATE (E,EDOT) IS BELOW LINE $E + 4 \times EDOT > -1.4$ DEG AND EDOT IS I
POSITIVE AND CONTINUE. OTHERWISE, SET JET FIRING TIME TO ZERO
IF STATE IS ABOVE LINE $E + 4 \times EDOT > +1.4$ DEG AND EDOT IS GREATER TH
AND CONTINUE. OTHERWISE, SET JET FIRING TIME TO ZERO AND CONTINUE.
#
5. INHIBITION LOGIC
IF OLDSENSE IS NON-ZERO:
A) RETURN IF JET TIME AS THE SAME SIGN AS OLDSENSE
B) SET INHIBITION COUNTER* AND RETURN IF JET TIME IS ZERO
C) SET INHIBITION COUNTER,* SET JET TIME TO ZERO AND RETURN I
OF JET TIME IS OPPOSITE TO THAT OF OLDSENSE
IF OLDSENSE IS ZERO:
A) RETURN IF INHIBITION COUNTER IS NOT POSITIVE
B) SET JET TIME TO ZERO AND RETURN IF INHIBITION COUNTER IS I
*NOTE: INHIBITION COUNTERS CAN BE SET TO 4 OR 10 FOR THE P AND UV AXES
RESPECTIVELY, IN SPSRCS. THEY ARE DECREMENTED BY ONE AT THE BEGINNING
Page 1508
EACH DAP PASS.
#
THE MINIMUM PULSE WIDTH OF THIS CONTROLLER IS DETERMINED BY THE REPETITION RATE
AND IS NOMINALLY 100 MS FOR ALL AXES IN DRIFTING FLIGHT. DURING POWERED FLIGHT
P AXIS AND 200 MS FOR THE CONTROL OF THE U AND V AXES.
#
CALLING SEQUENCE:
INHINT
TC IBNKCALL
CADR SPSRCE
#
EXIT:
TC Q
#
ALARM/ABORT MODE: NONE
#
SUBROUTINES CALLED: NONE
#

```

July 12, 2016

1765

```
INPUT: E, EDOT
TJP, TJV, TJU TJ MUST NOT BE NEGATIVE ZERO
#
OUTPUT: TJP, TJV, TJU

BANK 21
SETLOC DAPS4
BANK

COUNT* $$/DAPBU

EBANK= TJU
RATELIM2 OCT 00632 # 1.125 DEG/SEC
POSTHRST CA HALF

NDX AXISCTR
TS TJU
CCS OLDSENSE
TCF POSCHECK # JETS FIRING POSITIVELY
TCF CTRCHECK # JETS OFF. CHECK INHIBITION CTR
NEGCHECK INDEX AXISCTR # JETS FIRING NEGATIVELY
CS TJU
CCS A
TC Q # RETURN
TCF +2
TCF +1 # JETS COMMANDED OFF. SET CTR AND RETURN
SETCTR INDEX AXISCTR # JET FIRING REVERSAL COMMANDED. SET CTR,
CA UTIME # SET JET TIME TO ZER, AND RETURN

Page 1509

INDEX AXISCTR
TS UJETCTR
ZAPTJ CA ZERO
INDEX AXISCTR
TS TJU
TC Q
POSCHECK INDEX AXISCTR
CA TJU
TCF NEGCHECK +2
CTRCHECK INDEX AXISCTR # CHECK JET INHIBITION COUNTER
CCS UJETCTR
TCF +2
TC Q # CTR IS NOT POSITIVE. RETURN
TCF ZAPTJ # CTR IS POSITIVE. INHIBIT FIRINGS
TC Q # CTR IS NOT POSITIVE. RETURN
OCT 00004
UTIME OCT 00012
```

1766

July 12, 2016

|             |        |             |                                            |
|-------------|--------|-------------|--------------------------------------------|
|             | OCT    | 00012       |                                            |
| OLDSENSE    | EQUALS | DAPTRG1     |                                            |
| NEGFIRE     | CS     | ONE         | # JETS FIRING NEGATIVELY                   |
|             | TS     | OLDSENSE    |                                            |
|             | CA     | EDOT        |                                            |
|             | TCF    | +4          |                                            |
| PLUSFIRE    | CA     | ONE         |                                            |
|             | TS     | OLDSENSE    |                                            |
|             | CS     | EDOT        | # RATE DEAD BAND TEST                      |
|             | LXCH   | A           |                                            |
|             | CS     | DAPBOOLS    | # IF DRIFTBIT = 1, USE ZERO TARGET RATE    |
|             | MASK   | DRIFTBIT    | # IF DRIFTBIT = 0, USE 0.10 RATE TARGET    |
|             | CCS    | A           |                                            |
|             | CA     | RATEDB1     |                                            |
|             | AD     | L           |                                            |
|             | EXTEND |             |                                            |
|             | BZMF   | SPSSTART    |                                            |
|             | TCF    | POSTHRST +3 |                                            |
| SPSRCS      | INDEX  | AXISCTR     | # JET SENSE TEST                           |
|             | CCS    | TJU         |                                            |
|             | TCF    | PLUSFIRE    | # JETS FIRING POSITIVELY                   |
|             | TCF    | +2          |                                            |
|             | TCF    | NEGFIRE     | # JETS FIRING NEGATIVELY                   |
|             | TS     | OLDSENSE    | # JETS OFF                                 |
| SPSSTART    | CA     | EDOT        | # OUTER RATE LIMIT TEST                    |
|             | EXTEND |             |                                            |
|             | MP     | RATELIM1    |                                            |
|             | CCS    | A           |                                            |
|             | TCF    | NEGTHRST    | # OUTER RATE LIMIT EXCEEDED                |
|             | TCF    | +2          |                                            |
|             | TCF    | POSTHRST    | # OUTER RATE LIMIT EXCEEDED                |
|             | CA     | EDOT        | # COAST ZONE TEST                          |
| # Page 1510 | AD     | E           |                                            |
|             | EXTEND |             |                                            |
|             | MP     | DKDB        | # PAD LOADED DEADBAND. FRESHSTART: 1.4 DEG |
|             | EXTEND |             |                                            |
|             | BZF    | TJZERO      |                                            |
|             | EXTEND |             |                                            |
|             | BZMF   | +7          |                                            |
|             | CA     | EDOT        |                                            |
|             | AD     | RATELIM2    |                                            |
|             | EXTEND |             |                                            |
|             | BZMF   | TJZERO      |                                            |

July 12, 2016

1767

```
NEGTHRST CS HALF
 TCF POSTHRST +1
 CS RATELIM2
 AD EDOT
 EXTEND
 BZMF POSTHRST
TJZERO CA ZERO
 TCF POSTHRST +1

RATELIM1 = CALLCODE # = 00032, CORRESPONDING TO 1.73 DEG/SEC
RATEDB1 = TBUILDFX # = 00045, CORRESPONDS TO 0.101 DEG/SEC

*** END OF LMDAP .015 ***
```

This code is written to file `src/SPS-BACK-UP-RCS-CONTROL.s`.

## B.106 STABLE ORBIT

```

1768 <src/STABLE-ORBIT.s 1768>≡
 # Copyright: Public domain.
 # Filename: STABLE_ORBIT.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Hartmuth Gutsche <hgutsche@explornet.com>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 525-532
 # Mod history: 2009-05-10 HG Started adapting from the Colossus249/ file
 # of the same name, using Comanche055 page
 # images 0525.jpg - 0532.jpg.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A
 # Page 525
 # STABLE ORBIT RENDEZVOUS PROGRAMS (P38 AND P78)
 #
 # MOD NO -1 LOG SECTION - STABLE ORBIT - P38-P39
 # MOD BY RUDNICKI.S DATE 25JAN68
 #
 # FUNCTIONAL DESCRIPTION
 #
 # P38 AND P78 CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL
 # CONDITIONS REQUIRED BY THE AGC TO (1) PUT THE ACTIVE VEHICLE
 # ON A TRANSFER TRAJECTORY THAT INTERCEPTS THE PASSIVE VEHICLE
 # ORBIT A GIVEN DISTANCE, DELTA R, EITHER AHEAD OF OR BEHIND THE
 # PASSIVE VEHICLE AND (2) ACTUALLY PLACE THE ACTIVE VEHICLE IN THE
 # PASSIVE VEHICLE ORBIT WITH A DELTA R SEPARATION BETWEEN THE TWO

```



```

VEHICLES
#
CALLING SEQUENCE
#
ASTRONAUT REQUEST THRU DSKY
#
V37E38E IF THIS VEHICLE IS ACTIVE VEHICLE
V37E78E IF OTHER VEHICLE IS ACTIVE VEHICLE
#
INPUT
#
(1) SOI MANEUVER
#
(A) TIG TIME OF SOI MANEUVER
(B) CENTANG ORBITAL CENTRAL ANGLE OF THE PASSIVE VEHICLE
DURING THE TRANSFER FROM TIG TO TIME OF INTERCEPT
(C) DELTAR THE DESIRED SEPARATION OF THE TWO VEHICLES
SPECIFIED AS A DISTANCE ALONG THE PASSIVE VEHICLE
ORBIT
(D) OPTION EQUALS 1 FOR SOI
#
(2) SOR MANEUVER
#
(A) TIG TIME OF SOR MANEUVER
(B) CENTANG AN OPTIONAL RESPECIFICATION OF 1 (B) ABOVE
(C) OPTION EQUALS 2 FOR SOR
(D) DELTTIME THE TIME REQUIRED TO TRAVERSE DELTA R WHEN
TRAVELING AT A VELOCITY EQUAL TO THE HORIZONTAL
VELOCITY OF THE PASSIVE VEHICLE - SAVED FROM
SOI PHASE
(E) TINT TIME OF INTERCEPT (SOI) - SAVED FROM SOI PHASE
#
OUTPUT
#
(1) TRKMKCNT NUMBER OF MARKS
(2) TTOGO TIME TO GO
(3) +MGA MIDDLE GIMBAL ANGLE
Page 526
(4) DSPTM1 TIME OF INTERCEPT OF PASSIVE VEHICLE ORBIT
(FOR SOI ONLY)
(5) POSTTPI PERIGEE ALTITUDE OF ACTIVE VEHICLE ORBIT AFTER
THE SOI (SOR) MANEUVER
(6) DELVTPI MAGNITUDE OF DELTA V AT SOI (SOR) TIME
(7) DELVTPF MAGNITUDE OF DELTA V AT INTERCEPT TIME
(8) DELTA VELOCITY AT SOI (AND SOR) - LOCAL VERTICAL
COORDINATES

```

```

#
SUBROUTINES USED
#
AVFLAGA
AVFLAGP
VNDSPLY
BANKCALL
GOFLASHR
GOTOPOOH
BLANKET
ENDOFJOB
PREC/TT
SELECTMU
INTRPVP
MAINRTNE

```

```

 BANK 04
 SETLOC STBLEORB
 BANK

```

```

 EBANK= SUBEXIT
 COUNT* $$/P3879

```

```

P38 TC AVFLAGA # THIS VEHICLE ACTIVE
 TC +2
P78 TC AVFLAGP # OTHER VEHICLE ACTIVE
 TC P20FLGON # SET UPDATFLG, TRACKFLG
 CAF V06N33SR # DISPLAY TIG
 TC VNDSPLY
 CAF V06N55SR # DISPLAY CENTANG
 TCR BANKCALL
 CADR GOFLASHR
 TCF GOTOPOOH # TERMINATE
 TCF +5 # PROCEED
 TCF -5 # RECYCLE
 CAF THREE # IMMEDIATE RETURN -- BLANK R1, R2
 TCR BLANKET
 TCF ENDOFJOB
 CAF FIVE
 TS OPTION1
 CAF ONE
 TS OPTION2 # OPTION CODE IS SET TO 1

Page 527
 CAF V04N06SR # DISPLAY OPTION CODE -1 = SOI, 2 = SOR
 TCR BANKCALL
 CADR GOFLASHR

```

July 12, 2016

1771

```
TCF GOTOP00H # TERMINATE
TCF +5 # PROCEED
TCF -5 # RECYCLE
CAF BIT3 # IMMEDIATE RETURN -- BLANK R3
TCR BLANKET
TCF ENDOFJOB
TC INTPRET
SSP
NN
2
SLOAD SR1
 OPTION2
BHIZ DLOAD
 OPTN1
 TINT
STORE TINTSOI # STORE FOR SOR PHASE
CLRGO
 OPTNSW # OPTNSW: ON = SOI. OFF = SOR
 JUNCTN1
OPTN1 SET CLEAR # SOI
 OPTNSW
 UPDATFLG
CALL
 PREC/TT
SET DAD
 UPDATFLG
 TIG
STORE TINT # TI = TIG + TF
STORE DSPTEM1 # FOR DISPLAY
EXIT
CAF V06N57SR # DISPLAY DELTA R
TCR BANKCALL
CADR GOFLASHR
TCF GOTOP00H # TERMINATE
TCF +5 # PROCEED
TCF -5 # RECYCLE
CAF SIX # IMMEDIATE RETURN - BLANK R2, R3
TCR BLANKET
TCF ENDOFJOB
CAF V06N34SR # DISPLAY TIME OF INTERCEPT
TC VNDSPLY
TC INTPRET
JUNCTN1 CLEAR CALL
 P39/79SW
 SELECTMU # SELECT MU, CLEAR FINALFLG, GO TO VN1645
RECYCLE CALL
```

```

PREC/TT
Page 528
BOFF DLOAD
 OPTNSW
 OPTN2
 TINT
STCALL TDEC1 # PRECISION UPDATE PASSIVE VEHICLE TO
 INTRPVP # INTERCEPT TIME
VLOAD UNIT
 RATT # RP/(RP)
PDVL VXV
 VATT
ABVAL NORM # (VP X RP/(RP))
 X1
PDDL DDV
 DELTAR
SL* 0 -7,1 # DELTA R / (VP X RP/RP)
STCALL DELTTIME # DELTA T = (RP) DELTA R / (VP X RP)
 JUNCTN2
OPTN2 DLOAD DAD
 TINTSOI
 T
STORE TINT # TI = TI + TF
JUNCTN2 DLOAD DSU
 TINT
 DELTTIME
STORE TARGTIME # TT = TI - DELTA T

.... MAINRTNE
#
SUBROUTINES USED
#
S3435.25
PERIAP01
SHIFTR1
VNDSPLY
BANKCALL
GOFLASH
GOTOP00H
VN1645

MAINRTNE STCALL TDEC1 # PRECISION UPDATE PASSIVE VEHICLE TO
 INTRPVP # TARGET TIME
 DLOAD
 TIG

```

July 12, 2016

1773

```

 STORE INTIME
 SSP VLOAD
 SUBEXIT
 TEST3979
 RATT
 CALL
 S3435.25
TEST3979 BOFF BON
Page 529
 P39/79SW
 MAINRTN1
 FINALFLG
 P39P79
 SET
 UPDATFLG
P39P79 EXIT
 TC DSPLY81 # FOR P39 AND P79
MAINRTN1 VLOAD ABVAL
 DELVEET3
 STOVL DELVTPI # DELTA V
 VPASS4
 VSU ABVAL
 VTPRIME
 STOVL DELVTPI # DELTA V (FINAL) = V-T - VT
 RACT3
 PDVL CALL
 VIPRIME
 PERIAP01 # GET PERIGEE ALTITUDE
 CALL
 SHIFTR1
 STORE POSTTPI
 BON SET
 FINALFLG
 DSPLY58
 UPDATFLG
DSPLY58 EXIT
 CAF V06N58SR # DISPLAY HP, DELTA V, DELTA V (FINAL)
 TC VNDSPLY
DSPLY81 CAF V06N81SR # DISPLAY DELTA V (LV)
 TC VNDSPLY
 TC INTPRET
 CLEAR VLOAD
 XDELVFLG
 DELVEET3
 STCALL DELVSIN
 VN1645 # DISPLAY TRKMKCNT, TTOGO, +MGA
```

BON GOTO  
P39/79SW  
P39/P79B  
RECYCLE

# STABLE ORBIT MIDCOURSE PROGRAM (P39 AND P79)  
#  
# MOD NO -1 LOG SECTION - STABLE ORBIT - P38-P39  
# MOD BY RUDNICKI, S DATE 25JAN68  
#  
# FUNCTIONAL DESCRIPTION  
#  
# P39 AND P79 CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL  
# CONDITIONS REQUIRED BY THE AGC TO MAKE A MIDCOURSE CORRECTION  
# Page 530  
# MANEUVER AFTER COMPLETING THE SOI MANEUVER BUT BEFORE MAKING  
# THE SOR MANEUVER.  
#  
# CALLING SEQUENCE  
#  
# ASTRONAUT REQUEST THRU DSKY  
#  
# V37E39E IF THIS VEHICLE IS ACTIVE VEHICLE  
# V37E79E IF OTHER VEHICLE IS ACTIVE VEHICLE  
#  
# INPUT  
#  
# (1) TPASS4 TIME OF INTERCEPT -- SAVED FROM P38/P78  
# (2) TARGTIME TIME THAT PASSIVE VEHICLE IS AT INTERCEPT POINT --  
# SAVED FROM P38/P78  
#  
# OUTPUT  
#  
# (1) TRKMKCNT NUMBER OF MARKS  
# (2) TTOGO TIME TO GO  
# (3) +MGA MIDDLE GIMBAL ANGLE  
# (4) DELVLVC DELTA VELOCITY AT MID -- LOCAL VERTICAL COORDINATES  
#  
# SUBROUTINES USED  
#  
# AVFLAGA  
# AVFLAGP  
# LOADTIME  
# SELECTMU  
# PRECSET  
# S34/35.1

July 12, 2016

1775

```
MAINRTNE

P39 TC AVFLAGA # THIS VEHICLE ACTIVE
 EXTEND
 DCA ATIGINC
 TC P39/P79A
P79 TC AVFLAGP # OTHER VEHICLE ACTIVE
 EXTEND
 DCA PTIGINC
P39/P79A DXCH KT # TIME TO PREPARE FOR BURN
 TC P20FLGON # SET UPDATFLG, TRACKFLG
 TC INTPRET
 SET CALL
 P39/79SW
 SELECTMU
P39/P79B RTB DAD
 LOADTIME
 KT
 STORE TIG # TIG = T (PRESENT) + PREPARATION TIME
Page531
 STCALL TDEC1 # PRECISION UPDATE ACTIVE AND PASSIVE
 PRECSET # # VEHICLES TO TIG
 CALL
 S34/35.1 # GET UNIT NORMAL
 DLOAD GOTO
 TARGTIME
 MAINRTNE # CALCULATE DELTA V AND DELTA V (LV)

.... PREC/TT
#
SUBROUTINES USED
#
PRECSET
TIMETHET
S34/35.1

PREC/TT STQ DLOAD
 RTRN
 TIG
 STCALL TDEC1 # PRECISION UPDATE ACTIVE AND PASSIVE
 PRECSET # # VEHICLES TO TIG
 VLOAD VSR*
 RPASS3
 0,2
 STODL RVEC
 CENTANG
```

```

 PUSH COS
 STODL CSTDH
 SIN SET
 RVSW
 STOVL SNTH
 VPASS3
 VSR* 0,2
 STCALL VVEC # GET TRANSFER TIME BASED ON CENTANG OF
 TIMETHET # PASSIVE VEHICLE
 CALL S34/35.1 # GET UNIT NORMAL
 DLOAD GOTO
 T
 RTRN

.... INTRPVP
#
SUBROUTINES USED
#
CSMPREC
LEMPREC

INTRPVP STQ BOFF # PRECISION UPDATE PASSIVE VEHICLE TO
 RTRN # TDEC1
 AVFLAG
 OTHERV
 CALL
Page 532
 CSMPREC
 GOTO
 RTRN
OTHERV CALL LEMPREC
 GOTO
 RTRN

.... VNDSPLY
#
SUBROUTINES USED
#
BANKCALL
GOFLASH
GOTOPOOH

VNDSPLY EXTEND # FLASH DISPLAY

```



July 12, 2016

1777

|          |      |          |             |
|----------|------|----------|-------------|
|          | QXCH | RTRN     |             |
|          | TS   | VERBNOUN |             |
|          | CA   | VERBNOUN |             |
|          | TCR  | BANKCALL |             |
|          | CADR | GOFLASH  |             |
|          | TCF  | GOTOPOOH | # TERMINATE |
|          | TC   | RTRN     | # PROCEED   |
|          | TCF  | -5       | # RECYCLE   |
| V06N33SR | VN   | 0633     |             |
| V06N55SR | VN   | 0655     |             |
| V04N06SR | VN   | 0406     |             |
| V06N57SR | VN   | 0657     |             |
| V06N34SR | VN   | 0634     |             |
| V06N58SR | VN   | 0658     |             |
| V06N81SR | VN   | 0681     |             |

# \*\*\* END OF COMEKISS.020 \*\*\*

This code is written to file `src/STABLE-ORBIT.s`.

## B.107 STAR TABLES

```

1778 <src/STAR-TABLES.s 1778>≡
 # Copyright: Public domain.
 # Filename: STAR_TABLES.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 1389-1393
 # Mod history: 2009-05-10 SN (Sergio Navarro). Started adapting
 # from the Colossus249/ file of the same
 # name, using Comanche055 page images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 1389

 BANK 32
 SETLOC STARTAB
 BANK

 COUNT 14/STARS

 2DEC +.8342971408 B-1 # STAR 37 X
 2DEC -.2392481515 B-1 # STAR 37 Y
 2DEC -.4966976975 B-1 # STAR 37 Z

 2DEC +.8139832631 B-1 # STAR 36 X
 2DEC -.5557243189 B-1 # STAR 36 Y

```

July 12, 2016

1779

# Page 1390

|      |                  |           |   |
|------|------------------|-----------|---|
| 2DEC | +.1691204557 B-1 | # STAR 36 | Z |
| 2DEC | +.4541086270 B-1 | # STAR 35 | X |
| 2DEC | -.5392368197 B-1 | # STAR 35 | Y |
| 2DEC | +.7092312789 B-1 | # STAR 35 | Z |
| 2DEC | +.3201817378 B-1 | # STAR 34 | X |
| 2DEC | -.4436021946 B-1 | # STAR 34 | Y |
| 2DEC | -.8370786986 B-1 | # STAR 34 | Z |
| 2DEC | +.5520184464 B-1 | # STAR 33 | X |
| 2DEC | -.7933187400 B-1 | # STAR 33 | Y |
| 2DEC | -.2567508745 B-1 | # STAR 33 | Z |
| 2DEC | +.4537196908 B-1 | # STAR 32 | X |
| 2DEC | -.8779508801 B-1 | # STAR 32 | Y |
| 2DEC | +.1527766153 B-1 | # STAR 32 | Z |
| 2DEC | +.2069525789 B-1 | # STAR 31 | X |
| 2DEC | -.8719885748 B-1 | # STAR 31 | Y |
| 2DEC | -.4436288486 B-1 | # STAR 31 | Z |
| 2DEC | +.1217293692 B-1 | # STAR 30 | X |
| 2DEC | -.7702732847 B-1 | # STAR 30 | Y |
| 2DEC | +.6259880410 B-1 | # STAR 30 | Z |
| 2DEC | -.1124304773 B-1 | # STAR 29 | X |
| 2DEC | -.9694934200 B-1 | # STAR 29 | Y |
| 2DEC | +.2178116072 B-1 | # STAR 29 | Z |
| 2DEC | -.1146237858 B-1 | # STAR 28 | X |
| 2DEC | -.3399692557 B-1 | # STAR 28 | Y |
| 2DEC | -.9334250333 B-1 | # STAR 28 | Z |
| 2DEC | -.3516499609 B-1 | # STAR 27 | X |
| 2DEC | -.8240752703 B-1 | # STAR 27 | Y |
| 2DEC | -.4441196390 B-1 | # STAR 27 | Z |
| 2DEC | -.5326876930 B-1 | # STAR 26 | X |
| 2DEC | -.7160644554 B-1 | # STAR 26 | Y |
| 2DEC | +.4511047742 B-1 | # STAR 26 | Z |
| 2DEC | -.7861763936 B-1 | # STAR 25 | X |
| 2DEC | -.5217996305 B-1 | # STAR 25 | Y |
| 2DEC | +.3311371675 B-1 | # STAR 25 | Z |

# Page 1391

|      |                  |           |   |
|------|------------------|-----------|---|
| 2DEC | -.6898393233 B-1 | # STAR 24 | X |
| 2DEC | -.4182330640 B-1 | # STAR 24 | Y |
| 2DEC | -.5909338474 B-1 | # STAR 24 | Z |
| 2DEC | -.5812035376 B-1 | # STAR 23 | X |
| 2DEC | -.2909171294 B-1 | # STAR 23 | Y |
| 2DEC | +.7599800468 B-1 | # STAR 23 | Z |
| 2DEC | -.9170097662 B-1 | # STAR 22 | X |
| 2DEC | -.3502146628 B-1 | # STAR 22 | Y |
| 2DEC | -.1908999176 B-1 | # STAR 22 | Z |
| 2DEC | -.4523440203 B-1 | # STAR 21 | X |
| 2DEC | -.0493710140 B-1 | # STAR 21 | Y |
| 2DEC | -.8904759346 B-1 | # STAR 21 | Z |
| 2DEC | -.9525211695 B-1 | # STAR 20 | X |
| 2DEC | -.0593434796 B-1 | # STAR 20 | Y |
| 2DEC | -.2986331746 B-1 | # STAR 20 | Z |
| 2DEC | -.9656605484 B-1 | # STAR 19 | X |
| 2DEC | +.0525933156 B-1 | # STAR 19 | Y |
| 2DEC | +.2544280809 B-1 | # STAR 19 | Z |
| 2DEC | -.8608205219 B-1 | # STAR 18 | X |
| 2DEC | +.4636213989 B-1 | # STAR 18 | Y |
| 2DEC | +.2098647835 B-1 | # STAR 18 | Z |
| 2DEC | -.7742591356 B-1 | # STAR 17 | X |
| 2DEC | +.6152504197 B-1 | # STAR 17 | Y |
| 2DEC | -.1482892839 B-1 | # STAR 17 | Z |
| 2DEC | -.4657947941 B-1 | # STAR 16 | X |
| 2DEC | +.4774785033 B-1 | # STAR 16 | Y |
| 2DEC | +.7450164351 B-1 | # STAR 16 | Z |
| 2DEC | -.3612508532 B-1 | # STAR 15 | X |
| 2DEC | +.5747270840 B-1 | # STAR 15 | Y |
| 2DEC | -.7342932655 B-1 | # STAR 15 | Z |
| 2DEC | -.4118589524 B-1 | # STAR 14 | X |
| 2DEC | +.9065485360 B-1 | # STAR 14 | Y |
| 2DEC | +.0924226975 B-1 | # STAR 14 | Z |

July 12, 2016

1781

# Page 1392

|      |                  |           |   |
|------|------------------|-----------|---|
| 2DEC | -.1820751783 B-1 | # STAR 13 | X |
| 2DEC | +.9404899869 B-1 | # STAR 13 | Y |
| 2DEC | -.2869271926 B-1 | # STAR 13 | Z |
| 2DEC | -.0614937230 B-1 | # STAR 12 | X |
| 2DEC | +.6031563286 B-1 | # STAR 12 | Y |
| 2DEC | -.7952489957 B-1 | # STAR 12 | Z |
| 2DEC | +.1371725575 B-1 | # STAR 11 | X |
| 2DEC | +.6813721061 B-1 | # STAR 11 | Y |
| 2DEC | +.7189685267 B-1 | # STAR 11 | Z |
| 2DEC | +.2011399589 B-1 | # STAR 10 | X |
| 2DEC | +.9690337941 B-1 | # STAR 10 | Y |
| 2DEC | -.1432348512 B-1 | # STAR 10 | Z |
| 2DEC | +.3507315038 B-1 | # STAR 9  | X |
| 2DEC | +.8926333307 B-1 | # STAR 9  | Y |
| 2DEC | +.2831839492 B-1 | # STAR 9  | Z |
| 2DEC | +.4105636020 B-1 | # STAR 8  | X |
| 2DEC | +.4988110001 B-1 | # STAR 8  | Y |
| 2DEC | +.7632988371 B-1 | # STAR 8  | Z |
| 2DEC | +.7032235469 B-1 | # STAR 7  | X |
| 2DEC | +.7075846047 B-1 | # STAR 7  | Y |
| 2DEC | +.0692868685 B-1 | # STAR 7  | Z |
| 2DEC | +.5450107404 B-1 | # STAR 6  | X |
| 2DEC | +.5314955466 B-1 | # STAR 6  | Y |
| 2DEC | -.6484410356 B-1 | # STAR 6  | Z |
| 2DEC | +.0130968840 B-1 | # STAR 5  | X |

# Page 1393

|      |                  |          |   |
|------|------------------|----------|---|
| 2DEC | +.0078062795 B-1 | # STAR 5 | Y |
| 2DEC | +.9998837600 B-1 | # STAR 5 | Z |
| 2DEC | +.4917678276 B-1 | # STAR 4 | X |
| 2DEC | +.2204887125 B-1 | # STAR 4 | Y |
| 2DEC | -.8423473935 B-1 | # STAR 4 | Z |
| 2DEC | +.4775639450 B-1 | # STAR 3 | X |
| 2DEC | +.1166004340 B-1 | # STAR 3 | Y |
| 2DEC | +.8708254803 B-1 | # STAR 3 | Z |

1782

July 12, 2016

|      |              |     |          |   |
|------|--------------|-----|----------|---|
| 2DEC | +.9342640400 | B-1 | # STAR 2 | X |
| 2DEC | +.1735073142 | B-1 | # STAR 2 | Y |
| 2DEC | -.3115219339 | B-1 | # STAR 2 | Z |
| 2DEC | +.8748658918 | B-1 | # STAR 1 | X |
| 2DEC | +.0260879174 | B-1 | # STAR 1 | Y |
| 2DEC | +.4836621670 | B-1 | # STAR 1 | Z |

|        |     |      |
|--------|-----|------|
| CATLOG | DEC | 6970 |
|--------|-----|------|

This code is written to file `src/STAR-TABLES.s`.

## B.108 SXTMARK

1783

*<src/SXTMARK.s 1783>*≡

```
Copyright: Public domain.
Filename: SXTMARK.agc
Purpose: Part of the source code for Comanche, build 055. It
is part of the source code for the Command Module's
(CM) Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 222-235
Contact: Ron Burkey <info@sandroid.org>,
Fabrizio Bernardini <fabrizio@spacecraft.it>
Website: http://www.ibiblio.org/apollo.
Mod history: 16/05/09 FB Transcription Batch 2 Assignment.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.
#
Page 222
PROGRAM NAME: SXTMARK
PROGRAM MODIFIED BY 258/278 PROGRAMMERS
MOD BY: R. MELANSON TO ADD DOCUMENTATION
#
FUNCTIONAL DESCRIPTION:
#
SXTMARK IS CALLED FROM INTERNAL ROUTINES WHICH MAY REQUIRE STAR OR LANDMARK MARKINGS BY
THE MARK SYSTEM IS NOT IN USE, SXTMARK RESERVES A VAC AREA FOR MARKING AND REQUESTS EXE
ROUTINE VIA THE EXECUTIVE JOB PRIORITY LIST. R21 USES THIS ROUTINE TO DETERMINE IF THE
USED. IF YES, SXTMARK RETURNS TO R21 TO PERFORM ITS OWN MARK REQUESTS VIA THE V51 FLAS
#
CALLING SEQUENCE:
```

DATE: 5 APRIL 1967  
LOG SECTION SXTMARK  
ASSEMBLY SUNDISK REV. 116

```

#
CAF (NO. MARK REQUESTS IN BITS 1-3 OF A)
TC BANKCALL
CADR SXTMARK
#
NORMAL EXIT MODE:
#
SWRETURN
#
ALARM OR ABORT EXIT MODE:
#
ABORT
#
OUTPUT:
#
1) MARKSTAT CONTAINS MARK VALUE (BITS 14-12) AND VAC AREA ADDRESS
2) QPRET = VAC AREA POINTER VALUE
3) 1ST WORD OF RESERVED VAC AREA SET TO +0
4) PRIO32 PLACED IN A REGISTER
#
ERASABLE INITIALIZATION:
#
1) BITS 1-3 OF A = NO. MARKS REQUESTED
2) BITS 2,3 OF EXTVBACT = 0
3) A VAC AREA MUST BE AVAILABLE (WORD 1 = ADDRESS OF VAC AREA)
#
DEBRIS:
#
A,Q,L, RUPTREG1, MARKSTAT, QPRET, BIT2 OF EXTVBACT
#
BANK 13
SETLOC SXTMARKE
BANK
#
EBANK= MRKBUF1
COUNT 07/SXTMK
#
SXTMARK INHINT
TS RUPTREG1 # NUMBER OF MARKS WANTED
#
CAF SIX # BIT2 = MARKING SYSTEM IN USE
MASK EXTVBACT # BIT3 = EXTENDED VERB IN PROGRESS
CCS A
TC MKABORT
#
Page 223
CAF BIT2 # NOT SET

```



July 12, 2016

1785

|          |        |          |                                    |
|----------|--------|----------|------------------------------------|
|          | ADS    | EXTVBACT | # SET IT, RESET IN ENDMARK         |
|          | TC     | MARKOK   | # YES, FIND VAC AREA               |
| MKABORT  | TC     | BAILOUT  |                                    |
|          | OCT    | 01211    |                                    |
| MARKOK   | CCS    | VAC1USE  | # FIND VAC AREA                    |
|          | TC     | MKVACFND |                                    |
|          | CCS    | VAC2USE  |                                    |
|          | TC     | MKVACFND |                                    |
|          | CCS    | VAC3USE  |                                    |
|          | TC     | MKVACFND |                                    |
|          | CCS    | VAC4USE  |                                    |
|          | TC     | MKVACFND |                                    |
|          | CCS    | VAC5USE  |                                    |
|          | TC     | MKVACFND |                                    |
|          | TC     | BAILOUT  |                                    |
|          | OCT    | 01207    |                                    |
| MKVACFND | AD     | TWO      | # ADDRESS OF VAC AREA              |
|          | TS     | MARKSTAT |                                    |
|          | INDEX  | A        |                                    |
|          | TS     | QPRET    | # STORE NEXT AVAILABLE MARK SLOT   |
|          | CAF    | ZERO     | # STORE VAC AREA OCCUPIED          |
|          | INDEX  | MARKSTAT |                                    |
|          | TS     | 0 -1     |                                    |
|          | TC     | CHECKMM  | # BACKUP MARK ROUTINE USES SXTMARK |
|          | MM     | 53       |                                    |
|          | TCF    | +2       |                                    |
|          | TCF    | SWRETURN |                                    |
|          | TC     | CHECKMM  |                                    |
|          | MM     | 54       |                                    |
|          | TCF    | +2       |                                    |
|          | TCF    | SWRETURN |                                    |
|          | CAF    | BIT12    | # DESIRED NUMBER OF MARKS IN 12-14 |
|          | EXTEND |          |                                    |
|          | MP     | RUPTREG1 |                                    |
|          | XCH    | L        |                                    |
|          | ADS    | MARKSTAT |                                    |
|          | CAF    | PRI032   | # ENTER MARK JOB                   |
|          | TC     | NOVAC    |                                    |
|          | EBANK= | MARKSTAT |                                    |
|          | 2CADR  | MKVB51   |                                    |

July 12, 2016

```

Page 224
PROGRAM NAME: MKRELEAS
PROGRAM MODIFIED BY 248/278 PROGRAMMERS
MOD BY: R. MELANSON TO ADD DOCUMENTATION
#
FUNCTIONAL DESCRIPTION:
#
MKRELEAS IS EXECUTED BY INTERNAL ROUTINES TO RELEASE THE MARK SYSTEM TO MAKE
SYSTEM ROUTINES. IT ALSO CLEARS THE COARSE OPTICS FLAG BIT AND DISABLES THE
#
CALLING SEQUENCE:
#
TC BANKCALL
CADR MKRELEAS
#
NORMAL EXIT MODE:
#
SWRETURN
#
ALARM OR EXIT MODE: NONE
#
OUTPUT:
#
1) BIT9 OPTMODES SET TO 0
2) OPTIND SET TO -1
3) 1ST WORD OF VAC AREA SET TO VAC ADDRESS TO SIGNIFY AVAILABILITY
4) MARKSTAT CLEARED
5) BIT2 CHANNEL 12 SET TO 0
#
ERASABLE INITIALIZATION: NONE
#
DEBRIS:
#
A,MARKSTAT,BIT9,OPTMODES OPTIND,BIT2 CHANNEL 12
#
MKRELEAS CAF ZERO # SHOW MARK SYSTEM NOW AVAILABLE
XCH MARKSTAT
MASK LOW9
CCS A
INDEX A
TS 0
MKRLEES INHINT
CS BIT9 # COARSE OPTICS RETURN FLAG.

```

July 12, 2016

1787

MASK OPTMODES  
TS OPTMODES

CA NEGONE  
TS OPTIND

# KILL COARSE OPTICS

CS BIT2  
EXTEND  
WAND CHAN12

# DISABLE OPTICS ERROR COUNTER

RELINT  
TC SWRETURN

# Page 225

# PROGRAM NAME: MARKRUPT

DATE: 5 APRIL 1967

# PROGRAM MODIFIED BY 258/278 PROGRAMMERS

LOG SECTION SXTMARK

# MOD BY: R. MELANSON TO ADD DOCUMENTATION

ASSEMBLY SUNDISK REV. 116

#

# FUNCTIONAL DESCRIPTION:

#

# MARKRUPT STORES CDUS,OPTICS AND TIME AND TRANSFERS CONTROL TO THE MARKIT,MARK REJECT OR  
# BITS IN CHANNEL 16 ARE SET AS REQUIRED.

#

# CALLING SEQUENCE:

#

# ROUTINE ENTERED VIA KEYRUPT2 WHEN MARK,MARK REJECT OR DSKY KEYS DEPRESSED BY THE OPERAT

#

# NORMAL EXIT MODE:

#

# MARKIT, MKREJECT, OR POSTJUMP ROUTINES (MARK, MARK REJECT, OR DSKY CODE)

#

# ALARM OR ABORT EXIT MODE:

#

# ALARM AND RESUME

#

# OUTPUT:

#

# RUPTSTOR+5 = CDUT, RUPTSTOR+3 = CDUS, RUPTSTOR+2 = CDUY,  
# RUPTREG3 = CDUZ, RUPTSTOR+6 = CDUX, RUPTSTOR+1 AND SAMPTIME+1 = TIME1,  
# RUPTSTOR AND SAMPTIME = TIME2

#

# ERASABLE INITIALIZATION:

#

# CDUT,CDUS,CDUY,CDUZ,CDUX,TIME2,TIME1,CHANNEL 16 BITS 6,7 OR 1-5.

#

# DEBRIS:

#

```

A,QRUPT, RUPTREG3,SAMPTIME,SAMPTIME+1, RUPTSTOR TO RUPTSTOR+6 EXCEPT RUPTSTOR+
MARKRUPT TS BANKRUPT # STORE CDUS AND OPTICS NOW
 CA CDUT
 TS MKCDUT
 CA CDUS
 TS MKCDUS
 CA CDUY
 TS MKCDUY
 CA CDUZ
 TS MKCDUZ
 CA CDUX
 TS MKCDUX
 EXTEND
 DCA TIME2 # GET TIME
 DXCH MKT2T1
 EXTEND
 DCA MKT2T1
 DXCH SAMPTIME # RUPT TIME FOR NOUN 65.

 XCH Q
 TS QRUPT

 CAF BIT6 # SEE IF MARK OR MKREJECT
Page 226
 EXTEND
 RAND NAVKEYIN
 CCS A
 TC MARKIT # IT'S A MARK

 CAF BIT7 # NOT A MARK, SEE IF MKREJECT
 EXTEND
 RAND NAVKEYIN
 CCS A
 TC MKREJECT # IT'S A MARK REJECT

KEYCALL CAF OCT37 # NOT MARK OR MKREJECT, SEE IF KEYCOM
 EXTEND
 RAND NAVKEYIN
 EXTEND
 BZF +3 # IF NO INBITS
 TC POSTJUMP
 CADR KEYCOM # IT'S A KEY CODE, NOT A MARK.

 +3 TC ALARM # ALARM IF NO INBITS
 OCT 113

```

July 12, 2016

1789

TC RESUME

# Page 227

# PROGRAM NAME: MARKCONT

DATE: 19 SEPT 1967

# PROGRAM MODIFIED BY 258/278 PROGRAMMERS

LOG SECTION SXTMARK

# MOD BY: R. MELANSON TO ADD DOCUMENTATION

ASSEMBLY SUNDISK REV. 116

#

# FUNCTIONAL DESCRIPTION:

#

# MARKCONT IS USED TO PERFORM A SPECIAL MARK FUNCTION FOR R21, TO EXECUTE A SPECIAL DISPLAY  
# TO PERFORM A MARK OF THE STAR OR LAND SIGHTING BASED UPON FLASHING V-N.

#

# CALLING SEQUENCE:

#

# FROM MARKDIF

#

# NORMAL EXIT MODE:

#

# TASKOVER

#

# ALARM OR ABORT EXIT MODE:

#

# ALARM AND TASKOVER

#

# OUTPUT:

#

# 1) FOR R21:

# EBANK=EBANK7

# MRKBUF1 TO MRKBUF1+6 = TIME2,TIME1,CDUY,OPTICX,CDUZ,OPTICSY,CDUX OF CURRENT R21

# MRKBUF2 TO MRKBUF2+6 CONTAINS PREVIOUS R21 MARK VALUES.

# 2) FOR SPECIAL DISPLAY JOB:

# RUPTREG1 AND MRKBUF1 = CDUS, RUPTREG2, AND MRKBUF1+1 = CDUT.

# RUPTREG3 AND MRKBUF1+2 = TIME2, RUPTREG4, AND MRKBUF1+3 = TIME1.

# 3) FOR NORMAL MARKING:

# DECREMENT BITS14-12 OF MARKSTAT BY 1,

# BIT10 MARKSTAT SET TO 1, INCREMENT QPRET BY 7,

# STORE TIME2,TIME1,CDUY,CDUS,CDUZ,CDUT, AND CDUX IN VAC+1 TO VAC+7.

#

# ERASABLE INITIALIZATION:

#

# 1) FOR R21:

# BIT14 OF STATE+2 =1, MRKBUF1 TO MRKBUF1+6, ITEMP1, RUPTREG3,

# RUPTSTOR TO RUPTSTOR+6 EXCEPT RUPTSTOR+4.

# 2) FOR SPECIAL DISPLAY JOB:

# BIT14 OF STATE+2 =0, MARKSTAT =+0, RUPTREG1, RUPTREG2, RUPTREG3

# RUPTREG4, RUPTSTOR, RUPTSTOR+1, RUPTSTOR+3, RUPTSTOR+5,

1790

July 12, 2016

```

BIT12 OF STATE+5 (V59 FLAG), MRKBUF1 THRU MRKBUF1+3
3) FOR NORMAL MARKING:
BIT14 OF STATE+2 =0, MARKSTATE =VAC ADDRESS, A REG, ITEMP1, RUPTREG3,
RUPTSTOR TO RUPTSTOR+6 EXCEPT RUPTSTOR+4.
#
DEBRIS:
#
1) FOR R21:
A, ITEMP1, MRKBUF1, MRKBUF2
2) FOR SPECIAL DISPLAY JOB:
A, RUPTREG1, RUPTREG2, RUPTREG3, RUPTREG4, MPAC TO MPAC+3.
3) FOR NORMAL MARKING:
A, MARKSTAT, ITEMP1, QPRET, VAC+1 TO VAC+7 OF VAC AREA IN USE.

```

# Page 228

MARKCONT

MARKIT1

MARKET

```

CAF BIT14
MASK STATE +2 # R21 MARK (SPECIAL MARKING FOR R21)
EXTEND
BZF MARKET # NOT SET THEREFORE REGULAR MARKING
CAF SIX # SPECIAL FOR R21
TC GENTRAN # TRANSFER MRKBUF1 TO MRKBUF2
ADRES MRKBUF1
ADRES MRKBUF2

CAF SIX # TRANSFER CURRENT MARK DATE TO MARK
TC GENTRAN
ADRES MKT2T1
ADRES MRKBUF1

TCF TASKOVER

CCS MARKSTAT # SEE IF MARKS CALLED FOR
TC MARK2 # COLLECT MARKS

CAF TWO # IS MARKING SYSTEM IN USE (BIT2)
MASK EXTVBACT
EXTEND
BZF MARKET3 # MARKING NOT CALLED FOR
CAF BIT12
MASK STATE +5 # V59FLAG
EXTEND
BZF MARKET3 # IF V59FLAG NOT SET-MARK UNCALLED FOR
CAF PRI05 # CALIBRATION MARK (SET) FOR P23
TC NOVAC # SPECIAL DISPLAY JOB
EBANK= MRKBUF1
2CADR MARKDISP

```

July 12, 2016

1791

|         |       |          |                                         |
|---------|-------|----------|-----------------------------------------|
|         | CAF   | SIX      |                                         |
|         | TC    | GENTRAN  | # TRANSFER MARK DATE TO MARKDOWN        |
|         | ADRES | MKT2T1   |                                         |
|         | ADRES | MARKDOWN |                                         |
|         | CAF   | SIX      |                                         |
|         | TC    | GENTRAN  | # TRANSFER MARK DATA TO MRKBUF1 FOR     |
|         | ADRES | MKT2T1   | # SPECIAL DISPLAY OF SHAFT AND TRUNNION |
|         | ADRES | MRKBUF1  | # IF V59 ACTING                         |
|         | TCF   | TASKOVER |                                         |
| MARKET3 | TC    | ALARM    |                                         |
|         | OCT   | 122      | # MARKING NOT CALLED FOR                |
|         | TCF   | TASKOVER |                                         |
| 114ALM  | TC    | ALARM    | # MARK NOT WANTED                       |
|         | OCT   | 114      |                                         |
|         | TCF   | TASKOVER |                                         |

# Page 229

# STORE MARK DATA IN MKVAC AND INCREMENT POINTER

|         |        |          |                                           |
|---------|--------|----------|-------------------------------------------|
| MARK2   | AD     | 74K      | # SEE IF MARKS WANTED-REDUCE MARKS WANTED |
|         | EXTEND |          |                                           |
|         | BZMF   | 114ALM   | # MARK NOT WANTED-ALARM                   |
|         | TS     | MARKSTAT |                                           |
|         | COM    |          |                                           |
|         | MASK   | BIT10    | # SET BIT10 TO ENABLE REJECT              |
|         | ADS    | MARKSTAT |                                           |
|         | MASK   | LOW9     |                                           |
|         | TS     | ITEMP1   |                                           |
|         | INDEX  | A        |                                           |
|         | XCH    | QPRET    | # PICK UP MARK SLOT-POINTER               |
|         | TS     | ITEMP2   | # SAVE CURRENT POINTER                    |
|         | AD     | SEVEN    | # INCREMENT POINTER                       |
|         | INDEX  | ITEMP1   |                                           |
|         | TS     | QPRET    | # STORE ADVANCED POINTER                  |
| VACSTOR | EXTEND |          |                                           |
|         | DCA    | MKT2T1   |                                           |
|         | INDEX  | ITEMP2   |                                           |
|         | DXCH   | 0        |                                           |
|         | CA     | MKCDUY   |                                           |
|         | INDEX  | ITEMP2   |                                           |
|         | TS     | 2        |                                           |
|         | CA     | MKCDUS   |                                           |
|         | INDEX  | ITEMP2   |                                           |

```

TS 3
CA MKCDUZ
INDEX ITEMP2
TS 4
CA MKCDUT
INDEX ITEMP2
TS 5
CA MKCDUX
INDEX ITEMP2
TS 6

```

```

CAF PRI034
MASK MARKSTAT
EXTEND
BZF +2
TCF TASKOVER
CAF PRI032
TC NOVAC
EBANK= MARKSTAT
2CADR MKVB50

```

```
IF ALL MARKS MADE FLASH VB50
```

```
TCF TASKOVER
```

```
Page 230
```

```
PROGRAM NAME: MKREJECT
```

```
DATE: 5 APRIL 1967
```

```
PROGRAM MODIFIED BY 258/276 PROGRAMMERS
```

```
LOG SECTION SXTMARK
```

```
MOD BY: R. MELANSON TO ADD DOCUMENTATION
```

```
ASSEMBLY SUNDISK REV
```

```
#
```

```
FUNCTIONAL DESCRIPTION:
```

```
#
```

```
ROUTINE ALLOWS OPEATOR TO REJECT MARK MADE PRIOR TO ACCEPTANCE AND ALLOWS A M
```

```
#
```

```
CALLING SEQUENCE:
```

```
#
```

```
FROM MARKRUPT IF BIT7 OF CHANNEL 16 IS 1.
```

```
#
```

```
NORMAL EXIT MODE:
```

```
#
```

```
RESUME
```

```
#
```

```
ALARM OR ABORT EXIT MODE:
```

```
#
```

```
ALARM AND RESUME
```

```
#
```

```
OUTPUT:
```

```
#
```



July 12, 2016

1793

```
1) FOR R21:
MRKRUP1 SET TO -1
2) FOR NORMAL MARKING:
BIT10 MARKSTAT =0, INCREMENT NO. MARKS BY 1, DECREMENT QPRET BY 7
#
ERASABLE INITIALIZATION:
#
1) FOR R21:
BIT14 OF STATE+2 SET TO 1
2) FOR NORMAL MARKING:
BIT14 OF STATE+2 SET TO 0, MARKSTAT,QPRET
#
DEBRIS:
#
1) FOR R21:
A,MARKSTAT,EBANK
2) FOR NORMAL MARKING:
A,MARKSTAT,ITEMP1,QPRET

MKREJECT CAF BIT14
 MASK STATE +2 # R21 MARK (SPECIAL MARKING FOR R21)
 EXTEND
 BZF MRKREJCT # NOT SET THEREFORE REGULAR REJECT
 CA NEGONE # -1 (FOR R22)
 TS MRKBUF1 # -0 IN TIME IS FLAG TO R22 SIGNIFYING A
 TC RESUME # REJECTED MARK
MRKREJCT CCS MARKSTAT # SEE IF MARKS BEING ACCEPTED
 TC REJECT2
 TC ALARM # MARKS NOT BEING ACCEPTED
 OCT 112
 TC RESUME

REJECT2 CS BIT10 # SEE IF MARK HAD BEEN MADE SINCE LAST
 MASK MARKSTAT # REJECT, AND SET BIT10 TO ZERO TO
 XCH MARKSTAT # SHOW MARK REJECT

Page 231

 MASK BIT10
 CCS A
 TC REJECT3

 TC ALARM # DON'T ACCEPT TWO REJECTS TOGETHER
 OCT 110
 TC RESUME

REJECT3 CAF LOW9 # DECREMENT POINTER TO REJECT MARK
 MASK MARKSTAT
```

TS ITEMP1  
 CS SEVEN  
 INDEX ITEMP1  
 ADS QPRET

# NEW POINTER

CAF BIT12  
 AD MARKSTAT  
 XCH MARKSTAT  
 MASK PRI034  
 CCS A  
 TC RESUME  
 CAF PRI032  
 TC NOVAC  
 EBANK= MARKSTAT  
 2CADR MKVB51

# INCREMENT MARKS WANTED AND IF FIELD  
 # IS NOW NON-ZERO, CHANGE TO VB51 TO  
 # INDICATE MORE MARKS WANTED  
 # INDICATE MORE MARKS WANTED

TC RESUME

# Page 232

# PROGRAM DESCRIPTON MKVB51 AND MKVB50

#

# AUTHOR: BARNERT DATE: 2-15-67 MOD: 0

# PURPOSE: FLASH V51N70,V51N43, OR V51 TO REQUEST MARKING,  
 # AND V50N25 R1=16 TO REQUEST TERMINATE MARKING.

#

# CALLING SEQUENCE: AS JOB WITHIN SXTMARK

#

# EXIT TO ENDMARK UPON RECEIPT OF V33, V34 CAUSES GOTOP00H, ENTER  
 # RECYCLES THE DISPLAY

#

# NOTE: SXTMARK AUTOMATICALLY CHANGES FROM CALLING MKVB51 TO MKVB50 WHEN  
 # SUFFICIENT MARKS HAVE BEEN MADE, AND THE REVERSE WHEN A MARK  
 # REJECT REDUCES THE NUMBER MADE BELOW THAT REQUIRED

#

# SUBROUTINES CALLED: BANKCALL, GOMARK2, GOODEND, ENDMARK, WAITLIST

#

# ALARM OR ABORT MODES: NONE

#

# ERASABLE USED: VERBREG, MARKSTAT, QPRET, DSPTEM1

#

# OUTPUT MARKSTAT = VAC ADDRESS

# QPRET = NO. MARKS

MKVB51

TC BANKCALL  
 CADR KLEENEX

# CLEAR DISPLAY FOR MARK VERB

July 12, 2016

1795

|            |        |          |                                            |
|------------|--------|----------|--------------------------------------------|
|            | CAF    | VB51     | # DISPLAY MARK VB51                        |
|            | TC     | BANKCALL |                                            |
|            | CADR   | GOMARK4  |                                            |
|            | TCF    | TERMSXT  | # VB34-TERMINATE                           |
|            | TCF    | ENTANSWR | # V33-PROCEED-MARKING DONE                 |
|            | TCF    | MKVB5X   | # ENTER-RECYCLE TO INITIAL MARK DISPLAY    |
| TERMSXT    | TC     | CLEARMRK | # CLEAR MARK ACTIVITY.                     |
|            | TC     | CHECKMM  |                                            |
|            | MM     | 03       |                                            |
|            | TCF    | +2       |                                            |
|            | TC     | TERMP03  |                                            |
|            | TC     | POSTJUMP |                                            |
|            | CADR   | TERM52   |                                            |
| TERMP03    | TC     | UPFLAG   |                                            |
|            | ADRES  | TRM03FLG |                                            |
| ENTANSWR   | CAF    | LOW9     | # PUT VAC ADR IN MARKSTAT AND NO. OF       |
|            | MASK   | MARKSTAT | # MARKS MADE IN QPRET BEFORE LEAVING       |
|            | TS     | MARKSTAT | # SXTMARK                                  |
|            | COM    |          |                                            |
|            | INDEX  | MARKSTAT |                                            |
|            | AD     | QPRET    |                                            |
| # Page 233 |        |          |                                            |
|            | EXTEND |          |                                            |
|            | BZMF   | JAMIT    | # NO MARKS MADE, SHOW IT IN QPRET, R53     |
|            | EXTEND |          | # WILL PICK IT UP AND RECYCLE              |
|            | MP     | BIT12    | # THIS PUTS NUMBER MARKS-1 IN A            |
|            | AD     | ONE      |                                            |
| JAMIT      | INDEX  | MARKSTAT | # STORE NO OF MARKS MADE                   |
|            | TS     | QPRET    |                                            |
|            | INHINT |          | # SERVICE OPTSTALL INTERFACE WITH          |
|            | CAF    | FIVE     |                                            |
|            | TC     | WAITLIST |                                            |
|            | EBANK= | MARKSTAT |                                            |
|            | 2CADR  | ENDMARKS |                                            |
|            | TC     | ENDMARK  | # KNOCKS DOWN MARKING FLAG + DOES ENDOFJOB |
| ENDMARKS   | CAF    | ONE      |                                            |
|            | TC     | IBNKCALL |                                            |
|            | CADR   | GOODEND  |                                            |
| MKVB5X     | CAF    | PRI034   |                                            |
|            | MASK   | MARKSTAT | # RE-DISPLAY VB51 IF MORE MARKS WANTED     |
|            | CCS    | A        | # AND VB50 IF ALL IN                       |

1796

July 12, 2016

MKVB50            TCF      MKVB51  
                 CAF      R1D1  
                 TS        DSPTM1  
                 CAF      V50N25  
                 TCF      MKVB51 +3

# OCT 16

V50N25            VN        5025  
VB51              VN        5100  
OCT37             =        LOW5

# PROGRAM NAME:   MARKIT

DATE: 19 SEPT 1967

#

# CALLING SEQUENCE:

#

#            FROM MARKRUPT IF CHAN 16 BIT 6 = 1

#

# EXIT

#

#            RESUME

#

# INPUT

#

#            CDUCHKWD.    ALSO ALL INITIALIZATION FOR MARKCONT

#

# OUTPUT

#

#            MKT2T1,MKCDUX,MKCDUY,MKCDUZ,MKCDUS,MKCDUT

#

# ALARM EXIT

#

#            NONE

MARKIT            CCS        CDUCHKWD  
                 TCF        +3

# DELAY OF CDUCHKWD CS IF PNZ

# Page 234

TCF            +2  
CAF            ZERO  
AD             ONE  
TC             WAITLIST  
EBANK=        MRKBUF1  
2CADR         MARKDIF

# 10 MS IF NO CHECK

TCF            RESUME

SETLOC        SXTMARK1  
BANK

July 12, 2016

1797

COUNT 20/SXTMK

```
PROGRAM NAME: MARKDIF
#
CALLING SEQUENCE
#
WAITLIST FROM MARKIT
#
EXIT
#
TASKOVER TO IBNKCALL TO MARKCONT
#
INPUT
#
OUTPUT FROM MARKIT, INPUT TO MARKCONT, CDUCHKWD
#
OUTPUT
#
RUPTSTOR - RUPTSTOR+3, RUPTREG3, RUPTSTOR+5 - RUPTSTOR+6
#
ALARM EXIT
#
ALARM AND TASKOVER

MARKDIF CA CDUCHKWD # IF DELAY CHECK IS ZERO OR NEG, ACP MARK
 EXTEND
 BZMF MKACPT
 CS BIT1
 TS MKNDX # SET INDEX -1
 CA MKCDUX
 TC DIFCHK # SEE IF VEHICLE RATE TO MUCH AT MARK
 CA MKCDUY
 TC DIFCHK
 CA MKCDUZ
 TC DIFCHK

MKACPT TC IBNKCALL
 CADR MARKCONT # MARK DATA OK, WHAT DO WE DO WITH IT

DIFCHK INCR MKNDX # INCREMENT INDEX

 EXTEND
 INDEX MKNDX

Page 235
 MSU CDUX # GET MARK(ICDU) - CURRENT(ICDU)
```

1798

July 12, 2016

CCS      A  
TCF      +4  
TC        Q  
TCF      +2  
TC        Q  
AD        NEG2  
EXTEND  
BZMF      -3  
  
TC        ALARM  
OCT       00121  
  
TCF       TASKOVER

# SEE IF DIFFERENCE GREATER THAN 3 B

# NOT GREATER

# COUPLED WITH PROGRAM ALARM

# DO NOT ACCEPT

This code is written to file `src/SXTMARK.s`.

**B.109 SYSTEM TEST STANDARD LEAD INS**

```

1799 <src/SYSTEM-TEST-STANDARD-LEAD-INS.s 1799>≡
Copyright: Public domain.
Filename: SYSTEM_TEST_STANDARD_LEAD_INS.agc
Purpose: Part of the source code for Comanche, build 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 420-422
Contact: Onno Hommes <ohommes@cmu.edu>.
Website: www.ibiblio.org/apollo.
Mod history: 05/07/09 OH Transcription Batch 1 Assignment
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further information.
Please report any errors to info@sandroid.org.

Page 420

EBANK= XSM

BANK 33
SETLOC E/PROG1
BANK

COUNT* $$/P07

SPECIAL PROGRAMS TO EASE THE PANGS OF ERASABLE MEMORY PROGRAMS.
#
E/BKCALL FOR DOING BANKCALLS FROM AND RETURNING TO ERASABLE.
#

```

```

THIS ROUTINE IS CALLABLE FROM ERASABLE OR FIXED. LIKE BANKCALL, HOWEVER, SWITCHING
IS NOT POSSIBLE.
#
THE CALLING SEQUENCE IS:
#
TC BANKCALL
CADR E/BKCALL
CADR ROUTINE # WHERE TO WANT TO GO IN FIXED.
RETURN HERE FROM DISPLAY TERMINATE, BAD STALL OR TC Q.
RETURN HERE FROM DISPLAY PROCEED OR GOOD RETURN FROM STALL.
RETURN HERE FROM DISPLAY ENTER OR RECYCLE.
#
THIS ROUTINE REQUIRES TWO ERASABLES (EBUF2, +1) IN UNSWITCHED WHICH ARE UNSHARED BY
OTHER EMEMORY PROGRAMS.
#
A + L ARE PRESERVED THROUGH BANKCALL AND E/BKCALL.

E/BKCALL DXCH BUF2 # SAVE A,L AND GET DP RETURN.
 DXCH EBUF2 # SAVE DP RETURN.
 INCR EBUF2 # RETURN +1 BECAUSE DOUBLE CADR.
 CA BBANK
 MASK LOW10 # GET CURRENT EBANK. (SBANK SOMEDAY)
 ADS EBUF2 +1 # FORM BBCON. (WAS FBANK)
 NDX EBUF2
 CA 0 -1 # GET CADR OF ROUTINE.
 TC SWCALL # GO TO ROUTINE, SETTING Q TO SWRETURN
 # AND RESTORING A + L.
 TC +4 # TX Q, V34, OR BADD STALL RETURN.
 TC +2 # PROCEED OR GOOD STALL RETURN.
 INCR EBUF2 # ENTER OR RECYCLE RETURN.
 INCR EBUF2
E/SWITCH DXCH EBUF2
 DTCB

Page 421
E/CALL FOR CALLING A FIXED MEMORY INTERPRETIVE SUBROUTINE FROM ERASABLE AND
#
THE CALLING SEQUENCE IS...
#
RTB
#
CADR E/CALL
CADR ROUTINE # THE INTERPRETIVE SUBROUTINE YOU WANT
RETURNS HERE IN INTERPRETIVE.

E/CALL LXCH LOC # ADRES -1 OF CADR.
 INDEX L

```



July 12, 2016

1801

```
CA L # CADR IN A.
INCR L
INCR L # RETURN ADRES IN L.
DXCH EBUF2 # STORE CADR AND RETURN.
TC INTERPRET
CALL
 EBUF2 # INDIRECTLY EXECUTE ROUTING. IT MUST
EXIT EBUF2 # LEAVE VIA RVQ OR EQUIVALENT.
LXCH EBUF2 +1 # PICK UP RETURN.
TCF INTERPRET +2 # SET LOC AND RETURN TO CALLER.
```

# Page 422

# E/JOBWAK FOR WAKING UP ERASABLE MEMORY JOBS.

#

# THIS ROUTINE MUST BE CALLED IN INTERRUPT OR WITH INTERRUPTS INHIBITED.

#

# THE CALLING SEQUENCE IS

#

# INHINT

# ...

# CA WAKEADR # ADDRESS OF SLEEPING JOB

# TC IBNKCALL

# CADR E/JOBWAK

# ... # RETURNS HERE

# RELINT # IF YOU DID AND INHINT

BANK 33

SETLOC E/PROG

BANK

COUNT\* \$\$/P07

E/JOBWAK TC JOBWAKE # ARRIVE IWTH ADRES IN A.

CS BIT11

NDX LOCCTR

ADS LOC # KNOCK FIXED MEMORY BIT OUT OF ADRES.

TC RUPTREG3 # RETURN

# THESE PROGRAMS ARE PROVIDED TO ALLOW OVERLAY OF BANKS 30 THRU 33 OF THE 205 VERSIONS OF SYSTEM  
# PRELAUNCH ALIGN. THE INTENT IS TO ALLOW THE STG AND HYBRID LABS TO RUN ALL THE TESTS WITH CO

BANK 33

SETLOC TESTLEAD

BANK

|          |        |          |                      |
|----------|--------|----------|----------------------|
|          | COUNT  | 33/COMST |                      |
|          | EBANK= | QPLACE   |                      |
| COMPVER  | TC     | GCOMPVER | # MUST BE 33,2000.   |
| GTSCPSS1 | TC     | GTSCPSS  | # MUST BE AT 33,2001 |
| REDO     | TC     | NEWMODEX | # DISPLAY MM 07.     |
|          | MM     | 07       | # FALL INTO IMUTEST  |

This code is written to file `src/SYSTEM-TEST-STANDARD-LEAD-INS.s`.

## B.110 T4RUPT PROGRAM

```

1803 <src/T4RUPT-PROGRAM.s 1803>≡
Copyright: Public domain.
Filename: T4RUPT_PROGRAM.agc
Purpose: Part of the source code for Comanche, build 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), Apollo 11.
Assembler: yaYUL
Reference: pp. 133-169
Contact: Ron Burkey <info@sandroid.org>,
Fabrizio Bernardini <fabrizio@spacecraft.it>
Website: http://www.ibiblio.org/apollo.
Mod history: 10/05/09 FB Transcription of Batch FB-1 Assignment.
#
The contents of the "Comanche055" files, in general, are transcribed
from scanned documents.
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. April 1, 1969.
#
This AGC program shall also be referred to as Colossus 2A
#
Prepared by
#
Massachussets Institute of Technology
75 Cambridge Parkway
Cambridge, Massachusetts
#
under NASA contract NAS 9-4065.
#
Refer directly to the online document mentioned above for further
information. Please report any errors to info@sandroid.org.

Page 133

BANK 12
SETLOC T4RUP
BANK

COUNT 06/T4RPT

T4RUPT TS BANKRUPT
 EXTEND
 QXCH QRUPT

CCS DSRUPTSW # GOES 7(-1)0 AROUND AND AROUND
TCF NORMT4 +1

```

|        |       |          |
|--------|-------|----------|
|        | TCF   | NORMT4   |
|        | TCF   | QUIKDSP  |
| NORMT4 | CAF   | SEVEN    |
|        | TS    | RUPTREG1 |
|        | TS    | DSRUPTSW |
|        | COUNT | 02/T4RPT |
| 74K    | =     | HIGH4    |

# RELTAB IS A PACKED TABLE. RELAYWORD CODE IN UPPER 4 BITS, RELAY CODE  
# IN LOWER 5 BITS.

|          |        |         |
|----------|--------|---------|
|          | BLOCK  | 02      |
|          | SETLOC | FFTAG12 |
|          | BANK   |         |
| RELTAB   | OCT    | 04025   |
|          | OCT    | 10003   |
|          | OCT    | 14031   |
|          | OCT    | 20033   |
|          | OCT    | 24017   |
|          | OCT    | 30036   |
|          | OCT    | 34034   |
|          | OCT    | 40023   |
|          | OCT    | 44035   |
|          | OCT    | 50037   |
|          | OCT    | 54000   |
| RELTAB11 | OCT    | 60000   |

# Page 134  
# SWITCHED-BANK PORTION

|       |        |             |
|-------|--------|-------------|
|       | BANK   | 12          |
|       | SETLOC | T4RUP       |
|       | BANK   |             |
|       | COUNT  | 06/T4RPT    |
| CDRVE | CCS    | DSPTAB +11D |
|       | TC     | DSPOUT      |
|       | TC     | DSPOUT      |
|       | XCH    | DSPTAB +11D |

July 12, 2016

1805

```
MASK LOW11
TS DSPTAB +11D
AD RELTAB11
EXTEND
WRITE OUT0
TC HANG20
```

# Page 135

# DSPOUT PROGRAM, PUTS OUT DISPLAYS

```
DSPOUTSB TS NOUT
 CS ZERO
 TS DSRUPTM # SET TO -0 FOR 1ST PASS THRU DSPTAB
 XCH DSPCNT
 AD NEG0 # TO PREVENT +0
 TS DSPCNT
DSPSCAN INDEX DSPCNT
 CCS DSPTAB
 CCS DSPCNT # IF DSPTAB ENTRY +, SKIP
 TCF DSPSCAN -2 # IF DSPCNT +, AGAIN
 TCF DSPLY # IF DSPTAB ENTRY -, DISPLAY
TABLNTH OCT 12 # DEC 10 LENGTH OF DSPTAB
 CCS DSRUPTM # IF DSRUPTM=+0, 2ND PASS THRU DSPTAB
120MRUPT DEC 16372 # (DSPCNT = 0). +0 INTO NOUT.
 TS NOUT
 TC Q
 TS DSRUPTM # IF DSRUPTM=-0, 1ST PASS THRU DSPTAB
 CAF TABLNTH # (DSPCNT=0).+0 INTO DSRUPTM. PASS AGAIN
 TCF DSPSCAN -1

DSPLY AD ONE
 INDEX DSPCNT
 TS DSPTAB # REPLACE POSITIVELY
 MASK LOW11 # REMOVE BITS 12 TO 15
 TS DSRUPTM
 CAF HI5
 INDEX DSPCNT
 MASK RELTAB # PICK UP BITS 12 TO 15 OF RELTAB ENTRY
 AD DSRUPTM
 EXTEND
 WRITE OUT0 # WRITE CHANNEL 10
 TCF Q+1 # *** NORMAL RETURN SKIPS ONE

DSPOUT CCS FLAGWRD5 # DON'T DISPLAY UNLESS DSKY FLAG ON
 CAF ZERO
 TCF NODSPOUT
```

```

 CCS NOUT
 TC DSPOUTSB
 TCF NODSPOUT # NO DISPLAY REQUESTS

HANG20 CS 11,14,9
 ADS DSRUPTSW

 CAF 20MRUPT

SETTIME4 TS TIME4

```

# Page 136

```

THE STATUS OF THE PROCEED PUSHBUTTON IS MONITORED EVERY 120 MILLISECONDS VIA THE C
THE STATE OF THIS INBIT IS COMPARED WITH ITS STATE DURING THE PREVIOUS T4RUPT AND 1
IF PREV ON AND NOW ON -- BYPASS
IF PREV ON AND NOW OFF -- UPDATE IMODES33
IF PREV OFF AND NOW ON -- UPDATE IMODES33 AND PROCESS VIA PINBALL
IF PREV OFF AND NOW OFF -- BYPASS
THE LOGIC EMPLOYED REQUIRES ONLY 9 MCT (APPROX. 108 MICROSECONDS) OF COMPUTER TIME

```

```

PROCEEDE CA IMODES33 # MONITOR FOR PROCEED BUTTON
 EXTEND
 RXOR CHAN32 # CHECK IF BIT 14 DIFFERENT
 MASK BIT14
 EXTEND
 BZF T4JUMP # NO CHANGE

 LXCH IMODES33
 EXTEND
 RXOR LCHAN
 TS IMODES33 # UPDATE IMODES33
 MASK BIT14
 CCS A
 TCF T4JUMP # WAS ON -- NOW OFF

 CAF CHRPRIO # WAS OFF -- NOW ON
 TC NOVAC
 EBANK= DSPCOUNT
 2CADR PROCKEY

```

# Page 137

# JUMP TO APPROPRIATE ONCE-PER SECOND (0.96 SEC ACTUALLY) ACTIVITY

```

T4JUMP INDEX RUPTREG1
 TCF +1

```

July 12, 2016

1807

|          |        |          |                                         |
|----------|--------|----------|-----------------------------------------|
|          | TCF    | OPTTEST  |                                         |
|          | TCF    | OPTMON   |                                         |
|          | TCF    | IMUMON   |                                         |
|          | TCF    | RESUME   |                                         |
|          | TCF    | OPTTEST  |                                         |
|          | TCF    | OPTMON   |                                         |
|          | TCF    | IMUMON   |                                         |
|          | TCF    | RESUME   |                                         |
| OPTTEST  | TC     | IBNKCALL |                                         |
|          | CADR   | OPTDRIVE |                                         |
| 20MRUPT  | =      | OCT37776 | # (DEC 16382)                           |
| NODSPOUT | EXTEND |          | # TURN OFF RELAYS                       |
|          | WRITE  | OUT0     |                                         |
|          | CAF    | 120MRUPT | # SET FOR NEXT CDRVE                    |
|          | TCF    | SETTIME4 |                                         |
| QUIKDSP  | CAF    | BIT14    |                                         |
|          | MASK   | DSRUPTSW |                                         |
|          | EXTEND |          |                                         |
|          | BZF    | QUIKOFF  | # WROTE LAST TIME, NOW TURN OFF RELAYS. |
|          | CCS    | NOUT     |                                         |
|          | TC     | DSPOUTSB |                                         |
|          | TCF    | NODSPY   | # NOUT=0 OR BAD RETURN FROM DSPOUTSB    |
|          | CS     | BIT14    | # GOOD RETURN (WE DISPLAYED SOMETHING)  |
| QUIKRUPT | ADS    | DSRUPTSW |                                         |
|          | CAF    | 20MRUPT  |                                         |
|          | TS     | TIME4    |                                         |
|          | CAF    | BIT9     |                                         |
|          | ADS    | DSRUPTSW |                                         |
|          | TC     | RESUME   |                                         |
| NODSPY   | EXTEND |          |                                         |
|          | WRITE  | OUT0     |                                         |
| SYNCT4   | CAF    | 20MRUPT  |                                         |
|          | ADS    | TIME4    |                                         |
|          | CAF    | BIT9     |                                         |

# Page 138

```

 ADS DSRUPTSW
 CCS DSRUPTSW
 TC RESUME
OCT37737 OCT 37737
 TC SYNCT4
 TC RESUME

QUIKOFF EXTEND
 WRITE OUT0
 CAF BIT14 # RESET DSRUPTSW TO SEND DISPLAY NEXT PASS
 TCF QUIKRUP

```

```

11,14,9 OCT 22400

```

# Page 139

# PROGRAM NAME: IMUMON

#

```

FUNCTIONAL DESCRIPTION: THIS PROGRAM IS ENTERED EVERY 480 MS. IT DETECTS CHANGES
CHANNEL 30 AND CALLS THE APPROPRIATE SUBROUTINES. THE BITS PROCESSED AND THEIR RE

```

#

| FUNCTION            | BIT | SUBROUTINE CALLED  |
|---------------------|-----|--------------------|
| -----               | --- | -----              |
| TEMP IN LIMITS      | 15  | TLIM               |
| ISS TURN-ON REQUEST | 14  | ITURNON            |
| IMU FAIL            | 13  | IMUFAIL (SETISSW)  |
| IMU CDU FAIL        | 12  | ICDUFail (SETISSW) |
| IMU CAGE            | 11  | IMUCAGE            |
| IMU OPERATE         | 9   | IMUOP              |

#

```

THE LAST SAMPLED STATE OF THESE BITS IS LEFT IN IMODES30. ALSO, EACH SUBROUTINE CA
VALUE OF THE BIT IN A, WITH Q SET TO THE PROPER RETURN LOCATION NXTIFAIL.

```

#

```

CALLING SEQUENCE: T4RUPT EVERY 480 MILLISECONDS.

```

#

```

JOBS OR TASKS INITIATED: NONE.

```

#

```

SUBROUTINES CALLED: TLIM, ITURNON, SETISSW, IMUCAGE, IMUOP.

```

#

```

ERASABLE INITIALIZATION:

```

```

FRESH START OR RESTART WITH NO GROUPS ACTIVE: C(IMODES30) = OCT 37411.

```

```

RESTART WITH ACTIVE GROUPS: C(IMODES30) = (B(IMODES30)AND(OCT 00035)) PL
THIS LEAVES IMU FAIL BITS INTACT.

```

#

```

ALARMS: NONE.

```

#



July 12, 2016

1809

```
EXIT: TNONTEST.
#
OUTPUT: UPDATED IMODES30 WITH CHANGES PROCESSED BY APPROPRIATE SUBROUTINE.
```

```
IMUMON CA IMODES30 # SEE IF THERE HAS BEEN A CHANGE IN THE
EXTEND # RELEVANT BITS OF CHAN 30.
RXOR CHAN30 # CHECK IF BITS 9,11-15 CHANGED
MASK 3ORDMSK
EXTEND
BZF TNONTEST # NO CHANGE IN STATUS
```

```
TS RUPTREG1 # SAVE BITS WHICH HAVE CHANGED.
LXCH IMODES30 # UPDATE IMODES30.
EXTEND
RXOR LCHAN
TS IMODES30
```

```
CS ONE
XCH RUPTREG1
EXTEND
```

# Page 140

```
BZMF TLIM # CHANGE IN IMU TEMP.
TCF NXTIFBIT # BEGIN BIT SCAN.
```

```
 -1 AD ONE # (RE-ENTERS HERE FROM NXTIFAIL.)
NXTIFBIT INCR RUPTREG1 # ADVANCE BIT POSITION NUMBER.
 +1 DOUBLE
```

```
TS A # SKIP IF OVERFLOW.
TCF NXTIFBIT # LOOK FOR BIT.
```

```
XCH RUPTREG2 # SAVE OVERFLOW-CORRECTED DATA.
INDEX RUPTREG1 # SELECT NEW VALUE OF THIS BIT.
CAF BIT14
MASK IMODES30
INDEX RUPTREG1
TC IFAILJMP
```

```
NXTIFAIL CCS RUPTREG2 # PROCESS ANY ADDITIONAL CHANGES.
TCF NXTIFBIT -1
```

# Page 141

# PROGRAM NAME: TNONTEST.

#

# FUNCTIONAL DESCRIPTION: THIS PROGRAM HONORS REQUESTS FOR ISS INITIALIZATION. ISS TURN-ON (C  
# AND ISS OPERATE (CHANNEL 30 BIT 9) REQUESTS ARE TREATED AS A PAIR AND PROCESSING TAKES PLACE  
# AFTER EITHER ONE APPEARS. THIS INITIALIZATION TAKES ON ONE OF THE FOLLOWING THREE FORMS:

```

#
1) ISS TURN-ON: IN THIS SITUATION THE COMPUTER IS OPERATING WHEN THE ISS IS
BOTH ISS TURN-ON AND ISS OPERATE APPEAR. THE PLATFORM IS CAGED FOR 90 SECONDS
SO THAT AT THE END OF THE PROCESS THE GIMBAL LOCK MONITOR WILL FUNCTION PROPERLY.
#
2) ICDU INITIALIZATION: IN THIS CASE THE COMPUTER WAS PROBABLY TURNED ON WITH
A FRESH START WAS DONE WITH THE ISS IN OPERATE. IN THIS CASE ONLY ISS OPERATE
ZEROED SO THE GIMBAL LOCK MONITOR WILL FUNCTION. AN EXCEPTION IS IF THE ISS
A RESTART, THE ICDU'S WILL NOT BE ZEROED.
#
3) RESTART WITH RESTARTABLE PROGRAM USING THE IMU: IN THIS CASE, NO INITIAL
IT IS ASSUMED THAT THE USING PROGRAM DID THE INITIALIZATION AND THEREFORE T4
#
IMODES30 BIT 7 IS SET = 1 BY THE FIRST BIT (CHANNEL 30 BIT 14 OR 9) WHICH ARRIVES.
ENTERED, FINDS BIT 7 = 1 BUT BIT 8 = 0, SO IT SETS BIT 8 = 1 AND EXITS. THE NEXT
PROCEEDS, SETTING BITS 8 AND 7 = 0. AT PROCTNON, IF ISS TURN-ON REQUEST IS PRESENT
COARSE). IF ISS OPERATE IS NOT PRESENT PROGRAM ALARM 00213 IS ISSUED. AT THE END
OF IMODES30 IS TESTED. IF IT IS = 1, ISS TURN-ON WAS NOT PRESENT FOR THE ENTIRE 90
THE ISS TURN-ON REQUEST IS PRESENT THE 90 SECOND WAIT IS REPEATED, OTHERWISE NO AC
WAS WAITING FOR THE INITIALIZATION IN WHICH CASE THE PROGRAM IS GIVEN AN IMUSTALL
WENT PROPERLY, THE ISS DELAY OUTBIT IS SENT AND THE ICDU'S ZEROED. A TASK IS INITI
INHIBIT BIT IN 10.24 SECONDS. IF A MISSION PROGRAM WAS WAITING IT IS INFORMED VIA
#
AT PROCTNON, IF ONLY ISS OPERATE IS PRESENT (OPONLY), THE CDU'S ARE ZEROED UNLESS
ALIGN (= GIMBAL LOCK HERE) OR A MISSION PROGRAM IS USING THE IMU (INUSEFLG = 1).
#
CALLING SEQUENCE: T4RUPT EVERY 480 MILLISECONDS AFTER IMUMON.
#
JOBS OR TASKS INITIATED: 1) ENDTNON, 90 SECONDS AFTER CAGING STARTED. 2) ISSUP, 4
3) PFAILOK, 10.24 SECONDS AFTER INITIALIZATION COMPLETED. 4) UNZ2, 320 MILL
STARTED.
#
SUBROUTINES CALLED: CAGESUB, CAGESUB2, ZEROICDU, ENDIMU, IMUBAD, NOATTOFF, SETISSW
#
ERASABLE INITIALIZATION: SEE IMUMON.
#
ALARMS: PROGRAM ALARM 00213 IF ISS TURN-ON REQUESTED WITHOUT ISS OPERATE.
#
EXIT: ENDTNON EXITS TO C33TEST. TASKS HAVING TO DO WITH INITIALIZATION EXIT AS F
WAITING AND INITIALIZATION COMPLETE, EXIT TO ENDIMU, MISSION PROGRAM WAITING AND I
IMUBAD, IMU NOT IN USE, EXIT TO TASKOVER.
#
OUTPUT: ISS INITIALIZED.

```

```

TNONTEST CS IMODES30 # AFTER PROCESSING ALL CHANGES, SEE 1
Page 142

```

July 12, 2016

1811

```

MASK BIT7 # IS TIME TO ACT ON A TURN-ON SEQUENCE.
CCS A
TCF C33TEST # NO -- EXAMINE CHANNEL 33.

CAF BIT8 # SEE IF FIRST SAMPLE OR SECOND.
MASK IMODES30
CCS A
TCF PROCTNON # REACT AFTER A SECOND SAMPLE.

CAF BIT8 # IF FIRST SAMPLE, SET BIT TO REACT NEXT
ADS IMODES30 # TIME.
TCF C33TEST

PROCESS IMU TURN-ON REQUESTS AFTER WAITING 1 SAMPLE FOR ALL SIGNALS TO ARRIVE.

PROCTNON CS BITS7&8
 MASK IMODES30
 TS IMODES30
 MASK BIT14 # SEE IF TURN-ON REQUEST.
 CCS A
 TCF OPONLY # OPERATE ON ONLY.

 CS IMODES30 # IF TURN-ON REQUEST, WE SHOULD HAVE IMU
 MASK BIT9 # OPERATE.
 CCS A
 TCF +3

 TC ALARM # ALARM IF NOT
 OCT 213

+3 TC CAGESUB

 CAF 90SECS
 TC WAITLIST
 EBANK= CDUIND
 2CADR ENDTNON

 TCF C33TEST

RETNON CAF 90SECS
 TC VARDELAY

ENDTNON CS BIT2 # RESET TURN-ON REQUEST FAIL BIT.
 MASK IMODES30
 XCH IMODES30
 MASK BIT2 # IF IT WAS OFF, SEND ISS DELAY COMPLETE.
```

# Page 143

|          |        |          |                                       |
|----------|--------|----------|---------------------------------------|
|          | EXTEND |          |                                       |
|          | BZF    | ENDTNON2 |                                       |
|          | CAF    | BIT14    | # IF IT WAS ON AND TURN-ON REQUEST NO |
|          | MASK   | IMODES30 | # PRESENT, RE-ENTER 90 SEC DELAY IN V |
|          | EXTEND |          |                                       |
|          | BZF    | RETNON   |                                       |
|          | CS     | STATE    | # IF IT IS NOT ON NOW, SEE IF A PROG  |
|          | MASK   | IMUSEFLG | # WAITING.                            |
|          | CCS    | A        |                                       |
|          | TCF    | TASKOVER |                                       |
|          | TC     | POSTJUMP |                                       |
|          | CADR   | IMUBAD   | # UNSUCCESSFUL TURN-ON.               |
| ENDTNON2 | CAF    | BIT15    | # SEND ISS DELAY COMPLETE.            |
|          | EXTEND |          |                                       |
|          | WOR    | CHAN12   | # TURN OFF ISS DELAY COUNTER          |
|          | TC     | IBNKCALL | # TURN OFF NO ATT LAMP.               |
|          | CADR   | NOATTOFF |                                       |
| UNZ2     | TC     | ZEROICDU |                                       |
|          | CS     | BITS4&5  | # REMOVE ZERO AND COARSE.             |
|          | EXTEND |          |                                       |
|          | WAND   | CHAN12   |                                       |
|          | CAF    | BIT11    | # WAIT 10 SECS FOR CTRS TO FIND GIMBA |
|          | TC     | VARDELAY |                                       |
| ISSUP    | CS     | OCT54    | # REMOVE CAGING, IMU FAIL INHIBIT, AD |
|          | MASK   | IMODES30 | # ICDUFAIL INHIBIT FLAGS.             |
|          | TS     | IMODES30 |                                       |
|          | CS     | BIT6     | # ENABLE DAP                          |
|          | MASK   | IMODES33 |                                       |
|          | TS     | IMODES33 |                                       |
|          | TC     | SETISSW  | # ISS WARNING MIGHT HAVE BEEN INHIBIT |
|          | CS     | BIT15    | # REMOVE IMU DELAY COMPLETE DISCRETE  |
|          | EXTEND |          |                                       |
|          | WAND   | CHAN12   |                                       |
|          | CAF    | 4SECS    | # DONT ENABLE PROG ALARM ON PIP FAIL  |
|          | TC     | WAITLIST | # ANOTHER 4 SECS.                     |

July 12, 2016

1813

```
EBANK= CDUIND
2CADR PFAILOK

TCF TASKOVER

OPONLY
Page 144

CAF BIT4

EXTEND
RAND CHAN12
CCS A
TCF C33TEST

CAF IMUSEFLG
MASK STATE
CCS A
TCF C33TEST

TC CAGESUB2

TC IBNKCALL
CADR NOATTOFF

CAF BIT5
EXTEND
WOR CHAN12

TC ZEROICDU
CAF BIT6
TC WAITLIST
EBANK= OPTMODES
2CADR UNZ2

TCF C33TEST

IF OPERATE ON ONLY AND WE ARE IN COARSE
ALIGN, DON'T ZERO THE CDUS BECAUSE WE
MIGHT BE IN GIMBAL LOCK. USE V41N20 TO
RECOVER.

OTHERWISE, ZERO THE COUNTERS
UNLESS SOMEONE IS USING THE IMU.

SET TURNON FLAGS.

TURN OFF NO ATT LAMP.
IMU CAGE OFF ENTRY.

ISS CDU ZERO

WAIT 300 MS FOR AGS TO RECEIVE SIGNAL.

Page 145
PROGRAM NAME: C33TEST
#
FUNCTIONAL DESCRIPTION: THIS PROGRAM MONITORS THREE FLIP-FLOP INBITS OF CHANNEL 33 AND CALLS
SUBROUTINE TO PROCESS A CHANGE. IT IS ANALOGOUS TO IMUMON, WHICH MONITORS CHANNEL 30, EXCEPT
CHANNEL 33 WITH A WAND INSTRUCTION BECAUSE A 'WRITE' PULSE IS REQUIRED TO RESET THE FLIP-FLOP
PROCESSED AND THE SUBROUTINES CALLED ARE:
BIT FUNCTION SUBROUTINE
--- -
13 PIPA FAIL PIPFAIL
12 DOWNLINK TOO FAST DNTMFAST
11 UPLINK TOO FAST UPTMFAST
```

```

#
UPON ENTRY TO THE SUBROUTINE, THE NEW BIT STATE IS IN A.
#
CALLING SEQUENCE: EVERY 480 MILLISECONDS AFTER TNONTEST.
#
JOBS OR TASKS INITIATED: NONE.
#
SUBROUTINES CALLED: PIPFAIL, DNTMFAST AND UPTMFAST ON BIT CHANGES.
#
ERASABLE INITIALIZATION: C(IMODES33) = OCT 16000 ON A FRESH START OR RESTART, THEN
REAPPEAR IF THE CONDITIONS PERSIST.
#
ALARMS: NONE.
#
EXIT: GLOCKMON.
#
OUTPUT: UPDATED BITS 13, 12, AND 11 OF IMODES33 WITH CHANGES PROCESSED.

C33TEST CA IMODES33 # SEE IF RELEVANT CHAN33 BITS HAVE
 MASK 33RDMSK
 TS L # CHANGED.
 CAF 33RDMSK
 EXTEND
 WAND CHAN33 # RESETS FLIP-FLOP INPUTS
 EXTEND
 RXOR LCHAN
 EXTEND
 BZF GLOCKMON # ON NO CHANGE.

 TS RUPTREG1 # SAVE BITS WHICH HAVE CHANGED
 LXCH IMODES33
 EXTEND
 RXOR LCHAN
 TS IMODES33 # UPDATED IMODES33.

 CAF ZERO
 XCH RUPTREG1
 DOUBLE

Page 146
 TCF NXTIBT +1 # SCAN FOR BIT CHANGES.

 -1 AD ONE
NXTIBT INCR RUPTREG1
 +1 DOUBLE
 TS A # (CODING IDENTICAL TO CHAN 30).
 TCF NXTIBT

```

July 12, 2016

1815

```

XCH RUPTREG2
INDEX RUPTREG1 # GET NEW VALUE OF BIT WHICH CHANGED.
CAF BIT13
MASK IMODES33
INDEX RUPTREG1
TC C33JMP

NXTFL33 CCS RUPTREG2 # PROCESS POSSIBLE ADDITIONAL CHANGES.
TCF NXTIBT -1

Page 147
PROGRAM NAME: GLOCKMON
#
FUNCTIONAL DESCRIPTION: THIS PROGRAM MONITORS THE CDUZ COUNTER TO DETERMINE WHETHER THE ISS
AND TAKES ACTION IF IT IS. THREE REGIONS OF MIDDLE GIMBAL ANGLE (MGA) ARE USED:
#
1) ABS(MGA) LESS THAN OR EQUAL TO 70 DEGREES -- NORMAL MODE.
2) ABS(MGA) GREATER THAN 70 DEGREES AND LESS THAN OR EQUAL TO 85 DEGREES -- GIMBAL LOCK
3) ABS(MGA) GREATER THAN 85 DEGREES -- ISS PUT IN COARSE ALIGN AND NO ATT LAMP TURNED C
#
CALLING SEQUENCE: EVERY 480 MILLISECONDS AFTER C33TEST.
#
JOBS OR TASKS INITIATED: NONE.
#
SUBROUTINES CALLED: 1) SETCOARS WHEN ABS(MGA) GREATER THEN 85 DEGREES AND ISS NOT IN COARSE
2) LAMPTEST BEFORE TURNING OFF GIMBAL LOCK LAMP.
#
ERASABLE INITIALIZATION:
1) FRESH START OR RESTART WITH NO GROUPS ACTIVE: C(CDUZ) = 0, IMODES30 BIT 6 =
2) RESTART WTIH GROUPS ACTIVE: SAME AS FRESH START EXCEPT C(CDUZ) NOT CHANGED
PROCEEDS AS BEFORE.
#
ALARMS: 1) MGA REGION (2) CAUSES GIMBAL LOCK LAMP TO BE LIT.
2) MGA REGION (3) CAUSES THE ISS TO BE PUT IN COARSE ALIGN AND THE NO ATT LAMP
SO ALREADY.

GLOCKMON CCS CDUZ
TCF GLOCKCHK # SEE IF MAGNITUDE OF MGA IS GREATER THAN
TCF SETGLOCK # 70 DEGREES.
TCF GLOCKCHK
TCF SETGLOCK

GLOCKCHK AD -70DEGS
EXTEND
BZMF SETGLOCK -1 # NO LOCK.
```

```

AD -15DEGS # SEE IF ABS(MGA) GREATER THAN 85 DEG
EXTEND
BZMF NOGIMRUN

CAF BIT4 # IF SO, SYSTEM SHOULD BE IN COARSE A
EXTEND
RAND CHAN12
CCS A
TCF NOGIMRUN

TC IBNKCALL # GO INTO COARSE ALIGN.
CADR SETCOARS

CAF SIX # ENABLE ISS ERROR COUNTERS IN 60 MS
TC WAITLIST

EBANK= CDUIND
2CADR CA+ECE

NOGIMRUN CAF BIT6 # TURN ON GIMBAL LOCK LAMP.
TCF SETGLOCK

-1
SETGLOCK CAF ZERO
AD DSPTAB +11D # SEE IF PRESENT STATE OF GIMBAL LOCK
MASK BIT6 # AGREES WITH DESIRED STATE BY HALF A
EXTEND
BZF GLOCKOK # OK AS IS.

MASK DSPTAB +11D # IF OFF, DON'T TURN ON IF IMU BEING
CCS A
TCF GLAMPTST # TURN OFF UNLESS LAMP TEST IN PROGR

CAF BIT6
MASK IMODES30
CCS A
TCF GLOCKOK

GLINVERT CS DSPTAB +11D # INVERT GIMBAL LOCK LAMP.
MASK BIT6
AD BIT15 # TO INDICATE CHANGE IN DSPTAB +11D.
XCH DSPTAB +11D
MASK OCT37737
ADS DSPTAB +11D
TCF GLOCKOK

```



July 12, 2016

1817

|          |     |          |                                           |
|----------|-----|----------|-------------------------------------------|
| GLAMPTST | TC  | LAMPTST  | # TURN OFF UNLESS LAMP TEST IN PROGRESS.  |
|          | TCF | GLOCKOK  |                                           |
|          | TCF | GLINVERT |                                           |
| -70DEGS  | DEC | -.38888  | # -70 DEGREES SCALED IN HALF-REVOLUTIONS. |
| -15DEGS  | DEC | -.08333  |                                           |

# Page 149

# PROGRAM NAME: TLIM.

#

# FUNCTIONAL DESCRIPTION: THIS PROGRAM MAINTAINS THE TEMP LAMP (BIT 4 OF CHANNEL 11) ON THE DS  
# THE TEMP SIGNAL FROM THE ISS (BIT 15 OF CHANNEL 30). HOWEVER, THE LIGHT WILL NOT BE TURNED C  
# IS IN PROGRESS.

#

# CALLING SEQUENCE: CALLED BY IMUMON ON A CHANGE OF BIT 15 OF CHANNEL 30.

#

# JOBS OR TASKS INITIATED: NONE.

#

# SUBROUTINES CALLED: LAMPTST.

#

# ERASABLE INITIALIZATION: FRESH START AND RESTART TURN THE TEMP LAMP OFF.

#

# ALARMS: TEMP LAMP TURNED ON WHEN THE IMU TEMP GOES OUT OF LIMITS.

#

# EXIT: NXTIFAIL.

#

# OUTPUT: SERVICE OF TEMP LAMP.

IN A, EXCEPT FOR TLIM.

|      |      |          |                                           |
|------|------|----------|-------------------------------------------|
| TLIM | MASK | POSMAX   | # REMOVE BIT FROM WORD OF CHANGES AND SET |
|      | TS   | RUPTREG2 | # DSKY TEMP LAMP ACCORDINGLY.             |

|     |          |
|-----|----------|
| CCS | IMODES30 |
| TCF | TEMPOK   |
| TCF | TEMPOK   |

|        |          |
|--------|----------|
| CAF    | BIT4     |
| EXTEND |          |
| WOR    | DSALMOUT |
| TCF    | NXTIFAIL |

# TURN ON LAMP.

|        |     |          |                                          |
|--------|-----|----------|------------------------------------------|
| TEMPOK | TC  | LAMPTST  | # IF TEMP NOW OK, DON'T TURN OFF LAMP IF |
|        | TCF | NXTIFAIL | # LAMP TEST IN PROGRESS.                 |

|        |          |
|--------|----------|
| CS     | BIT4     |
| EXTEND |          |
| WAND   | DSALMOUT |

# TURN OFF LAMP

```
Page 150
PROGRAM NAME: ITURNON.
#
FUNCTIONAL DESCRIPTION: THIS PROGRAM IS CALLED BY IMUMON WHEN A CHANGE OF BIT 14 (
REQUEST) IS DETECTED. UPON ENTRY, ITURNON CHECKS IF A TURN-ON DELAY SEQUENCE HAS B
IF NOT, IT CHECKS WHETHER THE TURN-ON REQUEST CHANGE IS TO ON OR OFF. IF ON, IT SE
THAT TNONTEST WILL INITIATE THE ISS INITIALIZATION SEQUENCE. IF OFF, THE TURN-ON D
15, IS CHECKED AND IF IT IS ON, ITURNON EXITS. IF THE DELAY SIGNAL IS OFF, PROGRAM
OF IMODES30 IS SET TO 1 AND THE PROGRAM EXITS.
#
THE SETTING OF BIT 2 OF IMODES30 (ISS DELAY SEQUENCE FAIL) INHIBITS THIS ROUTINE AND
PROCESSING ANY CHANGES. THIS BIT WILL BE RESET BY THE ENDTNON ROUTINE WHEN THE CUR
ENDS.
#
CALLING SEQUENCE: FROM IMUMON WHEN ISS TURN-ON REQUEST CHANGES STATE.
#
JOBS OR TASKS INITIATED: NONE.
#
SUBROUTINES CALLED: ALARM, IF THE ISS TURN-ON REQUEST IS NOT PRESENT FOR 90 SECONDS
#
ERASABLE INITIALIZATION: FRESH START AND RESTART SET BIT 15 OF CHANNEL 12 AND BITS
AND BIT 14 OF IMODES30 TO 1.
#
ALARMS: PROGRAM ALARM 00207 IS ISSUED IF THE ISS TURN-ON REQUEST SIGNAL IS NOT PRESE
#
EXIT: NXTIFAIL.
#
OUTPUT: BIT 7 OF IMODES30 TO START ISS INITIALIZATION, OR BIT 2 OF IMODES30 AND PR
A FAILED TURN-ON SEQUENCE.
```

|         |        |          |                                           |
|---------|--------|----------|-------------------------------------------|
| ITURNON | CAF    | BIT2     | # IF DELAY REQUEST HAS GONE OFF           |
|         | MASK   | IMODES30 | # PREMATURELY, DO NOT PROCESS ANY CHANGES |
|         | CCS    | A        | # UNTIL THE CURRENT 90 SEC WAIT EXPIRES.  |
|         | TCF    | NXTIFAIL |                                           |
|         | CAF    | BIT14    | # SEE IF JUST ON OR OFF.                  |
|         | MASK   | IMODES30 |                                           |
|         | EXTEND |          |                                           |
|         | BZF    | ITURNON2 | # IF JUST ON.                             |
|         | CAF    | BIT15    |                                           |
|         | EXTEND |          | # SEE IF DELAY PRESENT DISCRETE HAS BEEN  |
|         | RAND   | CHAN12   | # SENT. IF SO, ACTION COMPLETE            |
|         | EXTEND |          |                                           |

July 12, 2016

1819

BZF +2  
TCF NXTIFAIL

CAF BIT2 # IF NOT, SET BIT TO INDICATE REQUEST NOT  
ADS IMODES30 # PRESENT FOR FULL DURATION.  
TC ALARM  
OCT 207  
TCF NXTIFAIL

# Page 151

ITURNON2 CS IMODES30 # SET BIT7 TO INDICATE WAIT OF 1 SAMPLE  
MASK BIT7  
ADS IMODES30  
TCF NXTIFAIL

# Page 152

# PROGRAM NAME: IMUCAGE.

#

# FUNCTIONAL DESCRIPTION: THIS PROGRAM PROCESSES CHANGES OF THE IMUCAGE INBIT, CHANNEL 30 BITS  
# CHANGES TO 0 (CAGE BUTTON PRESSED), THE ISS IS CAGED (ICDU ZERO + COARSE ALIGN + NO ATT LAMP)  
# ASTRONAUT SELECTS ANOTHER PROGRAM TO ALIGN THE ISS. ANY PULSE TRAINS TO THE ICDU'S AND GYRO'S  
# THE ASSOCIATE OUTCOUNTERS ARE ZEROED AND THE GYRO'S ARE DE-SELECTED. NO ACTION OCCURS WHEN T  
# RELEASED (INBIT CHANGES TO 1).

#

# CALLING SEQUENCE: BY IMUMON WHEN IMU CAGE BIT CHANGES.

#

# JOBS OR TASKS INITIATED: NONE.

#

# SUBROUTINES CALLED: CAGESUB.

#

# ERASABLE INITIALZATION: FRESH START AND RESTART SET BIT 11 OF IMODES30 TO 1.

#

# ALARMS: NONE.

#

# EXIT: NXTIFAIL.

#

# OUTPUT: ISS CAGED, COUNTERS ZEROED, PULSE TRAINS TERMINATED AND NO ATT LAMP LIT.

IMUCAGE CCS A # NO ACTION IF GOING OFF.  
TCF ISSZERO  
CS OCT77000 # TERMINATE ICDU, OPTICS, GYRO PULSE TRAINS  
EXTEND  
WAND CHAN14  
  
CS OCT272 # KNOCK DOWN TVC ENABLE, IMU ERROR COUNTER  
EXTEND # ENABLE, ZERO ICDU, COARSE ALIGN

```

WAND CHAN12 # ENABLE, OPTICS ERR CNTR ENABLE

CS BIT13 # TURN OFF ENGINE
EXTEND
WAND DSALMOUT

TC CAGESUB1

TC IBNKCALL # KNOCK DOWN TRACK, REFSMMAT, DRIFT FLAGS
CADR RNDREFDR

CS ZERO # ZERO COMMAND OUT-COUNTERS
TS CDUXCMD
TS CDUYCMD
TS CDUZCMD
TS GYROCMD

CS OCT740 # HAVING WAITED AT LEAST 27 MCT FROM
EXTEND # GYRO PULSE TRAIN TERMINATION, WE CAN
WAND CHAN14 # DE-SELECT THE GYROS.

Page 153
TCF NXTIFAIL

```

# Page 154

# PROGRAM NAME: IMUOP.

#

# FUNCTIONAL DESCRIPTION: THIS PROGRAM PROCESSES CHANGES IN THE ISS OPERATE DISCRETIONARY  
 # IF THE INBIT CHANGES TO 0, INDICATING ISS ON, IMUOP GENERALLY SETS BIT 7 OF IMODES30  
 # INITIALIZATION VIA TNONTEST. AN EXCEPTION IS DURING A FAILED ISS DELAY DURING WHICH  
 # TO 1 AND NO FURTHER INITIALIZATION IS REQUIRED. WHEN THE INBIT CHANGES TO 1, INDICATING  
 # TESTED TO SEE IF ANY PROGRAM WAS USING THE ISS. IF SO, PROGRAM ALARM 00214 IS ISSUED.

#

# CALLING SEQUENCE: BY IMUMON WHEN BIT 9 OF CHANNEL 30 CHANGES.

#

# JOBS OR TASKS INITIATED: NONE.

#

# SUBROUTINES CALLED: ALARM, IF ISS IS TURNED OFF WHILE IN USE.

#

# ERASABLE INITIALIZATION: ON FRESH START AND RESTART, BIT 9 OF IMODES30 IS SET TO 1  
 # LAMP IS ON, IN WHICH CASE IT IS SET TO 0. THIS PREVENTS ICPU ZERO BY TNONTEST WITH

#

# ALARMS: PROGRAM ALARM 00214 IF THE ISS IS TURNED OFF WHILE IN USE.

#

# EXIT: NXTIFAIL.

#

# OUTPUT: ISS INITIALIZATION REQUEST (IMODES30 BIT 7) OR PROGRAM ALARM 00214.

July 12, 2016

1821

|        |        |          |                                           |
|--------|--------|----------|-------------------------------------------|
| IMUOP  | EXTEND |          | # IF OPERATE JUST ON, WAIT 1 SAMPLE.      |
|        | BZF    | IMUOP2   |                                           |
|        | CS     | IMODES33 | # DISABLE DAP                             |
|        | MASK   | BIT6     |                                           |
|        | ADS    | IMODES33 |                                           |
|        | TC     | IBNKCALL | # KNOCK DOWN TRACK, REFSMMAT, DRIFT FLAGS |
|        | CADR   | RNDREFDR |                                           |
|        | CS     | BITS7&8  | # KNOCK DOWN RENDEZVOUS, IMUUSE FLAGS     |
|        | MASK   | STATE    |                                           |
|        | XCH    | STATE    | # IF GOING OFF, ALARM IF PROG USING IMU.  |
|        | COM    |          |                                           |
|        | MASK   | IMUSEFLG |                                           |
|        | CCS    | A        |                                           |
|        | TCF    | NXTIFAIL |                                           |
|        | TC     | ALARM    |                                           |
|        | OCT    | 214      |                                           |
|        | TCF    | NXTIFAIL |                                           |
| IMUOP2 | CAF    | BIT2     | # SEE IF FAILED ISS TURN-ON SEQ IN PROG.  |
|        | MASK   | IMODES30 |                                           |
|        | CCS    | A        |                                           |
|        | TCF    | NXTIFAIL | # IF SO, DON'T PROCESS UNTIL PRESENT 90   |
|        | TCF    | ITURNON2 | # SECONDS EXPIRES                         |

# Page 155

# PROGRAM NAME: PIPFAIL

#

# FUNCTIONAL DESCRIPTION: THIS PROGRAM PROCESSES CHANGES OF BIT 13 OF CHANNEL 33, PIPA FAIL.

# IMODES30 TO AGREE. IT CALLS SETISSW IN CASE A PIPA FAIL NECESSITATES AN ISS WARNING. IF NOT

# BIT 1 = 1, AND A PIPA FAIL IS PRESENT AND THE ISS NOT BEING INITIALIZED, PROGRAM ALARM 0212 I

#

# CALLING SEQUENCE: BY C33TEST ON CHANGES OF CHANNEL 33 BIT 13.

#

# JOBS OR TASKS INITIATED: NONE.

#

# SUBROUTINES CALLED: 1) SETISSW, AND 2) ALARM (SEE FUNCTIONAL DESCRIPTION).

#

# ERASABLE INITIALIZATION: SEE IMUMON FOR INITIALIZATION OF IMODES30. THE RELEVANT BITS ARE 5

#

# ALARMS: PROGRAM ALARM 00212 IF PIPA FAIL IS PRESENT BUT NEITHER ISS WARNING IS TO BE ISSUED

# BEING INITIALIZED.

```

#
EXIT: NXTFL33.
#
OUTPUT: PROGRAM ALARM 00212 AND ISS WARNING MAINTENANCE.

PIPFail CCS A # SET BIT10 IN IMODES30 SO ALL ISS W
CAF BIT10 # INFO IS IN ONE REGISTER.
XCH IMODES30
MASK -BIT10
ADS IMODES30

TC SETISSW

CS IMODES30 # IF PIP FAIL DOESN'T LIGHT ISS WARN
MASK BIT1 # A PROGRAM ALARM IF IMU OPERATING BU
CCS A # CAGED OR BEING TURNED ON.
TCF NXTFL33

CA IMODES30
MASK OCT1720
CCS A
TCF NXTFL33 # ABOVE CONDITION NOT MET.

TC ALARM
OCT 212
TCF NXTFL33

Page 156
PROGRAM NAMES: DNTMFAST, UPTMFAST
#
FUNCTIONAL DESCRIPTION: THESE PROGRAMS PROCESS CHANGES OF BITS 12 AND 11 OF CHANNEL
0, A PROGRAM ALARM IS ISSUED. THE ALARMS ARE:
#
BIT ALARM CAUSE
--- -
12 01105 DOWNLINK TOO FAST
11 01106 UPLINK TOO FAST
#
CALLING SEQUENCE: BY C33TEST ON A BIT CHANGE.
#
SUBROUTINES CALLED: ALARM, IF A BIT CHANGES TO A 0.
#
ERASABLE INITIALIZATION: FRESH START OR RESTART, BITS 12 AND 11 OF IMODES33 ARE SET
#
ALARMS: SEE FUNCTIONAL DESCRIPTION.
#

```

July 12, 2016

1823

# EXIT: NXTFL33.

#

# OUTPUT: PROGRAM ALARM ON A BIT CHANGE TO 0.

|          |     |         |                                 |
|----------|-----|---------|---------------------------------|
| DNTMFAST | CCS | A       | # DO PROG ALARM IF TM TOO FAST. |
|          | TCF | NXTFL33 |                                 |

|  |     |         |
|--|-----|---------|
|  | TC  | ALARM   |
|  | OCT | 1105    |
|  | TCF | NXTFL33 |

|          |     |         |                                          |
|----------|-----|---------|------------------------------------------|
| UPTMFAST | CCS | A       | # SAME AS DNLINK TOO FAST WITH DIFFERENT |
|          | TCF | NXTFL33 | # ALARM CODE.                            |

|  |     |         |
|--|-----|---------|
|  | TC  | ALARM   |
|  | OCT | 1106    |
|  | TCF | NXTFL33 |

# Page 157

# PROGRAM NAME: SETISSW

#

# FUNCTIONAL DESCRIPTION: THIS PROGRAM TURNS THE ISS WARNING LAMP ON AND OFF (CHANNEL 11 BIT 1  
# 0 FOR OFF) DEPENDING ON THE STATUS OF IMODES30 BITS 13 (IMU FAIL) AND 4 (INHIBIT IMU FAIL), 1  
# 3 (INHIBIT ICDU FAIL), AND 10 (PIPA FAIL) AND 1 (INHIBIT PIPA FAIL). THE LAMP IS LEFT ON IF  
# PROGRESS.

#

# CALLING SEQUENCE: CALLED BY IMUMON ON CHANGES TO IMU FAIL AND ICDU FAIL. CALLED BY IFAILOK  
# REMOVAL OF THE FAIL INHIBITS. CALLED BY PIPFAIL WHEN THE PIPA FAIL DISCRETE CHANGES. IT IS  
# SINCE THE PIPA FAIL PROGRAM ALARM MAY NECESSITATE AN ISS WARNING, AND LIKEWISE BY PIPFREE WHE  
# AND IT IS CALLED BY IMUZERO3 AND ISSUP AFTER THE FAIL INHIBITS HAVE BEEN REMOVED.

#

# JOBS OR TASKS INITIATED: NONE.

#

# SUBROUTINES CALLED: NONE.

#

# ERASABLE INITIALIZATION:

#

# 1) IMODES30 -- SEE IMUMON.

# 2) IMODES33 BIT 1 = 0 (LAMP TEST NOT IN PROGRESS).

#

# ALARMS: ISS WARNING.

#

# EXIT: VIA Q.

#

# OUTPUT: ISS WARNING LAMP SET PROPERLY.

|            |        |             |                                       |
|------------|--------|-------------|---------------------------------------|
| SETISSW    | CAF    | OCT15       | # SET ISS WARNING USING THE FAIL BITS |
|            | MASK   | IMODES30    | # BITS 13, 12, AND 10 OF IMODES30 AND |
|            | EXTEND |             | # FAILURE INHIBIT BITS IN POSITIONS   |
|            | MP     | BIT10       | # 4, 3, AND 1.                        |
|            | CA     | IMODES30    |                                       |
|            | EXTEND |             |                                       |
|            | ROR    | LCHAN       | # 0 INDICATES FAILURE                 |
|            | COM    |             |                                       |
|            | MASK   | OCT15000    |                                       |
|            | CCS    | A           |                                       |
|            | TCF    | ISSWON      | # FAILURE.                            |
| ISSWOFF    | CAF    | BIT1        | # DON'T TURN OFF ISS WARNING IF LAMP  |
|            | MASK   | IMODES33    | # IN PROGRESS.                        |
|            | CCS    | A           |                                       |
|            | TC     | Q           |                                       |
|            | CS     | BIT1        |                                       |
|            | EXTEND |             |                                       |
|            | WAND   | DSALMOUT    | # TURN OFF ISS WARNING.               |
|            | TC     | Q           |                                       |
| ISSWON     | EXTEND |             |                                       |
| # Page 158 | QXCH   | ITEMP6      |                                       |
|            | TC     | VARALARM    | # TELL EVERYONE WHAT CAUSED THE ISS W |
|            | CAF    | BIT1        |                                       |
|            | EXTEND |             |                                       |
|            | WOR    | DSALMOUT    | # TURN ON ISS WARNING                 |
|            | TC     | ITEMP6      |                                       |
| CAGESUB    | CS     | BIT15+6     | # SET OUTBITS + INTERNAL FLAGS FOR    |
|            | EXTEND |             | # SYSTEM TURN-ON OR CAGE. DISABLE T   |
|            | WAND   | CHAN12      | # ERROR COUNTER AND REMOVE THE IMU D  |
|            | CAF    | BITS4&5     | # SEND ZERO AND COARSE.               |
|            | EXTEND |             |                                       |
|            | WOR    | CHAN12      |                                       |
| CAGESUB1   | CS     | DSPTAB +11D | # TURN ON NO ATT LAMP                 |
|            | MASK   | OC40010     |                                       |
|            | ADS    | DSPTAB +11D |                                       |
| CAGESUB2   | CS     | IMODES30    | # SET FLAGS TO INDICATE CAGING OR TU  |
|            | MASK   | OCT75       | # AND INHIBIT ALL ISS WARNING INFO    |
|            | ADS    | IMODES30    |                                       |



July 12, 2016

1825

```

 CS IMODES33 # DISABLE DAP AUTO AND HOLD MODES
 MASK BIT6
 ADS IMODES33

 TC Q

IMUFAIL EQUALS SETISSW
ICDUFail EQUALS SETISSW

Page 159
JUMP TABLES AND CONSTANTS.

IFAILJMP TCF ITURNON # CHANNEL 30 DISPATCH.
 TCF IMUFAIL
 TCF ICDUFail
 TCF IMUCAGE

30RDMSK OCT 76400 # (BIT 10 NOT SAMPLED HERE).
 TCF IMUOP

C33JMP TCF PIPFAIL # CHANNEL 33 DISPATCH.
 TCF DNTMFAST
 TCF UPTMFAST

SUBROUTINE TO SKIP IF LAMP TEST NOT IN PROGRESS.

LAMPTEST CS IMODES33 # BIT 1 OF IMODES33 = 1 IF LAMP TEST IN
 MASK BIT1 # PROGRESS.
 TCF ZOPFIN3

33RDMSK EQUALS PRI016
OC40010 OCT 40010
OCT54 OCT 54
OCT75 OCT 75
OCT272 OCT 00272
BITS7&8 OCT 300
OCT1720 OCT 1720
OCT740 OCT 00740
OCT15000 EQUALS PRI015
OCT77000 OCT 77000
-BIT10 OCT -1000

90SECS DEC 9000
120MS = OCT14 # (DEC12)
GLOCKOK EQUALS RESUME

Page 160
```

## # OPTICS MONITORING AND ZERO ROUTINES

|            |        |          |                                        |
|------------|--------|----------|----------------------------------------|
| OPTMON     | CA     | OPTMODES | # MONITOR OPTICS INBITS IN CHAN 30 AND |
|            | EXTEND |          |                                        |
|            | RXOR   | CHAN30   | # LOOK FOR OCDU FAIL BIT CHANGE        |
|            | MASK   | BIT7     |                                        |
|            | TS     | RUPTREG1 | # STORE CHANGE BIT                     |
|            | CCS    | A        |                                        |
|            | TC     | OCDUFTST | # PROCESS OCDUFAIL BIT CHANGE          |
| 330OPTMON  | CCS    | OPTIND   | # BYPASS IF TVC TAKEOVER               |
|            | TCF    | +4       |                                        |
|            | TCF    | +3       |                                        |
|            | TCF    | +2       |                                        |
|            | TCF    | RESUME   |                                        |
|            | CA     | OPTMODES | # LOOK FOR OPTICS MODE SWITCH CHANGE   |
|            | EXTEND |          |                                        |
|            | RXOR   | CHAN33   |                                        |
|            | MASK   | OCTHIRTY |                                        |
|            | ADS    | RUPTREG1 | # STORE INBIT CHANGES                  |
|            | LXCH   | OPTMODES |                                        |
|            | EXTEND |          |                                        |
|            | RXOR   | LCHAN    |                                        |
|            | TS     | OPTMODES | # UPDATE OPTMODES TO SHOW BIT CHANGES  |
|            | COM    |          | # SAMPLE CURRENT SWITCH SETTING        |
|            | MASK   | OCTHIRTY |                                        |
|            | EXTEND |          |                                        |
|            | BZF    | SETSAMP  | # MANUAL-SET ZERO IN SWSAMPLE          |
|            | MASK   | BIT5     | # SEE IF CSC                           |
|            | CCS    | A        |                                        |
|            | TC     | +2       | # CSC-SET SWSAMPLE POS                 |
|            | CAF    | NEGONE   | # ZOPTICS-SET SWSAMPLE (-1)            |
| SETSAMP    | TS     | SWSAMPLE | # CURRENT OPTICS SWITCH SETTING        |
| PROCESSW   | CCS    | DESOPMOD | # BRANCH ON PREVIOUS SETTING           |
|            | TC     | CSCDES   | # CSC                                  |
|            | TC     | MANUDES  | # MANUAL                               |
|            | TC     | ZOPTDES  | # ZERO OPTICS                          |
| # Page 161 |        |          |                                        |
| ZOPTDES    | CCS    | SWSAMPLE | # IS SWITCH STILL AT ZOPTICS           |
|            | TC     | ZTOCSC   | # NOW AT CSC                           |
|            | TC     | ZTOMAN   | # MANUAL                               |
|            | TC     | ZOPFIN1  | # ZOPTICS-SEE IF ZOPT PROCESSING       |
|            | TC     | SETDESMD | # ZOPT NOT PROCESSING-NO ACTION        |

July 12, 2016

1827

|          |        |             |                                                |
|----------|--------|-------------|------------------------------------------------|
|          | CCS    | ZOPTCNT     | # ZOPT PROCESSING-CHECK COUNTER                |
|          | TC     | SETCNT      | # 32 SAMPLE NOT FINISHED-SET COUNTER           |
|          | TC     | SETZOEND    | # 32 SAMPLE WAIT COMPLETED-SET UP ZOP END      |
| ZTOMAN   | TC     | ZOPFIN1     | # ZOP TO MANUAL-IS ZOPT DONE // Shou           |
|          | TC     | SETDESMD    | # YES-NORMAL EXIT                              |
| ZOPALARM | TC     | ALARM       | # ALARM-SWITCHED ALTERED WHILE ZOPTICS         |
|          | OCT    | 00116       |                                                |
|          | CAF    | OCT13       | # PROCESSING-SET RETURN OPTION                 |
|          | TS     | WTOPTION    |                                                |
|          | TC     | CANZOPT     | # CANCEL ZOPT                                  |
|          | TC     | SETDESMD    |                                                |
| ZTOCSC   | TC     | ZOPFIN1     | # SEE IF ZOPT PROCESSING // Shou               |
|          | TC     | MANTOCSC +3 | # NO-CHECK RETURN TO COARS OPT                 |
|          | TC     | ALARM       | # ZOPT PROCESSING-ALARM                        |
|          | OCT    | 00116       |                                                |
|          | TC     | CANZOPT     | # CANCEL ZOPT                                  |
|          | TC     | MANTOCSC    | # ZERO CNT-LOOK FOR COARS OPT RETURN           |
| COARSLOK | CAF    | BIT9        | # IF COARS OPT SINCE FSTART GO TO L+2          |
|          | TCF    | ZOPFIN2     | # IF NOT GO TO L+1                             |
| ZOPFIN1  | CAF    | BIT1        | # SEE IF END ZOPT TASK WORKING // Label should |
|          | MASK   | OPTMODES    |                                                |
|          | CCS    | A           |                                                |
|          | TC     | RESUME      | # ZOPT TASK WORKING-WAIT ONE SAMPLE PERIOD     |
| ZOPFIN2  | CAF    | BIT3        | # TEST IF ZOPTICS PROCESSING                   |
| ZOPFIN3  | MASK   | OPTMODES    | # RETURNS TO L+1 PROCESSING AND                |
|          | CCS    | A           |                                                |
|          | INCR   | Q           | # L+2 IF NOT                                   |
|          | TC     | Q           |                                                |
| CANZOPT  | CS     | SIX         | # CANCEL ZERO OPTICS                           |
|          | MASK   | OPTMODES    | # ZERO ZOPT PROCESSING BIT-ENABLE OCDUFAIL     |
|          | TS     | OPTMODES    |                                                |
|          | CS     | BIT1        | # MAKE SURE ZERO OCDU IS OFF                   |
|          | EXTEND |             |                                                |
|          | WAND   | CHAN12      |                                                |
|          | TC     | Q           |                                                |

1828

July 12, 2016

|            |        |          |                                       |
|------------|--------|----------|---------------------------------------|
| MANUDES    | CCS    | SWSAMPLE | # SEE IF SWITCH STILL IN MANUAL MODE  |
|            | TC     | MANTOCSC | # NOW AT CSC                          |
|            | TC     | MANTOMAN | # STILL MANUAL                        |
|            | CCS    | WTOPTION | # ZOPTICS-LOOK AT ZOPTICS RETURN OPT  |
|            | TC     | +2       | # 5 SEC RETURN GOOD-CONTINUE ZOPTICS  |
|            | TC     | OPTZERO  | # ZOPTICS MUST START ANEW             |
|            | TC     | INITZOPT | # SHOW ZERO OPTICS PROCESSING         |
|            | TC     | SETDESMD | # NORMAL EXIT                         |
|            |        |          |                                       |
| MANTOMAN   | CCS    | WTOPTION | # DECREMENT RETURN OPTION TIME        |
|            | TS     | WTOPTION |                                       |
|            | TC     | SETDESMD |                                       |
| MANTOCSC   | CAF    | ZERO     | # CANCEL ZOPT RETURN OPTION IF SET    |
|            | TS     | WTOPTION |                                       |
|            | TS     | ZOPTCNT  |                                       |
|            |        |          |                                       |
|            | TC     | COARSLOK | # CHECK FOR COARS OPT RETURN          |
|            | TC     | SETDESMD | # NO COARS TASK-NO ACTION             |
|            |        |          |                                       |
|            | CAF    | ONE      | # SET COARS OPT WORKING               |
|            | TS     | OPTIND   |                                       |
|            | CAF    | BIT2     | # ENABLE OPTICS CDU ERROR CNTS        |
|            | EXTEND |          |                                       |
|            | WOR    | CHAN12   |                                       |
|            | TC     | SETDESMD |                                       |
| CSCDES     | CCS    | SWSAMPLE | # SEE IF SWITCH STILL AT CSC          |
|            | TC     | SETDESMD | # STILL AT CSC                        |
|            | TC     | CSCTOMAN | # MANUAL                              |
| CSCTOZOP   | CAF    | OCT40    | # ZOPTICS-INITIALIZE FOR ZOPT         |
|            | TS     | ZOPTCNT  |                                       |
|            | TC     | INITZOPT |                                       |
| CSCTOMAN   | CCS    | OPTIND   | # SEE IF COARS WORKING                |
|            | TC     | CANCOARS | # COARS WORKING-SWITCH NOT CSC-KILL C |
|            | TC     | CANCOARS |                                       |
|            | TC     | +1       | # NO COARS-NORMAL EXIT                |
|            | TC     | SETDESMD |                                       |
| # Page 163 |        |          |                                       |
| CANCOARS   | CA     | NEGONE   |                                       |
|            | TS     | OPTIND   | # SET OPTIND (-1) TO SHOW NOT WORKING |
|            | CS     | BIT2     | # DISABLE OCDU ERR CNTS               |
|            | EXTEND |          |                                       |

July 12, 2016

1829

|            |        |          |                                         |
|------------|--------|----------|-----------------------------------------|
|            | WAND   | CHAN12   |                                         |
|            | CS     | OPTMODES | # SET RETURN-TO-COARS BIT               |
|            | MASK   | BIT9     |                                         |
|            | ADS    | OPTMODES |                                         |
|            | TC     | SETDESMD |                                         |
| OPTZERO    | TC     | INITZOPT | # INITIALIZE ZERO OPTICS                |
|            | CA     | OCT40    | # SET UP 32 SAMPLE WAIT                 |
| SETCNT     | TS     | ZOPTCNT  |                                         |
| SETDESMD   | CA     | SWSAMPLE | # SET CURRENT SWITCH INDICATION-RESUME  |
|            | TS     | DESOPMOD |                                         |
|            | TC     | RESUME   |                                         |
| SETZOEND   | CAF    | BIT1     | # SEND ZERO OPTICS CDU                  |
|            | EXTEND |          |                                         |
|            | WOR    | CHAN12   |                                         |
|            | CA     | 200MS    | # HOLD ZERO CDU FOR 200 MS              |
|            | TC     | WAITLIST |                                         |
|            | EBANK= | OPTMODES |                                         |
|            | 2CADR  | ENDZOPT  |                                         |
|            | CS     | OPTMODES | # SHOW ZOPTICS TASK WORKING             |
|            | MASK   | BIT1     |                                         |
|            | ADS    | OPTMODES |                                         |
|            | TC     | SETDESMD |                                         |
| ENDZOPT    | TC     | ZEROPCDU | # ZERO OCDU COUNTERS                    |
|            | CS     | BIT1     | # TURN OFF ZERO OCDU                    |
|            | EXTEND |          |                                         |
|            | WAND   | CHAN12   |                                         |
|            | CAF    | 200MS    | # DELAY 200MS FOR CDUS TO RESYNCHRONIZE |
|            | TC     | VARDELAY |                                         |
|            | CS     | OPTMODES | # SHOW ZOPTICS SINCE LAST FRESH START   |
|            | MASK   | BIT10    | # OR RESTART                            |
|            | ADS    | OPTMODES |                                         |
|            | CS     | SEVEN    | # ENABLE OCDUFAIL-SHOW OPTICS COMPLETE  |
|            | MASK   | OPTMODES |                                         |
|            | TS     | OPTMODES |                                         |
|            | TC     | OCDUFTST | # CHECK OCDU FAIL BIT AFTER ENABLE.     |
| # Page 164 | TC     | TASKOVER |                                         |

1830

July 12, 2016

|            |        |             |                                       |
|------------|--------|-------------|---------------------------------------|
| ZEROPCDU   | CAF    | ZERO        |                                       |
|            | TS     | CDUS        | # ZERO IN CDUS, -20 IN CDUT           |
|            | TS     | ZONE        | # INITIALZE SHAFT MONITOR ZONE.       |
|            | CS     | 20DEGS      |                                       |
|            | TS     | CDUT        |                                       |
|            | TC     | Q           |                                       |
| INITZOPT   | CAF    | ZERO        | # INITIALIZE ZOPTICS-INHIBIT OCDUFAIL |
|            | TS     | WTOPTION    | # AND SHOW OPTICS PROCESSING          |
|            | CS     | OPTMODES    | # SET ZERO OPTICS PROCESSING          |
|            | MASK   | SIX         | # OPTICS CDU FAIL INHIBITED           |
|            | ADS    | OPTMODES    |                                       |
|            | TC     | Q           |                                       |
| # Page 165 |        |             |                                       |
| OCDUFTST   | CAF    | BIT7        | # SEE IF OCDUFAIL ON OR OFF           |
|            | EXTEND |             |                                       |
|            | RAND   | CHAN30      |                                       |
|            | CCS    | A           |                                       |
|            | TCF    | OPFAILOF    | # OCDUFAIL LIGHT OFF                  |
|            | CAF    | BIT2        | # OCDUFAIL LIGHT ON UNLESS INHIBITED  |
|            | MASK   | OPTMODES    |                                       |
|            | CCS    | A           |                                       |
|            | TC     | Q           | # OCDUFAIL INHIBITED                  |
| OPFAILON   | CAF    | BIT8        | # ON BIT                              |
|            | AD     | DSPTAB +11D |                                       |
|            | MASK   | BIT8        |                                       |
| SETOFF     | EXTEND |             |                                       |
|            | BZF    | TCQ         | # NO CHANGE                           |
|            | TS     | L           |                                       |
|            | CA     | DSPTAB +11D |                                       |
|            | EXTEND |             |                                       |
|            | RXOR   | LCHAN       |                                       |
|            | MASK   | POS MAX     |                                       |
|            | AD     | BIT15       | # SHOW ACTION WANTED                  |
|            | TS     | DSPTAB +11D |                                       |
|            | TC     | Q           |                                       |
| OPFAILOF   | CAF    | BIT1        | # DON'T TURN OFF IF LAMP TEST         |
|            | MASK   | IMODES33    |                                       |
|            | CCS    | A           |                                       |
|            | TC     | Q           | # LAMP TEST IN PROGRESS               |

1831

```

CAF BIT8 # TURN OFF OCDUFAIL LIGHT
MASK DSPTAB +11D
TCF SETOFF

OCT13 = ELEVEN
OCTHIRTY EQUALS BITS4&5
20DEGS DEC 7199
OCT40 EQUALS BIT6
200MS EQUALS OCT24

Page 166
OPTICS CDU DRIVING PROGRAM

BANK 10
SETLOC OPTDRV
BANK
COUNT* $$/SXT

SHAFT STOP MONITOR-ZONE UPDATE

OPTDRIVE CA CDUS # GRAB OPTIC SHAFT CDU
 TS L
 CCS A # GET ABS(CDUS)
 AD 13,14,15
 TCF +2 # ABS(CDUS) - 45 DEG
 TCF -2
 EXTEND
 BZMF OZONE # LESS THAN 45 DEG-SET ZONE 0
 CA ZONE # IF ZONE ZERO, CHANGE TO + OR - OTHERWISE
 EXTEND
 BZF +2 # DON'T MESS WITH ZONE
 TCF CONTDRVE # JUST CONTINUE
 XCH L # GREATER THAN 45 DEG-SET ZONE TO SIGN CDU
 TCF OZONE +1
OZONE CAF ZERO # ABS(CDUS) LESS THAN 90 DEG-ZONE ZERO
 TS ZONE
 COUNT* $$/T4RUPT
CONTDRVE CCS OPTIND
 TC +4 # WORK COARS OPTICS
 TC +3 # WORK COARS OPTICS
 TC RESUME # NO OPT
 TC RESUME # NO OPT

 CA SWSAMPLE # SEE IF SWITCH AT CMC
 EXTEND

```

|                        |        |          |                                       |
|------------------------|--------|----------|---------------------------------------|
|                        | BZMF   | RESUME   | # ZERO (-1)      MANUAL (+0)          |
|                        | CAF    | BIT10    | # SEE IF OCDUS ZEROED SINCE LAST FSTA |
|                        | MASK   | OPTMODES |                                       |
|                        | CCS    | A        |                                       |
|                        | TC     | +3       |                                       |
|                        | TC     | ALARM    | # OPTICS NOT ZEROED                   |
|                        | OCT    | 00120    |                                       |
|                        | CA     | BIT2     | # SEE IF ERR CNTS ENABLED             |
|                        | EXTEND |          |                                       |
|                        | RAND   | CHAN12   |                                       |
|                        | EXTEND |          |                                       |
|                        | BZF    | SETBIT   | # CNTS NOT ENABLED-DO IT AND RESUME   |
|                        | CAF    | ONE      | # INITIALIZE OPTIND                   |
| # Page 167             | TS     | OPTIND   |                                       |
| OPT2                   | EXTEND |          |                                       |
|                        | BZF    | TRUNCMD  | # CHECK TRUNION COMMAND               |
| GETOPCMD               | INDEX  | OPTIND   |                                       |
|                        | CA     | DESOPTT  | # PICK UP DESIRED OPT ANGLE           |
|                        | EXTEND |          |                                       |
|                        | INDEX  | OPTIND   |                                       |
|                        | MSU    | CDUT     | # GET DIFFERENCE                      |
|                        | EXTEND |          |                                       |
|                        | MP     | BIT13    |                                       |
|                        | XCH    | L        |                                       |
|                        | DOUBLE |          |                                       |
|                        | TS     | ITEMP1   |                                       |
|                        | TCF    | +2       | # NO OVFL                             |
|                        | ADS    | L        | # WITH OVFL                           |
| STORCMD                | INDEX  | OPTIND   |                                       |
|                        | LXCH   | COMMANDO | # STORE COMMAND                       |
|                        | CCS    | OPTIND   |                                       |
|                        | TCF    | OPT2     | # GET NEXT COMMAND                    |
|                        | TS     | ITEMP1   | # INITIALIZE SEND INDICATOR TO ZERO   |
|                        | COUNT* | \$\$/SXT |                                       |
| # SHAFT STOP AVOIDANCE |        |          |                                       |
|                        | CCS    | CDUS     | # IF CDUS GREATER THAN + OR - 90 DEG  |
|                        | AD     | NEG1/2   | # FOR POSSIBLE STOP PROBLEM           |



July 12, 2016

1833

```
TCF +2
TCF -2
EXTEND
BZMF CMDSETUP # CDU LESS THAN 90 DEG, NO PROBLEMS

CA ZONE
EXTEND
BZF CMDSETUP # ZONE=3, NORMAL COMMAND
MASK BIT15 # GRAB SIGN OF ZONE
TS L
CA COMMANDO +1
MASK BIT15 # GRAB SIGN OF SHAFT COMMAND
EXTEND
RXOR LCHAN
CCS A
TCF CMDSETUP # SIGN ZONE NOT EQUAL TO SIGN COMMAND
CCS DESOPTS # SEE IF DESOPTS BETWEEN -90 AND +90
AD NEG1/2
TCF +2 # ABS(DESOPTS) - 90 DEG
TCF -2
EXTEND

Page 168
BZMF +2 # DESOPTS IN FIRST OR FOURTH QUAD
TCF CMDSETUP
CS COMMANDO +1 # REVERSE REGULAR COMMAND
TS COMMANDO +1

COUNT* $$/T4RPT

CMDSETUP CAF ONE # SET OPTIND
TS OPTIND
INDEX A
CCS COMMANDO # GET SIGN OF COMMAND
TC POSOPCMD
TC NEXTOPT +1 # ZERO COMMAND-SKIP SEND INDICATOR
TC NEGOPCMD
TC NEXTOPT +1 # ZERO COMMAND

TRUNCMD CS CDUT # IF COMMAND GREATER THAN 45 DEG-COMMAND
AD DESOPTT # 45 DEG
TS Q
TC GETOPCMD # LESS THAN 45 DEG-NORMAL OPERATION

CCS A # GREATER THAN 45 DEG-USE OPSMAX WITH
CA POSMAX # CORRECT SIGN
TC +2
```

|            |        |             |                                       |
|------------|--------|-------------|---------------------------------------|
|            | CS     | POSMAX      |                                       |
|            | TS     | L           |                                       |
|            | TC     | STORCMD     |                                       |
| POSOPCMD   | AD     | MAXPLS1     |                                       |
|            | EXTEND |             |                                       |
|            | BZMF   | DELOPCMD    | # COMMAND LESS THAN MAX PULSE         |
|            | CS     | MAXPLS      | # GREATER THAN MAX PULSE-USE MAX PULS |
| NEXTOPT    | INCR   | ITEMP1      | # SET SEND INDICATOR                  |
|            | AD     | NEGO        | # MAKE SURE ZERO COMMAND IS -ZERO     |
|            | INDEX  | OPTIND      |                                       |
|            | TS     | CDUTCMD     | # STORE PULSE IN SEND REG             |
|            | CCS    | OPTIND      |                                       |
|            | TC     | CMDSETUP +1 | # GET NEXT OPT                        |
|            | CCS    | ITEMP1      | # ARE ANY PULSES TO GO                |
|            | TCF    | SENDOCMD    | # YES-SEND EM                         |
|            | TC     | RESUME      | # NO                                  |
| NEGOPCMD   | AD     | MAXPLS1     |                                       |
|            | EXTEND |             |                                       |
|            | BZMF   | DELOPCMD    | # LESS THAN MAX PULSE                 |
|            | CA     | MAXPLS      | # MAX PULSES                          |
|            | TCF    | NEXTOPT     |                                       |
| # Page 169 |        |             |                                       |
| DELOPCMD   | INDEX  | OPTIND      |                                       |
|            | XCH    | COMMANDO    | # SET UP SMALL COMMAND                |
|            | TCF    | NEXTOPT     |                                       |
| SENDOCMD   | CAF    | 11,12       | # SEND OCDU DRIVE COMMANDS            |
|            | EXTEND |             |                                       |
|            | WOR    | CHAN14      |                                       |
|            | TC     | RESUME      |                                       |
| SETBIT     | CAF    | BIT2        | # ENABLE OCDU ERR CNTS                |
|            | EXTEND |             |                                       |
|            | WOR    | CHAN12      |                                       |
|            | TC     | RESUME      | # START COARS NEXT TIME AROUND        |
| MAXPLS     | DEC    | -165        | # WAS -80                             |
| MAXPLS1    | DEC    | -164        | # WAS -79                             |
| 11,12      | EQUALS | PRI06       |                                       |

This code is written to file `src/T4RUPT-PROGRAM.s`.

## B.111 T6-RUPT PROGRAMS

1835

*<src/T6-RUPT-PROGRAMS.s 1835>≡*

```
Copyright: Public domain.
Filename: T6-RUPT_PROGRAMS.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1403-1405
Mod history: 2009-05-10 SN (Sergio Navarro). Started adapting
from the Luminary131/ file of the same
name, using Luminary099 page images.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 1403
PROGRAM NAMES: (1) T6JOBCHK MOD. NO. 5 OCTOBER 2, 1967
(2) DOT6RUPT
MODIFICATION BY: LOWELL G. HULL (A.C.ELECTRONICS)
#
THESE PROGRAMS ENABLE THE LM DAP TO CONTROL THE THRUST TIMES OF THE REACTION CONTROL SYSTEM J
SINCE THE LM DAP MAINTAINS EXCLUSIVE CONTROL OVER TIME6 AND ITS INTERRUPTS, THE FOLLOWING CON
ESTABLISHED AND MUST NOT BE TAMPERED WITH:
1. NO NUMBER IS EVER PLACED INTO TIME6 EXCEPT BY LM DAP.
2. NO PROGRAM OTHER THAN LM DAP ENABLES THE TIME6 COUNTER.
3. TO USE TIME6, THE FOLLOWING SEQUENCE IS ALWAYS EMPLOYED:
A. A POSITIVE (NON-ZERO) NUMBER IS STORED IN TIME6.
B. THE TIME6 CLOCK IS ENABLED.
C. TIME6 IS INTERROGATED AND IS:
I. NEVER FOUND NEGATIVE (NON-ZERO) OR +0.
```

```

II. SOMETIMES FOUND POSITIVE (BETWEEN 1 AND 240D) INDICAT
III. SOMETIMES FOUND POSMAX INDICATING THAT IT IS INACTIV
IV. SOMETIMES FOUND NEGATIVE ZERO INDICATING THAT:
A. A T6RUPT IS ABOUT TO OCCUR AT THE NEXT DINC,
B. A T6RUPT IS WAITING IN THE PRIORITY CHAIN, OR
C. A T6RUPT IS IN PROCESS NOW.
4. ALL PROGRAMS WHICH OPERATE IN EITHER INTERRUPT MODE OR WITH INTERRUPT
EVERY 5 MILLISECONDS TO PROCESS A POSSIBLE WAITING T6RUPT BEFORE IT C
(5. PROGRAM JTLST, IN Q,R-AXES, HANDLES THE INPUT LIST.)
#
T6JOBCHK CALLING SEQUENCE:
L TC T6JOBCHK
L+1 (RETURN)
#
DOT6RUPT CALLING SEQUENCE:
DXCH ARUPT # T6RUPT LEAD IN AT LOCATION 4004.
EXTEND
DCA T6ADR
DTCB
#
SUBROUTINES CALLED: DOT6RUPT CALLS T6JOBCHK.
#
NORMAL EXIT MODES: T6JOBCHK RETURNS TO L +1.
DOT6RUPT TRANSFERS CONTROL TO RESUME.
#
ALARM/ABORT MODES: NONE.
#
INPUT: TIME6 NXT6ADR OUTPUT: TIME6 NXT6A
T6NEXT T6NEXT +1 T6NEXT T6NE
T6FURTHA T6FURTHA +1 T6FURTHA T6FU
#
DEBRIS: T6JOBCHK CLOBBERS A. DOT6RUPT CLOBBERS NOTHING.
#
Page 1404
BLOCK 02
#
BANK 17
SETLOC DAPS2
BANK
EBANK= T6NEXT
COUNT* $$/DAPT6
#
T6JOBCHK CCS TIME6 # CHECK TIME6 FOR WAITING T6RUPT:
TC Q # NONE: CLOCK COUNTING DOWN.
TC CCSHOLE
TC T6JOBCHK +3

```

July 12, 2016

1837

# CONTROL PASSES TO T6JOB ONLY WHEN C(TIME6) = -0 (I.E., WHEN A T6RUPT MUST BE PROCESSED).

|       |        |        |                                           |
|-------|--------|--------|-------------------------------------------|
| T6JOB | CAF    | POSMAX | # DISABLE CLOCK: NEEDED SINCE RUPT OCCURS |
|       | EXTEND |        | # 1 DINC AFTER T6 = 77777. FOR 625 MUSECS |
|       | WAND   | CHAN13 | # MUST NOT HAVE T6 = +0 WITH ENABLE SET   |

|      |          |
|------|----------|
| CA   | POSMAX   |
| ZL   |          |
| DXCH | T6FURTHA |
| DXCH | T6NEXT   |
| LXCH | NXT6ADR  |
| TS   | TIME6    |

|          |        |           |
|----------|--------|-----------|
|          | AD     | PRI037    |
|          | TS     | A         |
|          | TCF    | ENABLET6  |
|          | CA     | POSMAX    |
|          | TS     | TIME6     |
| ENABLET6 | TCF    | GOCH56    |
|          | CA     | BIT15     |
|          | EXTEND |           |
|          | WOR    | CHAN13    |
|          | CA     | T6NEXT    |
|          | AD     | PRI037    |
|          | TS     | A         |
|          | TCF    | GOCH56    |
|          | CA     | POSMAX    |
|          | TS     | T6NEXT    |
| GOCH56   | INDEX  | L         |
|          | TCF    | WRITEP -1 |

|        |            |
|--------|------------|
| BLOCK  | 02         |
| SETLOC | FFTAG9     |
| BANK   |            |
| EBANK= | CDUXD      |
| COUNT* | \$\$/DAPT6 |

|        |        |       |
|--------|--------|-------|
|        | CA     | NEXTP |
| WRITEP | EXTEND |       |
|        | WRITE  | CHAN6 |

# Page 1405

|    |   |
|----|---|
| TC | Q |
|----|---|

|        |    |          |
|--------|----|----------|
|        | CA | NEXTU    |
| WRITEU | TS | L        |
|        | CS | 003140CT |

```

 EXTEND
 RAND CHAN5
 AD L
 EXTEND
 WRITE CHAN5
 TC Q

 CA NEXTV
WRITEV TS L
 CA 00314OCT
 TCF -9D
00314OCT OCT 00314

 BANK 17
 SETLOC DAPS2
 BANK

 EBANK= T6NEXT
 COUNT* $$/DAPT6

DOT6RUPT LXCH BANKRUPT # (INTERRUPT LEAD INS CONTINUED)
 EXTEND
 QXCH QRUPT

 TC T6JOBCHK # CALL T6JOBCHK.

 TCF RESUME # END TIME6 INTERRUPT PROCESSOR.

```

This code is written to file `src/T6-RUPT-PROGRAMS.s`.

## B.112 TAGS FOR RELATIVE SETLOC

```

1839 <src/TAGS-FOR-RELATIVE-SETLOC.s 1839>≡
Copyright: Public domain.
Filename: TAGS_FOR_RELATIVE_SETLOC.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Mod history: 2009-05-05 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
2009-05-20 RSB Corrected R32 -> R31.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 27
TAGS FOR RELATIVE SETLOC AND BLANK BANK CARDS

FIXED MEMORY 120000 - 167777
 COUNT BANKSUM

MODULE 1 CONTAINS BANKS 0 THROUGH 5

 BLOCK 02
FFTAG1 EQUALS
FFTAG2 EQUALS
FFTAG3 EQUALS
FFTAG4 EQUALS
FFTAG7 EQUALS

```

|         |        |    |
|---------|--------|----|
| FFTAG8  | EQUALS |    |
| FFTAG9  | EQUALS |    |
| FFTAG10 | EQUALS |    |
| FFTAG12 | EQUALS |    |
| P30SUBS | EQUALS |    |
| STOPRAT | EQUALS |    |
| P23S    | EQUALS |    |
|         | BNKSUM | 02 |

|         |        |    |
|---------|--------|----|
|         | BLOCK  | 03 |
| FFTAG5  | EQUALS |    |
| FFTAG6  | EQUALS |    |
| DAPS9   | EQUALS |    |
| FFTAG13 | EQUALS |    |
|         | BNKSUM | 03 |

|         |        |    |
|---------|--------|----|
|         | BANK   | 00 |
| DLAYJOB | EQUALS |    |
|         | BNKSUM | 00 |

|         |        |    |
|---------|--------|----|
|         | BANK   | 01 |
| RESTART | EQUALS |    |
|         | BNKSUM | 01 |

|          |        |   |
|----------|--------|---|
|          | BANK   | 4 |
| VERB37   | EQUALS |   |
| CONICS1  | EQUALS |   |
| PINBALL4 | EQUALS |   |
| CSI/CDH1 | EQUALS |   |
| INTPRET2 | EQUALS |   |
| IMUCAL1  | EQUALS |   |

# Page 28

|          |        |    |
|----------|--------|----|
| STBLEORB | EQUALS |    |
| E/PROG   | EQUALS |    |
| MIDDGIM  | EQUALS |    |
|          | BNKSUM | 04 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 5  |
| FRANDRES | EQUALS |    |
| DOWNTLM  | EQUALS |    |
| DAPMASS  | EQUALS |    |
| CDHTAG   | EQUALS |    |
|          | BNKSUM | 05 |



# MODULE 2 CONTAINS BANKS 6 THROUGH 13

|         |        |    |
|---------|--------|----|
|         | BANK   | 6  |
| IMUCOMP | EQUALS |    |
| T4RUP   | EQUALS |    |
| IMUCAL2 | EQUALS |    |
| CSIPROG | EQUALS |    |
|         | BNKSUM | 06 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 7  |
| SXTMARKE | EQUALS |    |
| R02      | EQUALS |    |
| MODESW   | EQUALS |    |
| XANG     | EQUALS |    |
| KEYRUPT  | EQUALS |    |
| CSIPROG6 | EQUALS |    |
|          | BNKSUM | 07 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 10 |
| DISPLAYS | EQUALS |    |
| PHASETAB | EQUALS |    |
| COMGEOM2 | EQUALS |    |
| SXTMARK1 | EQUALS |    |
| P60S4    | EQUALS |    |
| OPTDRV   | EQUALS |    |
| CSIPROG8 | EQUALS |    |
|          | BNKSUM | 10 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 11 |
| ORBITAL  | EQUALS |    |
| ORBITAL1 | EQUALS |    |

# CONSTANTS

# Page 29

|          |        |    |
|----------|--------|----|
| INTVEL   | EQUALS |    |
| S52/2    | EQUALS |    |
| CSIPROG5 | EQUALS |    |
| INTINIT1 | EQUALS |    |
|          | BNKSUM | 11 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 12 |
| CONICS   | EQUALS |    |
| CSIPROG2 | EQUALS |    |
| CSI/CDH2 | EQUALS |    |
| MODCHG2  | EQUALS |    |

|          |        |    |
|----------|--------|----|
|          | BNKSUM | 12 |
|          | BANK   | 13 |
| P76LOC   | EQUALS |    |
| LATLONG  | EQUALS |    |
| INTINIT  | EQUALS |    |
| SR52/1   | EQUALS |    |
| ORBITAL2 | EQUALS |    |
| CDHTAGS  | EQUALS |    |
| E/PROG1  | EQUALS |    |
| MODCHG3  | EQUALS |    |
|          | BNKSUM | 13 |

# MODULE 3 CONTAINS BANKS 14 THROUGH 21

|          |        |    |
|----------|--------|----|
|          | BANK   | 14 |
| STARTAB  | EQUALS |    |
| RT53     | EQUALS |    |
| P50S1    | EQUALS |    |
| MEASINC2 | EQUALS |    |
| CSI/CDH3 | EQUALS |    |
|          | BNKSUM | 14 |

|         |        |    |
|---------|--------|----|
|         | BANK   | 15 |
| P50S    | EQUALS |    |
| ETRYDAP | EQUALS |    |
| S52/3   | EQUALS |    |
|         | BNKSUM | 15 |

|       |        |    |
|-------|--------|----|
|       | BANK   | 16 |
| P40S1 | EQUALS |    |

# Page 30

|         |        |    |
|---------|--------|----|
| DAPROLL | EQUALS |    |
| P50S2   | EQUALS |    |
| P23S1   | EQUALS |    |
| RTE2    | EQUALS |    |
|         | BNKSUM | 16 |

|       |        |    |
|-------|--------|----|
|       | BANK   | 17 |
| DAPS4 | EQUALS |    |
| DAPS5 | EQUALS |    |
| DAPS7 | EQUALS |    |
| P50S3 | EQUALS |    |
|       | BNKSUM | 17 |

July 12, 2016

1843

|          |        |    |
|----------|--------|----|
|          | BANK   | 20 |
| DAPS6    | EQUALS |    |
| DAPS1    | EQUALS |    |
| DAPS2    | EQUALS |    |
| MANUSTUF | EQUALS |    |
| R36CM    | EQUALS |    |
| VAC5LOC  | EQUALS |    |
|          | BNKSUM | 20 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 21 |
| DAPS3    | EQUALS |    |
| MYSUBS   | EQUALS |    |
| KALCMON3 | EQUALS |    |
|          | BNKSUM | 21 |

# MODULE 4 CONTAINS BANKS 22 THROUGH 27

|          |        |    |
|----------|--------|----|
|          | BANK   | 22 |
| RTBCODES | EQUALS |    |
| RTBCODE1 | EQUALS |    |
| DAPS8    | EQUALS |    |
| AOPERI   | EQUALS |    |
| P40S5    | EQUALS |    |
| KALCMON2 | EQUALS |    |
| KALCMON1 | EQUALS |    |
| CSIPROG3 | EQUALS |    |
|          | BNKSUM | 22 |

# Page 31

|          |        |    |
|----------|--------|----|
|          | BANK   | 23 |
| P20S2    | EQUALS |    |
| INFLIGHT | EQUALS |    |
| COMGEOM1 | EQUALS |    |
| POWFLITE | EQUALS |    |
| POWFLIT1 | EQUALS |    |
| RENDGUID | EQUALS |    |
| POWFLIT2 | EQUALS |    |
| R30LOC   | EQUALS |    |
| P11FOUR  | EQUALS |    |
| CSIPROG4 | EQUALS |    |
|          | BNKSUM | 23 |

|         |        |    |
|---------|--------|----|
|         | BANK   | 24 |
| LOADDAP | EQUALS |    |

1844

July 12, 2016

P40S                   EQUALS  
CSIPROG7           EQUALS  
                      BNKSUM   24

                      BANK      25  
REENTRY            EQUALS  
CDHTAG1            EQUALS  
                      BNKSUM   25

                      BANK      26  
INTPRET1           EQUALS  
REENTRY1           EQUALS  
P60S                EQUALS  
P60S1               EQUALS  
P60S2               EQUALS  
P60S3               EQUALS  
PLANTIN            EQUALS  
EPHEM               EQUALS  
P05P06             EQUALS  
26P50S             EQUALS  
                      BNKSUM   26

                      BANK      27  
TOF-FF             EQUALS  
TOF-FF1            EQUALS  
MANUVER            EQUALS  
MANUVER1           EQUALS

# Page 32

VECPT               EQUALS  
UPDATE1            EQUALS  
UPDATE2            EQUALS  
R22S1               EQUALS  
P60S5               EQUALS  
P40S2               EQUALS  
                      BNKSUM   27

# MODULE 5 CONTAINS BANKS 30 THROUGH 35

                      BANK      30  
IMUSUPER           EQUALS  
LOWSUPER           EQUALS  
FCSTART            EQUALS  
LOPC                EQUALS  
P20S1               EQUALS

# LUNAR ROT

# STANDARD LOCATION FOR THIS. (FOR EXTIV8)

July 12, 2016

1845

|       |        |    |
|-------|--------|----|
| P20S6 | EQUALS |    |
| P40S3 | EQUALS |    |
| R35A  | EQUALS |    |
|       | BNKSUM | 30 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 31 |
| R35      | EQUALS |    |
| RT23     | EQUALS |    |
| P30S1A   | EQUALS |    |
| R34      | EQUALS |    |
| CDHTAG2  | EQUALS |    |
| CSIPROG9 | EQUALS |    |
| R31      | EQUALS |    |
| P22S     | EQUALS |    |
| RTE3     | EQUALS |    |
|          | BNKSUM | 31 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 32 |
| MSGSCAN1 | EQUALS |    |
| RTE      | EQUALS |    |
| DELRSP11 | EQUALS |    |
| IMUCAL3  | EQUALS |    |
|          | BNKSUM | 32 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 33 |
| TESTLEAD | EQUALS |    |

# Page 33

|        |        |    |
|--------|--------|----|
| IMUCAL | EQUALS |    |
|        | BNKSUM | 33 |

|        |        |    |
|--------|--------|----|
|        | BANK   | 34 |
| P110NE | EQUALS |    |
| P20S3  | EQUALS |    |
| P20S4  | EQUALS |    |
| RTECON | EQUALS |    |
|        | BNKSUM | 34 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 35 |
| RTECON1  | EQUALS |    |
| CSI/CDH  | EQUALS |    |
| P30S1    | EQUALS |    |
| P30S     | EQUALS |    |
| P17S1    | EQUALS |    |
| MEASINC3 | EQUALS |    |

1846

July 12, 2016

|          |        |    |
|----------|--------|----|
| INTINIT2 | EQUALS |    |
|          | BNKSUM | 35 |

# MODULE 6 CONTAINS BANKS 36 THROUGH 43

|          |        |    |
|----------|--------|----|
|          | BANK   | 36 |
| MEASINC  | EQUALS |    |
| MEASINC1 | EQUALS |    |
| P17S     | EQUALS |    |
| RTE1     | EQUALS |    |
| S3435LOC | EQUALS |    |
|          | BNKSUM | 36 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 37 |
| P20S     | EQUALS |    |
| BODYATT  | EQUALS |    |
| RENDEZ   | EQUALS |    |
| SERVICES | EQUALS |    |
| P11TWO   | EQUALS |    |
| CDHTAG3  | EQUALS |    |
|          | BNKSUM | 37 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 40 |
| PINSUPER | EQUALS |    |

# Page 34

|          |        |    |
|----------|--------|----|
| SELSUPR  | EQUALS |    |
| PINBALL1 | EQUALS |    |
| R36CM1   | EQUALS |    |
|          | BNKSUM | 40 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 41 |
| PINBALL2 | EQUALS |    |
| R36LM    | EQUALS |    |
|          | BNKSUM | 41 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 42 |
| SBAND    | EQUALS |    |
| PINBALL3 | EQUALS |    |
| EXTVBS   | EQUALS |    |
| R36LM1   | EQUALS |    |
|          | BNKSUM | 42 |

|          |        |    |
|----------|--------|----|
|          | BANK   | 43 |
| SELFCHEC | EQUALS |    |

July 12, 2016

1847

```
EXTVERBS EQUALS
BNKSUM 43

HI6ZEROS EQUALS ZEROVECS # ZERO VECTOR ALWAYS IN HIGH MEMORY
LO6ZEROS EQUALS ZEROVEC # ZERO VECTOR ALWAYS IN LOW MEMORY
HIDPHALF EQUALS UNITX
LODPHALF EQUALS XUNIT
HIDP1/4 EQUALS DP1/4TH
LODP1/4 EQUALS D1/4 # 2DEC .25
HIUNITX EQUALS UNITX
HIUNITY EQUALS UNITY
HIUNITZ EQUALS UNITZ
LOUNITX EQUALS XUNIT # 2DEC .5
LOUNITY EQUALS YUNIT # 2DEC 0
LOUNITZ EQUALS ZUNIT # 2DEC 0
3/4LOWDP EQUALS 3/4 # 2DEC 3.0 B-2
```

SBANK= LOWSUPER

# ROPE-SPECIFIC ASSIGNS OBVIATING NEED TO CHECK COMPUTER FLAG IN DETERMINING(?) INTEGRATION ARE

```
OTHPREC EQUALS LEMPREC
ATOPOTH EQUALS ATOPLEM
ATOPTHIS EQUALS ATOPCSM
MOONTHIS EQUALS CMOONFLG
```

# Page 35

```
MOONOTH EQUALS LMOONFLG
MOVATHIS EQUALS MOVEACSM
STATEST EQUALS V83CALL # * TEMPORARY
THISPREC EQUALS CSMPREC
THISAXIS = UNITX
ERASID EQUALS LOW10 # DOWNLINK ERASABLE DUMP ID
DELAYNUM EQUALS THREE
```

\*\*\*\*\*

# THE FOLLOWING ECADRS ARE DEFINED TO FACILITATE EBANK SWITCHING. THEY ALSO MAKE IT EASIER FOR  
# ERASABLE CONTROL TO REARRANGE ERASABLE MEMORY WITHOUT DISRUPTING THE PROGRAMS WHICH SET EBANK  
# PRIOR TO ROPE RELEASE FIXED MEMORY CAN BE SAVED BY SETTING EACH EBXXXX =EBANKX (X=4,5,6,7).  
# WILL BE THE BANK WHERE THE ERASABLES REFERENCED IN EBXXXX WILL BE STORED.

```
BANK 7
EBANK= MARKDOWN
EBMARKDO ECADR MARKDOWN
```

|          |        |         |
|----------|--------|---------|
|          | EBANK= | MRKBUF1 |
| EBMRKBUF | ECADR  | MRKBUF1 |

|          |        |        |
|----------|--------|--------|
|          | BANK   | 24     |
|          | EBANK= | DVCNTR |
| EBDVCNTR | ECADR  | DVCNTR |
|          | EBANK= | P40TMP |
| EBP40TMP | ECADR  | P40TMP |

|          |        |         |
|----------|--------|---------|
|          | BANK   | 34      |
|          | EBANK= | DVCNTR  |
| EBDVCNT  | ECADR  | DVCNTR  |
|          | EBANK= | QPLACES |
| EBQPLACE | ECADR  | QPLACES |

|       |        |     |
|-------|--------|-----|
|       | BANK   | 37  |
|       | EBANK= | RN1 |
| EBRN1 | ECADR  | RN1 |

\*\*\*\*\*

# Page 36

# \*\*\* END OF MAIN PROGRAM \*\*\*

This code is written to file `src/TAGS-FOR-RELATIVE-SETLOC.s`.



## B.113 THE LUNAR LANDING

1849

*<src/THE-LUNAR-LANDING.s 1849>≡*

```
Copyright: Public domain.
Filename: THE_LUNAR_LANDING.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Hartmuth Gutsche<hgutsche@explornet.com>.
Website: www.ibiblio.org/apollo.
Pages: 785-792
Mod history: 2009-05-20 HG Transcribed from page images.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969

Page 785

 BANK 32
 SETLOC F2DPS*32
 BANK

 EBANK= E2DPS

P63: THE LUNAR LANDING, BRAKING PHASE

 COUNT* $$/P63

P63LM TC PHASCHNG
 OCT 04024

 TC BANKCALL # DO IMU STATUS CHECK ROUTINE R02
```

|            |        |          |                                          |
|------------|--------|----------|------------------------------------------|
|            | CADR   | R02BOTH  |                                          |
|            | CAF    | P63ADRES | # INITIALIZE WHICH FOR BURNBABY          |
|            | TS     | WHICH    |                                          |
|            | CAF    | DPSTHRSH | # INITIALIZE DVMON                       |
|            | TS     | DVTHRUSH |                                          |
|            | CAF    | FOUR     |                                          |
|            | TS     | DVCNTR   |                                          |
|            | CS     | ONE      | # INITIALIZE WCHPHASE AND FLPASSO        |
|            | TS     | WCHPHASE |                                          |
|            | CA     | ZERO     |                                          |
|            | TS     | FLPASSO  |                                          |
|            | CS     | BIT14    |                                          |
|            | EXTEND |          |                                          |
|            | WAND   | CHAN12   | # REMOVE TRACK-ENABLE DISCRETE.          |
| FLAGORGY   | TC     | INTPRET  | # DIONYSIAN FLAG WAVING                  |
|            | CLEAR  | CLEAR    |                                          |
|            |        | NOTHROTL |                                          |
|            |        | REDFLAG  |                                          |
|            | CLEAR  | SET      |                                          |
|            |        | LRBYPASS |                                          |
|            |        | MUNFLAG  |                                          |
|            | CLEAR  | CLEAR    |                                          |
|            |        | P25FLAG  | # TERMINATE P25 IF IT IS RUNNING.        |
|            |        | RNDVZFLG | # TERMINATE P20 IF IT IS RUNNING.        |
|            |        |          | # *****                                  |
| IGNALG     | SETPD  | VLOAD    | # FIRST SET UP INPUTS FOR RP-TO-R:       |
| # Page 786 |        | 0        | # AT 0D LANDING SITE IN MOON FIXED FRAME |
|            |        | RLS      | # AT 6D ESTIMATED TIME OF LANDING        |
|            | PDDL   | PUSH     | # MPAC NON-ZERO TO INDICATE LUNAR CASE   |
|            |        | TLAND    |                                          |
|            | STCALL | TPIP     | # ALSO SET TPIP FOR FIRST GUIDANCE PASS  |
|            |        | RP-TO-R  |                                          |
|            | VSL4   | MXV      |                                          |
|            |        | REFSMMAT |                                          |
|            | STCALL | LAND     |                                          |
|            |        | GUIDINIT | # GUIDINIT INITIALIZES WM AND /LAND/     |
|            | DLOAD  | DSU      |                                          |

July 12, 2016

1851

```

 TLAND
 GUIDDURN
STCALL TDEC1 # INTEGRATE STATE FORWARD TO THAT TIME
 LEMPREC
SSP VLOAD
 NIGNLOOP
 40D
 UNITX
STOVL CG
 UNITY
STOVL CG +6
 UNITZ
STODL CG +14
 99999CON
STOVL DELTAH # INITIALIZE DELTAH FOR V16N68 DISPLAY
 ZEROVECS
STODL UNFC/2 # INITIALIZE TRIM VELOCITY CORRECTION TERM
 HI6ZEROS
STORE TTF/8

IGNALOOP DLOAD
 TAT
STOVL PIPTIME1
 RATT1
VSL4 MXV
 REFSMMAT
STCALL R
 MUNGRAV
STCALL GDT/2
 ?GUIDSUB # WHICH DELIVERS N PASSES OF GUIDANCE

DDUMCALC IS PROGRAMMED AS FOLLOWS:
#
2
(RIGNZ - RGU)/16 + 16(RGU)KIGNY/B8 + (RGU - RIGNX)KIGNX/B4 + (ABVAL(VGU) - VI
2 1 0
DDUM = -----
10
2 (VGU - 16 VGU KIGNX/B4)
2 0
Page 787 new page is actually one line earlier but this would put the indices on a seperate l
disconnected from their respective variables
THE NUMERATOR IS SCALED IN METERS AT 2(28). THE DENOMINATOR IS A VELOCITY IN UNITS OF 2(10)
THE QUOTIENT IS THUS A TIME IN UNITS OF 2(18) CENTISECONDS. THE FINAL SHIFT RESCALES TO UNIT
THERE IS NO DAMPING FACTOR. THE CONSTANTS KIGNX/B4, KIGNY/B8 AND KIGNV/B4 ARE ALL NEGATIVE I

DDUMCALC TS NIGNLOOP
```

|       |          |                                        |
|-------|----------|----------------------------------------|
| TC    | INTPRET  |                                        |
| DLOAD | DMPR     | # FORM DENOMINATOR FIRST               |
|       | VGU      |                                        |
|       | KIGNX/B4 |                                        |
| SL4R  | BDSU     |                                        |
|       | VGU +4   |                                        |
| PDDL  | DSU      |                                        |
|       | RIGNZ    |                                        |
|       | RGU +4   |                                        |
| SR4R  | PDDL     |                                        |
|       | RGU +2   |                                        |
| DSQ   | DMPR     |                                        |
|       | KIGNY/B8 |                                        |
| SL4R  | PDDL     |                                        |
|       | RGU      |                                        |
| DSU   | DMPR     |                                        |
|       | RIGNX    |                                        |
|       | KIGNX/B4 |                                        |
| PDVL  | ABVAL    |                                        |
|       | VGU      |                                        |
| DSU   | DMPR     |                                        |
|       | VIGN     |                                        |
|       | KIGNV/B4 |                                        |
| DAD   | DAD      |                                        |
| DAD   | DDV      |                                        |
| SRR   |          |                                        |
|       | 10D      |                                        |
| PUSH  | DAD      |                                        |
|       | PIPTIME1 |                                        |
| STODL | TDEC1    | # STORE NEW GUESS FOR NEXT INTEGRATION |
| ABS   | DSU      |                                        |
|       | DDUMCRIT |                                        |
| BMN   | CALL     |                                        |
|       | DDUMGOOD |                                        |
|       | INTSTALL |                                        |
| SET   | SET      |                                        |
|       | INTYPFLG |                                        |
|       | MOONFLAG |                                        |
| DLOAD |          |                                        |
|       | PIPTIME1 |                                        |
| STOVL | TET      | # HOPEFULLY ?GUIDSUB DID NOT           |
|       | RATT1    | # CLOBBER RATT1 AND VATT1              |
| STOVL | RCV      |                                        |
|       | VATT1    |                                        |

1853

|          |        |          |                                          |
|----------|--------|----------|------------------------------------------|
|          | STCALL | VCV      |                                          |
|          |        | INTEGRVS |                                          |
|          | GOTO   |          |                                          |
|          |        | IGNALOOP |                                          |
| DDUMGOOD | SLOAD  | SR       |                                          |
|          |        | ZOOMTIME |                                          |
|          |        | 14D      |                                          |
|          | BDSU   |          |                                          |
|          |        | TDEC1    |                                          |
|          | STOVL  | TIG      | # COMPUTE DISTANCE LANDING SITE WILL BE  |
|          |        | V        | # OUT OF LM'S ORBITAL PLANE AT IGNITION: |
|          | VXV    | UNIT     | # SIGN IS + IF LANDING SITE IS TO THE    |
|          |        | R        | # RIGHT, NORTH; - IF TO THE LEFT, SOUTH. |
|          | DOT    | SL1      |                                          |
|          |        | LAND     |                                          |
| R60INIT  | STOVL  | OUTOFPLN | # INITIALIZATION FOR CALCMANU            |
|          |        | UNFC/2   |                                          |
|          | STORE  | R60VSAVE | # STORE UNFC/2 TEMPORARILY IN R60SAVE    |
|          | EXIT   |          |                                          |
|          |        |          | # *****                                  |
| IGNALGRT | TC     | PHASCHNG | # PREVENT REPEATING IGNALG               |
|          | OCT    | 04024    |                                          |
| ASTNCLOK | CS     | ASTNDEX  |                                          |
|          | TC     | BANKCALL |                                          |
|          | CADR   | STCLOK2  |                                          |
|          | TCF    | ENDOFJOB | # RETURN IN NEW JOB AND IN EBANK FIVE    |
| ASTNRET  | TC     | INTPRET  |                                          |
|          | SSP    | RTB      | # GO PICK UP DISPLAY AT END OF R51:      |
|          |        | QMAJ     | # "PROCEED" WILL DO A FINE ALIGNMENT     |
|          | FCADR  | P63SPOT2 | # "ENTER" WILL RETURN TO P63SPOT2        |
|          |        | R51P63   |                                          |
| P63SPOT2 | VLOAD  | UNIT     | # INITIALIZE KALCMANU FOR BURN ATTITUDE  |
|          |        | R60VSAVE |                                          |
|          | STOVL  | POINTVSM |                                          |
|          |        | UNITX    |                                          |
|          | STORE  | SCAXIS   |                                          |
|          | EXIT   |          |                                          |
|          | CAF    | EBANK7   |                                          |
|          | TS     | EBANK    |                                          |
|          | INHINT |          |                                          |

```

Page 789
TC IBNKCALL
CADR PFLITEDB

RELINT

TC BANKCALL
CADR R60LEM

TC PHASCHNG # PREVENT RECALLING R60
OCT 04024

P63SPOT3 CA BIT6 # IS THE LR ANTENNA IN POSITION 1 YET
 EXTEND
 RAND CHAN33
 EXTEND
 BZF P63SPOT4 # BRANCH IF ANTENNA ALREADY IN POSITION 1

 CAF CODE500 # ASTRONAUT: PLEASE CRANK THE
 TC BANKCALL # SILLY THING AROUND
 CADR GOPERF1
 TCF GOTOP00H # TERMINATE
 TCF P63SPOT3 # PROCEED SEE IF HE'S LYING

P63SPOT4 TC BANKCALL # ENTER INITIALIZE LANDING RADAR
 CADR SETPOS1

 TC POSTJUMP # OFF TO SEE THE WIZARD ...
 CADR BURNBABY

CONSTANTS FOR P63LM AND IGNALG

P63ADRES GENADR P63TABLE

ASTNDEX = MD1 # OCT 25: INDEX FOR CLOKTASK

CODE500 OCT 00500

99999CON 2DEC 30479.7 B-24

GUIDDURN 2DEC +66440 # GUIDDURN +6.64400314 E+2
DDUMCRIT 2DEC +8 B-28 # CRITERION FOR IGNALG CONVERGENCE

Page 790

```

July 12, 2016

1855

# Page 791

```

P68: LANDING CONFIRMATION

```

```

 BANK 31
 SETLOC F2DPS*31
 BANK
 COUNT* $$/P6567

LANDJUNK TC PHASCHNG
 OCT 04024

 INHINT
 TC BANKCALL # ZERO ATTITUDE ERROR
 CADR ZATTEROR

 TC BANKCALL # SET 5 DEGREE DEADBAND
 CADR SETMAXDB

 TC INTERPRET # TO INTERPRETIVE AS TIME IS NOT CRITICAL
 SET CLEAR
 SURFFLAG
 LETABORT
 SET VLOAD
 APSFLAG
 RN
 STODL ALPHAV
 PIPTIME
 SET CALL
 LUNAFLAG
 LAT-LONG
 SETPD VLOAD # COMPUTE RLS AND STORE IT AWAY
 0
 RN
 VSL2 PDDL
 PIPTIME
 PUSH CALL
 R-TO-RP
 STORE RLS
 EXIT
 CAF V06N43* # ASTRONAUT: NOW LOOK WHERE TO ENDED UP
 TC BANKCALL
 CADR GOFLASH
```

1856

July 12, 2016

```
TCF GOTOPOOH # TERMINATE
TCF +2 # PROCEED
TCF -5 # RECYCLE

TC INTPRET

Page 792
VLOAD # INITIALIZE GSAV AND (USING REFMF)
UNITX # YNBSAV, ZNBSAV AND ATTFLAG FOR P57
STCALL GSAV
REFMF
EXIT

TCF GOTOPOOH # ASTRONAUT: PLEASE SELECT P57

V06N43* VN 0643
```

This code is written to file `src/THE-LUNAR-LANDING.s`.



## B.114 THROTTLE CONTROL ROUTINES

1857 *<src/THROTTLE-CONTROL-ROUTINES.s 1857>≡*

```
Copyright: Public domain.
Filename: THROTTLE_CONTROL_ROUTINES.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: HARTMUTH GUTSCHE <hgutsche@explornet.com>.
Website: www.ibiblio.org/apollo.
Pages: 793-797
Mod history: 2009-05-20 HG Transcribed from page images.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
```

# Page 793

```
BANK 31
SETLOC FTHROT
BANK
EBANK= PIF
COUNT* $$/THROT
```

```
* * * * *
HERE FC, DESIRED THRUST, AND FP, PRESENT THRUST, UNWEIGHTED, ARE COMPUTED.
```

```
THROTTLE CA ABDELV # COMPUTE PRESENT ACCELERATION IN UNITS OF
EXTEND # 2(-4) M/CS/CS, SAVING SERVICER TROUBLE
MP /AF/CNST
+3 EXTEND
QXCH RTNHOLD
AFDUMP TC MASSMULT
DXCH FP # FP = PRESENT THRUST
```

```

EXTEND
DCA /AFC/
TC MASSMULT
TS FC # FC = THRUST DESIRED BY GUIDANCE
DXCH FCODD # FCODD = WHAT IT IS GOING TO GET

IF IT HAS BEEN LESS THAN 3 SECONDS SINCE THE LAST THROTTLING, AUGMENT FP USING THE

CS TTHROT # THIS CODING ASSUMES A FLATOUT WITHIN
AD TIME1 # 80 SECONDS BEFORE FIRST THROTTLE CALL
MASK POSMAX
COM
AD 3SECS
EXTEND
BZMF WHERETO # BRANCH IF (TIME1-TTHROT +1) > 3 SECONDS
EXTEND
DCA FWEIGHT
DAS FP

THIS LOGIC DETERMINES THE THROTTLING IN THE REGION 10% - 94%. THE MANUAL THROTTLE
MINIMUM BY ASTRONAUT OR MISSION CONTROL PROGRAMS, PROVIDES THE LOWER BOUND. A STOP
PROVIDES THE UPPER.

WHERETO CA EBANK5 # INITIALIZE L*WCR*T AND H*GHCR*T FROM
TS EBANK # PAD LOADED ERASABLES IN W-MATRIX

#Page 794
EBANK= LOWCRIT
EXTEND
DCA LOWCRIT
DXCH L*WCR*T
CA EBANK7
TS EBANK
EBANK= PIF
CS ZERO # INITIALIZE PIFPSET
TS PIFPSET
CS H*GHCR*T
AD FCOLD
EXTEND
BZMF LOWFCOLD # BRANCH IF FCOLD < OR = HIGHCRIT
CS L*WCR*T
AD FCODD
EXTEND
BZMF FCOMPSET # BRANCH IF FC < OR = LOWCRIT
CA FP # SEE NOTE 1
TCF FLATOUT1

```

July 12, 2016

1859

```
FCOMPSET CS FMAXODD # SEE NOTE 2
 AD FP
 TCF FLATOUT2

LOWFCOLD CS H*GHCR*T
 AD FCODD
 EXTEND
 BZMF DOPIF # BRANCH IF FC < OR = HIGHCRIT

 CA FMAXPOS # NO: THROTTLE-UP
FLATOUT1 DXCH FCODD
 CA FEXTRA
FLATOUT2 TS PIFPSET

NOTE 1 FC IS SET EQUAL TO FP SO PIF WILL BE ZERO. THIS IS DESIRABLE
AS THERE IS ACTUALLY NO THROTTLE CHANGE.
#
NOTE2 HERE, SINCE WE ARE ABOUT TO RETURN TO THE THROTTLEABLE REGION
(BELOW 55%) THE QUANTITY -(FMAXODD-FP) IS COMPUTED AND PUT
INTO PIFPSET TO COMPENSATE FOR THE DIFFERENCE BETWEEN THE
NUMBER OF BITS CORRESPONDING TO FULL THROTTLE (FMAXODD) AND THE
NUMBER CORRESPONDING TO ACTUAL THRUST (FP). THUS THE TOTAL
THROTTLE COMMAND PIF = FC - FP - (FMAXODD - FP) = FC - FMAXODD.

DOPIF TC FASTCHNG
 EXTEND
 DCA FCODD
 TS FCOLD
 DXCH PIF
 EXTEND

#Page 795 DCS FP
 DAS PIF # PIF = FC - FP, NEVER EQUALS +0

DOIT CA PIF
 AD PIFPSET # ADD IN PIFPSET, WITHOUT CHANGING PIF
 TS PSEUDO55
 TS THRUST
 CAF BIT4
 EXTEND
 WOR CHAN14
 CA TIME1
 TS TTHROT
```

```
SINCE /AF/ IS NOT AN INSTANTANEOUS ACCELERATION, BUT RATHER AN "AVERAGE" OF THE ACCELERATION I
THE PRECEEDING PIPA INTERVAL, AND SINCE FP IS COMPUTED DIRECTLY FROM /AF/, FP IN ORDER TO COR
```

# ACTUAL THRUST LEVEL AT THE END OF THE INTERVAL MUST BE WEIGHTED BY

#

#  
#  
#
$$\text{FWEIGHT} = \frac{\text{PIF}(\text{PPROCESS} + \text{TL})}{\text{PGUID}} + \frac{\text{PIF} / \text{PIF/}}{2 \text{ PGUID FRATE}}$$
  
#

# WHERE PROCESS IS THE TIME BETWEEN PIPA READING AND THE START OF THROTTLING, PGUID IS

# FRATE IS THE THROTTLING RATE (32 UNITS PER CENTISECOND). PGUID IS EITHER 1 OR 2 SE

# FIRST TERM REPRESENTS THE ENGINE'S RESPONSE LAG. HERE FWEIGHT IS COMPUTED FOR USE

CA THISTPIP +1 # INITIALIZE FWEIGHT COMP AS IF FOR P  
TS BUF

CS MODREG # ARE WE IN FACT IN P66?  
AD DEC66

EXTEND  
BZF FWCOMP # YES

CA PIPTIME +1 # NO: INITIALIZE FOR TWO SECOND PER.  
TS BUF  
CAF 4SECS  
TCF FWCOMP +1

FWCOMP  
+1

CAF 2SECS  
TS Q  
EXTEND  
MP BIT6  
LXCH BUF +1  
CS BUF # TIME OF LAST PIPA READING.  
AD TIME1  
AD THROTLAG # COMPENSATE FOR ENGINE RESPONSE LAG  
MASK LOW8 # MAKE SURE SMALL AND POSITIVE  
ZL  
EXTEND

#Page 796

DV Q  
EXTEND  
MP PIF  
DOUBLE  
DXCH FWEIGHT  
CCS PIF  
AD ONE  
TCF +2  
AD ONE  
EXTEND  
MP PIF

July 12, 2016

1861

```

 EXTEND
 DV BUF +1
 ZL
 DAS FWEIGHT

THDUMP TC RTNHOLD
```

# FLATOUT THROTTLES UP THE DESCENT ENGINE, AND IS CALLED AS A BASIC SUBROUTINE.

```

FLATOUT CAF BIT13 # 4096 PULSES
WHATOUT TS PIFPSET # USE PIFPSET SO FWEIGHT WILL BE ZERO
 CS ZERO
 TS FCOLD
 TS PIF
 EXTEND
 QXCH RTNHOLD
 TCF DOIT
```

# MASSMULT SCALES ACCELERATION, ARRIVING IN A AND L IN UNITS OF 2(-4) M/CS/CS, TO FORCE IN PULSES

```

MASSMULT EXTEND
 QXCH BUF
 DXCH MPAC
 TC DMP
 ADRES MASS
 TC DMP # LEAVES PROPERLY SCALED FORCE IN MPAC
 ADRES SCALEFAC
 TC TPAGREE
 CA MPAC
 EXTEND
 BZF +3
 CAF POSMAX
 TC BUF
 DXCH MPAC +1
 TC BUF
```

#Page 797

# CONSTANTS --

```

FEXTRA = BIT13 # FEXT +5.13309020 E+4
```

```

/AF/CNST DEC .13107
```

# \* \* \* \* \*

This code is written to file `src/THROTTLE-CONTROL-ROUTINES.s`.

## B.115 TIME OF FREE FALL

```

1862 <src/TIME-OF-FREE-FALL.s 1862>≡
 # Copyright: Public domain.
 # Filename: TIME_OF_FREE_FALL.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 1373-1388
 # Mod history: 2009-05-10 SN (Sergio Navarro). Started adapting
 # from the Colossus249/ file of the same
 # name, using Comanche055 page images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 1373
 # THE TFF SUBROUTINES MAY BE USED IN EITHER EARTH OR MOON CENTERED COORDINATES. THE
 # KNOW WHICH ORIGIN APPLIES. IT IS THE USER WHO KNOWS, AND WHO SUPPLIES RONE, VONE,
 # APPROPRIATE SCALE LEVEL FOR THE PROPER PRIMARY BODY.
 #
 # EARTH ORIGIN POSITION -29 METERS
 # VELOCITY -7 METERS/CENTISECOND
 # 1/SQRT(MU) +17 SQRT(CS SQ/METERS CUBED)
 #
 # MOON ORIGIN POSITION -27 METERS
 # VELOCITY -5 METERS/CENTISECONDS
 # 1/SQRT(MU) +14 SQRT(CS SQ/METERS CUBED)
 #

```

```

ALL DATA PROVIDED TO AND RECEIVED FROM ANY TFF SUBROUTINE WILL BE AT ONE OF THE LEVELS ABOVE.
THE FREE FALL TIME IS RETURNED IN CENTISECONDS AT (-28). PROGRAM TFF/CONIC WILL GENERATE VONE
LEAVE IT IN VONE' AT (+10) IF EARTH ORIGIN AND (+9) IF MOON ORIGIN.
#
THE USER MUST STORE THE STATE VECTOR IN RONE, VONE, AND MU IN THE FORM 1/SQRT(MU) IN TFF/RTMU
AT THE PROPER SCALE BEFORE CALLING TFF/CONIC. SINCE RONE, VONE ARE IN THE EXTENDED VERB STORAGE
THE USER MUST ALSO LOCK OUT THE EXTENDED VERBS, AND RELEASE THEM WHEN FINISHED.
#
PROGRAMS CALC/TFF AND CALC/TPER ASSUME THAT THE TERMINAL RADIUS IS LESS THAN THE PRESENT
RADIUS. THIS RESTRICTION CAN BE REMOVED BY A 15 W CODING CHANGE, BUT AT PRESENT IT IS NOT DONE.
#
THE FOLLOWING ERASABLE QUANTITIES ARE USED BY THE TFF ROUTINES, AND ARE LOCATED IN THE PUSH DOWN
#
BELOW E: IS USED FOR EARTH ORIGIN SCALE
M: IS USED FOR MOON ORIGIN SCALE
#
#TFFSW = 119D # BIT1 0 = CALCTFF 1 = CALCTPER
TFFDELQ = 10D # Q2-Q1 E: (-16) M: (-15)
RMAG1 = 12D # ABVAL(RN) M E: (-29) M: (-27)
#RPER = 14D # PERIGEE RADIUS M E: (-29) M: (-27)
TFFQ1 = 14D # R.V / SQRT(MUE) E: (-16) M: (-15)
#SDELF/2 = # SIN(THETA) /2
CDELF/2 = 14D # COS(THETA) /2
#RAPO = 16D # APOGEE RADIUS M E: (-29) M: (-27)
NRTERM = 16D # TERMINAL RADIUS M E: (-29+NR)
M: (-27+NR)
RTERM = 18D # TERMINAL RADIUS M E: (-29) M: (-27)
TFFVSQ = 20D # -(V SQUARED/MU) 1/M E: (20) M: (18)
TFF1/ALF = 22D # SEMI MAJ AXIS M E: (-22-2 NA)
M: (-20-2 NA)
TFFRTALF = 24D # SQRT(ALFA) E: (10+NA) M: (9+NA)
TFFALFA = 26D # ALFA 1/M E: (26-NR) M: (24-NR)
TFFNP = 28D # SEMI LATUS RECTUM M E: (-38+2 NR)
M: (-36+2 NR)
TFF/RTMU = 30D # 1/SQRT(MU) E: (17) M: (14)
NRMAG = 32D # PRESENT RADIUS M E: (-29+NR)
M: (-27+NR)
TFFX = 34D #
TFFTEM = 36D # TEMPORARY
Page 1374
#
REGISTERS S1, S2 ARE UNTOUCHED BY ANY TFF SUBROUTINE
#
INDEX REGISTERS X1, X2 ARE USED BY ALL TFF SUBROUTINES. THEY ARE ESTABLISHED IN TFF/CONIC AND MUST BE PRESERVED BETWEEN CALLS TO SUBSEQUENT
SUBROUTINES.
#
-NR C(X1) = NORM COUNT OF RMAG
-NA C(X2) = NORM COUNT OF SQRT(ABS(ALFA))

```

```

Page 1375
SUBROUTINE NAME: TFFCONIC DATE: 01.29.67
MOD NO: 0 LOG SECTION: TIME OF FREE FALL
MOD BY: RR BAIRNSFATHER
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 11 APR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 21 NOV 67 ADD MOON MU.
MOD NO: 3 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 68 ACCEPT DIFFER
#
FUNCTIONAL DESCRIPTION: THIS SUBROUTINE IS CALLED TO COMPUTE THOSE CONIC PARAMETERS
SUBROUTINES AND TO ESTABLISH THEM IN THE PUSH LIST AREA. THE PARAMETERS ARE
THE EQUATIONS ARE:
#

$$\bar{H} = \bar{R}\bar{N}*\bar{V}\bar{N}$$
 ANGULAR MOMENTUM
#

$$\text{LCP} = \bar{H}.\bar{H} / \mu$$
 SEMI LATUS RECTUM
#

$$\text{ALFA} = 2/\bar{R}\bar{N} - \bar{V}\bar{N}.\bar{V}\bar{N} / \mu$$
 RECIPROCAL SEMI MAJ AXIS, SIGNED
#
AND ALFA IS POS FOR ELLIPTIC ORBITS
0 FOR PARABOLIC ORBITS
NEG FOR HYPERBOLIC ORBITS.
SUBROUTINE ALSO COMPUTES AND SAVES RMAG.
#
CALLING SEQUENCE:
TFFCONIC EXPECTS CALLER TO ENTER WITH CORRECT GRAVITATIONAL CONSTANT IN MPAC,
1/SQRT(MU). THE PROGRAM WILL SAVE IN TFF/RTMU. THE SCALE IS DETERMINED BY V
ORIGIN IS USED. THE CALLER MUST LOCK OUT THE EXTENDED VERBS BEFORE PROVIDING
VONE AT PROPER SCALE. THE EXTENDED VERBS MUST BE RESTORED WHEN THE CALLER IS
TFF ROUTINES.
#
ENTRY POINT TFFCONMU EXPECTS THAT TFF/RTMU IS ALREADY LOADED.
#
TO SPECIFY MU: DLOAD CALL # IF MU ALREADY STORED: CALL
YOURMU # 1/RTMU E:(17) M:(14)
TFFCONIC
PUSHLOC = PDL+0, ARBITRARY IF LEQ 18D
#
SUBROUTINES CALLED: NONE
#
NORMAL EXIT MODES: RVQ
#
ALARMS: NONE
#
OUTPUT: THE FOLLOWING ARE STORED IN THE PUSH LIST AREA.
RMAG1 E:(-29) M:(-27) M RN, PRESENT RADIUS LENGTH

```



July 12, 2016

1865

```
NRMAG E: (-29+NR) M RMAG, NORMALIZED
M: (-27+NR)
X1
TFFNP E: (-38+2NR) M LCP, SEMI LATUS RECTUM, WEIGHTED BY
M: (-36+2NR)
TFF/RTMU E: (17) M: (14) 1/SQRT(MU)
TFFVSQ E: (20) M: (18) 1/M -(V SQ/MU): PRESENT VELOCITY, NOR
TFFALFA E: (26-NR) 1/M ALFA, WEIGHTED BY NR
M: (24-NR)
TFFRTALF E: (10+NA) SQRT(ALFA), NORMALIZED
M: (9+NA)
Page 1376
X2
TFF1/ALF E: (-22-2NA) -NA, NORM COUNT
M: (-20-2NA) SIGNED SEMI MAJ AXIS, WEIGHTED BY NA
PUSHLOC AT PDL+0
#
THE FOLLOWING IS STORED IN GENERAL ERASABLE
VONE' E: (10) M: (9) V/RT(MU), NORMALIZED VELOCITY
#
ERASABLE INITIALIZATION REQUIRED:
RONE E: (-29) M: (-27) M STATE VECTOR LEFT BY CALLER
VONE E: (-7) M: (-5) M/CS STATE VECTOR LEFT BY CALLER
TFF/RTMU E: (17) M: (14) 1/RT(CS SQ/M CUBE) IF ENTER VIA TR
#
DEBRIS: QPRET PDL+0 ... PDL+3
#
BANK 33
SETLOC TOF-FF
BANK
#
COUNT* $$/TFF
#
TFFCONIC STORE TFF/RTMU # 1/SQRT(MU) E: (17) M: (14)
#
TFFCONMU VLOAD UNIT # COME HERE WITH TFFRTMU LOADED.
RONE # SAVED RN. M E: (-29) M: (-27)
PDDL # UR/2 TO PDL+0, +5
36D # MAGNITUDE
STORE RMAG1 # M E: (-29) M: (-27)
#
NORM
X1 # -NR
STOVL NRMAG # RMAG M E: (-29+NR) M: (-27+NR)
VONE # SAVED VN. M/CS E: (-7) M: (-5)
VXSC
```

```

STORE TFF/RTMU # E:(17) M:(14)
 VONE' # VN/SQRT(MU) E:(10) M:(9)

VXSC VXV
 NRMAG # E:(-29+NR) M:(-27+NR)
 # UR/2 FROM PDL

VSL1 VSQ # BEFORE: E:(-19+NR) M:(-18+NR)
STODL TFFNP # LC P M E:(-38+2NR) M:(-36+2NR)
 # SAVE ALSO FOR VGAMCALC

 TFF1/4
DDV PDVL # (2/RMAG) 1/M E:(26-NR) M:(24-NR)
 NRMAG # RMAG M E:(-29+NR) M:(-27+NR)
 VONE' # SAVED VN. E:(10) M:(9)
VSQ DCOMP # KEEP MPAC+2 HONEST FOR SQRT.
STORE TFFVSQ # -(V SQ/MU) E:(20) M:(18)
 # SAVE FOR VGAMCALC

SR* DAD

Page 1377

 0 -6,1 # GET -VSQ/MU E:(26-NR) M:(24-NR)
STADR

 # 2/RMAG FROM PDL+2
STORE TFFALFA # ALFA 1/M E:(26-NR) M:(24-NR)
SL* PUSH # TEMP SAVE ALFA E:(20) M:(18)
 0 -6,1
ABS SQRT # E:(10) M:(9)
NORM

 X2 # X2 = -NA
STORE TFFRTALF # SQRT(ABS(ALFA)) E:(10+NA) M:(9+NA)
DSQ SIGN # NOT SO ACCURATE, BUT OK
 # ALFA FROM PDL+2 E:(20) M:(18)
BZE BDDV # SET 1/ALFA =0, TO SHOW SMALL ALFA
 +2
 TFF1/4
+2 STORE TFF1/ALF # 1/ALFA E:(-22-2NA) M:(-20-2NA)
DUMPCNIC RVQ

#
#
Page 1378
SUBROUTINE NAME: TFFRP/RA DATE: 01.17.67
MOD NO: 0 LOG SECTION: TIME OF FREE FALL
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 11 APR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 68 ACCEPT DIFFE
#
ALSO IMPROVE
#
FUNCTIONAL DESCRIPTION: USED BY CALCTPER AND TFF DISPLAYS TO CALCULATE PERIGEE RA
APOGEE RADIUS FOR A GENERAL CONIC.
PROGRAM GIVES PERIGEE RADIUS AS APOGEE RADIUS IS GIVEN BY

```

July 12, 2016

1867

```
RP = P/(1+E) RA = (1+E) / ALFA
WHERE 2
E = 1 - P ALFA
IF RA IS NEGATIVE OR SHOWS DIVIDE OVERFLOW, THEN RA = POSMAX BECAUSE
1. APOGEE RADIUS IS NOT MEANINGFUL FOR HYPERBOLA
2. APOGEE RADIUS IS NOT DEFINED FOR PARABOLA
3. APOGEE RADIUS EXCEEDS THE SCALING FOR ELLIPSE.
#
THIS SUBROUTINE REQUIRED THE SIGNED RECIPROCAL SEMI MAJ AXIS, ALFA, AND SEMI LATUS RECT
#
CALLING SEQUENCE: CALL
TFFRP/RA
PUSHLOC = PDL+0, ARBITRARY IF LEQ 10D
C(MPAC) UNSPECIFIED
#
SUBROUTINES CALLED: NONE
#
NORMAL EXIT MODE: RVQ
IF ELLIPSE, WITHIN NORMAL SCALING, RAPO IS CORRECT.
OTHERWISE, RAPO = POSMAX.
#
ALARMS: NONE
#
OUTPUT: STORED IN PUSH LIST AREA. SCALE OF OUTPUT AGREES WITH DATA SUPPLIED TO TFF/CON
RPER E:(-29) M:(-27) M PERIGEE RADIUS DESTROYED BY CALCTFF/CA
RAPO E:(-29) M:(-27) M APOGEE RADIUS WILL BE DESTROYED BY CA
PUSHLOC AT PDL+0
#
ERASABLE INITIALIZATION REQUIRED:
TFFALFA E:(26-NR) M 1/SEMI MAJ AXIS LEFT BY TFFCONIC
M:(24-NR)
TFFNP E:(-38+2NR) M LC P, SEMI LATUS RECTUM LEFT BY TFFCONIC
M:(-36+2NR)
X1 -NR, NORM COUNT OF RMAG LEFT BY TFFCONIC
X2 -NA, NORM COUNT OF ALFA LEFT BY TFFCONIC
#
DEBRIS: QPRET, PDL+0 ... PDL+1

Page 1379
RAPO = 16D # APOGEE RADIUS M E:(-29) M:(-27)
RPER = 14D # PERIGEE RADIUS M E:(-29) M:(-27)

TFFRP/RA DLOAD DMP
 TFFALFA # ALFA 1/M E:(26-NR) M:(24-NR)
 TFFNP # LC P M E:(-38+2NR) M:(-36+2NR)
SR* DCOMP # ALFA P (-12+NR)
```

```

 DAD 0 -8D,1 # ALFA P (-4)
 # (DCOMP GIVES VALID TP RESULT FOR SQRT)
 # (ABS PROTECTS SQRT IF E IS VERY NEAR 0)

 SQRT DP2(-4)
 DAD # E SQ = (1- P ALFA) (-4)
 TFF1/4

 PUSH BDDV # (1+E) (-2) TO PDL+0
 TFFNP # LCP M E:(-38+2NR) M:(-36+2NR)

 SR* SR* # (DOES SR THEN SL TO AVOID OVFL)
 0,1 # X1=-NR
 0 -7,1 # (EFFECTIVE SL)

 STODL RPER # PERIGEE RADIUS M E:(-29) M:(-27)
 # (1+E) (-2) FROM PDL+0

 DMP BOVB
 TFF1/ALF # E:(-22-2NA) M:(-20-2NA)
 TCDANZIG # CLEAR OVFLND, IF ON.

 BZE SL*
 MAXRA # SET POSMAX IF ALFA=0
 0 -5,2 # -5+NA

 SL* BOV
 0,2
 MAXRA # SET POSMAX IF OVFL.

 BPL # CONTINUE WITH VALID RAPO.

 +3

MAXRA DLOAD # RAPO CALC IS NOT VALID. SET RAPO =
 # POSMAX AS A TAG.

 +3 STORE # APOGEE RADIUS M E:(-29) M:(-27)
DUMPRPRA RVQ

#
#
Page 1380
SUBROUTINE NAME: CALCTPER / CALCTFF DATE: 01.29.67
MOD NO: 0 LOG SECTION: TIME OF FREE FALL
MOD BY: RR BAIRNSFATHER
MOD NO: 1 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 67
MOD NO: 2 MOD BY: RR BAIRNSFATHER DATE: 14 APR 67
MOD BY: 3 MOD BY: RR BAIRNSFATHER DATE: 8 JUL 67 NEAR EARTH M
MOD BY: 4 MOD BY: RR BAIRNSFATHER DATE: 21 NOV 67 ADD VARIABLE
MOD BY: 5 MOD BY: RR BAIRNSFATHER DATE: 21 MAR 68 ACCEPT DIFFER
#
FUNCTIONAL DESCRIPTION: PROGRAM CALCULATES THE FREE-FALL TIME OF FLIGHT FROM PRES
VELOCITY VN TO A RADIUS LENGTH SPECIFIED BY RTERM, SUPPLIED BY THE USER. TH
RN MAY BE ON EITHER SIDE OF THE CONIC, BUT RTERM IS CONSIDERED ON THE INBOUND
THE EQUATIONS ARE:
#
Q2 = -SQRT(RTERM (2-RTERM ALFA) - LCP) (INBOUND SIDE) LEQ +- LCE/S
#

```

```

Q1 = RN.VN / SQRT(MU) LEQ +- LCE/SQRT(ALFA)
#
Z = NUM / DEN LEQ +- 1/SQRT(ALFA)
#
WHERE, IF INBOUND
NUM = RTERM -RN LEQ +- 2 LCE/ALFA
DEN = Q2+Q1 LEQ +- 2 LCE/SQRT(ALFA)
#
AND, IF OUTBOUND
NUM = Q2-Q1 LEQ +- 2 LCE/SQRT(ALFA)
DEN = 2 - ALFA (RTERM + RN). LEQ +- 2 LCE
#
IF ALFA ZZ < 1.0 (FOR ALL CONICS EXCEPT ELLIPSES HAVING ABS(DEL ECC ANOM)
THEN X = ALFA Z Z
AND TFF = (RTERM +RN -2 ZZ T(X)) Z/SQRT(MU)
EXCEPT IF ALFA PNZ, AND IF TFF NEG,
THEN TFF = 2 PI /(ALFA SQRT(ALFA)) + TFF
OR IF ALFA ZZ GEQ 1.0 (FOR ELLIPSES HAVING ABS(DEL ECC ANOM) GEQ 90 DEG)
THEN X = 1/ALFA Z Z
AND TFF = (PI/SQRT(ALFA) -Q2 +Q1 +2(X T(X) -1) /ALFA Z) /ALFA SQRT(MU)
WHERE T(X) IS A POLYNOMIAL APPROXIMATION TO THE SERIES
2 3 2
1/3 - X/5 + X /7 - X /8 ... (X < 1.0)
#
CALLING SEQUENC: TIME TO RTERM TIME TO PERIGEE
CALL CALL
CALCTFF CALCTPER
C(MPAC) = TERMNL RAD M C(MPAC) = PERIGEE RAD M
FOR EITHER, E: (-29) M: (-27)
FOR EITHER, PUSHLOC = PDL+0, ARBITRARY IF LEQ 8D.
Page 1381
#
SUBROUTINES CALLED: T(X), VIA RTB
#
NORMAL EXIT MODE: RVQ
HOWEVER, PROGRAM EXITS WITH ONE OF THE FOLLOWING VALUES FOR TFF (-28) CS IN MPAC. USER
A. TFF = FLIGHT TIME. NORMAL CASE FOR POSITIVE FLIGHT TIME LESS THAN ONE ORBIT
B. (THIS OPTION IS NO LONGER USED.)
C. TFF = POSMAX. THIS INDICATES THAT THE CONIC FROM THE PRESENT POSITION WILL
THE SPECIFIED ALTITUDE. ALSO INDICATES OUTBOUND PARABOLA OR HYPERBOLA.
#
OUTPUT: C(MPAC) (-28) CS TIME OF FLIGHT, OR TIME TO PERIGEE
TFFX (0) X,
NRTERM E: (-29+NR) M RTERM, WEIGHTED BY NR LEFT FO
M: (-27+NR)
TFFTEM E: (-59+2NR) LCP Z Z SGN(SDELF) LEFT FO

```

```

M:(-55+2NR) LCP /ALFA SGN(SDELF)
NOTE: TFFTEM = PDL 36D AND WILL BE DESTROYED BY .:UNIT:.
RMAG1 E:(-29) M:(-27) PDL 12 NOT TOUCHED.
TFFQ1 E:(-16) M:(-15) PDL 14D
TFFDELQ E:(-16) M:(-15) PDL 10D
PUSHLOC AT PDL+0
#
ERASABLE INITIALIZATION REQUIRED:
RONE E:(-29) M:(-27) M STATE VECTOR
VONE' E:(+10) M:(+9) VN/SQRT(NU)
RMAG1 E:(-29) M:(-27) PRESENT RADIUS, M
C(MPAC) E:(-29) M:(-27) RTERM, TERMINAL RADIUS LENGTH, M
#
THE FOLLOWING ARE STORED IN THE PUSH LIST AREA.
TFF/RTMU E:(17) M:(14) 1/SQRT(MU)
NRMAG E:(-29+NR) M RMAG, NORMALIZED
M:(-27+NR)
X1 -NR, NORM COUNT
TFFNP E:(-38+2NR) M LCP, SEMI LATUS RECTUM, WEIGHT NR
M:(-36+2N4)
TFFALFA E:(26-NR) 1/M ALFA, WEIGHT NR
M:(24-NR)
TFFRTALF E:(10+NA) SQRT(ALFA), NORMALIZED
M:(9+NA)
X2 -NA, NORM COUNT
TFF1/ALF E:(-22-2NA) SIGNED SEMIMAJ AXIS, WEIGHTED BY NA
M:(-20-2NA)
#
DEBRIS: QPRET, PDL+0 ... PDL+3
RTERM E:(-29) M(-27) RTERM, TERMINAL RADIUS LENGTH
RAPO E:(-29) M(-27) PDL 16D (=NRTERM)
RPER E:(-29) M(-27) PDL 14D (=TFFQ1)
#
Page 1382
CALCTPER SETGO # ENTER WITH RPER IN MPAC
 TFFSW
 +3
CALCTFF CLEAR # ENTER WITH RTERM IN MPAC
 TFFSW
 +3 STORE RTERM # E:(-29) M:(-27)
 SL*
 0,1 # X1=-NR
 STORE NRTERM # RTERM E:(-29+NR) M:(-27+NR)
 DMP BDSU
 TFFALFA # ALFA E:(26-NR) M:(24-NR)
 TFF1/4

```

July 12, 2016

1871

```

PUSH DMP # (2-ALFA RTERM) (-3) TO PDL+0
 NRTERM # E: (-29+NR) M: (-27+NR)
PDDL SR* # RTERM(2-ALFA RTERM) TO PDL+2
 SR* # E: (-32+NR) M: (-30+NR)
 TFFNP # LC P E: (-38+2NR) M: (-36+2NR)
 0 -6,1 # X1 = -NR
DCOMP DAD # DUE TO SHIFTS, KEEP PRECISION FOR SQRT
 DAD # RTERM(2-ALFA RTERM) FROM PDL +2
 SR* # E: (-32+NR) M: (-30+NR)
 SR* # LEAVE E: (-32) M: (-30)
 0,1 # X1 = -NR
BOFF DLOAD # CHECK TFF /TPER SWITCH
 TFFSW
 +2 # IF TFF, CONTINUE
 TFFZEROS # IF TPER, SET Q2 = 0
+2 BMN # E: (-16) M: (-15)
 SQRT # NO FREE FALL CONIC TO RTERM FROM HERE
 MAXTFF1 # RESET PDL, SET TFF=POSMAX, AND EXIT.

DCOMP BOVB # RT IS ON INBOUND SIDE. ASSURE OVFIN=0
 TCDANZIG # ANY PORT IN A STORM.
STOVL TFFTEM # Q2 E: (-16) M: (-15)
 VONE' # VN/SQRT(MU) E: (10) M: (9)
DOT SL3
 RONE # SAVED RN. E: (-29) M: (-27)
STORE TFFQ1 # Q1, SAVE FOR GONEPAST TEST.
 TFFQ1 # E: (-16) M: (-15)
BMN BDSU
 INBOUND # USE ALTERNATE Z
 TFFTEM # Q2 E: (-16) M: (-15)

OUTBOUND Z CALC CONTINUES HERE

STODL TFFX # NUM=Q2-Q1 E: (-16) M: (-15)
 TFFALFA # ALFA E: (26-NR) M: (24-NR)
DMP BDSU
Page 1383
 NRMAG # RMAG E: (-29+NR) M: (-27+NR)
 NRMAG # (2-RTERM ALFA) (-3) FROM PDL+0
SAVEDEN PUSH ABS # DEN TO PDL+0 E: (-3) OR (-16)
 ABS # M: (-3) OR (-15)
 DAD # INDETERMINANCY TEST
 BOV # =1.0-B(-22)
 LIM(-22) # GO IF DEN >= B(-22)
 TFFXTEST # SET DEN=0 OTHERWISE
DLOAD PDDL
 TFFZEROS
```

```

 # XCH ZERO WITH PDL+0
 # ALFA E:(26-NR) M:(24-NR)
 # FOR TPER: Z INDET AT DELE/2=0 AND 90.
 # ASSUME 90, AND LEAVE 0 IN PDL: 1/Z=D/N

 # Z INDET. AT PERIGEE FOR PARAB OR HYPERB.
 # RETURN TFF =0

DUMPTFF1 RVQ

INBOUND Z CALC CONTINUES HERE

INBOUND DLOAD DCOMP # RESET PDL+0
 DLOAD DSU # ALTERNATE Z CALC
 RTERM # E:(-29) M:(-27)
 RMAG1 # E:(-29) M:(-27)
 STODL TFFX # NUM=RTERM-RN E:(-29) M:(-27)
 TFFTEM # Q2 E:(-16) M:(-15)
 DAD GOTO
 TFFQ1 # Q1 E:(-16) M:(-15)
 SAVEDEN # DEN = Q2+Q1 E:(-16) M:(-15)

TFFXTEST DAD PDDL # (ABS(DEN) TO PDL+2) E:(-3) OR (-16)
 DP(-22) # M:(-3) OR (-15)
 TFFX # RESTORE ABS(DEN) TO MPAC
 SR* # NUM E:(-16) OR (-29) M:(-15) OR (-27)
 TFFRTALF # SQRT(ALFA) E:(10+NA) M:(9+NA)
 0 -3,2 # X2=-NA
 DDV # C(MPAC) =NUM SQRT(ALFA) E:(-3) OR (-16)
 # M:(-3) OR (-15)
 # ABS(DEN) FROM PDL+2 E:(-3) OR (-16)
 # M:(-3) OR (-15)
 DLOAD BOV # (THE DLOAD IS SHARED WITH TFFELL)
 TFFX # NUM E:(-16) OR (-29) M:(-15) OR (-27)
 TFFELL # USE EQN FOR DELE GEQ 90, LEQ -90

OTHERWISE, CONTINUE FOR GENERAL CONIC FOR TFF EQN

 DDV STADR # DEN FROM PDL+0 E:(-3) OR (-16)
 # M:(-3) OR (-15)
 STORE TFFTEM # Z SAVE FOR SIGN OF SDELF.

Page 1384
 # E:(-13) M:(-12)
 PUSH DSQ # Z TO PDL+0
 PUSH DMP # Z SQ TO PDL+2 E:(-26) M:(-24)

```



July 12, 2016

1873

```

 TFFNP # LC P E:(-38+2NR) M:(-36+NR)
SL SIGN
 5
 TFFTEM # AFFIX SIGN FOR SDELF (ENTRY DISPLAY)
STODL TFFTEM # P ZSQ E:(-59+2NR) M:(-55+2NR)
 # (ARG IS USED IN TFF/TRIG)
 # ZSQ FROM PDL+2 E:(-26) M:(-24)
PUSH DMP # RESTORE PUSH LOC
 TFFALFA # ALFA E:(26-NR) M:(24-NR)
SL*
 0,1 # X1=-NR
STORE TFFX # X
RTB DMP
 T(X) # POLY
 # ZSQ FROM PDL+2 E:(-26) M:(-24)
SR2 BDSU # 2 ZSQ T(X) E:(-29) M:(-27)
 RTERM # RTERM E:(-29) M:(-27)
DAD DMP
 RMAG1 # E:(-29) M:(-27)
 # Z FROM PDL+0 E:(-13) M:(-12)
SR3 BPL # TFF SQRT(MU) E:(-45) M:(-42)
 ENDTFF # (NO PUSH UP)
PUSH SIGN # TFF SQRT(MU) TO PDL+0
 TFFQ1 # Q1 FOR GONEPAST TEST
BPL DLOAD # GONE PAST ?
 NEGTTFF # YES. TFF < 0.
 TFF1/ALF # 1/ALFA E:(-22-2NA) M:(-20-2NA)
DCOMP BPL # ALFA > 0 ?
 NEGTTFF # NO. TFF IS NEGATIVE.

CORRECT FOR ORBITAL PERIOD.

DCOMP # YES. CORRECT FOR ORB PERIOD.
DMP DDV
 PI/16 # 2 PI (-5)
 TFFRTALF # SQRT(ALFA) E:(10+NA) M:(9+NA)
SL* SL*
 0 -4,2 # X2=-NA
 0 -4,2
SL* DAD
 0,2
 # TFF SQRT(MU) FROM PDL+0 E:(-45) M:(-42)
ENDTFF DMP BOV # TFF SQRT(MU) IN MPAC E:(-45) M:(-42)
 TFF/RTMU # E:(17) M:(14)
 MAXTFF # SET POSMAX IN OVFL.
```

1874

July 12, 2016

```

DUMPTFF2 RVQ # RETURN TFF (-28) CS IN MPAC.

Page 1385
NEGTTFF DLOAD
 GOTO # TFF SQRT(MU) FROM PDL+0, NEGATIVE.
 ENDTFF

MAXTFF1 DLOAD # RESET PDL
MAXTFF DLOAD RVQ
 NEARONE

TIME OF FLIGHT ELLIPSE WHEN DEL (ECCENTRIC ANOM) GEQ 90 AND LEQ -90.

TFFELL SL2 # NUM FROM TFFX. E: (-16) OR (-29)
 BDDV # M: (-15) OR (-27)
 PUSH # NUM E: (-14) OR (-27) M: (-13) OR (-25)
 # TEMP SAVE D/N IN PDL+0
 # DEN FROM PDL+0 E: (-3)/(-16) M: (-3)/(-15)
 # N/D TO PDL+0 E: (11) M: (10)
TFFEL1 DLOAD # (ENTER WITH D/N=0 IN PDL+0)
 DSU # Q2 E: (-16) M: (-15)
 TFFTEM # Q1 E: (-16) M: (-15)
 TFFQ1 # Q2-Q1 E: (-16) M: (-15)
 STODL # D/N FROM PDL+0
 TFFDELQ
 STADR
 STORE # D/N E: (11) M: (10)
 TFFTEM
 DMP SL*
 TFF1/ALF # 1/ALFA E: (-22-2NA) M: (-20-2NA)
 0,2 # 1/ALFA Z E: (-11-NA) M: (-10-NA)
 PUSH # TO PDL+0
 DMP # 1/Z E: (11) M: (10)
 TFFTEM
 SL* BOVB
 0,2 # X2= -NA
 SIGNMPAC # IN CASE X= 1.0, CONTINUE
 STORE TFFX # X=1/ALFA ZSQ
 RTB DMP
 T(X) # POLY
 TFFX
 SR3 DSU
 DP2(-3)
 DMP PUSH # 2(X T(X)-1) /Z ALFA E: (-15-NA)
 # M: (-14-NA)
 # 1/ALFA Z FROM PDL+0 E: (-11-NA)
 # M: (-10-NA)
 DLOAD DMP # GET SIGN FOR SDELF

```

July 12, 2016

1875

```
Page 1386
 TFFTEM # 1/Z E:(11) M:(10)
 RMAG1 # E:(-29) M:(-27)
SL2 DAD
 TFFQ1 # Q1 E:(-16) M:(-15)
STODL TFFTEM # (Q1+R 1/Z) =SGN OF SDELF E:(-16) M:(-15)
 TFFNP # LC P E:(-38+2NR) M:(-36+2NR)
DMP SL* # CALC FOR ARG FOR TFF/TRIG.

 TFF1/ALF # 1/ALFA E:(-22-2NA) M:(-20-2NA)
 1,2 # X2=-NA
SIGN SL*
 TFFTEM # AFFIX SIGN FOR SDELF
 0,2
STODL TFFTEM # P/ALFA E:(-59+2NR) M:(-55+2NR)
 TFF1/ALF # (ARG FOR USE IN TFF/TRIG)
 DMP # 1/ALFA E:(-22-2NA) M:(-20-2NA)
SQRT PI/16 # PI (-4)
DAD
 # 2(XT(X)-1)/Z ALFA FROM PDL E:(-15-NA)
 # M:(-14-NA)
SL* DSU
 0 -1,2
 TFFDELQ # Q2-Q1 E:(-16) M:(-15)
DMP SL*
 TFF1/ALF # 1/ALFA E:(-22-2NA) M:(-20-2NA)
 0 -3,2
SL* GOTO
 0 -4,2
 ENDTFF # TFF SQRT(MU) IN MPAC E:(-145) M:(-42)
```

# Page 1387

# PROGRAM NAME: T(X)

DATE: 01.17.67

# MOD NO: 0

LOG SECTION: TIME OF FREE FALL

# MOD BY: RR BAIRNSFATHER

#

# FUNCTIONAL DESCRIPTION: THE POLYNOMIAL T(X) IS USED BY TIME OF FLIGHT SUBROUTINES CALCTFF AND

# CALCTPER TO APPROXIMATE THE SERIES

```
2 3
1/3 -X/5 +X /7 -X /9 ...
```

#

```
WHERE X = ALFA Z Z IF ALFA Z Z LEQ 1
X = 1/(ALFA Z Z) IF ALFA Z Z G 1
```

#

# ALSO X IS NEG FOR HYPERBOLIC ORBITS

# X = 0 FOR PARABOLIC ORBITS

```

X IS POSITIVE FOR ELLIPTIC ORBITS
#
FOR FLIGHT 278, THE POLYNOMIAL T(X) IS FITTED OVER THE RANGE (0,+1) AND HAS A
DEVIATION FROM THE SERIES OF 2 E-5. (T(X) IS A CHEBYCHEV TYPE FIT AND WAS OBTAINED BY THE
MAC PROGRAM AUTCURFIT294RRB AND IS VALID TO THE SAME TOLERANCE OVER THE RANGE (0,+1)
#
CALLING SEQUENCE: RTB
T(X)
C(MPAC) = X
#
SUBROUTINE CALLED: NONE
#
NORMAL EXIT MODE: TC DANZIG
#
ALARMS: NONE
#
OUTPUT: C(MPAC) = T(X)
#
ERASABLE INITIALIZATION REQUIRED:
C(MPAC) = X
#
DEBRIS: NONE

T(X) TC POLY
 DEC 4 # N-1
 2DEC 3.333333333 E-1
 2DEC* -1.999819135 E-1*
 2DEC* 1.418148467 E-1*
 2DEC* -1.01310997 E-1*
 2DEC* 5.609004986 E-2*
 2DEC* -1.536156925 E-2*

ENDT(X) TC DANZIG

TCDANZIG = ENDT(X)

Page 1388
TFF CONSTANTS

 BANK 32

 SETLOC TOF-FF1
 BANK

#
#MUE = 3.990815471 E10 # NOTE _ NOTE _ ADJUSTED MUE FOR NEUTRON
M CUBE/CS SQ

```

July 12, 2016

1877

|          |        |                      |                                                |
|----------|--------|----------------------|------------------------------------------------|
| #RTMUE   | =      | 1.997702549 E5 B-18* | # MODIFIED EARTH MU                            |
| 1/RTMU   | 2DEC*  | .5005750271 E-5 B17* | # MODIFIED EARTH MU                            |
| #        |        |                      | # NOTE _ NOTE _ ADJUSTED MUE FOR NEAR EARTH TR |
| #MUM     | =      | 4.902778 E8          | # M CUBE/CS SQ                                 |
| #RTMUM   | 2DEC*  | 2.21422176 E4 B-18*  |                                                |
| PI/16    | 2DEC   | 3.141592653 B-4      |                                                |
| LIM(-22) | 2OCT   | 3777737700           | # 1.0 -B(-22)                                  |
| DP(-22)  | 2OCT   | 0000000100           | # B(-22)                                       |
| DP2(-3)  | 2DEC   | 1 B-3                |                                                |
| DP2(-4)  | 2DEC   | 1 B-4                | # 1/16                                         |
| # RPAD1  | 2DEC   | 6373338 B-29         | # M (-29) = 20909901.57 FT                     |
| RPAD1    | =      | RPAD                 |                                                |
| R300K    | 2DEC   | 6464778 B-29         | # (-29) M                                      |
| NEARONE  | 2DEC   | .999999999           |                                                |
| TFFZEROS | EQUALS | HI6ZEROS             |                                                |
| TFF1/4   | EQUALS | HIDP1/4              |                                                |

This code is written to file `src/TIME-OF-FREE-FALL.s`.

**B.116 TJET LAW**1878  $\langle \text{src}/\text{TJET-LAW.s } 1878 \rangle \equiv$ 

```

Copyright: Public domain.
Filename: TJET_LAW.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1460-1469
Mod history: 2009-05-27 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
2009-06-06 RSB Eliminated a stray instruction that had crept
in somehow.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
#
Page 1460
PROGRAM DESCRIPTION
DESIGNED BY: R. D. GOSS AND P. S. WEISSMAN
CODED BY: P. S. WEISSMAN, 28 FEBRUARY 1968
#
TJETLAW IS CALLED AS A SUBROUTINE WHEN THE LEM IS NOT DOCKED AND THE AUTOPILOT IS IN
ATTITUDE-HOLD MODE TO CALCULATE THE JET-FIRING-TIME (TJET) REQUIRED FOR THE AXIS IN
#
-1 INDICATES THE P-AXIS
+0 INDICATES THE U-AXIS
+1 INDICATES THE V-AXIS
#
THE REGISTERS E AND EDOT CONTAIN THE APPROPRIATE ATTITUDE ERROR AND ERROR RATE AND
UNBALANCED COUPLES ARE PREFERRED. TJETLAW ALSO USES VARIOUS FUNCTIONS OF ACCELERATION
COMPUTED IN THE 1/ACCONT SECTION OF 1/ACCS AND ARE STORED IN SUCH AN ORDER THAT THE

```

July 12, 2016

1879

```
ACCESSED BY INDEXING.
#
THE SIGN OF THE REQUIRED ROTATION IS CARRIED THROUGH TJETLAW AS ROTSENSE AND IS FINALLY APPLIED
PREVIOUS TO ITS STORAGE IN THE LOCATION CORRESPONDING TO THE AXIS (TJP, TJU, OR TJV). THE NUMBER
TJETLAW ASSUMES WILL BE USED AS INDICATED BY THE SETTING OF NUMBERT FOR THE U- OR V-AXIS. TWO
ASSUMED FOR THE P-AXIS ALTHOUGH FOUR JETS WILL BE FIRED WHEN FIREFCT IS MORE NEGATIVE THAN -4
(FIREFCT IS THE DISTANCE TO A SWITCH CURVE IN THE PHASE PLANE) AND A LONG FIRING IS CALLED FOR
#
IN ORDER TO AVOID SCALING DIFFICULTIES, SIMPLE ALGORITHMS TAGGED RUFLAW1, -2 AND -3 ARE RESORTED TO
ERROR AND/OR ERROR RATE ARE LARGE.
#
CALLING SEQUENCE:
TC TJETLAW # (MUST BE IN JASK)
OR
INHINT # (MUST BE IN JASK)
TC IBNKCALL
CADR TJETLAW
RELINT
#
EXIT: RETURN TO Q.
#
INPUT:
FROM THE CALLER: E, EDOT, AXISCTR, SENSETYP, TJP, -U, -V.
FROM 1/ACCONT: 48 ERASABLES BEGINNING AT BLOCKTOP (INCLUDING FLAT, ZONE3LIM AND ACCSWITCH)
#
OUTPUT:
TJP, -U OR -V, NUMBERT (DAPTEMP5), FIREFCT (DAPTEMP3).
#
DEBRIS:
A, L, Q, E, EDOT, DAPTEMP1-6, DAPTEMP1-4.
#
ALARM: NONE

 BANK 17
 SETLOC DAPS2
 BANK
 EBANK= TJP

Page 1461

 COUNT* $$/DAPTJ

TJETLAW EXTEND # SAVE Q FOR RETURN.
 QXCH HOLDQ

SET INDEXERS TO CORRESPOND TO THE AXIS AND TO THE SIGN OF EDOT

 INDEX AXISCTR # AXISDIFF(-1)=NO OF LOCATIONS BET P AND U
```

```

CAF AXISDIFF # AXISDIFF(0)=0
TS ADRSDIF1 # AXISDIFF(+1)=NO OF LOCATIONS BET V AND U

CAE EDOT # IF EDOT NEGATIVE, PICK UP SET OF VALUES
EXTEND # THAT ALLOW USE OF SAME CODING AS FOR
BZMF NEGEDOT # POSITIVE EDOT.
CAE ADRSDIF1 # SET A SECOND INDEXER WHICH MAY BE
TS ADRSDIF2 # MODIFIED BY A DECISION FOR MAX JETS.
CAF SENSOR # FOR POSITIVE EDOT, ROTSENSE IS
TCF SETSENSE # INITIALIZED POSITIVE.

NEGEDOT CS E # IN ORDER FOR NEG EDOT CASE TO USE CODING
TS E # OF POS EDOT, MUST MODIFY AS FOLLOWS:
CS EDOT # 1. COMPLEMENT E AND EDOT.
TS EDOT # 2. SET SENSE OF ROTATION TO NEGATIVE
CAF BIT1 # (REVERSED LATER IF NECESSARY).
ADS ADRSDIF1 # 3. INCREMENT INDEXERS BY ONE SO THAT
TS ADRSDIF2 # THE PROPER PARAMETERS ARE ACCESE
CS SENSOR
SETSENSE TS ROTSENSE

TEST MAGNITUDE OF E (ATTITUDE ERROR, SINGLE-PRECISION, SCALED AT PI RADIANS):
IF GREATER THAN (OR EQUAL TO) PI/16 RADIANS, GO TO THE SIMPLIFIED TJET ROUTIN
IF LESS THAN PI/16 RADIANS, RESCALE TO PI/4

CAE E # PICK UP ATTITUDE ERROR FOR THIS AXIS
EXTEND
MP BIT5 # SHIFT RIGHT TEN BITS: IF A-REGISTER IS
CCS A # ZERO, RESCALE AND TEST EDOT.
TCF RUFLAW2
TCF SCALEE
TCF RUFLAW1
SCALEE CAF BIT13 # ERROR IS IN L SCALED AT PI/16. RESCALE
EXTEND # IT TO PI/4 AND SAVE IT.
MP L
TS E

TEST MAGNITUDE OF EDOT (ERROR RATE SCALED AT PI/4 RADIANS/SECOND)
IF GREATER THAN (OR EQUAL TO) PI/32 RADIANS/SECOND, GO TO THE SIMPLIFIED TJET
IF LESS THAN PI/32 RADIANS/SECOND, THEN RESCALE TO PI/32 RADIANS/SECOND.

CAE EDOT # PICK UP SINGLE-PRECISION ERROR-RATE
Page 1462
EXTEND # FOR THIS AXIS=
MP BIT4 # SHIFT RIGHT ELEVEN BITS, IF THE A-REG IS
EXTEND # ZERO, THEN RESCALE AND USE FINELAW.

```



July 12, 2016

1881

```

 BZF SCALEDOT
 TCF RUFLAW3

*** FINELAW STARTS HERE ***

SCALEDOT LXCH EDOT # EDOT IS SCALED AT PI/32 RADIANS/SECOND.

 CAE EDOT # COMPUTE (EDOT)(EDOT)
 EXTEND
 SQUARE
 EXTEND # PRODUCT SCALED AT PI(2)/2(10) RAD/SEC.
 MP BIT13 # SHIFT RIGHT TWO BITS TO RESCALE TO EDOTSQ
 TS EDOTSQ # TO PI(2)/2(8) RAD(2)/SEC(2).

ERRTEST CCS E # DOES BIG ERROR (THREE DEG BEYOND THE
 AD -3DEG # DEADBAND) REQUIRE MAXIMUM JETS?
 TCF +2
 AD -3DEG
 EXTEND
 INDEX ADRSDIF1
 SU FIREDB
 EXTEND
 BZMF SENSTEST # IF NOT: ARE UNBALANCED JETS PREFERRED?
MAXJETS CAF TWO # IF YES: INCRMENT ADDRESS LOCATOR AND
 ADS ADRSDIF2 # SET SWITCH FOR JET SELECT LOGIC TO 4.
 CAF FOUR # (ALWAYS DO THIS FOR P-AXIS)
 TCF TJCALC

SENSTEST CCS SENSETYP # DOES TRANSLATION PREFER MIN JETS.
 TCF TJCALC # YES. USE MIN-JET PARAMETERS
 TCF MAXJETS # NO. GET THE MAX-JET PARAMETERS.
TJCALC TS NUMBERT # SET TO +0,1,4 FOR (U,V-AXES) JET SELECT.

BEGINNING OF TJET CALCULATIONS:

 CS EDOTSQ # SCALED AT PI(2)/2(8).
 EXTEND
 INDEX ADRSDIF2
 MP 1/ANET1 # .5/ACC SCALED AT 2(6)/PI SEC(2)/RADIAN.
 INDEX ADRSDIF1
 AD FIREDB # DEADBAND SCALED AT PI/4 RADIAN.
 EXTEND
 SU E # ATTITUDE ERROR SCALED AT PI/4 RADIAN.
 TS FIREFCT # -E-.5(EDOTSQ)/ACC-DB AT PI/4 RADIAN.
 EXTEND
 BZMF ZON1,2,3
```

```

ZONE4,5 INDEX ADRSDIF1
CAE 1/ACOST # .5/ACC SCALED AT 2(6)/PI WHERE

Page 1463

EXTEND
MP EDOTSQ # ACC = MAX(AMIN, AOS-).
AD E # SCALED AT PI/2(8).
INDEX ADRSDIF1 # SCALED AT PI/4
AD COASTDB # SCALED AT PI/4 POS. FOR NEG. INTERCEPT.
EXTEND
BZMF ZONE5 # TEST E+.5(EDOTSQ)/ACC+DB AT PI/4 RADIANT.
 # IF FUNCTION NEGATIVE, FIND TJET.
 # IF FUNCTION POSITIVE, IN ZONE 4.

ZONE 4 IS THE COAST REGION. HOWEVER, IF THE JETS ARE ON AND DRIVING TOWARD
A. THE AXIS WITHIN + OR - (DB + FLAT) FOR DRIFTING FLIGHT, OR
B. THE USUAL TARGET PARABOLA FOR POWERED FLIGHT
THEN THE THRUSTERS ARE KEPT ON.

ZONE4 INDEX AXISCTR # IS THE CURRENT VALUE IN TJET NON-ZERO
CS TJETU # WITH SENSE OPPOSITE TO EDOT,
EXTEND
MP ROTSENSE # (I.E., ARE JETS ON AND FIRING TOWARD
EXTEND
BZMF COASTTJ # THE DESIRABLE STATE).
 # NO. COAST.

JETSON CCS FLAT # YES. IS THIS DRIFTING OR POWERED FLIGHT?
TCF DRIFT/ON # DRIFTING. GO MAKE FURTHER TEST.

CS FIREFCT # POWERED (OR ULLAGE). CAN TARGET PARABOLA
INDEX ADRSDIF1 # BE REACHED FROM THIS POINT IN THE
AD AXISDIST # PHASE PLANE?
EXTEND
BZMF COASTTJ # NO. SET TJET = 0.
TC Z123COMP # YES. CALCULATE TJET AS THOUGH IN ZONE 1
CAE FIREFCT # AFTER COMPUTING THE REQUIRED
TCF ZONE1 # PARAMETERS.

DRIFT/ON INDEX ADRSDIF1 # CAN TARGET STRIP OF AXIS BE REACHED FROM
CS FIREDDB # THIS POINT IN THE PHASE PLANE?
DOUBLE
AD FIREFCT
EXTEND
BZMF +3
COASTTJ CAF ZERO # NO. SET TJET = 0.
TCF RETURNJTJ

TC Z123COMP # YES. CALCULATE TJET AS THOUGH IN ZONE 2

```

July 12, 2016

1883

```

 TCF ZONE2,3 # OR 3 AFTER COMPUTING REQUIRED VALUES.

ZONE5 TS L # TEMPORARILY STORE FUNCTION IN L.
 CCS ROTSENSE # MODIFY ADRSDIF2 FOR ACCESSING 1/ANET2
 TCF +4 # AND ACCFCTZ5, WHICH MUST BE PICKED UP
 TC CSHOLE # FROM THE NEXT LOWER REGISTER IF THE
 CS TWO # (ACTUAL) ERROR RATE IS NEGATIVE.

Page 1464

 ADS ADRSDIF2

 +4 CAE L
 EXTEND
 INDEX ADRSDIF2 # TTOAXIS AND HH ARE THE PARAMETERS UPON
 MP ACCFCTZ5 # WHICH THE APPROXIMATIONS TO TJET ARE
 DDOUBL # ABASED.
 DDOUBL
 DXCH HH # DOUBLE PRECISION H SCALED AT 8 SEC(2).
 INDEX ADRSDIF2
 CAE 1/ANET2 # SCALED AT 2(7)/PI SEC(2)/RAD.
 EXTEND
 MP EDOT # SCALED AT PI/2(5)
 TS TTOAXIS # SCALED AT 4 SEC.

TEST WHETHER TJET GREATER THAN 50 MSEC.

 EXTEND
 MP -.05AT2 # H - .05 TTOAXIS - .00125 G.T. ZERO
 AD HH # (SCALED AT 8 SEC(2)).
 AD NEG2
 EXTEND
 BZMF FORMULA1

TEST WHETHER TJET GREATER THAN 150 MSEC.

 CAE TTOAXIS
 EXTEND
 MP -.15AT2 # H - .15 TTOAXIS - .01125 G.T. ZERO
 AD HH # (SCALED AT 8 SEC(2))
 AD -.0112A8
 EXTEND
 BZMF FORMULA2

IF TJET GREATER THAN 150 MSEC, ASSIGN IT VALUE OF 250 MSEC, SINCE THIS
IS ENOUGH TO ASSURE NO SKIP NEXT CSP (100 MSEC).

FULLTIME CAF BIT11 # 250 MSEC SCALED AT 4 SEC.
```

# RETURN TO CALLING PROGRAM WITH JET TIME SCALED AS TIME6 AND SIGNED.

```

RETURN TJ EXTEND # ALL BRANCHES TERMINATE HERE WITH TJET
 MP ROTSENSE # (SCALED AT 4 SEC) IN THE ACCUMULATOR.
 INDEX AXISCTR # ROTSENSE APPLIES SIGN AND CHANGES SCALE.
 TS TJETU
 EXTEND
 INDEX AXISCTR
 MP ACCSWU # SET SWITCH FOR JET SELECT IF ROTATION IS
 CAE L
 EXTEND # IN A SENSE FOR WHICH 1/ACCS HAS FORCE
 BZMF +3 # A MAX-JET CALCULATION.
 CAF FOUR
Page 1465
 TS NUMBERT
 TC HOLDQ # RETURN VIA SAVED Q.

```

# TJET = H/(.025 + TTOAXIS) FOR TJET LESS THAN 50 MSEC.

```

FORMULA1 CS -.025AT4 # .025 SEC SCALED AT 4.
 AD TTOAXIS # SCALED AT 4 SECONDS.
 DXCH HH # STORE DENOMINATOR IN FIRST WORD OF H,
 EXTEND # WHICH NEED NOT BE PRESERVED. PICK UP
 DV HH # DP H AND DIVIDE BY DENOMINATOR.
 EXTEND
 MP BIT14 # RESCALE TJET FROM 2 TO USUAL 4 SEC.
 TCF CHKMINTJ # CHECK THAT TJET IS NOT LESS THAN MINIMUM

```

# TJET = (H + .00375)/(0.1 + TTOAXIS) FOR TJET GREATER THAN 50 MSEC.

```

FORMULA2 EXTEND
 DCA .00375A8 # .00375 SEC(2) SCALED AT 8.
 DAS HH # STORE NUMERATOR IN DP H, WHICH NEED NOT
 CAE TTOAXIS # BE PRESERVED.
 AD .1AT4 # SCALED AT 4 SEC.
 DXCH HH # 0.1 SEC SCALED AT 4.
 EXTEND # STORE DENOMINATOR IN FIRST WORD OF H,
 DV HH # WHICH NEED NOT BE PRESERVED. PICK UP
 EXTEND # DP NUMERATOR AND DIVIDE BY DENOMINATOR
 MP BIT14 # RESCALE TJET FROM 2 TO USUAL 4 SEC.
 TCF RETURN TJ # END SUBROUTINE.

```

# SUBROUTINIZED COMPUTATIONS REQUIRED FOR ALL ENTRIES INTO CODING FOR ZONES 1, 2, AND 3  
# REACHED BY TC FROM 3 POINTS IN TJETLAW.

July 12, 2016

1885

```
Z123COMP CS ROTSENSE # USED IN RETURN TJ SECTION TO RESCALE TJET
 TS ROTSENSE # AS TIME6 AND GIVE IT PROPER SIGN.
 CAE EDOT # SCALED AT PI/2(5) RAD/SEC.
 EXTEND
 INDEX ADRSDIF2
 MP 1/ANET1 # SCALED AT 2(7)/PI SEC(2)/RAD.
 TS TTOAXIS # STORE TIME-TO-AXIS SCALED AT 4 SECONDS.
 AD -TJMAX
 EXTEND
 BZMF +2 # IS TIME TO AXIS LESS THAN 150 MSEC.
 TCF FULLTIME # NO. FIRE JETS, DO NOT CALCULATE TJET.
 RETURN # YES. GO ON TO FIND TJET

ZON1,2,3 TC Z123COMP # SUBROUTINIZED PREPARATION FOR ZONE1,2,3.

IF THE (NEG) DISTANCE BEYOND PARABOLA IS LESS THAN FLAT, USE SPECIAL
LOGIC TO ACQUIRE MINIMUM IMPULSE LIMIT CYCLE. DURING POWERED FLIGHT
Page 1466
OR ULLAGE, FLAT = 0

 CAE FIREFCT # SCALED AT PI/4 RAD.
 AD FLAT
 EXTEND
 BZMF ZONE1 # NOT IN SPECIAL ZONES.

FIRE FOR AXIS OR, IF CLOSE, FIRE MINIMUM IMPULSE. IF ON AXIS, COAST.

ZONE2,3 CS ZONE3LIM # HEIGHT OF MIN-IMPULSE ZONE SET BY 1/ACCS
 AD TTOAXIS # 35 MSEC IN DRIFTING FLIGHT
 EXTEND # ZERO WHEN TRYING TO ENTER GTS CONTROL.
 BZMF ZONE3
ZONE2 CAE TTOAXIS # FIRE TO AXIS.
 TCF RETURN TJ
ZONE3 CCS EDOT # CHECK IF EDOT IS ZERO.
 CAF BIT6 # FIRE A ONE-JET MINIMUM IMPULSE.
 TCF RETURN TJ # TJET = +0.
 TC CCSHOLE # CANNOT BE BECAUSE NEG EDOT COMPLEMENTED.
 TCF RETURN TJ # TJET = +0.

ZONE1 EXTEND
 INDEX ADRSDIF1
 SU AXISDIST # SCALED AT PI/4 RAD.
 EXTEND
 INDEX ADRSDIF2
 MP ACCFCTZ1 # SCALED AT 2(7)/PI SEC(2)/RAD.
```

```

DDOUBL
DDOUBL
DXCH HH # DOUBLE PRECISION H SCALED AT 8 SEC(2).

TEST WHETHER TOTAL TIME REQUIRED GREATER THAN 150 MSEC:
2 2
IS .5(.150 - TTOAXIS) - H NEGATIVE (SCALED AT 8 SECONDS)

 CAE TTOAXIS # TTOAXIS SCALED AT 4 SECONDS.
 AD -TJMAX # -.150 SECOND SCALED AT 4.
 EXTEND
 SQUARE
 EXTEND
 SU HH # HIGH WORD OF H SCALED AT 8 SEC(2).
 EXTEND
 BZMF FULLTIME # YES. NEED NOT CALCULATE TJET.

TEST WHETHER TIME BEYOND AXIS GREATER THAN 50 MSEC TO DETERMINE WHICH APPROXIMATION

 CAE HH
 AD NEG2
 EXTEND
 BZMF FORMULA3

Page 1467
TJET = H/0.1 + TTOAXIS + .0375 FOR APPROXIMATION OVER MORE THAN 50 MSEC.

 CAF .1AT2 # STORE .1 SEC SCALED AT 2 FOR DIVISION.
 DXCH HH # DP H SCALED AT 8 SEC(2) NEED NOT BE
 EXTEND # PRESERVED.
 DV HH # QUOTIENT SCALED AT 4 SECONDS.
 AD TTOAXIS # SCALED AT 4 SEC.
 AD .0375AT4 # .0375 SEC SCALED AT 4.
 TCF RETURN TJ # END COMPUTATION.

TJET - H/.025 + TTOAXIS FOR APPROXIMATION OVER LESS THAN 50 MSEC.

FORMULA3 CS -.025AT2 # STORE +.25 SEC SCALED AT 2 FOR DIVISION
 DXCH HH # PICK UP DP H AT 8, WHICH NEED NOT BE
 EXTEND # PRESERVED.
 DV HH # QUOTIENT SCALED AT 4 SECONDS.
 AD TTOAXIS # SCALED AT 4 SEC.

IF COMPUTED JET TIME IS LESS THAN TJMIN, TJET IS SET TO ZERO.
MINIMUM IMPULSES REQUIRED IN ZONE 3 ARE NOT SUBJECT TO THIS CONSTRAINT, NATURALLY.

```

July 12, 2016

1887

```
CHKMINTJ AD -TJMIN # IS COMPUTED TIME LESS THAN THE MINIMUM.
 EXTEND
 BZMF COASTTJ # YES, SET TIME TO ZERO.
 AD TJMIN # NO, RESTORE COMPUTED TIME.
 TCF RETURN TJ # END COMPUTATION.
```

# Page 1468

# \*\*\* ROUGHLAW \*\*\*

#

# BEFORE ENTRY TO RUFLAW:

# 1. INDEXERS ADRSDIF1 AND ADRSDIF2 ARE SET ON BASIS OF AXIS, AND SIGN OF EDOT.

# 2. IF EDOT WAS NEGATIVE, E AND EDOT ARE ROTATED INTO UPPER HALF-PLANE AND ROTSENSE IS M

# 3. E IS SCALED AT PI RADIANS AND EDOT AT PI/4 RAD/SEC.

# (EXCEPT THE RUFLAW3 ENTRY WHEN E IS AT PI/4)

#

# RUFLAW1: ERROR MORE NEGATIVE THAN PI/16 RAD. FIRE TO A RATE OF 6.5 DEG/SEC (IF JET TIME

# RUFLAW2: ERROR MORE POSITIVE THAN PI/16 RAD. FIRE TO AN OPPOSING RATE OF 6.5 DEG/SEC.

# RUFLAW3: ERROR RATE GREATER THAN PI/32 RAD/SEC AND ERROR WITHIN BOUNDS. COAST IF BELOW

```
RUFLAW1 CS RUFRATE # DECREMENT EDOT BY .1444 RAD/SEC AT PI/4
 ADS EDOT # WHICH IS THE TARGET RATE
 EXTEND
 BZMF SMALRATE # BRANCH IF RATE LESS THAN TARGET.
 TC RUFSETUP # REVERSE ROTSENSE AND INDICATE MAX JETS.
 CAE EDOT # PICK UP DESIRED RATE CHANGE.
```

```
RUFLAW12 EXTEND # COMPUTE TJET
 INDEX ADRSDIF2 # = (DESIRED RATE CHANGE)/(2-JET ACCEL.)
 MP 1/ANET1 +2
 AD -1/8 # IF TJET, SCALED AT 32 SEC, EXCEEDS
 EXTEND # 4 SECONDS, SET TJET TO TJMAX.
 BZMF +2
 TCF FULLTIME
 EXTEND
 BZF FULLTIME
 AD BIT12 # RESTORE COMPUTED TJET TO ACCUMULATOR
 DAS A
 DAS A
 DAS A # RESCALED TJET AT 4 SECONDS.
 TCF CHKMINTJ # RETURN AS FROM FINELAW.
```

```
SMALRATE TC RUFSETUP +2 # SET NUMBERT AND FIREFCT FOR MAXIMUM JETS
 CCS ROTSENSE
 CAF ONE # MODIFY INDEXER TO POINT TO 1/ANET
 TCF +2 # CORRESPONDING TO THE PROPER SENSE.
 CAF NEGONE
```

```

ADS ADRSDIF2

CS EDOT # (.144 AT PI/4 - EDOT) = DESIRED RATE CHNG.
TCF RUFLAW12

RUFLAW2 TC RUFSETUP # REVERSE ROTSENSE AND INDICATE MAX JETS.
CAF RUFRATE
AD EDOT # (.144 AT PI/4 + EDOT) = DESIRED RATE CHNG.
TS A # IF OVERFLOW SKIP, FIRE FOR FULL TIME.
TCF RUFLAW12 # OTHERWISE, COMPUTE JET TIME.
TCF FULLTIME

Page 1469
RUFLAW3 TC RUFSETUP # EXECUTE COMMON RUFLAW SUBROUTINE.
INDEX ADRSDIF1
CS FIREDDB # CALCULATE DISTANCE FROM SWITCH CURVE
AD E # 1/ANET1*EDOT*EDOT +E - FIREDDB = 0
EXTEND # SCALED AT 4 PI RADIANS
MP BIT11
XCH EDOT
EXTEND
SQUARE
EXTEND
INDEX ADRSDIF1
MP 1/ANET1 +2
AD EDOT
EXTEND
BZMF COASTTJ # COAST IF BELOW IT.
TCF FULLTIME # FIRE FOR FULL PERIOD IF ABOVE IT.

SUBROUTINE USED IN ALL ENTRIES TO ROUGHLAW.

RUFSETUP CS ROTSENSE # REVERSE ROTSENSE WHEN ENTER HERE.
 TS ROTSENSE
 +2 CAF FOUR # REQUIRE MAXIMUM (2) JETS IN U,V-AXES.
 TS NUMBERT
 CAF NEGMAX # SUGGEST MAXIMUM (4) JETS IN P-AXIS.
 TS FIREFCT
 TC Q

CONSTANTS FOR TJETLAW

 DEC -16 # AXISDIFF(INDEX) = NUMBER OF REGISTERS
AXISDIFF DEC +0 # BETWEEN STORED 1/ACCS PARAMETERS FOR
 DEC 16 # THE INDEXED AXIS AND THE U-AXIS.
SENSOR OCT 14400 # RATIO OF TJET SCALING WITHIN TJETLAW

```



July 12, 2016

1889

|          |      |            |                                            |
|----------|------|------------|--------------------------------------------|
|          |      |            | # (4 SEC) TO SCALING FOR T6 (10.24 SEC).   |
| -3DEG    | DEC  | -.06667    | # -3.0 DEGREES SCALED AT 45.               |
| -.0112A8 | DEC  | -.00141    | # -.01125 SEC(2) SCALED AT 8.              |
| .1AT4    | DEC  | .025       | # 0.1 SECOND SCALED AT 4.                  |
| .1AT2    | DEC  | .05        | # .1 SEC SCALED AT 2.                      |
| .0375AT4 | DEC  | .00938     | # .0375 SEC SCALED AT 4.                   |
| -.025AT2 | DEC  | -.0125     | # -.025 SEC SCALED AT 2.                   |
| -.025AT4 | DEC  | -.00625    |                                            |
| -.05AT2  | DEC  | -.025      |                                            |
| -.15AT2  | DEC  | -.075      |                                            |
| .00375A8 | 2DEC | .00375 B-3 |                                            |
| -TJMAX   | DEC  | -.0375     | # LARGEST CALCULATED TIME. .150 SEC AT 4.  |
| TJMIN    | DEC  | .005       | # SMALLEST ALLOWABLE TIME. .020 SEC AT 4.  |
| -TJMIN   | DEC  | -.005      |                                            |
| RUFRATE  | DEC  | .1444      | # CORRESPONDS TO TARGET RATE OF 6.5 DEG/S. |

This code is written to file `src/TJET-LAW.s`.

## B.117 TPI SEARCH

```

1890 <src/TPI-SEARCH.s 1890>≡
 # Copyright: Public domain.
 # Filename: TPI_SEARCH.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 551-561
 # Mod history: 2009-05-15 RSB Adapted from the Colossus249/ file of the
 # same name, using Comanche055 page images.
 # 2009-05-20 RSB Corrections: On p. 551, "SETLOC P17S" -> P17S1.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A

 # Page 551
 # TPI SEARCH
 #
 # PROGRAM DESCRIPTION S17.1 AND S17.2
 #
 # FUNCTIONAL DESCRIPTION
 #
 # THE TPI SEARCH ROUTINE DETERMINES THE MINIMUM TOTAL VELOCITY TRANSFER TRAJECTORY FOR
 # MANEUVER TIME WITHIN THE CONSTRAINT OF A SAFE PERICENTER. THIS VELOCITY IS THE SUM
 # FOR THE TPI AND TPF MANEUVERS.
 #
 # THE S17.1 ROUTINE EXTRAPOLATES THE STATE VECTORS OF BOTH VEHICLES TO THE TPI TIME AND
 # RELATIVE PHASE ANGLE BETWEEN THE VEHICLES, THE ALTITUDE DIFFERENCE (I.E., THE MAGNITUDE

```

July 12, 2016

1891

```
POSITION VECTORS) AND SELECTS A SEARCH SECTOR BASED ON THE SIGN OF THE ALTITUDE DIFFERENCE.
#
THE S17.2 ROUTINE FURTHER DEFINES THE SEARCH SECTOR BY COMPUTING ANGULAR LIMITS AND USES THE
SUBROUTINE TO COMPUTE THE SEARCH START AND END TIMES. THE SEARCH IS THEN MADE IN AN ITERATIVE
LAMBERT SUBROUTINE TO COMPUTE THE VELOCITIES REQUIRED AT TPI TIME AND AT TPF TIME. EXIT FROM
IS MADE WHEN SOLUTION CRITERIA ARE MET (NORMAL EXIT) OR AS SOON AS IT IS EVIDENT THAT NO SOLUTION
THE SECTOR SEARCHED.
#
CALLING SEQUENCE
#
BOTH ROUTINES ARE CALLED IN INTERPRETIVE CODE AND RETURN VIA QPRET. S17.1 HAS ONLY A NORMAL
S17.2 RETURNS VIA QPRET FOR NORMAL EXIT AND TO ALARMS FOR ERROR EXIT.
#
SUBROUTINES CALLED
#
CSMCONIC
LEMCONIC
TIMETHET
INITVEL
#
BANK 36
SETLOC P17S1
BANK
#
COUNT 36/TPI
#
EBANK= RACT3
#
***** TEMPORARY *****
#
HPE 2DEC 157420.0 B-29 # EARTH'S MIN. PERICENTER ALTITUDE 85 N.M.
#
HPL 2DEC 10668.0213 B-29 # MOON'S MIN. PERICENTER ALTITUDE 35000 FT.
#
CDSEC 2DEC 40000
#
CLSEC 2DEC 15000
#
PIINVERS 2DEC .3183098862
#
SEC1THET 2DEC .1944444444
#
Page 552
SEC2THET 2DEC .9166666667
#
MANYFEET 2DEC -1.0 B-2
```

1892

July 12, 2016

|         |        |               |                                  |      |
|---------|--------|---------------|----------------------------------|------|
| LIMVEL  | 2DEC   | .6096 E-2 B-7 | # 2FPS                           |      |
| DFTMOON | 2DEC   | .1524 E3 B-29 | # 500 FEET                       |      |
| DP-.002 | 2DEC   | 0.002         |                                  |      |
| S17.1   | SETLOC | P17S          |                                  |      |
|         | BANK   |               |                                  |      |
|         | STQ    | DLOAD         |                                  |      |
|         |        | NORMEX        |                                  |      |
|         |        | TTPI          |                                  |      |
|         | STCALL | TDEC1         | # ADVANCE PASSIVE VEHICLE TO TPI |      |
|         |        | LEMCONIC      |                                  |      |
|         | CALL   |               |                                  |      |
|         |        | LEMSTORE      |                                  |      |
|         | DLOAD  |               |                                  |      |
|         |        | TTPI          |                                  |      |
|         | STCALL | TDEC1         | # ADVANCE ACTIVE VEHICLE TO TPI  |      |
|         |        | CSMCONIC      |                                  |      |
|         | CALL   |               |                                  |      |
|         |        | CSMSTORE      |                                  |      |
|         | VLOAD  |               |                                  |      |
|         |        | RACT3         |                                  |      |
|         | ABVAL  | PDVL          | # /RA/ OD                        | PL 2 |
|         |        | RPASS3        |                                  |      |
|         | UNIT   | PDDL          | # UNIT RP OD                     | PL 6 |
|         | BDSU   | SET           |                                  |      |
|         |        | 36D           | # /RP/ -/RA/                     |      |
|         |        | KFLAG         | # OFF = +                        |      |
|         | BMN    | CLEAR         |                                  |      |
|         |        | +2            |                                  |      |
|         |        | KFLAG         | # ON = -                         |      |
|         | STOVL  | DELHITE       |                                  |      |
|         |        | OD            |                                  |      |
|         | VXV    | UNIT          |                                  |      |
|         |        | VPASS3        |                                  |      |
|         | STOVL  | E2            | # ALMOST IT SAVE FOR 17.2        |      |
|         |        | RACT3         |                                  |      |
|         | PUSH   | VPROJ         |                                  |      |
|         |        | E2            |                                  |      |
|         | VSL2   | BVSU          | # RPA                            |      |
|         | UNIT   | DOT           |                                  |      |
|         |        | OD            |                                  |      |
|         | SL1    | ACOS          |                                  |      |
|         | PDVL   |               |                                  |      |

July 12, 2016

1893

```
Page 553
VXV DOT
 RACT3
 E2
PDDL SIGN
STADR
STODL THETZERO # CENTRAL ANGLE
 X1
STCALL XRS # SAVE INDICES FOR FURTHER USE
 NORMEX # += ACTIVE AHEAD -= ACTIVE BEHIND
S17.2 STQ # COMPUTE SEARCH SECTOR LIMITS
 VLOAD
 QTEMP
 RACT3
UNIT DOT
 E2
ABS SQRT
SL1 DAD
 DP-.002 # ADD .002 RADIANS TO IT
 DCOMP # GIVES CORRECT SINE, COSINE MUST BE
 KFLAG # COMP. ADD .5 FOR ANGLE
 +1
PHI(0)=180-(-(THETAZERO +K5IT)), PHI(I)=180-(-(THETAZERO+K2IT))
SIN(180-ALPHA)=SIN(ALPHA) ETC
DMP SETPD
 PIINVERS # REVOLUTIONARY HERES TWO IT
 OD
PUSH DSU
 THETZERO
STORE IT # PHI(I) , -(THETZERO + K2IT)
PDDL PUSH
SR1 DAD
DAD PUSH # PHI(0) , -(THETZERO + K5IT)
SIN SET
 RVSW
STODL SNTH
COS BMN
 +2
DCOMP
STODL CSTH
 XRS
STOVL X1
 RPASS3
VSR*
 0,2
STOVL RVEC
 VPASS3
```

```

VSR* 0,2
STCALL VVEC
 TIMETHET
DLOAD # SAVE START TIME AND GET END TIME
 T
Page 554
STORE TF
STODL TFO
 IT
PUSH SIN
STODL SNTH
COS BMN
 +2
DCOMP
STORE Csth
LXA,1 CALL
 XRS
 TIMETHET
INITIALIZE LOOP
DLOAD CLEAR
 T
 ITSWICH
STODL TFI # SAVE TIME FOR LOOP TEST
 DPPOS MAX
STODL DELVEE
 MANYFEET
STODL HP
 SEC1THET # 70 DEGREES
BON DLOAD
 KFLAG
 +2
 SEC2THET # 330 DEGREES
STCALL THETL
 CONCAUL
BIS DLOAD SR1
 Csth
STODL COSTH
 SNTH
SR1
STCALL SINTH # GET 4 QUADRANT THETA
 ARCTRIG
BPL DAD
 +2
 DPPOS MAX # PUT THETA BETWEEN 0,1
BDSU PDDL

```

July 12, 2016

1895

```

 THETL
 TF
 DSU SIGN # FAST TIMES
 TFI
 BMN
 RNETEST # TIME MUST HAVE A STOP
ADVANCE PASSIVE FOR TARGET VECTOR
CONCAUL DLOAD
 TTPI
 DAD BON
 TF
Page 555
 AVFLAG
 ADVCSM
 STCALL TDEC1
 LEMCONIC
 GOTO
 JUNCT3
ADVCSM STCALL TDEC1
 CSMCONIC
SAVE BACK VALUES OF HP AND DELVEE
JUNCT3 VLOAD
 VATT
 STOVL VPASS4
 RATT
 STORE RPASS4
 STODL RTARG
 TF
 STODL DELLT4
 HP
 STODL HPO
 DELVEE
 STODL DELVEO
PREPARE FOR LAMBERT
 TTPI
 STODL INTIME
 XRS
 STODL RTX1
 HI6ZEROS
 SETPD PDDL
 OD
 EPSFOUR
 PDVL
 RACT3
 STOVL RINIT
 VACT3
```

```

 STCALL VINIT
 INITVEL
COMPUTE H ET CETERA
 VLOAD VSU
 VTPRIME
 VPASS4
 ABVAL PUSH
 STOVL RELDELV # /V2-VP(TPI+TF)/
 DELVEET3 # V1-VA
 ABVAL # /V1-VA/
 STORE MAGVTPI
 DAD STADR
 STODL DELVEE
 XRS
 STOVL X1
 VIPRIME
Page 556
 VSR*
 0,2
 STOVL VVEC
 RACT3
 VSR*
 0,2
 STCALL RVEC
 PERIAPO
 LXA,2 DLOAD
 XRS +1
 SL*
 0,2
 STORE HP
ITSWICH DENOTES INTERPOLATION -- SOLUTION ACCEPTANCE IS FORCED
 BON DLOAD
 ITSWICH
 ENDEN
 HPERMIN
 DSU BMN
 HP
 HALFSAFE
 PDDL DSU # WAS PERICENTER ALT SAFE
 HPERMIN
 HPO
 BMN DSU # (HPLIM-HPO)-(HPLIM-HP)=HP-HPO
 INTERP # SOLUTION AT HAND
 BMN DLOAD
 ALARUMS # IT'S GETTING WORSE -- SOUND THE AL
 CDSEC

```



July 12, 2016

1897

|            |        |         |      |                              |
|------------|--------|---------|------|------------------------------|
| JUNCT1     | BOFF   | DCOMP   |      | # OFF IS PLUS, ON IS MINUS   |
|            |        | KFLAG   |      |                              |
|            |        | +1      |      |                              |
| JUNCT2     | STORE  | DELTEE  |      |                              |
|            | DLOAD  | DAD     |      |                              |
|            |        | DELTEE  |      |                              |
|            |        | TF      |      |                              |
|            | STCALL | TF      |      |                              |
|            |        | BIS     |      | # RECYCLE                    |
| INTERP     | SET    | DSU     |      | # HP-HPO                     |
|            |        | ITSWICH |      |                              |
|            | NORM   | PDDL    |      |                              |
|            |        | X1      |      |                              |
|            |        | DFTMOON |      |                              |
|            | DAD    | DSU     |      |                              |
|            |        | HPERMIN |      |                              |
|            |        | HP      |      |                              |
|            | NORM   | SR1     |      |                              |
|            |        | X2      |      |                              |
|            | XSU,2  | DDV     |      |                              |
|            |        | X1      |      |                              |
| # Page 557 |        |         |      |                              |
|            | DMP    | SR*     |      |                              |
|            |        | DELTEE  |      |                              |
|            |        | 0       | -1,2 |                              |
|            | STCALL | DELTEE  |      |                              |
|            |        | JUNCT2  |      |                              |
| HALFSAFE   | PDDL   | DSU     |      | # SAVE HP-HPLIM FOR POSSIBLE |
|            |        | DELVEE  |      |                              |
|            |        | DELVEO  |      | # SAVE THIS TOO              |
|            | PUSH   | ABS     |      |                              |
|            | DSU    | BMN     |      |                              |
|            |        | LIMVEL  |      | # 2 FT PS                    |
|            |        | ENDEN   |      |                              |
|            | DLOAD  | DSU     |      |                              |
|            |        | HPERMIN |      |                              |
|            |        | HPO     |      |                              |
|            | PDDL   |         |      |                              |
|            | BMN    | DLOAD   |      |                              |
|            |        | LRGRDVO |      |                              |
|            | BPL    | DLOAD   |      |                              |
|            |        | INTERP  |      |                              |
|            |        | DELTEE  |      |                              |
|            | SR1    | DCOMP   |      |                              |
|            | STCALL | DELTEE  |      |                              |
|            |        | JUNCT2  |      |                              |

1898

July 12, 2016

```

LRGRDVO DLOAD
 BMN DLOAD
 JUNCT2
 CLSEC
 GOTO JUNCT1
TIME RAN OUT ASSUME SOLUTION IF SAVE PERICENTER
RNGETEST DLOAD DSU
 HP
 HPERMIN
 BMN DLOAD
 ALARUMS
 TF
 DSU
 DELTEE
 STORE TF # TIME OF SOLUTION
 VLOAD
 VTPRIME
 DOT PDDL # SG2 WITH MAGNITUDE
 RPASS4
 RELDELV
 SIGN STADR # NOW SIGN(RELDELV)=SIGN(SG2)
 STCALL RELDELV
 TRANSANG # COMPUTE OMEGA T, CENTRAL ANGLE
 VLOAD DOT
 RACT3
Page 558
 VIPRIME # SG1
 SIGN BPL # IF POSITIVE THEN SG1 = SG2 OTHERWISE
 RELDELV
 USEKAY # SIGN(SG2-SG1)=SIGN(SG2)=SIGN(RELDELV)
 SLOAD DCOMP
 DECTWO
 SIGN BPL
 RELDELV
 NEXUS
 DCOMP GOTO
 USEKAY +4
 USEKAY SLOAD BON
 DECTWO
 KFLAG
 NEXUS
 DSU
 P210NENN
 NEXUS STODL NN1
 HP

```

July 12, 2016

1899

STCALL POSTTPI  
QTEMP  
BANK 07  
SETLOC XANG  
BANK  
COUNT 07/XANG

# CENTRAL ANGLE SUBROUTINE

#

# THIS SUBROUTINE COMPUTES THE CENTRAL ANGLE OF TRAVEL OF THE  
# PASSIVE VEHICLE DURING THE TRANSFER.

|            |        |           |      |                        |                  |
|------------|--------|-----------|------|------------------------|------------------|
| TRANSANG   | STQ    | SETPD     |      |                        |                  |
|            |        | SUBEXIT   |      |                        |                  |
|            |        | 0         |      |                        |                  |
|            | LXA,1  | LXA,2     |      |                        |                  |
|            |        | XRS       |      |                        |                  |
|            |        | XRS       | +1   |                        |                  |
|            | VLOAD  | VSR*      |      |                        |                  |
|            |        | VPASS4    |      |                        |                  |
|            |        | 0,2       |      |                        |                  |
|            | STODL* | VVEC      |      |                        |                  |
|            |        | MUTABLE   | +2,1 |                        |                  |
|            | PDVL   | VSR*      |      | # SQRT MU (+18 OR +15) | 00D              |
|            |        | RPASS4    |      |                        |                  |
|            |        | 0,2       |      |                        |                  |
|            | ABVAL  | PDDL*     |      | # MAGNITUDE OF R       | (+29 OR +27) 02D |
|            |        | MUTABLE,1 |      |                        |                  |
|            | PDVL   | VSQ       |      | # 1/MU                 | (+34 OR +28) 04D |
|            |        | VVEC      |      |                        |                  |
|            | NORM   | DMPR      |      | # PUSH LIST AT 02D     |                  |
|            |        | X1        |      |                        |                  |
| # Page 559 |        |           |      |                        |                  |
|            | DMP    | SRR*      |      |                        |                  |
|            |        | 02D       |      |                        |                  |
|            |        | 0         | -3,1 |                        |                  |
|            | BDSU   |           |      | # R V**/MU             | (+6)             |
|            |        | D1/32     |      |                        |                  |
|            | NORM   | PDDL      |      |                        |                  |
|            |        | X1        |      | # (2 - R V**/MU)       | (+6-N)           |
|            | SR1R   | DDV       |      | # MAGNITUDE OF R       | (+30 OR +28)     |
|            | SL*    | PUSH      |      | # R/(2 - R V**/MU)     | (+29 OR +27) 02D |
|            |        | 0         | -5,1 |                        |                  |
|            | SR1    | SQRT      |      |                        |                  |
|            | DMP    |           |      |                        |                  |
|            | NORM   | PDDL      |      | # ASUBP***             | 00D              |

```

 X1
SR1 DDV
DMP SL*
 TF
 0,1
PDDL NORM
 2PISC
 X1
PDDL DDV
SL*
 0 -3,1
STCALL CENTANG
 SUBEXIT
BANK 35
SETLOC P17S1
BANK
COUNT 35/P17

TPI SEARCH DISPLAY ROUTINE

P17 TC AVFLAGA # AVFLAG = CSM, SET TRACK + UPDATE FI
 TC P17.1
P77 TC AVFLAGP # AVFLAG = LEM, SET TRACK + UPDATE FI
P17.1 TC P20FLGON # SET UPDATE FLAG
 CAF V06N37 # DISPLAY TTPI TIME
 TC VNP00H
 TC INTPRET
CLEAR CALL
 UPDATFLG
 S17.1 # UPDATE STATE VECTORS TO TTPI
 SET AXT,1
 UPDATFLG
 DEC 2 # DELTA H = 2 K POSITIVE, KFLAG OFF
BOFF AXT,1
 KFLAG
 +2
 DEC 1 # DELTA H = 1 K NEGATIVE, KFLAG ON

Page 560
SXA,1 EXIT
 OPTION2
CAF V06N72 # DISPLAY PHI, DELTA H, SEARCH OPTION
TC VNCOMP17
TC INTPRET
CLEAR SET
 UPDATFLG
 KFLAG

```

July 12, 2016

1901

|            |        |             |                                            |
|------------|--------|-------------|--------------------------------------------|
|            | SLOAD  | DSU         |                                            |
|            |        | OPTION2     | # RESET KFLAG ON FOR OPTION =1             |
|            |        | P21ONENN    | # OFF FOR OPTION =2                        |
|            | BHIZ   | CLEAR       |                                            |
|            |        | +2          |                                            |
|            |        | KFLAG       |                                            |
|            | SLOAD  | BHIZ        |                                            |
|            |        | XRS +1      |                                            |
|            |        | +4          |                                            |
|            | DLOAD  | GOTO        |                                            |
|            |        | HPL         |                                            |
|            |        | P17.2       |                                            |
|            | DLOAD  |             |                                            |
|            |        | HPE         |                                            |
| P17.2      | STCALL | HPERMIN     |                                            |
|            |        | S17.2       |                                            |
|            | SET    | EXIT        |                                            |
|            |        | UPDATFLG    |                                            |
| P17.3      | CAF    | V06N58      | # DISPLAY DELTA VTPI, DELTA VTPF, AND H    |
|            | TC     | VNCOMP17    |                                            |
|            | CAF    | V06N55      | # DISPLAY PERICENTER CODE AND CENTRAL ANG, |
|            | TC     | BANKCALL    |                                            |
|            | CADR   | GOFLASHR    |                                            |
|            | TC     | GOTOP00H    | # TERMINATE PROGRAM                        |
|            | TC     | GOTOP00H    | # END PROGRAM                              |
|            | TC     | P17.1       | # RECYCLE WITH NEW TTPI OR SEARCH OPTION   |
|            | CAF    | TWO         | # BLANK R2                                 |
|            | TC     | BLANKET     |                                            |
|            | TCF    | ENDOFJOB    |                                            |
|            | EBANK= | RTRN        |                                            |
| VNCOMP17   | EXTEND |             |                                            |
|            | QXCH   | QSAVED      |                                            |
|            | TS     | VERBNOUN    |                                            |
|            | CA     | VERBNOUN    |                                            |
|            | TCR    | BANKCALL    |                                            |
|            | CADR   | GOFLASH     |                                            |
|            | TC     | -3          | # TERMINATE ILLEGAL REDISPLAY              |
|            | TC     | QSAVED      | # PROCEED                                  |
|            | CS     | MPAC        | # RECYCLE WITH NEW TPI TIME                |
|            | AD     | BIT6        | # OR PROCEED WITH NEW SEARCH OPTION        |
|            | EXTEND |             |                                            |
| # Page 561 | BZF    | P17.1       |                                            |
|            | TC     | VNCOMP17 +3 |                                            |
| ALARUMS    | SET    | EXIT        |                                            |

1902

July 12, 2016

|        |     |          |                                     |
|--------|-----|----------|-------------------------------------|
|        |     | UPDATFLG |                                     |
|        | TC  | ALARM    |                                     |
|        | OCT | 00124    | # NO SAFE PERICENTER IN THIS SECTOR |
|        | CAF | V05N09   |                                     |
|        | TC  | VNCOMP17 |                                     |
|        | TC  | GOTOP00H | # PROCEED ILLEGAL TERMINATE PROGRAM |
| V06N72 | VN  | 0672     |                                     |

This code is written to file `src/TPI-SEARCH.s`.

## B.118 TRIM GIMBAL CNTROL SYSTEM

1903

*<src/TRIM-GIMBAL-CNTROL-SYSTEM.s 1903>≡*

```
Copyright: Public domain.
Filename: TRIM_GIMBAL_CNTLROL_SYSTEM.agc
Purpose: Part of the source code for Luminary 1A build 099.
It is part of the source code for the Lunar Module's (LM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1472-1485
Mod history: 2009-05-27 RSB Adapted from the corresponding
Luminary131 file, using page
images from Luminary 1A.
#
This source code has been transcribed or otherwise adapted from
digitized images of a hardcopy from the MIT Museum. The digitization
was performed by Paul Fjeld, and arranged for by Deborah Douglas of
the Museum. Many thanks to both. The images (with suitable reduction
in storage size and consequent reduction in image quality as well) are
available online at www.ibiblio.org/apollo. If for some reason you
find that the images are illegible, contact me at info@sandroid.org
about getting access to the (much) higher-quality images which Paul
actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
16:27 JULY 14, 1969
```

# Page 1472

```
BANK 21
EBANK= QDIFF
SETLOC DAPS4
BANK

COUNT* $$/DAPGT
```

```
CONTROL REACHES THIS POINT UNDER EITHER OF THE FOLLOWING TWO CONDITIONS ONCE THE DESCENT ENGI
AUTOPILOT ARE BOTH ON:
A) THE TRIM GIMBAL CONTROL LAW WAS ON DURING THE PREVIOUS Q,R-AXIS TIME5 INTERRUPT (OR
INITIALIZATION WAS SET FOR TRIM GIMBAL CONTROL AND THIS IS THE FIRST PASS), OR
B) THE Q,R-AXES RCS AUTOPILOT DETERMINED THAT THE VEHICLE WAS ENTERING (OR HAD JUST ENT
ZONE WITH A SMALL OFFSET ANGULAR ACCELERATION.
GTS IS THE ENTRY TO THE GIMBAL TRIM SYSTEM FOR CONTROLLING ATTITUDE ERRORS AND RATES AS WELL
```

```

GTS CAF NEGONE # MAKE THE NEXT PASS THROUGH THE DAP BE
 TS COTROLER # THROUGH RCS CONTROL,
 CAF FOUR # AND ENSURE THAT IT IS NOT A SKIP.
 TS SKIPU
 TS SKIPV

 CAF TWO
 TS INGTS # SET INDICATOR OF GTS CONTROL POSITIVE.
 TS QGIMTIMR # SET TIMERS TO 200 MSEC TO AVOID BOTH
 TS RGIMTIMR # RUNAWAY AND INTERFERENCE BY NULLING.

THE DRIVE SETTING ALGORITHM
#
DEL = SGN(OMEGA + ALPHA*ABS(ALPHA)/(2*K))
#
NEGUSUM = ERROR*K + ALPHA*(DEL*OMEGA + ALPHA / (3*K)) + DEL*K (DEL*OMEGA + A
#
DRIVE = -SGN(NEGUSUM)

 CA SR # SAVE THE SR. SHIFT IT LEFT TO CORRECT
 AD A # FOR THE RIGHT SHIFT DUE TO EDITING.
 TS SAVESR

GTSGO+DN CAF TWO # SET INDEXER FOR R-AXIS CALCULATIONS.
 TCF GOQTRIMG +1

GOQTRIMG CAF ZERO # SET INDEXER FOR Q-AXIS CALCULATIONS
 TS QRCNTR

Page 1473
RSB 2009 -----
Everything between this line and the similar line below was simply filled-in
as-is from Luminary 131, and then verified to assemble to the proper binary
values. This area is blank on the Luminary 099 print-out, as if the
printer ribbon had run out.
 INDEX QRCNTR # AOS SCALED AT PI/2
 CA AOSQ
 EXTEND
 MP BIT2 # RESCALE AOS TO PI/4
 EXTEND
 BZF GTSQAXIS -3 # USE FULL SCALE FOR LARGER AOS ESTIMATES.

 INDEX A
 CS LIMITS # LIMITS +1 CONTAINS NEGMAX.
 XCH L # LIMITS -1 CONTAINS POSMAX.

```



July 12, 2016

1905

```
GTSQAXIS CCS QRCNTR # PICK UP RATE FOR THIS AXIS. RATE CELLS
 INDEX A # USE ADJACENT, NOT SEPARATED. AT PI/4
 CA EDOTQ
 DXCH WCENTRAL

 INDEX QRCNTR # COLLECT K FOR THIS AXIS
 CA KQ
 TS KCENTRAL

 EXTEND
 BZF POSDRIVE +1 # CONTROL AUTHORITY ZERO. AVOID DRIVING
 # ENGINE BELL TO THE STOPS.

 INDEX QRCNTR # QDIFF, RDIFF ARE STORED IN D.P.
 CAE QDIFF

ALGORITHM EXTEND
 MP KCENTRAL # Q(R)DIFF IS THETA (ERROR) SCALED AT PI.
 LXCH K2THETA # FORM K*ERROR AT PI(2)/2(8), IN D.P.
 EXTEND
 MP BIT5 # RESCALE TO 4*PI(2)
 DXCH K2THETA
 EXTEND
 MP BIT5 # FIRST TERM OF NEGUSUM IN K2THETA.
 ADS K2THETA +1 # NO CARRY NEEDED D.P. AT 4*PI(2)

 CS ACENTRAL # FORM ALPHA(2)/(2*K) AT 16*PI, IN D.P.,
 EXTEND # LIMITING QUOTIENT TO AVOID OVERFLOW.
 MP BIT14 # -ALPHA/2 IN A, SCALED AT PI/4
 EXTEND
 MP ACENTRAL # -ALPHA(2)/2 IN A,L, SCALED AT PI(2)/16)
 AD KCENTRAL
 EXTEND
 BZMF HUGEQUOT # K-ALPHA(2)/2 SHOULD BE PNZ FO DIVISION

 EXTEND
 DCS A # ALPHA(2)/2 - K
 AD KCENTRAL

RSB 2009 -----
 EXTEND
 DV KCENTRAL # HIGH ORDER OF QUOTIENT.
 XCH A2CNTRAL
 CA L # SHIFT UP THE REMAINDER.
 LXCH 7 # ZERO LOW-ORDER DIVIDEND.
 EXTEND

Page 1474
 DV KCENTRAL
```

1906

July 12, 2016

|          |        |             |                                           |
|----------|--------|-------------|-------------------------------------------|
|          | XCH    | A2CNTRAL +1 | # QUOTIENT STORED AT 16*PI, D.P.          |
|          | TCF    | HAVEQUOT    |                                           |
| HUGEQUOT | CA     | POSMAX      |                                           |
|          | TS     | L           |                                           |
|          | DXCH   | A2CNTRAL    | # LIMITED QUOTIENT STORED AT 16*PI, D.P.  |
| HAVEQUOT | CA     | WCENTRAL    |                                           |
|          | EXTEND |             |                                           |
|          | MP     | BIT9        | # RESCALE OMEGA AT 16*PI IN D.P.          |
|          | DXCH   | K2CNTRAL    | # LOWER WORD OVERLAYS OMEGA IN WCENTRAL   |
|          | EXTEND |             |                                           |
|          | DCA    | K2CNTRAL    |                                           |
|          | DXCH   | FUNCTION    |                                           |
|          | CA     | ACENTRAL    | # GET ALPHA*ABS(ALPHA)/(2*K)              |
|          | EXTEND |             |                                           |
|          | BZMF   | +4          |                                           |
|          | EXTEND |             |                                           |
|          | DCA    | A2CNTRAL    |                                           |
|          | TCF    | +3          |                                           |
|          | EXTEND |             |                                           |
|          | DCS    | A2CNTRAL    |                                           |
|          | DAS    | FUNCTION    | # OMEGA + ALPHA*ABS(ALPHA)/(2*K) AT 16*PI |
|          | CCS    | FUNCTION    | # DEL = +1 FOR FUNCT1 GREATER THAN ZERO.  |
|          | TCF    | POSFNCT1    | # OTHERWISE DEL = -1                      |
|          | TCF    | +2          |                                           |
|          | TCF    | NEGFNCT1    |                                           |
| POSFNCT1 | CCS    | FUNCTION +1 | # USE LOW ORDER WORD SINCE HIGH IS ZERO   |
|          | CAF    | BIT1        |                                           |
|          | TCF    | +2          |                                           |
| NEGFNCT1 | CS     | BIT1        |                                           |
|          | TS     | DEL         |                                           |
|          | CCS    | DEL         | # REPLACE OMEGA BY DEL*OMEGA              |
|          | TCF    | FUNCT2      | # POSITIVE DEL VALUE. PROCEED.            |
|          | TCF    | DEFUNCT     |                                           |
|          | TCF    | NEGFNCT2    |                                           |
| DEFUNCT  | TS     | K2CNTRAL    |                                           |

July 12, 2016

1907

|             |        |             |                                             |
|-------------|--------|-------------|---------------------------------------------|
|             | TS     | K2CNTRAL +1 |                                             |
|             | TCF    | FUNCT2      |                                             |
| # Page 1475 |        |             |                                             |
| NEG1/3      | DEC    | -.33333     |                                             |
| NEGFUNCT2   | EXTEND |             |                                             |
|             | DCS    | K2CNTRAL    |                                             |
|             | DXCH   | K2CNTRAL    |                                             |
| FUNCT2      | EXTEND |             |                                             |
|             | DCA    | A2CNTRAL    |                                             |
|             | DAS    | K2CNTRAL    | # DEL*OMEGA + ALPHA(2)/(2*K) AT 16*PI, D.P. |
| FUNCT3      | CA     | A2CNTRAL    |                                             |
|             | EXTEND |             |                                             |
|             | MP     | NEG1/3      |                                             |
|             | DXCH   | A2CNTRAL    |                                             |
|             | CA     | L           |                                             |
|             | EXTEND |             |                                             |
|             | MP     | NEG1/3      |                                             |
|             | ADS    | A2CNTRAL +1 |                                             |
|             | TS     | L           |                                             |
|             | TCF    | +2          | # A2CNTRAL NOW CONTAINS -ALPHA(2)/(6*K),    |
|             | ADS    | A2CNTRAL    | # SCALED AT 16*PI, IN D.P.                  |
|             | EXTEND |             |                                             |
|             | DCA    | K2CNTRAL    | # DEL*OMEGA + ALPHA(2)/(3*K) IN A2CNTRAL,   |
|             | DAS    | A2CNTRAL    | # SCALED AT 16*PI, D.P.                     |
|             | CA     | A2CNTRAL    |                                             |
|             | EXTEND |             |                                             |
|             | MP     | ACENTRAL    |                                             |
|             | DAS    | K2THETA     |                                             |
|             | CA     | A2CNTRAL +1 |                                             |
|             | EXTEND |             |                                             |
|             | MP     | ACENTRAL    | # ACENTRAL MAY NOW BE OVERLAID.             |
|             | ADS    | K2THETA +1  |                                             |
|             | TS     | L           |                                             |
|             | TCF    | +2          | # TWO TERMS OF NEGUSUM ACCUMULATED, SO FAR  |
|             | ADS    | K2THETA     | # SCALED AT 4*PI(2), IN D.P.                |
| GETROOT     | CA     | K2CNTRAL    | # K*(DEL*OMEGA + ALPHA(2)/(2*K)) IS THE     |
|             | EXTEND |             | # TERM FOR WHICH A SQUARE ROOT IS NEEDED.   |
|             | MP     | KCENTRAL    | # K AT PI/2(8)                              |
|             | DXCH   | FUNCTION    |                                             |

```

 CA K2CNTRAL +1
 EXTEND
 MP KCENTRAL
 ADS FUNCTION +1
 TS L
 TCF +2
 ADS FUNCTION # DESIRED TERM IN FUNCTION, AT PI(2)/16
Page 1476
 CCS DEL
 TCF RSTOFGTS
 TCF NEGUSUM
 TCF NEGATE
 TCF NEGUSUM

NEGATE EXTEND
 DCS K2CNTRAL
 DXCH K2CNTRAL
 TCF RSTOFGTS

 BANK 16
 EBANK= NEGUQ
 SETLOC DAPS1
 BANK

THE WRCHN12 SUBROUTINE SETS BITS 9,10,11,12 OF CHANNEL 12 ON THE BASIS OF THE CONT
THE NEGATIVES OF THE DESIRED ACCELERATION CHANGES. ACDT+C12 SETS Q(R)ACCDOT TO RE
#
WARNING: ACDT+C12 AND WRCHN12 MUST BE CALLED WITH INTERRUPT INHIBITED.

BGIM OCTAL 07400
CHNL12 EQUALS ITEM6
ACDT+C12 CS NEGUQ
 EXTEND # GIMBAL DRIVE REQUESTS.
 MP ACCDOTQ
 LXCH QACCDOT
 CS NEGUR
 EXTEND
 MP ACCDOTR
 LXCH RACCDOT

 CCS NEGUQ
 CAF BIT10
 TCF +2
 CAF BIT9
 TS CHNL12

```

July 12, 2016

1909

```
CCS NEGUR
CAF BIT12
TCF +2
CAF BIT11
ADS CHNL12 # (STORED RESULT NOT USED AT PRESENT)
```

```
CS BGIM
EXTEND
RAND CHAN12
AD CHNL12
EXTEND
WRITE CHAN12
```

# Page 1477

```
CS CALLGMBL # TURN OFF REQUEST FOR ACDT+C12 EXECUTION.
MASK RCSFLAGS
TS RCSFLAGS
```

```
TC Q # RETURN TO CALLER.
```

```
BANK 21
EBANK= QDIFF
SETLOC DAPS4
BANK
```

# Page 1478

# SUBROUTINE TIMEGMBL: MOD 0, OCTOBER 1967, CRAIG WORK

#

# TIMEGMBL COMPUTES THE DRIVE TIME NEEDED FOR THE TRIM GIMBAL TO POSITION THE DESCENT ENGINE NO  
# THE OFFSET ANGULAR ACCELERATION ABOUT THE Q (OR R) AXIS. INSTEAD OF USING AOSQ(R), TIMEGMBL  
# SCALED AT PI/8. FOR EACH AXIS, THE DRIVE TIME IS COMPUTED AS ABS(ALPHA/ACCDOT). A ZERO  
# ALPHA OR ACCDOT OR A ZERO QUOTIENT TURNS OFF THE GIMBAL DRIVE IMMEDIATELY. OTHERWISE, THE GI  
# DRIVING IN THE CORRECT DIRECTION. THE Q(R)GIMTIMR IS SET TO TERMINATE THE DRIVE AND Q(R)ACCD  
# IS STORED TO REFLECT THE NEW ACCELERATION DERIVATIVE. NEGUQ(R) WILL CONTAIN +1,+0,-1 FOR A Q  
# WHICH IS NEGATIVE, ZERO, OR POSITIVE.

#

# INPUTS: AOSQ,AOSR, SCALED AT P1/2, AND ACCDOTQ, ACCDOTR AT PI/2(7). PI/2(7).

#

# OUTPUTS: NEW GIMBAL DRIVE BITS IN CHANNEL 12, NEGUQ, NEGUR, QACCDOT, AND RACCDOT, THE LA  
# Q(R)GIMTIMR WILL BE SET TO TIME AND TERMINATE GIMBAL DRIVE(S).

#

# DEBRIS: A, L, Q, ITEMPS 2, 3, 6, AND RUPTREG2 AND ACDT+C12 DEBRIS.

#

# EXITS: VIA TC Q.

#

# ALARMS, ABORTS: NONE.

#

```

SUBROUTINES: ACDT+C12, IBNKCALL
#
WARNING: THIS SUBROUTINE WRITES INTO CHANNEL 12 AND USES THE ITEMS. THEREFOR
INTERRUPT INHIBITED.
#
ERASABLE STORAGE CONFIGURATION (NEEDED BY THE INDEXING METHODS):
NEGUQ ERASE +2 # NEGATIVE OF Q-AXIS GIMBAL DRIVE
(SPWORD) EQUALS NEGUQ +1 # ANY S.P. ERASABLE NUMBER, NOW THRST
NEGUR EQUALS NEGUQ +2 # NEGATIVE OF R-AXIS GIMBAL DRIVE
ACCDOTQ ERASE +2 # Q-JERK TERM SCALED AT PI/2(7) RAD/S
(SPWORD) EQUALS ACCDOTQ +1 # ANY S.P. ERASABLE NUMBER NOW QACCDOT
ACCDOTR EQUALS ACCDOTQ +2 # R-JERK TERM SCALED AT PI/2(7) RAD/S
ACCDOTQ, ACCDOTR ARE MAGNITUDES.
AOSQ ERASE +4 # Q-AXIS ACC., D.P. AT PI/2 R/SEC(2)
AOSR EQUALS AOSQ +2 # R-AXIS ACCELERATION SCALED AT PI/2

QRNDXER EQUALS ITEMP6
OCT23146 OCTAL 23146 # DECIMAL .6
NZACCDOT EQUALS ITEMP3

TIMEGMBL CAF ONE # INITIALZE ALLOWGTS.
 TS ALLOWGTS

 CAF TWO
 LXCH Q # SET UP LOOP FOR R AXIS.
 LXCH RUPTREG2 # SAVE RETURN ADDRESS.

Page 1479

TIMQGMBL TCF +2
 CAF ZERO # NOW DO THE Q-AXIS
 TS QRNDXER
 INDEX QRNDXER
 CA ACCDOTQ # ACCDOT IS PRESUMED TO BE AT PI/2(7)
 EXTEND
 BZMF TGOFFNOW # IS ACCDOT LESS THAN OR EQUAL TO 0?
 TS NZACCDOT # NO. STORE NON-ZERO, POSITIVE ACCDOT

ALPHATRY INDEX QRNDXER
 CS AOSQ
 EXTEND
 BZF TGOFFNOW # IS ALPHA ZERO?

 TS Q # SAVE A COPY OF -AOS.
 EXTEND # NO. RESCALE FOR TIMEGMBL USE.
 MP OCT23146 # OCTAL 23146 IS DECIMAL .6
 AD Q # -1.6*AOS AT PI/2 = -.4*AOS AT PI/8
 TS L # WAS THERE OVERFLOW?

```

July 12, 2016

1911

|             |        |            |                                         |
|-------------|--------|------------|-----------------------------------------|
|             | TCF    | SETNEGU    | # NO. COMPUTE DRIVE TIME.               |
|             | CS     | A          | # RECOVER -SGN(AOS) IN THE A REGISTER.  |
|             | INDEX  | QRNDXER    | # YES. START DRIVE WITHOUT WAITLIST.    |
|             | XCH    | NEGUQ      |                                         |
|             | TCF    | NOTALLOW   | # KNOCK DOWN THE ALLOWGTS FLAG.         |
| SETNEGU     | EXTEND |            |                                         |
|             | BZMF   | POSALPH    |                                         |
|             | COM    |            |                                         |
|             | TS     | ITEMP2     | # STORE -ABS(.4*AOS) SCALED AT PI/8.    |
|             | CS     | BIT1       |                                         |
|             | TCF    | POSALPH +2 |                                         |
| POSALPH     | TS     | ITEMP2     | # STORE -ABS(.4*AOS) SCALED AT PI/8.    |
|             | CA     | BIT1       |                                         |
| +2          | INDEX  | QRNDXER    | # SGN(AOS) INTO NEGU                    |
|             | TS     | NEGUQ      | # STORE SGN(ALPHA) AS NEGU              |
|             | CA     | NZACCDOT   |                                         |
|             | EXTEND |            |                                         |
|             | MP     | BIT12      | # 2*ACCDOT, SCALED AT PI/8.             |
|             | AD     | ITEMP2     | # -ABS(ALPHS) + 2*ACCDOT, AT PI/8.      |
|             | EXTEND |            |                                         |
|             | BZMF   | NOTALLOW   | # IS DRIVE TIME MORE THAN TWO SECONDS?  |
|             | CS     | ITEMP2     | # NO. COMPUTE DRIVE TIME.               |
|             | EXTEND |            | # ABS(ALPHA) AT PI/8.                   |
|             | MP     | OCT00240   | # DECIMAL 10/1024                       |
|             | EXTEND |            | # QUOTIENT IS DRIVE TIME AT WAITLIST.   |
|             | DV     | NZACCDOT   | # ABS(ALPHA)/ACCDOT AT 2(14)/100        |
| # Page 1480 | EXTEND |            |                                         |
|             | BZF    | TGOFFNOW   | # DRIVE TIME MUST BE GREATER THAN ZERO. |
|             | TCF    | DRIVEON    |                                         |
| TGOFFNOW    | CAF    | ZERO       | # TURN OFF GIMBAL NOW.                  |
|             | INDEX  | QRNDXER    |                                         |
|             | TS     | NEGUQ      |                                         |
|             | TCF    | DONEYET    |                                         |
| NOTALLOW    | CAF    | OCT31      |                                         |
|             | INDEX  | QRNDXER    |                                         |
|             | TS     | QGIMTIMR   |                                         |
|             | CAF    | ZERO       | # DRIVE TIME IS MORE THAN 2 SECONDS, SO |

1912

July 12, 2016

|          |       |          |                                       |
|----------|-------|----------|---------------------------------------|
|          | TS    | ALLOWGTS | # DO NOT PERMIT FURTHER GTS ATTITUDE  |
|          | TCF   | DONEYET  | # CONTROL UNTIL AOSTASK APPROVES.     |
|          |       |          | # NO WAITLIST CALL IS MADE.           |
| DRIVEON  | INDEX | QRNDXER  |                                       |
|          | TS    | QGIMTIMR | # CHOOSE Q OR R AXIS.                 |
| DONEYET  | CCS   | QRNDXER  |                                       |
|          | TCF   | TIMQGMBL |                                       |
|          | DXCH  | RUPTREG3 | # PROTECT IBNKCALL ERASABLES. ACDT+C  |
|          | DXCH  | ITEMP2   | # LEAVES ITEMS2,3 ALONE.              |
|          | TC    | IBNKCALL | # TURN OFF CHANNEL BITS, SET Q(R)ACCD |
|          | CADR  | ACDT+C12 |                                       |
|          | DXCH  | ITEMP2   | # RESTORE ERASABLES FOR IBNKCALL.     |
|          | DXCH  | RUPTREG3 |                                       |
|          | TC    | RUPTREG2 | # RETURN TO CALLER.                   |
| OCT00240 | OCTAL | 00240    | # DECIMAL 10/1024                     |

# Page 1481

# THE FOLLOWING SECTION IS A CONTINUATION OF THE TRIM GIMBAL CONTROL FROM THE LAST G

# IS COMPUTED FOR EACH AXIS (Q,R),  $.707 * \Delta * \text{FUNCTION}(3/2) + K2\theta = \text{NEGUSUM}$ . NEW I

#

# THE SUBROUTINE GTSQRT ACCEPTS A DOUBLE PRECISION VALUE IN FUNCTION, FUNCTION +1 AND

# SQUARE ROOT OF THE FOURTEEN MOST SIGNIFICANT BITS OF THE ARGUMENT. ALSO, THE CELL

# EXPONENT S, SUCH THAT THE SQUARE ROOT (RETURNED IN THE A REGISTER) MUST BE SHIFTED

# POWER (-S)) IN ORDER TO BE THE TRUE SQUARE ROOT OF THE FOURTEEN MOST SIGNIFICANT B

# SQUARE ROOT ERROR IS NOT MORE THAN 2 IN THE 14TH SIGNIFICANT BIT. CELLS CLOBBE

# HALFARG, SCRATCH, SR, FUNCTION, FUNCTION +1. GTSQRT IS CALLED BY TC GTSQRT AND RE

# ZERO OR NEGATIVE ARGUMENTS YIELD ZERO FOR SQUARE ROOTS.

|        |        |             |                                           |
|--------|--------|-------------|-------------------------------------------|
| GTSQRT | CCS    | FUNCTION    |                                           |
|        | TCF    | GOODARG     | # FUNCTION IS POSITIVE. TAKE SQUARE ROOT. |
|        | TCF    | +2          | # HIGH ORDER WORD IS ZERO. TRY THE LOWER. |
|        | TCF    | ZEROOT      | # NEGATIVE. USE ZERO FOR 1/2 POWER.       |
|        | CA     | FUNCTION +1 |                                           |
|        | EXTEND |             |                                           |
|        | BZMF   | ZEROOT      |                                           |
|        | TCF    | ZEROHIGH    | # PROCEED.                                |
| ZEROOT | CA     | ZERO        |                                           |



July 12, 2016

1913

|             |        |             |                                            |
|-------------|--------|-------------|--------------------------------------------|
|             | TS     | SHFTFLAG    |                                            |
|             | TC     | Q           |                                            |
| ZEROHIGH    | XCH    | FUNCTION    | # 14 MOST SIGNIFICANT BITS ARE IN THE      |
|             | XCH    | FUNCTION +1 | # LOWER WORD. EXCHANGE THEM.               |
|             | CA     | SEVEN       |                                            |
|             | TCF    | GOODARG +1  |                                            |
| GOODARG     | CA     | ZERO        |                                            |
|             | TS     | SHFTFLAG    |                                            |
|             | CA     | TWELVE      | # INITIALIZE THE SCALING LOOP.             |
|             | TS     | ININDEX     |                                            |
|             | TCF    | SCALLOOP    |                                            |
| SCALSTRT    | CA     | FUNCTION    |                                            |
|             | TCF    | SCALDONE    |                                            |
| MULBUSH     | CA     | NEG2        | # IF ARG IS NOT LESS THAN 1/4, INDEX IS    |
|             | ADS    | ININDEX     | # ZERO, INDICATING NO SHIFT NEEDED.        |
|             | EXTEND |             | # BRANCH IF ARG IS NOT LESS THAN 1/4.      |
|             | BZMF   | SCALSTRT    | # OTHERWISE COMPARE ARG WITH A REFERENCE   |
|             |        |             | # WHICH IS 4 TIMES LARGER THAN THE LAST.   |
| SCALLOOP    | CS     | FUNCTION    |                                            |
|             | INDEX  | ININDEX     |                                            |
|             | AD     | BIT15       | # REFERENCE MAGNITUDE LESS OR EQUAL TO 1/4 |
|             | EXTEND |             |                                            |
|             | BZMF   | MULBUSH     | # IF ARG IS NOT LESS THAN REFERENCE, GO    |
|             |        |             | # AROUND THE MULBERRY BUSH ONCE MORE.      |
| # Page 1482 |        |             |                                            |
|             | INDEX  | ININDEX     |                                            |
|             | CA     | BIT15       | # THIS IS THE SCALE MAGNITUDE              |
|             | XCH    | HALFARG     | # 2*(-ININDEX) IS THE SHIFT DIVISOR.       |
|             | EXTEND |             | # RESCALE ARGUMENT.                        |
|             | DCA    | FUNCTION    |                                            |
|             | EXTEND |             |                                            |
|             | DV     | HALFARG     |                                            |
|             |        |             | # ININDEX AND SHFTFLAG PRESERVE INFO FOR   |
|             |        |             | # RESCALING AFTER ROOT PROCESS.            |
| SCALDONE    | EXTEND |             |                                            |
|             | QXCH   | FUNCTION +1 | # SAVE Q FOR RETURN                        |
|             | EXTEND |             |                                            |
|             | MP     | BIT14       |                                            |
|             | TS     | HALFARG     |                                            |
|             | MASK   | BIT13       |                                            |
|             | CCS    | A           |                                            |
|             | CA     | OCT11276    |                                            |

```

AD ROOTHALF # INITIAL GUESS IS ROOT 1/2 OR POSMAX
TC ROOTCYCL
TC ROOTCYCL
TC ROOTCYCL
TC FUNCTION +1

```

```

```

```

RSTOFGTS TC GTSQRT
PRODUCT XCH K2CNTRAL
 EXTEND
 MP K2CNTRAL
 DXCH K2CNTRAL
 EXTEND
 MP L # THE PRODUCT OF
 ADS K2CNTRAL +1 # 1/2 2 1/2
 TS L # K *(DEL*OMEGA + ALPHA /(2*K))
 TCF +2 # AND
 ADS K2CNTRAL # 2
 # DEL*(DEL*OMEGA + ALPHA /(2*K)) NOW IN
 # K2CNTRAL

DOSHIFT CA ININDEX
 EXTEND
 MP BIT14 # MULTIPLY IN THE FACTOR 2(-S), RETURNED
 ADS SHFTFLAG # BY THE GTSQRT SUBROUTINE
 EXTEND
 BZF ADDITIN
 INDEX SHFTFLAG
 CA BIT15

Page 1483

 XCH K2CNTRAL
 EXTEND
 MP K2CNTRAL
 DAS K2THETA
 XCH K2CNTRAL
 EXTEND
 MP K2CNTRAL +1
 ADS K2THETA +1
 TS L
 TCF +2
 ADS K2THETA

 TCF NEGUSUM

ADDITIN EXTEND
 DCA K2CNTRAL

```

July 12, 2016

1915

|             |        |            |                                           |
|-------------|--------|------------|-------------------------------------------|
| NEGUSUM     | DAS    | K2THETA    | # NO ADD IN THE K2THETA TERM.             |
|             | CCS    | K2THETA    | # TEST SIGN OF HIGH ORDER PART.           |
|             | TCF    | NEGDRIVE   |                                           |
|             | TCF    | +2         |                                           |
|             | TCF    | POSDRIVE   |                                           |
| NEGDRIVE    | CCS    | K2THETA +1 | # SIGN TEST FOR LOW ORDER PART.           |
|             | CA     | BIT1       |                                           |
| POSDRIVE    | TCF    | +2         | # STOP GIMBAL DRIVE FOR A ZERO NEGUSUM.   |
|             | CS     | BIT1       |                                           |
|             | TS     | L          | # SAVE FOR DRIVE REVERSAL TEST.           |
|             | INDEX  | QRCNTR     |                                           |
|             | XCH    | NEGUQ      |                                           |
|             | EXTEND |            |                                           |
|             | MP     | L          | # MULTIPLY OLD NEGU AND NEW NEGU.         |
|             | CCS    | L          |                                           |
|             | TCF    | LOUPE      | # NON-ZERO GIMBAL DRIVE BEING CONTINUED.  |
|             | TCF    | ZEROLOUP   | # NO REVERSAL PROBLEM HERE.               |
|             | TCF    | REVERSAL   | # NON-ZERO GIMBAL DRIVE BEING REVERSED.   |
|             | TCF    | ZEROLOUP   | # NO REVERSAL PROBLEM HERE.               |
| REVERSAL    | INDEX  | QRCNTR     | # A ZERO-DRIVE PAUSE IS NEEDED HERE. ZERO |
|             | TS     | QACCDOT    | # IS IN A REGISTER FROM CCS ON (-1).      |
|             | INDEX  | QRCNTR     |                                           |
|             | CS     | GMBLBITA   |                                           |
|             | EXTEND |            |                                           |
|             | WAND   | CHAN12     |                                           |
| ZEROLOUP    | CS     | RCSFLAGS   | # SET UP REQUEST FOR ACDT+C12 CALL.       |
|             | MASK   | CALLGMBL   |                                           |
|             | ADS    | RCSFLAGS   |                                           |
| # Page 1484 |        |            |                                           |
| LOUPE       | CCS    | QRCNTR     | # HAVE BOTH AXES BEEN PROCESSED?          |
|             | TCF    | GOQTRIMG   | # NO. DO Q AXIS NEXT.                     |
|             | CA     | SAVESR     | # RESTORE THE SR                          |
|             | TS     | SR         |                                           |
| GOCLOSE     | EXTEND |            | # TERMINATE THE JASK.                     |
|             | DCA    | CLOSEADR   |                                           |
|             | DTCB   |            |                                           |
|             | EBANK= | AOSQ       |                                           |

```

CLOSEADR 2CADR CLOSEOUT # TERMINATE THE JASK.

TWELVE EQUALS OCT14
ROOTHALF OCTAL 26501 # SQUARE ROOT OF 1/2
GMBLBITA OCTAL 01400 # INDEXED WRT GMBLBITB DO NOT MOVE *****
OCT11276 OCTAL 11276 # POSMAX -- ROOTHALF
GMBLBITB OCTAL 06000 # INDEXED WRT GMBLBITA DO NOT MOVE *****

SUBROUTINE ROOTCYCL: BY CRAIG WORK, 3 APRIL 68
#
ROOTCYCL IS A SUBROUTINE WHICH EXECUTS ONE NEWTON SQUARE ALGORITHM ITERATION. THE
SQUARE ROOT IS PRESUMED TO BE IN THE A REGISTER AND ONE-HALF THE SQUARE IS TAKEN FR
TO THE SQUARE ROOT IS RETURNED IN THE A REGISTER. DEBRIS: A, L, SR, SCRATCH. RO
LOCATION (LOC) BY A TC ROOTCYCL, AND RETURNS (TC Q) TO LOC +1.
#
WARNING: IF THE INITIAL GUESS IS NOT GREATER THAN THE SQUARE, DIVIDE OR ADD OVERFI

ROOTCYCL TS SCRATCH # STORE X
 TS SR # X/2 NOW IN SR
 CA HALFARG # ARG/2 IN THE A REG
 ZL
 EXTEND
 DV SCRATCH # (ARG/X)/2
 AD SR # (X + ARG/X)/2 IN THE A REG
 TC Q

```

This code is written to file `src/TRIM-GIMBAL-CNTROL-SYSTEM.s`.

**B.119 TVCDAPS**1917 *<src/TVCDAPS.s 1917>≡*

```

Copyright: Public domain.
Filename: TVCDAPS.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 961-978
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
2009-05-20 RSB Corrections: Eliminated an extraneous EXTEND,
added a missing instruction to PFORWARD.
2000-05-21 RSB Wrong opcode was used with DELBRTMP and
DELBRTMP +1 operands in 4 places. Corrected
an MP operation in 2CASFLTR.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 961
PROGRAM NAME....TVCDAP, CONSISTING OF PITCHDAP, YAWDAP, ETC.
LOG SECTION....TVCDAPS SUBROUTINE....DAPCSM
MODIFIED BY SCHLUNDT 21 OCTOBER 1968
#
FUNCTIONAL DESCRIPTION
#
SELF-PERPETUATING T5 TASKS WHICH GENERATE THE COMMAND SIGNALS
FOR THE PITCH AND YAW SPS GIMBAL ACTUATORS DURING TVC (SPS) BURNS,

```

```

IN RESPONSE TO BODY-AXIS RATE COMMANDS FROM CROSS-PRODUCT STEERING
(S40.8). IF NO STEERING (IMPULSIVE BURNS) MAINTAINS ATTITUDE-HOLD
ABOUT THE REFERENCE (INITIAL) DIRECTIONS (ZERO RATE COMMANDS).
#
THE PITCH AND YAW LOOPS ARE SEPARATE, BUT STRUCTURED IDENTICALLY.
EACH ATTITUDE-RATE LOOP INCLUDES GIMBAL ANGLE RATE DERIVATION,
GIMBAL/BODY AXIS TRANSFORMATION, BODY-AXIS ATTITUDE ERROR
INTEGRATION WITH ERROR LIMITING, THE GENERALIZED 6TH-ORDER FILTERS
FOR CSM OR CSM/LM OPERATION. A FILTER OUTPUT LIMITER.
CG-OFFSET TRACKER FILTER, AND THE CG-TRACKER MINOR LOOP.
#
THE DAPS ARE CYCLIC, CALLING EACH OTHER AT 1/2 THE DAP SAMPLE
TIME, AS DETERMINED BY T5TVCDT. THE ACTUATOR COMMANDS ARE
REGENERATED AS ANALOG VOLTAGES BY THE OPTICS ERROR COUNTERS, WHICH
TRANSMIT THE SIGNAL TO THE ACTUATOR SERVOS WHEN THERE IS PROPER CDU
MODING.
#
CALLING SEQUENCE.... (TYPICALLY)
#
T5 CALL OF TVCDAPON (TVCINITIALIZE) BY DOTVCON (P40)
T5 CALL OF DAPINIT (TVCDAPS) BY TVCINIT4 (TVCINITIALIZE)
T5 CALL OF PITCHDAP BY DAPINIT
T5 CALL OF YAWDAP BY PITCHDAP
T5 CALL OF PITCHDAP BY YAWDAP
ETC.
(AUTOMATIC SEQUENCING FROM TVCDAPON)
#
NORMAL EXIT MODE....RESUME
#
ALARM OR ABORT EXIT MODES....NONE
#
SUBROUTINES CALLED....
#
HACK FOR STROKE TEST (V68) WAVEFORM GENERATION
PCOPY, YCOPY FOR COPY-CYCLES (USED ALSO BY TVC RESTART PACKAGE)
DAPINIT FOR INITIAL CDUS FOR RATE MEASUREMENTS
ERRORLIM, ACTLIM FOR INPUT (ATTITUDE-ERROR INTEGRATION) AND
OUTPUT (ACTUATOR COMMAND) LIMITING, COMMON TO PITCH AND
YAW DAPS
FWDFLTR (INCLUDING OPTVARK) AND PRECOMP, TO COMPUTE FILTER
OUTPUTS AND STORAGE VALUES
RESUME
#
Page 962
OTHER INTERFACES
#

```

July 12, 2016

1919

```
S40.8 CROSS-PRODUCT STEERING FOR BODY AXIS RATE COMMANDS OMEGAY,ZC
S40.15 FOR THE INITIAL DAP GAINS VARK AND 1/CONACC
TVCEXECUTIVE FOR DAP GAIN UPDATES AND TMC LOOP OPERATIONS
TVCRESTART PACKAGE FOR TVC RESTART PROTECTION.
#
ERASABLE INITIALIZATION REQUIRED....
#
PAD-LOAD ERASABLES (SEE ERASABLE ASSIGNMENTS)
CONFIGURATION BITS (14, 13) OF DAPDATR1 AS IN R03
ENGINE-ON BIT (11.13) FOR RESTARTS
TVCPHASE FOR RESTARTS (SEE DOTVCON, AND TVCINIT4)
T5 BITS (15,14 OF FLAGWRD6) FOR RESTARTS
MISCELLANEOUS VARIABLES SET UP OR COMPUTED BY TVCDAPON...TVCINIT4,
INCLUDING THE ZEROING OF TEMPORARIES BY MRCLEAN
CDUX,Y,Z AND SINCDEX... COSCDUX AS PREPARED BY QUICTRIG1 (WITH
UPDATES EVERY 1/2 SECOND)
ALSO G+N PRIMARY, TVC ENABLE, AND OPTICS ERROR COUNTER ENABLE
UNLESS BENCH-TESTING.
#
OUTPUT....
#
TVCPITCH AND TVCYAW WITH COUNTER RELEASE (11.14 AND 11.13 INCREMENTAL
COMMANDS TO OPTICS ERROR COUNTERS), FILTER NODES, BODY-
AXIS ATTITUDE ERROR INTEGRATOR, TOTAL ACTUATOR COMMANDS,
OFFSET-TRACKER-FILTER OUTPUTS, ETC.
#
DEBRIS....
#
MUCH, SHAREABLE WITH RCS/ENTRY, IN EBANK6 ONLY
#
BANK 17
SETLOC DAPS2
BANK
#
EBANK= BZERO
#
COUNT* $$/DAPS
#
Page 963
PITCH TVCDAP STARTS HERE....(INCOPORATES CSM/LEM DAP FILTER, MODOR DESIGN)
#
PITCHDAP LXCH BANKRUPT # T5 ENTRY, NORMAL OR VIA DAPINIT
EXTEND
QXCH QRUPT
#
CAF YAWT5 # SET UP T5 CALL FOR YAW AUTOPILOT (LOW-
```

|            |        |           |   |                                          |
|------------|--------|-----------|---|------------------------------------------|
|            | TS     | T5LOC     | # | ORDER PART OF 2CADR ALREADY THERE)       |
|            | CAE    | T5TVCDT   |   |                                          |
|            | TS     | TIME5     |   |                                          |
| PSTROKER   | CCS    | STROKER   | # | (STRKFLG) CHECK FOR STROKE TEST          |
|            | TC     | HACK      | # | TEST-START OR TEST-IN-PROGRESS           |
|            | TCF    | +2        | # | NO-TEST                                  |
|            | TC     | HACK      | # | TEST-IN-PROGRESS                         |
| PCDUDOTS   | CAE    | CDUY      | # | COMPUTE CDUYDOT (USED BY PITCH AND YAW)  |
|            | XCH    | PCDUYPST  |   |                                          |
|            | EXTEND |           |   |                                          |
|            | MSU    | PCDUYPST  |   |                                          |
|            | TCR    | RLIMTEST  | # | RATE TEST                                |
|            | TS     | MCDUYDOT  | # | (MINUS, SC.AT 1/2TVCDT REVS/SEC)         |
|            | CAE    | CDUZ      | # | COMPUTE CDUZDOT (USED BY PITCH AND YAW)  |
|            | XCH    | PCDUZPST  |   |                                          |
|            | EXTEND |           |   |                                          |
|            | MSU    | PCDUZPST  |   |                                          |
|            | TCR    | RLIMTEST  | # | RATE TEST                                |
|            | TS     | MCDUZDOT  | # | (MINUS, SC.AT 1/2TVCDT REVS/SEC)         |
|            | TCF    | PINTEGRAL |   |                                          |
| RLIMTEST   | TS     | TTMP1     | # | TEST FOR EXCESSIVE CDU RATES (GREATER    |
|            | EXTEND |           | # | THAN 2.33 DEG IN ONE SAMPLE PERIOD       |
|            | MP     | 1/RTLIM   |   |                                          |
|            | EXTEND |           |   |                                          |
|            | BZF    | +3        |   |                                          |
|            | CAF    | ZERO      |   |                                          |
|            | TS     | TTMP1     |   |                                          |
|            | CAE    | TTMP1     |   |                                          |
|            | TC     | Q         |   |                                          |
| PINTEGRAL  | EXTEND |           | # | COMPUTE INTEGRAL OF BODY-AXIS PITCH-RATE |
|            | DCA    | PERRB     | # | ERROR, SC.AT B-1 REVS                    |
|            | DXCH   | ERRBTMP   |   |                                          |
|            | EXTEND |           |   |                                          |
|            | DCA    | OMEGAYC   |   |                                          |
|            | DAS    | ERRBTMP   |   |                                          |
| # Page 964 | CS     | COSCDUZ   | # | PREPARE BODY-AXIS PITCH RATE, OMEGAYB    |
|            | EXTEND |           |   |                                          |
|            | MP     | COSCDUX   |   |                                          |



July 12, 2016

1921

|          |        |          |                                            |
|----------|--------|----------|--------------------------------------------|
|          | DDOUBL |          |                                            |
|          | EXTEND |          |                                            |
|          | MP     | MCDUYDOT |                                            |
|          | DDOUBL |          |                                            |
|          | DXCH   | OMEGAYB  |                                            |
|          | CS     | MCDUZDOT |                                            |
|          | EXTEND |          |                                            |
|          | MP     | SINCDEX  |                                            |
|          | DDOUBL |          |                                            |
|          | DAS    | OMEGAYB  | # (COMPLETED OMEGAYB, SC.AT 1/2TVCDT REVS) |
|          | EXTEND |          | # PICK UP -OMEGAYB (SIGN CHNG, INTEGRATE)  |
|          | DCS    | OMEGAYB  |                                            |
|          | DAS    | ERRBTMP  |                                            |
| PERORLIM | TCR    | ERRORLIM | # PITCH BODY-AXIS-ERROR INPUT LIMITER      |
| PFORWARD | EXTEND |          | # PREPARE THE FILTER STORAGE LOCATIONS     |
|          | DCA    | PTMP1    | # FOR THE PITCH CHANNEL                    |
|          | DXCH   | TMP1     |                                            |
|          | EXTEND |          |                                            |
|          | DCA    | PTMP3    |                                            |
|          | DXCH   | TMP3     |                                            |
|          | EXTEND |          |                                            |
|          | DCA    | PTMP5    |                                            |
|          | DXCH   | TMP5     |                                            |
|          | TCR    | FWDFLTR  | # GO COMPUTE PRESENT OUTPUT                |
|          |        |          | # (INCLUDES VARIABLE GAIN PACKAGE)         |
| POFFSET  | EXTEND |          |                                            |
|          | DCA    | PDELOFF  |                                            |
|          | DAS    | CMDTMP   | # NO SCALED AT B+0 ASCREV                  |
| PACLIM   | TCR    | ACTLIM   | # ROUND OFF & LIMIT PITCH ACTUATOR COMMAND |
| POUT     | CS     | PCMD     | # INCREMENTAL PITCH COMMAND                |
|          | AD     | CMDTMP   |                                            |
|          | ADS    | TVCPITCH | # UPDATE THE ERROR COUNTER (NO RESTART-    |
|          |        |          | # PROTECT. SINCE ERROR CNTR ZEROED)        |
|          | CAF    | BIT11    | # BIT FOR TVCPITCH COUNT RELEASE           |
|          | EXTEND |          |                                            |
|          | WOR    | CHAN14   |                                            |

```

PPRECOMP EXTEND # PREPARE THE FILTER STORAGE FOR PITCH
Page 965

 DCA PTMP2
 DXCH TMP2
 EXTEND
 DCA PTMP4
 DXCH TMP4
 EXTEND
 DCA PTMP6
 DXCH TMP6

 TCR PRECOMP # TO THE FILTER FOR PRECOMPUTATION

DELBARP CAE DELPBAR +1
 EXTEND
 MP E(-AT)
 TS DELBRTMP +1
 CAE DELPBAR
 EXTEND
 MP E(-AT)
 DAS DELBRTMP
 CAE CMDTMP
 EXTEND
 MP 1-E(-AT)
 DAS DELBRTMP

PCOPYCYC TCR PCOPY # PITCH COPYCYCLE

PDAPEND TCF RESUME # PITCH DAP COMPLETED
Page 966
PITCH TVCDAP COPYCYCLE SUBROUTINE (CALLED VIA PITCH TVCDAP OR TVC RESTART PACKAGE)

PCOPY INCR TVCPHASE # RESTART-PROTECT THE COPYCYCLE. (1)
 # NOTE POSSIBLE RE-ENTRY FROM RESTART
 # PACKAGE, SHOULD A RESTART OCCUR
 # DURING PITCH COPYCYCLE.

 EXTEND
 DCA TMP1
 DXCH PTMP1
 EXTEND
 DCA TMP2
 DXCH PTMP2
 EXTEND
 DCA TMP3
 DXCH PTMP3

```

July 12, 2016

1923

```
EXTEND
DCA TMP4
DXCH PTMP4
EXTEND
DCA TMP5
DXCH PTMP5
EXTEND
DCA TMP6
DXCH PTMP6

PMISC EXTEND # MISC...PITCH-RATE-ERROR INTEGRATOR
DCA ERRBTMP
TS AK1 # FOR PITCH NEEDLES, SC.AT B-1 REVS
DXCH PERRB

CAE CMDTMP # PITCH ACTUATOR COMMAND
TS PCMD

EXTEND # PITCH OFFSET-TRACKER-FILTER
DCA DELBRTMP
DXCH DELPBAR

INCR TVCPHASE # PITCH COPYCYCLE COMPLETED (2)

TC Q

Page 967
YAW TVCDAP STARTS HERE...(INCORPORATES CSM/LEM DAP FILTER, MODOR DESIGN)

YAWDAP LXCH BANKRUPT # T5 ENTRY, NORMAL
EXTEND
QXCH QRUP

CAF PITCHT5 # SET UP T5 CALL FOR PITCH AUTOPILOT (LOW-
TS T5LOC # ORDER PART OF 2CADR ALREADY THERE)
CAE T5TVCDT
TS TIME5

YSTROKER CCS STROKER # (STRKFLG) CHECK FOR STROKE TEST
TC HACK # TEST-START OR TEST-IN-PROGRESS
TCF +2 # NO-TEST
TC HACK # TEST-IN-PROGRESS

USE BODY RATES FROM PITCHDAP (PCDUDOTS)

YINTEGRL EXTEND # COMPUTE INTEGRAL OF BODY-AXIS YAW-RATE
```

1924

July 12, 2016

|            |        |          |   |                                          |
|------------|--------|----------|---|------------------------------------------|
|            | DCA    | YERRB    | # | ERROR, SC.AT B-1 REVS                    |
|            | DXCH   | ERRBTMP  |   |                                          |
|            | EXTEND |          |   |                                          |
|            | DCA    | OMEGAZC  |   |                                          |
|            | DAS    | ERRBTMP  |   |                                          |
|            | CAE    | COSCDUZ  | # | PREPARE BODY-AXIS YAW-RATE, OMEGAZB      |
|            | EXTEND |          |   |                                          |
|            | MP     | SINCDEX  |   |                                          |
|            | DDOUBL |          |   |                                          |
|            | EXTEND |          |   |                                          |
|            | MP     | MCDUYDOT |   |                                          |
|            | DDOUBL |          |   |                                          |
|            | DXCH   | OMEGAZB  |   |                                          |
|            | CS     | MCDUZDOT |   |                                          |
|            | EXTEND |          |   |                                          |
|            | MP     | COSCDUX  |   |                                          |
|            | DDOUBL |          |   |                                          |
|            | DAS    | OMEGAZB  | # | (COMPLETED OMEGAZB, SC.AT 1/2TVCDT REVS) |
|            | EXTEND |          | # | PICK UP -OMEGAZB (SIGN CHNG, INTEGRATE)  |
|            | DCS    | OMEGAZB  |   |                                          |
|            | DAS    | ERRBTMP  |   |                                          |
| YERORLIM   | TCR    | ERRORLIM | # | YAW BODY-AXIS-ERROR INPUT LIMITER        |
| YFORWARD   | EXTEND |          | # | PREPARE THE FILTER STORAGE LOCATIONS     |
|            | DCA    | YTMP1    | # | FOR THE YAW CHANNEL                      |
| # Page 968 | DXCH   | TMP1     |   |                                          |
|            | EXTEND |          |   |                                          |
|            | DCA    | YTMP3    |   |                                          |
|            | DXCH   | TMP3     |   |                                          |
|            | EXTEND |          |   |                                          |
|            | DCA    | YTMP5    |   |                                          |
|            | DXCH   | TMP5     |   |                                          |
|            | TCR    | FWDFLTR  | # | GO COMPUTE PRESENT OUTPUT                |
|            |        |          | # | (INCLUDES VARIABLE GAIN PACKAGE)         |
| YOFFSET    | EXTEND |          |   |                                          |
|            | DCA    | YDELOFF  |   |                                          |
|            | DAS    | CMDTMP   | # | NOW SCALED AT B+0 ASCREV                 |

July 12, 2016

1925

|                                                                              |        |             |                                                                                |
|------------------------------------------------------------------------------|--------|-------------|--------------------------------------------------------------------------------|
| YACLIM                                                                       | TCR    | ACTLIM      | # YAW ACTUATOR-COMMAND-LIMITER                                                 |
| YOUT                                                                         | CS     | YCMD        | # INCRMENTAL YAW COMMAND                                                       |
|                                                                              | AD     | CMDTMP      |                                                                                |
|                                                                              | ADS    | TVCYAW      | # UPDATE THE ERROR COUNTER (NO RESTART-<br># PROTECT, SINCE ERROR CNTR ZEROED) |
|                                                                              | CAF    | BIT12       | # BIT FOR TVCYAW COUNT RELEASE                                                 |
|                                                                              | EXTEND |             |                                                                                |
|                                                                              | WOR    | CHAN14      |                                                                                |
| YPRECOMP                                                                     | EXTEND |             | # PREPARE THE FILTER STORAGE FOR YAW                                           |
|                                                                              | DCA    | YTMP2       |                                                                                |
|                                                                              | DXCH   | TMP2        |                                                                                |
|                                                                              | EXTEND |             |                                                                                |
|                                                                              | DCA    | YTMP4       |                                                                                |
|                                                                              | DXCH   | TMP4        |                                                                                |
|                                                                              | EXTEND |             |                                                                                |
|                                                                              | DCA    | YTMP6       |                                                                                |
|                                                                              | DXCH   | TMP6        |                                                                                |
|                                                                              | TCR    | PRECOMP     | # TO THE FILTER FOR PRECOMPUTATION                                             |
| DELBAR                                                                       | CAE    | DELYBAR +1  | # UPDATE YAW OFFSET-TRACKER-FILTER                                             |
|                                                                              | EXTEND |             |                                                                                |
|                                                                              | MP     | E(-AT)      |                                                                                |
|                                                                              | TS     | DELBRTMP +1 |                                                                                |
|                                                                              | CAE    | DELYBAR     |                                                                                |
|                                                                              | EXTEND |             |                                                                                |
|                                                                              | MP     | E(-AT)      |                                                                                |
|                                                                              | DAS    | DELBRTMP    |                                                                                |
|                                                                              | CAE    | CMDTMP      |                                                                                |
|                                                                              | EXTEND |             |                                                                                |
|                                                                              | MP     | 1-E(-AT)    |                                                                                |
|                                                                              | DAS    | DELBRTMP    |                                                                                |
| # Page 969                                                                   |        |             |                                                                                |
| YCOPYCYC                                                                     | TCR    | YCOPY       | # YAW COPYCYCLE                                                                |
| YDAPEND                                                                      | TCF    | RESUME      | # YAW DAP COMPLETED                                                            |
| # Page 970                                                                   |        |             |                                                                                |
| # TVCDAP COPYCYCLE SUBROUTINE (CALLED VIA YAW TVCDAP OR TVC RESTART PACKAGE) |        |             |                                                                                |
| YCOPY                                                                        | INCR   | TVCPHASE    | # RESTART-PROTECT THE COPYCYCLE. (3)                                           |
|                                                                              |        |             | # NOTE POSSIBLE RE-ENTRY FROM RESTART                                          |

```
PACKAGE, SHOULD A RESTART OCCUR
DURING YAW COPYCYCLE.
```

```
EXTEND
DCA TMP1
DXCH YTMP1
EXTEND
DCA TMP2
DXCH YTMP2
EXTEND
DCA TMP3
DXCH YTMP3
EXTEND
DCA TMP4
DXCH YTMP4
EXTEND
DCA TMP5
DXCH YTMP5
EXTEND
DCA TMP6
DXCH YTMP6
```

```
YMISC EXTEND # MISC...YAW-RATE-ERROR INTEGRATOR
DCA ERRBTMP
TS AK2 # FOR YAW NEEDLES, SC.AT B-1 REVS
DXCH YERRB

CAE CMDTMP
TS YCMD

EXTEND
DCA DELBRTMP
DXCH DELYBAR

CAF ZERO # YAW COPYCYCLE COMPLETED
TS TVCPHASE # RESET TVCPHASE

TC Q
```

```
Page 971
```

```
SUBROUTINES COMMON TO BOTH PITCH AND YAW DAPS....
```

```
INITIALIZATION PACKAGE FOR CDURATES....
```

```
DAPINIT LXCH BANKRUPT # T5 RUPT ENTRY (CALLED BY TVCINT4)
```

July 12, 2016

1927

```
CAF NEGONE # SET UP
AD T5TVCDT # T5 CALL FOR PITCHDAP IN TVCDT SECS
AD NEGMAX # (T5TVCDT = POSMAX - TVCDT/2 +1)
AD T5TVCDT
TS TIME5
CAF PITCHT5 # (BBCON ALREADY THERE)
TS T5LOC

CAE CDUY # READ AND STORE CDUS FOR DIFFERENTIATOR
TS PCDUYPST # PAST-VALUES
CAE CDUZ
TS PCDUZPST

TCF NOQRSM
```

# BODY-AXIS-ERROR INPUT LIMITER PACKAGE....

```
ERRORLIM CAE ERRBTMP # CHECK FOR INPUT-ERROR LIMIT
 EXTEND # CHECKS UPPER WORD ONLY
 MP 1/ERRLIM
 EXTEND
 BZF +6
 CCS ERRBTMP
 CAF ERRLIM
 TCF +2
 CS ERRLIM
 TS ERRBTMP # LIMIT WRITES OVER UPPER WORD ONLY

 TC Q
```

# ACTUATOR-COMMAND LIMITER PACKAGE....

```
ACTLIM CAE CMDTMP +1 # ROUND UP FOR OUTPUT
 DOUBLE
 TS L
 CAF ZERO
 AD CMDTMP

 EXTEND # CHECK FOR ACTUATOR COMMAND LIMIT
 MP 1/ACTSAT
 EXTEND
```

# Page 972

```
BZF +6
CCS CMDTMP # APPLY LIMITS
CAF ACTSAT
TCF +2
```

```

CS ACTSAT
TS CMDTMP # LIMITS WRITE OVER CMDTMP

TC Q

FILTER COMPUTATIONS FOR PRESENT OUTPUT.....

FWDFLTR CAF ZERO
 TS DAP1
 TS DAP2
 TS DAP3
 TS CMDTMP

 TS DELBRTMP

1DAPCAS CAE ERRBTMP +1 # FIRST DAP CASCADE
 EXTEND
 MP N10 # N10
 TS DAP1 +1
 CA ERRBTMP
 EXTEND
 MP N10 # N10
 DAS DAP1
 DXCH TMP1
 DAS DAP1

2DAPCAS CAE DAP1 +1 # SECOND DAP CASCADE
 EXTEND
 MP N10 +5 # N20
 TS DAP2 +1
 CA DAP1
 EXTEND
 MP N10 +5 # N20
 DAS DAP2
 DXCH TMP3
 DAS DAP2

 CAE DAPDATR1 # TEST FOR LEM ON OR OFF
 MASK BIT14
 CCS A
 TCF 3DAPCAS # LEM ON
 EXTEND # LEM OFF
 DCA DAP2
 DXCH DAP3
 TCF OPTVARK

```



# Page 973

3DAPCAS

```

CAE DAP2 +1 # THIRD DAP CASCADE
EXTEND
MP N10 +10D # N30
TS DAP3 +1
CA DAP2
EXTEND
MP N10 +10D # N30
DAS DAP3
DXCH TMP5
DAS DAP3

```

OPTVARK

```

CS DAP3 +1 # VARIABLE GAIN PACKAGE
EXTEND # (ALSO, SIGN CHANGE IN FORWARD LOOP)
MP VARK # SCALED AT 1/(8 ASCREV) OF ACTUAL VALUE
TS CMDTMP +1
CS DAP3
EXTEND
MP VARK
DAS CMDTMP

DXCH CMDTMP # FIX UP SCALING -- SCALED B+3 ASCREVS
DDOUBL
DDOUBL
DXCH CMDTMP # -- SCALED B+1 ASCREVS
NOTE -- THERE IS AN INHERENT GAIN OF
(B+1 ASCREVS) ON THE OUTPUT DACs.

```

TC Q

# FILTER PRECOMPUTATIONS FOR NEXT PASS.....

PRECOMP

```

CAF ZERO # **** FIRST CASCADE FILTER *****
TS TTMP1
TS TTMP2

CA ERRBTMP +1 # MULTIPLY INPUT BY
EXTEND
MP N10 +1 # N11/2
TS TTMP1 +1
CA ERRBTMP
EXTEND
MP N10 +1 # N11/2
DAS TTMP1

CS DAP1 +1 # MULTIPLY OUTPUT BY

```

```

EXTEND
MP N10 +3 # D11/2
TS TTMP2 +1
CS DAP1

Page 974

EXTEND
MP N10 +3 # D11/2
DAS TTMP2

DXCH TTMP2
DAS TTMP1
DXCH TTMP1
DDOUBL
DAS TMP2

DXCH TMP2
DXCH TMP1

CAF ZERO
TS TTMP1
TS TMP2

CA ERRBTMP +1 # MULTIPLY INPUT BY
EXTEND # SECOND-ORDER NUMERATOR COEFF.
MP N10 +2 # N12
TS TTMP1 +1
CA ERRBTMP
EXTEND
MP N10 +2 # N12
DAS TTMP1

CS DAP1 +1 # MULTIPLY OUTPUT BY
EXTEND
MP N10 +4 # D12
TS TMP2 +1
CS DAP1
EXTEND
MP N10 +4 # D12
DAS TMP2

DXCH TTMP1
DAS TMP2

2CASFLTR CAF ZERO # **** SECOND CASCADE FILTER ****
 TS TTMP1

```

# Page 975

|        |       |     |   |                               |
|--------|-------|-----|---|-------------------------------|
| TS     | TTMP2 |     |   |                               |
| CA     | DAP1  | +1  | # | MULTIPLY INPUT BY             |
| EXTEND |       |     |   |                               |
| MP     | N10   | +6  | # | N21/2                         |
| TS     | TTMP1 | +1  |   |                               |
| CA     | DAP1  |     |   |                               |
| EXTEND |       |     |   |                               |
| MP     | N10   | +6  | # | N21/2                         |
| DAS    | TTMP1 |     |   |                               |
| CS     | DAP2  | +1  | # | MULTIPLY OUTPUT BY            |
| EXTEND |       |     |   |                               |
| MP     | N10   | +8D | # | D21/2                         |
| TS     | TTMP2 | +1  |   |                               |
| CS     | DAP2  |     |   |                               |
| EXTEND |       |     |   |                               |
| MP     | N10   | +8D | # | D21/2                         |
| DAS    | TTMP2 |     |   |                               |
| DXCH   | TTMP2 |     |   |                               |
| DAS    | TTMP1 |     |   |                               |
| DXCH   | TTMP1 |     |   |                               |
| DDOUBL |       |     |   |                               |
| DAS    | TMP4  |     |   |                               |
| DXCH   | TMP4  |     |   |                               |
| DXCH   | TMP3  |     |   |                               |
| CAF    | ZERO  |     |   |                               |
| TS     | TTMP1 |     |   |                               |
| TS     | TMP4  |     |   |                               |
| CA     | DAP1  | +1  | # | MULTIPLY INPUT BY             |
| EXTEND |       |     | # | SECOND-ORDER NUMERATOR COEFF. |
| MP     | N10   | +7  | # | N22                           |
| TS     | TTMP1 | +1  |   |                               |
| CA     | DAP1  |     |   |                               |
| EXTEND |       |     |   |                               |
| MP     | N10   | +7  | # | N22                           |
| DAS    | TTMP1 |     |   |                               |
| CS     | DAP2  | +1  | # | MULTIPLY OUTPUT BY            |
| EXTEND |       |     |   |                               |
| MP     | N10   | +9D | # | D22                           |

```

TS TMP4 +1
CS DAP2
EXTEND
MP N10 +9D # D22
DAS TMP4

DXCH TTMP1
DAS TMP4

CAE DAPDATR1 # TEST FOR LEM ON OR OFF
MASK BIT13
CCS A
TC Q # EXIT IF LEM OFF

Page 976
3CASFLTR CAF ZERO # **** THIRD CASCADE FILTER ****
TS TTMP1
TS TTMP2

CA DAP2 +1 # MULTIPLY INPUT BY (1/2)
EXTEND
MP N10 +11D # N31/2
TS TTMP1 +1
CA DAP2
EXTEND
MP N10 +11D # N31/2
DAS TTMP1

CS DAP3 +1
EXTEND
MP N10 +13D # D31/2
TS TTMP2 +1
CS DAP3
EXTEND
MP N10 +13D # D31/2
DAS TTMP2

DXCH TTMP2
DAS TTMP1
DXCH TTMP1
DDOUBL
DAS TMP6

DXCH TMP6
DXCH TMP5

```

July 12, 2016

1933

```
CAF ZERO
TS TTMP1
TS TMP6

CA DAP2 +1 # MULTIPLY INPUT BY
EXTEND
MP N10 +12D # N32
TS TTMP1 +1
CA DAP2
EXTEND
MP N10 +12D # N32
DAS TTMP1

CS DAP3 +1
EXTEND
MP N10 +14D # D32
TS TMP6 +1
CS DAP3
EXTEND

MP N10 +14D # D32
DAS TMP6

DXCH TTMP1
DAS TMP6

TC Q
```

# Page 977

# Page 978

# CONSTANTS FOR AUTOPILOTS

# NOTE....1 ASCREV (ACTUATOR CMD SCALING) = 85.41 ARCSEC/BIT OR 1.07975111 REVS (85.41x16384/360)  
# 1 SPASCREV (SPECIAL ACTUATOR CMD SCALING) = 1.04620942 REVS

```
ACTSAT DEC 253 # ACTUATOR LIMIT (6 DEG), SC.AT 1ASCREV
1/ACTSAT DEC .0039525692 # RECIPROCAL (1/253)

ERRLIM EQUALS BIT13 # FILTER INPUT LIMIT....B-3 REVS (45DEG),
1/ERRLIM EQUALS BIT3 # SC.AT B-1 REV, AND ITS RECIPROCAL

PITCHT5 GENADR PITCHDAP # UPPER WORDS OF T5 2CADRS, LOWER WORDS
DAPT5 GENADR DAPINIT # (BBCON) ALREADY THERE. ORDER IS
YAWT5 GENADR YAWDAP # REQUIRED.

1/RTLIM DEC 0.004715 # .004715(CDUDIF) = 0 IF CDUDIF < 2.33 DEG
1-E(-AT) OCT 00243 # AT = .01SEC....EITHER(1/A=4SEC, T=40MS),
```

1934

July 12, 2016

E(-AT)

OCT

37535

#

OR(1/A=8SEC, T=80MS)

This code is written to file `src/TVCDAPS.s`.

**B.120 TVCEXECUTIVE**

1935

*<src/TVCEXECUTIVE.s 1935>*≡

```

Copyright: Public domain.
Filename: TVCEXECUTIVE.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 945-950
Mod history: 2009-05-12 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
2009-05-20 RSB Corrections: CAE -> CAF in one place.
2009-05-21 RSB In 1SHOTCHK, a CAF SEVEN was corrected to
CAF SIX.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 945
PROGRAM NAME.... TVCEXECUTIVE, CONSISTING OF TVCEXEC, NEEDLEUP, VARGAINS
1SHOTCHK, REPCHEK, CG.CORR, COPYCYCLES, ETC.
LOG SECTION.... TVCEXECUTIVE SUBROUTINE ...DAPCSM
MOD BY SCHLUNDT 21 OCTOBER 1968
#
FUNCTIONAL DESCRIPTION....
*A SELF-PERPETUATING WAITLIST TASK AT 1/2 SECOND INTERVALS WHICH:
PREPARES THE ROLL WITH OGA (CDUX)
PREPARES THE ROLL FDAI NEEDLE (FLY-TO OGA ERROR)
PREPARES THE ROLL PHASE PLANE OGAERR (FLY-FROM OGA ERROR)

```

```

PREPARES THE TVC ROLLDAP TASK WAITLIST CALL (3 CS DELAY)
UPDATES THE NEEDLES DISPLAY
UPDATES THE VEHICLE MASS AND CALLS MASSPROP TO UPDATE INERTIA DATA
UPDATES PITCH, YAW, AND ROLL DAP GAINS FROM MASSPROP DATA
PERFORMS ONE-SHOT CORRECTION FOR TMC LOOP 0-3 SEC AFTER IGNITION
PERFORMS REPETITIVE UPDATES FOR THE TMC LOOP AFTER THE ONE-SHOT CORR.
#
CALLING SEQUENCE....
*TVCEXEC CALLED AS A WAITLIST TASK, IN PARTICULAR BY TVCINIT4 AND BY
ITSELF, BOTH AT 1/2 SECOND INTERVALS
#
NORMAL EXIT MODE.... TASKOVER
#
ALARM OR ABORT EXIT MODES.... NONE
#
SUBROUTINES CALLED....NEEDLER, S40.15, MASSPROP, TASKOVER, IBNKCALL
#
OTHER INTERFACES....
*TVCRESTART PACKAGE FOR RESTARTS
*PITCHDAP, YAWDAP FOR VARIABLE GAINS AND ENGINE TRIM ANGLES
#
ERASABLE INITIALIZATION REQUIRED....
*SEE TVCDAPON....TVCINIT4
*VARK AND 1/CONACC (S40.15 OF R03)
*PAD LOAD EREPFRAC
*BITS 15,14 OF FLAGWRD6 (T5 BITS)
*TVCEXPHS FOR RESTARTS
*ENGINE-ON BIT (11.13) FOR RESTARTS
*CDUX, OGAD
#
OUTPUT....
*ROLL DAP OGANOW, FDAI NEEDLE= (AK). AND PHASE PLANE OGAERR
*VARIABLE GAINS FOR PITCH/YAW AND ROLL TVC DAPS
*SINGLE-SHOT AND REPETITIVE CORRECTIONS TO ENGINE TRIM ANGLES
PACTOFF AND YACTOFF
#
DEBRIS.... MUCH, BUT SHAREABLE WITH RCS/ENTRY, ALL IN EBANK6

Page 946
#
BANK 16
SETLOC DAPROLL
BANK
EBANK= BZERO
COUNT* $$/TVCX

TVCEXEC CS FLAGWRD6 # CHECK FOR TERMINATION (BITS 15,14 READ

```



July 12, 2016

1937

```

 MASK OCT60000 # 10 FROM TVCDAPON TO RCSDAPON)
 EXTEND
 BZMF TVCEXFIN # TERMINATE

 CAF .5SEC # W.L. CALL TO PERPETUATE TVCEXEC
 TC WAITLIST
 EBANK= BZERO
 2CADR TVCEXEC

ROLLPREP CAE CDUX # UPDATE ROLL LADDERS (NO NEED TO RESTART-
 XCH OGANOW # PROTECT, SINCE ROLL DAPS RE-START)
 XCH OGAPAST

 CAE OGAD # PREPARE ROLL FDAI NEEDLE WITH FLY-TO
 EXTEND # ERROR (COMMAND - MEASURED)
 MSU OGANOW
 TS AK # FLY-TO OGA ERROR, SC.AT B-1 REVS

 EXTEND # PREPARE ROLL DAP PHASE PLANE OGAERR
 MP -BIT14
 TS OGAERR # PHASE-PLANE (FLY-FROM) OGAERROR,
 # SC.AT B+0 REVS

 CAF THREE # SET UP ROLL DAP TASK (ALLOW SOME TIME)
 TC WAITLIST
 EBANK= BZERO
 2CADR ROLLDAP

NEEDLEUP TC IBNKCALL # DO A NEEDLES UPDATE (RETURNS AFTER CADR)
 CADR NEEDLER # (NEEDLES RESTARTS ITSELF)

VARGAINS CAF BIT13 # CHECK ENGINE-ON BIT TO INHIBIT VARIABLE
 EXTEND # GAINS AND MASS IF ENGINE OFF
 RAND DSALMOUT # CHANNEL 11
 CCS A
 TCF +4 # ON, SO OK TO UPDATE GAINS AND MASS
+5 CAF TWO # OFF, SO BYPASS MASS/GAIN UPDATES,
 TS TVCEXPHS # ALSO ENTRY FROM CCS BELOW WITH
 TCF 1SHOTCHK # VCNTR = -0 (V97 R40 ENGFAIL)
 CCS VCNTR # TEST FOR GAIN OF UPDATE TIME
 TCF +4 # NOT YET

Page 947
 TCF GAINCHNG # NOW
 TCF +0 # NOT USED
 TCF VARGAINS +5 # NO, LOTHRUST (S40.6 R40)
```

|          |    |        |            |   |                                          |     |
|----------|----|--------|------------|---|------------------------------------------|-----|
|          | +4 | TS     | VCNTRTMP   | # | PROTECT VCNTR AND                        |     |
|          |    | CAE    | CSMMASS    | # | CSMMASS DURING AN IMPULSIVE BURN         |     |
|          |    | TS     | MASSTMP    |   |                                          |     |
|          |    | TCF    | EXECCOPY   |   |                                          |     |
| GAINCHNG |    | TC     | IBNKCALL   | # | UPDATE IXX, IAVG, IAVG/TLX               |     |
|          |    | CADR   | FIXCW      | # | MASSPROP ENTRY (ALREADY INITIALIZED)     |     |
|          |    | TC     | IBNKCALL   | # | UPDATE 1/CONACC, VARK                    |     |
|          |    | CADR   | S40.15     | # | (S40.15 IS IN TVCINITIALIZE)             |     |
|          |    | CS     | TENMDOT    | # | UPDATE MASS FOR NEXT 10 SEC. OF BURN     |     |
|          |    | AD     | CSMMASS    |   |                                          |     |
|          |    | TS     | MASSTMP    | # | KG B+16                                  |     |
|          |    | CAF    | NINETEEN   | # | RESET THE VARIABLE-GAIN UPDATE COUNTER   |     |
|          |    | TS     | VCNTRTMP   |   |                                          |     |
| EXECCOPY |    | INCR   | TVCEXPHS   | # | RESTART-PROTECT TEH COPYCYCLE            | (1) |
|          |    | CAE    | MASSTMP    | # | CSMMASS KG B+16                          |     |
|          |    | TS     | CSMMASS    |   |                                          |     |
|          |    | CAE    | VCNTRTMP   | # | VCNTR                                    |     |
|          |    | TS     | VCNTR      |   |                                          |     |
|          |    | TS     | V97VCNTR   | # | FOR ENGFALL (R41) MASS UPDATES AT SPSOFF |     |
|          |    | INCR   | TVCEXPHS   | # | COPYCYCLE OVER                           | (2) |
| 1SHOTCHK |    | CCS    | CNTR       | # | CHECK FOR ONE-SHOT OR REPCORR            |     |
|          |    | TCF    | +4         | # | NOT YET                                  |     |
|          |    | TCF    | 1SHOTOK    | # | NOW                                      |     |
|          |    | TCF    | REPCHEK    | # | ONE-SHOT OVER, ON TO REPCORR             |     |
|          |    | TCF    | 1SHOTOK    | # | NOW (ONE-SHOT ONLY, NO REPCORR)          |     |
|          | +4 | TS     | CNTRTMP    | # | COUNT DOWN                               |     |
|          |    | CAF    | SIX        | # | SETUP TVCEXPHS FOR ENTRY AT CNTRCOPY     |     |
|          |    | TS     | TVCEXPHS   |   |                                          |     |
|          |    | TCF    | CNTRCOPY   |   |                                          |     |
| REPCHEK  |    | CAE    | REPFRAC    | # | CHECK FOR REPETITIVE UPDATES             |     |
|          |    | EXTEND |            |   |                                          |     |
|          |    | BZMF   | TVCEXFIN   | # | NO (NEG OR +-ZERO)                       |     |
|          |    | TS     | TEMPDAP +1 | # | YES, SET UP CORRECTION FUNCTION          |     |
|          |    | CAF    | FOUR       | # | SET UP TVCEXPHS FOR ENTRY AT CORSETUP    |     |
|          |    | TS     | TVCEXPHS   |   |                                          |     |
|          |    | TCF    | CORSETUP   |   |                                          |     |

July 12, 2016

1939

```
1SHOTOK CAF BIT13 # CHECK ENGINE-ON BIT, NOT PERMITTING
 EXTEND # SWITCHOVER DURING ENGINE-SHUTDOWN
 RAND DSALMOUT
 CCS A
 TCF +2 # ONE-SHOT OK
 TCF TVCEXFIN # NO, TERMINATE

 INCR TVCEXPHS # (3)
```

# RSB 2009. The following instruction was previously "CAE FCORFRAC", but FCORFRAC  
# is not in erasable memory as implied by the use of CAE. I've accordingly changed  
# it to CAF instead to indicate fixed memory.

```
TEMPSET CAF FCORFRAC # SET UP CORRECTION FRACTION
 TS TEMPDAP +1

 INCR TVCEXPHS # ENTRY FROM REPCHECK AT NEXT LOCATION (4)
```

```
CORSETUP CAE DAPDATR1 # CHECK FOR LEM-OFF/ON
 MASK BIT13 # (NOTE, SHOWS LEM-OFF)
 EXTEND
 BZF +2 # LEM IS ON, PICK UP TEMPdap+1
 CAE TEMPdap +1 # LEM IS OFF, PICK UP 2(TEMPdap+1)
 AD TEMPdap +1
 TS TEMPdap # CG.CORR USES TEMPdap

 CAF NEGONE # SET UP FOR CNTR = -1 (SWTCHOVR DONE)
 TS CNTRTMP # (COPYCYCLE AT "CNTRCOPY")
```

```
CG.CORR EXTEND # PITCH TMC LOOP
 DCA PDELOFF
 DXCH PACTTMP
 EXTEND
 DCS PDELOFF
 DDOUBL
 DDOUBL
 DXCH TTMP1
 EXTEND
 DCA DELPBAR
 DDOUBL
 DDOUBL
 DAS TTMP1
 EXTEND
 DCA TTMP1
 EXTEND
 MP TEMPdap
 DAS PACTTMP
```

```

EXTEND
DCA YDELOFF
DXCH YACTTMP
EXTEND
DCS YDELOFF
DDOUBL

Page 949

DDOUBL
DXCH TTMP1
EXTEND
DCA DELYBAR
DDOUBL
DDOUBL
DAS TTMP1
EXTEND
DCA TTMP1
EXTEND
MP TEMPDAP
DAS YACTTMP

CORCOPY INCR TVCEXPHS # RESTART-PROTECT THE COPYCYCLE (5)

EXTEND
DCA PACTTMP
TS PACTOFF
DXCH PDELOFF

EXTEND
DCA YACTTMP
TS YACTOFF
DXCH YDELOFF

INCR TVCEXPHS # ENTRY FROM 1SHOTCHK AT NEXT LOCATION (6)

CNTRCOPY CAE CNTRTMP # UPDATE CNTR (RESTARTS OK, FOLLOWS CPYCY)
 TS CNTR

TVCEXFIN CAF ZERO # RESET TVCEXPHS
 TS TVCEXPHS
 TCF TASKOVER # TVCEXECUTIVE FINISHED

FCORFRAC OCT 10000 # ONE-SHOT CORRECTION FRACTION

Page 950 (page is empty)

```

July 12, 2016

1941

This code is written to file `src/TVCEXECUTIVE.s`.

## B.121 TVCINITIALIZE

```

1942 <src/TVCINITIALIZE.s 1942>≡
Copyright: Public domain.
Filename: TVCINITIALIZE.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Jim Lawton <jim.lawton@gmail.com>.
Website: www.ibiblio.org/apollo.
Pages: 936-944
Mod history: 2009-05-11 JVL Adapted from the Colossus249/ file
of the same name, using Comanche055 page
images.
2009-05-20 RSB Corrections: +80 -> +8D, added 4 missing
lines in TVCINIT1, changed the capitalization
of a couple of the "Page N" comments,
corrected a couple of lines in LOADCOEFF.
2009-05-22 RSB In LOADCOEF, DXCH N10 +14D corrected to
TS N10 +14D. Also, various comment-marks
were added to comments following this
change.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A

Page 937
NAME TVCDAPON (TVC DAP INITIALIZATION AND STARTUP CALL)
LOG SECTION...TVCINITIALIZE SUBROUTINE...DAPCSM
MODIFIED BY SCHLUNDT 21 OCTOBER 1968
FUNCTIONAL DESCRIPTION

```

```

PERFORMS TVCDAP INITIALIZATION (GAINS, TIMING PARAMETERS, FILTER VARIABLES, ETC.)
COMPUTES STEERING (S40.8) GAIN KPRIMEDT, AND ZEROES PASTDELV,+1 VARIABLE
MAKES INITIALIZATION CALL TO "NEEDLER" FOR TVC DAP NEEDLES-SETUP
PERFORMS INITIALIZATION FOR ROLL DAP
CALLS TVCEXECUTIVE AT TVCEXEC, VIA WAITLIST
CALLS TVCDAP CDU-RATE INITIALIZATION PKG AT DAPINIT VIA T5
PROVIDES FOR LOADING OF LOW-BANDWIDTH COEFFS AND GAINS AT SWICHOVR
CALLING SEQUENCE - T5LOC=2CADR(TVCDAPON,EBANK=BZERO), T5=.6SECT5
IN PARTICULAR, CALLED BY "DOTVCON" IN P40
MRCLEAN AND TVCINIT4 ARE POSSIBLE TVC-RESTART ENTRY POINTS
NORMAL EXIT MODE
TCF RESUME
SUBROUTINES CALLED
NEEDLER, MASSPROP
ALARM OR ABORT EXIT MODES
NONE
ERASABLE INITIALIZATION REQUIRED
CSMMASS, LEMMASS, DAPDATR1 (FOR MASSPROP SUBROUTINE)
TVC PAD LOADS (SEE EBANK6 IN ERASABLE ASSIGNMENTS)
PACTOFF, YACTOFF, CDUX
TVCPHASE AND THE T5 BITS OF FLAGWRD6 (SET AT DOTVCON IN P40)
OUTPUT
ALL TVC AND ROLL DAP ERASABLES, FLAGWRD6 (BITS 13,14), T5, WAITLIST
DEBRIS
NONE

COUNT* $$/INIT
BANK 17
SETLOC DAPS7
BANK

EBANK= BZERO

TVCDAPON LXCH BANKRUPT # T5 RUPT ARRIVAL (CALL BY DOTVCON - P40)
 EXTEND # SAVE Q REQUIRED IN RESTART (MRCLEAN AND
MRCLEAN QXCH QRUPT # TVCINIT4 ARE ENTRIES)
 CAF NZERO # NUMBER TO ZERO, LESS ONE (MUST BE ODD)
 # TVC RESTARTS ENTER HERE (NEW BANK)

 +1 CCS A
 TS CNTR
 CAF ZERO
 TS L
 INDEX CNTR
 DXCH OMEGAYC # FIRST (LAST) TWO LOCATIONS
 CCS CNTR
 TCF MRCLEAN +1

```

# Page 938

|          |        |          |                                             |
|----------|--------|----------|---------------------------------------------|
|          | EXTEND |          | # SET UP ANOTHER T5 RUPT TO CONTINUE        |
|          | DCA    | INITLOC2 | #       INITIALIZATION AT TVCINIT1          |
|          | DXCH   | T5LOC    | # THE PHSCHK2 ENTRY (REDOTVC) AT TVCDAPON   |
|          | CAF    | POSMAX   | #       +3 IS IN ANOTHER BANK. MUST RESET   |
|          | TS     | TIME5    | #       BBCON TOO (FULL 2CADR), FOR THAT    |
| ENDMRC   | TCF    | RESUME   | #       ENTRY.                              |
| TVCINIT1 | LXCH   | BANKRUPT |                                             |
|          | EXTEND |          |                                             |
|          | QXCH   | QRUPT    |                                             |
|          | TC     | IBNKCALL | # UPDATE IXX, IAVG/TLX FOR DAP GAINS (R03   |
|          | CADR   | MASSPROP | #       OR NOUNS 46 AND 47 MUST BE CORRECT) |
|          | CAE    | EMDOT    | # SPS FLOW RATE, SCALED B+3 KG/CS           |
|          | EXTEND |          |                                             |
|          | MP     | ONETHOU  |                                             |
|          | TS     | TENMDOT  | # 10-SEC MASS LOSS B+16 KG                  |
|          | COM    |          |                                             |
|          | AD     | CSMMASS  |                                             |
|          | TS     | MASSTMP  | # DECREMENT FOR FIRST 10 SEC OF BURN        |
|          | CAE    | DAPDATR1 | # CHECK LEM-ON/OFF                          |
|          | MASK   | BIT14    |                                             |
|          | CCS    | A        |                                             |
|          | CAF    | BIT1     | # LEM-ON (BIT1)                             |
|          | TS     | CNTR     | # LEM-OFF (ZERO)                            |
|          | INDEX  | CNTR     | # LOAD THE FILTER COEFFICIENTS              |
|          | CAF    | CSMCFADR |                                             |
|          | TS     | COEFFADR |                                             |
|          | TC     | LOADCOEF |                                             |
|          | INDEX  | CNTR     | # PICK UP LM-OFF,-ON KTLX/I                 |
|          | CAE    | EKTLX/I  | # SCALED AT 1/(8 ASCREV) OF ACTUAL VALUE    |
|          | TS     | KTLX/I   |                                             |
|          | TCR    | S40.15   | # COMPUTE 1/CONACC, VARK                    |
| TVCINIT2 | CS     | CNTR     | # PICK LM-OFF,-ON VALUE FOR FILTER PERIOD   |
|          | INDEX  | A        | # DETERMINATION:                            |
|          | CAF    | BIT2     | #       BIT2 FOR CSM ONLY 40MS FILTER       |
|          | TS     | KPRIMEDT | #       BIT3 FOR CSM/LM 80MS FILTER         |
|          | COM    |          | # PREPARE T5TVCDT                           |



July 12, 2016

1945

# Page 939

|          |        |           |                                          |
|----------|--------|-----------|------------------------------------------|
|          | AD     | POSMAX    |                                          |
|          | AD     | BIT1      |                                          |
|          | TS     | T5TVCDT   |                                          |
|          | CS     | BIT15     | # RESET SWTOVER FLAG                     |
|          | MASK   | FLAGWRD9  |                                          |
|          | TS     | FLAGWRD9  |                                          |
|          | INDEX  | CNTR      | # PICK UP LEM-OFF,-ON KPRIME             |
|          | CAE    | EKPRIME   | # SCALED (100 PI)/16                     |
|          | EXTEND |           |                                          |
|          | MP     | KPRIMEDT  | # (TVCDT/2, SC.AT B+14 CS)               |
|          | LXCH   | A         | # SC.AT PI/8 (DIMENSIONLESS)             |
|          | DXCH   | KPRIMEDT  |                                          |
|          | INDEX  | CNTR      | # PICK UP LEM-OFF,-ON REPFRAC            |
|          | CAE    | EREPPFRAC |                                          |
|          | TS     | REPFRAC   |                                          |
|          | INDEX  | CNTR      | # PICK UP ONE-SHOT CORRECTION TIME       |
|          | CAF    | TCORR     |                                          |
|          | TS     | CNTR      |                                          |
|          | CAF    | NEGONE    | # PREVENT STROKE TEST UNTIL CALLED       |
|          | TS     | STRKTIME  |                                          |
|          | CAF    | NINETEEN  | # SET VCNTR FOR VARIABLE-GAIN UPDATES IN |
|          | TS     | VCNTR     | # 10 SECONDS (TVCEXEC 1/2 SEC RATE)      |
|          | TS     | V97VCNTR  | # FOR ENGFAIL (R41) LOGIC                |
| TVCINIT3 | CAE    | PACTOFF   | # TRIM VALUES TO TRIM-TRACKERS, OUTPUT   |
|          | TS     | PDELOFF   | # TRACKERS, OFFSET-UPDATES, AND          |
|          | TS     | PCMD      | # OFFSET-TRACKER FILTERS                 |
|          | TS     | DELPBAR   | # NOTE, LO-ORDER DELOFF,DELBAR ZEROED    |
|          | CAE    | YACTOFF   |                                          |
|          | TS     | YDELOFF   |                                          |
|          | TS     | YCMD      |                                          |
|          | TS     | DELYBAR   |                                          |
| ATTINIT  | CAE    | DAPDATR1  | # ATTITUDE-ERROR INITIALIZATION LOGIC    |
|          | MASK   | BIT13     | # TEST FOR CSM OR CSM/LM                 |
|          | EXTEND |           |                                          |
|          | BZF    | NEEDLEIN  | # BYPASS INITIALIZATION FOR CSM/LM       |

|            |        |            |   |                                          |
|------------|--------|------------|---|------------------------------------------|
|            | CAF    | BIT1       | # | SET UP TEMPORARY COUNTER                 |
| +5         | TS     | TTMP1      |   |                                          |
|            | INDEX  | TTMP1      |   |                                          |
|            | CA     | ERRBTMP    | # | ERRBTMP CONTAINS RCS ATTITUDE ERRORS     |
|            | EXTEND |            | # | ERRORY & ERRORZ (P40 AT DOTVCON)         |
|            | MP     | 1/ATTLIM   | # | .007325(ERROR) = 0 IF ERROR < 1.5 DEG    |
|            | EXTEND |            |   |                                          |
|            | BZF    | +8D        | # | ERROR  LESS THAN 1.5 DEG                 |
|            | EXTEND |            |   |                                          |
| # Page 940 | BZMF   | +3         | # | ERROR  > 1.5 DEG, AND NEG                |
|            | CA     | ATTLIM     | # | ERROR  > 1.5 DEG, AND POS                |
|            | TCF    | +2         |   |                                          |
| +3         | CS     | ATTLIM     |   |                                          |
| +2         | INDEX  | TTMP1      |   |                                          |
|            | TS     | ERRBTMP    |   |                                          |
| +8         | CCS    | TTMP1      | # | TEST TEMPORARY COUNTER                   |
|            | TCF    | ATTINIT +5 | # | BACK TO REPEAT FOR PITCH ERROR           |
|            | CA     | ERRBTMP    | # | ERROS ESTABLISHED AND LIMITED            |
|            | TS     | PERRB      |   |                                          |
|            | CA     | ERRBTMP +1 |   |                                          |
|            | TS     | YERRB      |   |                                          |
| NEEDLEIN   | CS     | RCSFLAGS   | # | SET BIT 3 FOR INITIALIZATION PASS AND GO |
|            | MASK   | BIT3       | # | TO NEEDLER. WILL CLEAR FOR TVC DAP       |
|            | ADS    | RCSFLAGS   | # | (RETURNS AFTER CADR)                     |
|            | TC     | IBNKCALL   |   |                                          |
|            | CADR   | NEEDLER    |   |                                          |
| TVCINIT4   | CAF    | ZERO       | # | SET TVCPHASE TO INDICATE TVCDAPON-THRU-  |
|            | TS     | TVCPHASE   | # | NEEDLEIN INITIALIZATION FINISHED.        |
|            |        |            | # | (POSSIBLE TVC-RESTART ENTRY)             |
|            | CAE    | CDUX       | # | PREPARE ROLL DAP                         |
|            | TS     | OGANOW     |   |                                          |
|            | CAF    | BIT13      | # | IF ENGINE IS ALREADY OFF, ENGINOFF HAS   |
|            | EXTEND |            | # | ALREADY ESTABLISHED THE POST-BURN        |
|            | RAND   | DSALMOUT   | # | CSMMASS (MASSBACK DOES IT). DON'T        |
|            | EXTEND |            | # | TOUCH CSMMASS. IF ENGINE IS ON,          |
|            | BZF    | +3         | # | THEN IT'S OK TO DO THE COPYCYCLE         |
|            |        |            | # | EVEN BURNS LESS THAN 0.4 SEC ARE AOK     |
|            | CAE    | MASSTMP    | # | COPYCYCLE                                |

July 12, 2016

1947

|            |        |              |                                            |
|------------|--------|--------------|--------------------------------------------|
|            | TS     | CSMMASS      |                                            |
| +3         | CAF    | .5SEC        | # CALL TVCEXECUTIVE (ROLLDAP CALL, ETC)    |
|            | TC     | WAITLIST     |                                            |
|            | EBANK= | BZERO        |                                            |
|            | 2CADR  | TVCEXEC      |                                            |
|            | EXTEND |              | # CALL FOR DAPINIT                         |
|            | DCA    | DAPINIT5     |                                            |
|            | DXCH   | T5LOC        |                                            |
|            | CAE    | T5TVCDT      | # (ALLOW TIME FOR RESTART COMPUTATIONS)    |
|            | TS     | TIME5        |                                            |
| # Page 941 |        |              |                                            |
| ENDTVCIN   | TCF    | RESUME       |                                            |
| PRESWTCH   | TCR    | SWICHOVR     | # ENTRY FROM V46                           |
|            | TC     | POSTJUMP     | # THIS PROVIDES AN EXIT FROM SWITCH-OVER   |
|            | CADR   | PINBRNCH     | # (PINBRNCH DOES A RELINT)                 |
| SWICHOVR   | INHINT |              |                                            |
|            | CA     | TVCPHASE     | # SAVE TVCPHASE                            |
|            | TS     | PHASETMP     |                                            |
|            | CS     | BIT2         | # SET TVCPHASE = -2 (INDICATES SWITCH-OVER |
|            | TS     | TVCPHASE     | # TO RESTART LOGIC)                        |
| +5         | EXTEND |              | # SAVE Q FOR RETURN (RESTART ENTRY POINT,  |
|            | QXCH   | RTRNLOC      | # TVCPHASE AND PHASETMP ALREADY SET)       |
|            | CAF    | NZEROJR      | # ZEROING LOOP FOR FILTER STORAGE LOCS     |
| +8         | TS     | CNTRTMP      |                                            |
| MCLEANJR   | CA     | ZERO         |                                            |
|            | TS     | L            |                                            |
|            | INDEX  | CNTRTMP      |                                            |
|            | DXCH   | PTMP1 -1     |                                            |
|            | CCS    | CNTRTMP      |                                            |
|            | CCS    | A            |                                            |
|            | TCF    | SWICHOVR +8D |                                            |
|            | CS     | FLAGWRD9     | # SET SWITCHOVER FLAG FOR DOWNLINK         |
|            | MASK   | BIT15        |                                            |
|            | ADS    | FLAGWRD9     |                                            |
|            | CAE    | EKTLX/I +2   | # LOW BANDWIDTH GAINS - DAP                |
|            | TS     | KTLX/I       |                                            |

|            |        |          |    |                                            |
|------------|--------|----------|----|--------------------------------------------|
|            | TCR    | S40.15   | +7 |                                            |
|            | CAF    | FKPRIMDT |    | #                                          |
|            | TS     | KPRIMEDT |    | - STEERING                                 |
|            | CAF    | FREPFRAC |    | #                                          |
|            | TS     | REPFRAC  |    | - TMC LOOP                                 |
|            | EXTEND |          |    | # UPDATE TRIM ESTIMATES                    |
|            | DCA    | DELPBAR  |    |                                            |
|            | DXCH   | PDELOFF  |    |                                            |
|            | EXTEND |          |    |                                            |
|            | DCA    | DELYBAR  |    |                                            |
|            | DXCH   | YDELOFF  |    |                                            |
|            | CA     | LBCFADR  |    |                                            |
| # Page 942 | TS     | COEFFADR |    |                                            |
|            | TC     | LOADCOEF |    |                                            |
|            | CAE    | PHASETMP |    | # RESTORE TVCPHASE                         |
|            | TS     | TVCPHASE |    |                                            |
|            | TC     | RTRNLOC  |    | # BACK TO PRESWTCH OR TVCRESTARTS          |
| LOADCOEF   | EXTEND |          |    | # LOAD DAP FILTER COEFFICIENTS             |
|            | INDEX  | COEFFADR |    | # FROM: ERASABLE FOR CSM/LM HB             |
|            | DCA    | 0        |    | # FIXED FOR CSM/LM LB                      |
|            | DXCH   | N10      |    | # FIXED FOR CSM                            |
|            | EXTEND |          |    | # NOTE: FOR CSM/LM, NORMAL COEFFICIENT     |
|            | INDEX  | COEFFADR |    | # LOAD WILL BE HIGH BANDWIDTH PAD LOAD     |
|            | DCA    | 2        |    | # ERASABLES. DURING CSM/LM SWITCHOVER,     |
|            | DXCH   | N10      | +2 | # THIS LOGIC IS USED TO LOAD LOW BANDWIDTH |
|            |        |          |    | # COEFFICIENTS FROM FIXED MEMORY.          |
|            | EXTEND |          |    |                                            |
|            | INDEX  | COEFFADR |    |                                            |
|            | DCA    | 4        |    |                                            |
|            | DXCH   | N10      | +4 |                                            |
|            | EXTEND |          |    |                                            |
|            | INDEX  | COEFFADR |    |                                            |
|            | DCA    | 6        |    |                                            |
|            | DXCH   | N10      | +6 |                                            |

July 12, 2016

1949

```
EXTEND
INDEX COEFFADR
DCA 8D
DXCH N10 +8D

EXTEND
INDEX COEFFADR
DCA 10D
DXCH N10 +10D

EXTEND
INDEX COEFFADR
DCA 12D
DXCH N10 +12D

INDEX COEFFADR
CA 14D
TS N10 +14D

TC Q

Page 943
S40.15 CAE IXX # GAIN COMPUTATIONS (1/CONACC, VARK)
EXTEND # ENTERED FROM TVCINITIALIZE AND TVCEXEC
MP 2PI/M # 2PI/M SCALED 1/(B+8 N M)
DDOUBL # IXX SCALED B+20 KG-MSQ
DDOUBL
DDOUBL
TS 1/CONACC # SCALED B+9 SEC-SQ/REV

+7 CAE KTLX/I # ENTRY FROM CSM/LM V46 SWITCH-OVER
EXTEND # SCALED (B+3 ASCREV) 1/SECSQ
MP IAVG/TLX # SCALED B+2 SECSQ
DDOUBL
DDOUBL
TS VARK # SCALED (B+3 ASCREV)
TC Q

CSMN10 DEC .99999 # N10 CSM ONLY FILTER COEFFICIENTS
DEC -.2549 # N11/2
DEC .0588 # N12
DEC -.7620 # D11/2
DEC .7450 # D12

DEC .99999 # N20
DEC -.4852 # N21/2
DEC 0 # N22
```

|            |        |               |                                           |
|------------|--------|---------------|-------------------------------------------|
|            | DEC    | -.2692        | # D22/2                                   |
|            | DEC    | 0             | # D22                                     |
| LBN10      | DEC    | +.99999       | # N10 LOW BANDWIDTH FILTER COEFFICIENTS   |
|            | DEC    | -.3285        | # N11/2                                   |
|            | DEC    | -.3301        | #N12                                      |
|            | DEC    | -.9101        | #D11/2                                    |
|            | DEC    | +.8460        | #D12                                      |
|            | DEC    | +.03125       | #N20                                      |
|            | DEC    | 0             | #N21/2                                    |
|            | DEC    | 0             | #N22                                      |
|            | DEC    | -.9101        | #D21/2                                    |
|            | DEC    | +.8460        | #D22                                      |
|            | DEC    | +.50000       | #N30                                      |
|            | DEC    | -.47115       | #N31/2                                    |
|            | DEC    | +.4749        | #N32                                      |
|            | DEC    | -.9558        | #D31/2                                    |
|            | DEC    | +.9372        | #D32                                      |
| CSMCFADR   | GENADR | CSMN10        | # CSM ONLY COEFFICIENTS ADDRESS           |
| HBCFADR    | GENADR | HBN10         | # HIGH BANDWIDTH COEFFICIENTS ADDRESS     |
| # Page 944 |        |               |                                           |
| LBCFADR    | GENADR | LBN10         | # LOW BANDWIDTH COEFFICIENTS ADDRESS      |
| NZERO      | DEC    | 51            | # MUST BE ODD FOR MRCLEAN                 |
| NZEROJR    | DEC    | 23            | # MUST BE ODD FOR MCLEANJR                |
| ATTLM      | DEC    | 0.00833       | # INITIAL ATTITUDE EROR LIMIT (1.5 DEG)   |
| 1/ATTLM    | DEC    | 0.007325      | # .007325(ERROR) = 0 IF ERROR < 1.5 DEG   |
| TCORR      | OCT    | 00005         | # CSM                                     |
| +1         | OCT    | 00000         | # CSM/LM (HB,LB)                          |
| FKPRIMDT   | DEC    | .0102         | # CSM/LM (LB), (.05 X .08) SCALED AT PI/8 |
| FREPFRAC   | DEC    | .0375 B-2     | # CSM/LM (LB), 0.0375 SCALED AT B+2       |
| NINETEEN   | =      | VD1           |                                           |
| 2PI/M      | DEC    | .00331017 B+8 | # 2PI/M, SCALED AT 1/(B+8 N-M)            |
| ONETHOU    | DEC    | 1000 B-13     | # KG/CS B3 TO KG/10SEC B16 CONVERSION     |
|            | EBANK= | BZERO         |                                           |
| DAPINIT5   | 2CADR  | DAPINIT       |                                           |

July 12, 2016

1951

```
 EBANK= BZERO
INITLOC2 2CADR TVCINIT1
```

This code is written to file `src/TVCINITIALIZE.s`.

## B.122 TVCMASSPROP

```

1952 <src/TVCMASSPROP.s 1952>≡
 # Copyright: Public domain.
 # Filename: TVCMASSPROP.agc
 # Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
 # It is part of the source code for the Command Module's (CM)
 # Apollo Guidance Computer (AGC), for Apollo 11.
 # Assembler: yaYUL
 # Contact: Ron Burkey <info@sandroid.org>.
 # Website: www.ibiblio.org/apollo.
 # Pages: 951-955
 # Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
 # same name, using Comanche055 page images.
 #
 # This source code has been transcribed or otherwise adapted from digitized
 # images of a hardcopy from the MIT Museum. The digitization was performed
 # by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
 # thanks to both. The images (with suitable reduction in storage size and
 # consequent reduction in image quality as well) are available online at
 # www.ibiblio.org/apollo. If for some reason you find that the images are
 # illegible, contact me at info@sandroid.org about getting access to the
 # (much) higher-quality images which Paul actually created.
 #
 # Notations on the hardcopy document read, in part:
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. 10:28 APR. 1, 1969
 #
 # This AGC program shall also be referred to as
 # Colossus 2A
 #
 # Page 951
 # PROGRAM NAME...MASSPROP
 # LOG SECTION...TVCMASSPROP PROGRAMMER...MELANSON (ENGEL, SCHLUNDT)
 #
 # FUNCTIONAL DESCRIPTION:
 #
 # MASSPROP OPERATES IN TWO MODES: (1) IF LEM MASS OR CONFIGURATION ARE UPDATED
 # FOR THIS) THE ENTIRE PROGRAM MUST BE RUN THROUGH, BREAKPOINT VALUES AND DERIV
 # RESPECT TO CSM MASS BEING CALCULATED PRIOR TO CALCULATION OF THE OUTPUTS. (2
 # CALCULATED USING PREVIOUSLY COMPUTED BREAKPOINT VALUES AND DERIVATIVES.
 #
 # CALLING SEQUENCES
 #
 # IF LEM MASS OR CONFIGURATION HAS BEEN UPDATED, TRANSFER TO MASSPROP, OTHERWIS

```



July 12, 2016

1953

```
L TC BANKCALL or IBNKCALL
L+1 CADR MASSPROP
OR
L+1 CADR FIXCW
L+2 RETURNS VIA Q
#
CALLED: IN PARTICULAR BY DONOUN47 (JOB) AND TVCEXECUTIVE (TASK)
#
JOBS OR TASKS INITIATED: NONE
#
SUBROUTINES CALLED: NONE
#
ERASABLE INITIALIZATION REQUIRED
#
LEMMASS MUST CONTAIN LEM MASS SCALED AT B+16 KILOGRAMS
CSMMASS MUST CONTAIN CSM MASS SCALED AT B+16 KILOGRAMS
DAPDATR1 MUST BE SET TO INDICATE VEHICLE CONFIGURATION.
BITS (15,14,13) = (0 , 0 , 1) LEM OFF
(0 , 1 , 0) LEM ON (ASCNT,DSCNT)
(1 , 1 , 0) LEM ON (ASCNT ONLY)
#
ALARMS: NONE
#
EXIT: TC Q
#
OUTPUTS:
#
(1) IXX, SINGLE PRECISION SCALED AT B+20 IN KG-M SQ.
(2) IAVG, SINGLE PRECISION SCALED AT B+20 IN KG-M SQ.
(3) IAVG/TLX, SINGLE PRECISION, SCALED AT B+2 SEC-SQD
#
THEY ARE STORED IN CONSECUTIVE REGISTERS IXX0, IXX1, IXX2
CONVERSION FACTOR: (SLUG-FTSQ) = 0.737562 (KG-MSQ)
Page 952
#
OUTPUTS ARE CALCULATED AS FOLLOWS:
#
(1) IF LEM DOCKED, LEMMASS IS FIRST ELIMINATED AS A PARAMETER
#
VARST0 = INTVALUE0 + LEMMASS(SLOPEVAL0) IXX BREAKPOINT VALU
VARST1 = INTVALUE1 + LEMMASS(SLOPEVAL1) IAVG BREAKPOINT VALU
VARST2 = INTVALUE2 + LEMMASS(SLOPEVAL2) IAVG/TLX BREAKPOINT VALU
#
VARST3 = INTVALUE3 + LEMMASS(SLOPEVAL3) IAVG/TLX SLOPE FOR CSMMA
VARST4 = INTVALUE4 + LEMMASS(SLOPEVAL4) IAVG SLOPE FOR CSMMA
#
```

```

VARST5 = INTVALUE5 + LEMMASS(SLOPEVAL5) IXX SLOPE
#
VARST6 = INTVALUE6 + LEMMASS(SLOPEVAL6) IAVG SLOPE
VARST7 = INTVALUE7 + LEMMASS(SLOPEVAL7) IAVG/TLX SLOPE
#
VARST8 = INTVALUE8 + LEMMASS(SLOPEVAL8) IAVG DECRE
VARST9 = INTVALUE9 + LEMMASS(SLOPEVAL9) IAVG/TLX DECRE
#
(2) IF LEM NOT DOCKED
#
VARST0 = NOLEMVAL0 WHERE THE MEANING AND SCALING OF VARST0
. TO VARST9 ARE THE SAME AS GIVEN ABOVE
.
. NOTE... FOR THIS CASE, VARST8,9 HAVE NO
VARST9 = NOLEMVAL9 MEANING (THEY ARE COMPUTED BUT NOT USED)
#
(3) THE FINAL OUTPUT CALCULATIONS ARE THEN DONE
#
IXX0 = VARST0 + (CSMASS + NEGBPW)VARST5 IXX
#
IXX1 = VARST1 + (CSMASS + NEGBPW)VARST(4 OR 6) IAVG
#
IXX2 = VARST2 + (CSMASS + NEGBPW)VARST(3 OR 7) IAVG/TLX
#
THE DATA USED CAME FROM THE CSM/LM SPACECRAFT OPERATIONAL DATA BOOK
VOL. 3, NASA DOCUMENT SNA-8-D-027 (MARCH 1968)
#
PERTINENT MASS DATA: CSM WEIGHT (FULL) 64100 LBS.
(EMPTY) 23956 LBS.
LEM WEIGHT (FULL) 32000 LBS.
(EMPTY) 14116 LBS.
#
(WEIGHTS ARE FROM AMMENDMENT #1 (APRIL 24, 1968) TO ABOVE DATA BOOK)
Page 953

```

```

BANK 25
SETLOC DAPMASS
BANK
EBANK= BZERO
COUNT* $$/MASP

```

```

MASSPROP CAF NINE # MASSPROP USES TVC/RCS INTERRUPT TEMPS
 TS PHI333 # SET UP TEN PASSES

LEMTEST CAE DAPDATR1 # DETERMINE LEM STATUS
 MASK BIT13

```

|            |        |             |                                           |
|------------|--------|-------------|-------------------------------------------|
|            | EXTEND |             |                                           |
|            | BZF    | LEMYES      |                                           |
| LEMNO      | INDEX  | PHI333      | # LEM NOT ATTACHED                        |
|            | CAF    | NOLEMVAL    |                                           |
|            | TCF    | STOINST     |                                           |
| LEMYES     | CAE    | LEMASS      | # LEM IS ATTACHED                         |
|            | DOUBLE |             |                                           |
|            | EXTEND |             |                                           |
|            | INDEX  | PHI333      |                                           |
|            | MP     | SLOPEVAL    |                                           |
|            | DDOUBL |             |                                           |
|            | INDEX  | PHI333      |                                           |
|            | AD     | INTVALUE    |                                           |
| STOINST    | INDEX  | PHI333      | # STORAGE INST BEGIN HERE                 |
|            | TS     | VARSTO      |                                           |
|            | CCS    | PHI333      | # ARE ALL TEN PASSES COMPLETED            |
|            | TCF    | MASSPROP +1 | # NO: GO DECREMENT PHI333                 |
| DXTEST     | CCS    | DAPDATR1    | # IF NEG, BIT15 IS 1, LEM DSCNT STAGE OFF |
|            | TCF    | FIXCW       |                                           |
|            | TCF    | FIXCW       |                                           |
|            | DXCH   | VARSTO +8D  |                                           |
|            | DAS    | VARSTO +1   |                                           |
|            | CA     | DXITFIX     |                                           |
|            | ADS    | VARSTO +7   |                                           |
| FIXCW      | CAF    | BIT2        | # COMPUTATION PHASE BEGINS HERE. SET UP   |
|            | TS     | PHI333      | # THREE PASSES                            |
|            | TS     | PSI333      |                                           |
|            | CAE    | CSMASS      | # GET DELTA CSM WEIGHT: SIGN DETERMINES   |
|            | AD     | NEGBPW      | # SLOPE LOCATIONS.                        |
|            | DOUBLE |             |                                           |
|            | TS     | TEMP333     |                                           |
| # Page 954 | EXTEND |             |                                           |
|            | BZMF   | PEGGY       | # DETERMINE CORRECT SLOPE                 |
|            | CAF    | NEG2        |                                           |
|            | TS     | PHI333      |                                           |
| PEGGY      | INDEX  | PHI333      | # ALL IS READY: CALCULATE OUTPUTS NOW     |
|            | CAE    | VARST5      | # GET SLOPE                               |
|            | EXTEND |             |                                           |

|            |        |                  |                                          |
|------------|--------|------------------|------------------------------------------|
|            | MP     | TEMP333          | # MULT BY DELTA CSM WEIGHT               |
|            | DOUBLE |                  |                                          |
|            | INDEX  | PSI333           |                                          |
|            | AD     | VARST0           | # ADD BREAKPOINT VALUE                   |
|            | INDEX  | PSI333           |                                          |
|            | TS     | IXX              | # ***** OUTPUTS (IXX0, IXX1, IXX2) ***** |
|            | CCS    | PSI333           | # BOOKKEEPING: MASSPROP FINISHED OR NOT  |
|            | TCF    | BOKKEP2          | # NO: GO TAKE CARE OF INDEXING REGISTERS |
|            | CAE    | DAPDATR1         | # UPDATE WEIGHT/G                        |
|            | MASK   | BIT14            |                                          |
|            | CCS    | A                |                                          |
|            | CA     | LEMMASS          |                                          |
|            | AD     | CSMMASS          |                                          |
|            | TS     | WEIGHT/G         | # SCALED AT B+16 KILOGRAMS               |
| ENDMASSP   | TC     | Q                |                                          |
| BOKKEP2    | TS     | PSI333           | # REDUCE PSI BY ONE                      |
|            | EXTEND |                  |                                          |
|            | DIM    | PHI333           |                                          |
|            | TCF    | PEGGY            |                                          |
| # Page 955 |        |                  |                                          |
| NOLEMVAL   | DEC    | 25445 B-20       |                                          |
|            | DEC    | 87450 B-20       |                                          |
|            | DEC    | .30715 B-2       |                                          |
|            | DEC    | 1.22877 E-5 B+12 |                                          |
|            | DEC    | 1.6096 B-6       |                                          |
|            | DEC    | 1.54 B-6         |                                          |
|            | DEC    | 7.77177 B-6      |                                          |
|            | DEC    | 3.46458 E-5 B+12 |                                          |
| INTVALUE   | DEC    | 26850 B-20       |                                          |
|            | DEC    | 127518 B-20      |                                          |
|            | DEC    | .54059 B-2       |                                          |
|            | DEC    | .153964 E-4 B+12 |                                          |
|            | DEC    | -.742923 B-6     |                                          |
|            | DEC    | 1.5398 B-6       |                                          |
|            | DEC    | 9.68 B-6         |                                          |
|            | DEC    | .647625 E-4 B+12 |                                          |
|            | DEC    | -27228 B-20      |                                          |
|            | DEC    | -.206476 B-2     |                                          |
| SLOPEVAL   | DEC    | 1.96307 B-6      |                                          |
|            | DEC    | 27.5774 B-6      |                                          |

July 12, 2016

1957

|         |                         |
|---------|-------------------------|
| DEC     | 2.3548 E-5 B+12         |
| DEC     | 2.1777 E-9 B+26         |
| DEC     | 1.044 E-3 B+8           |
| DEC     | 0                       |
| DEC     | 2.21068 E-3 B+8         |
| DEC     | 1.5166 E-9 B+26         |
| DEC     | -1.284 B-6              |
| DEC     | 2 E-5 B+12              |
| NEGBPW  | DEC -15402.17 B-16      |
| DXITFIX | DEC* -1.88275 E-5 B+12* |

This code is written to file `src/TVCMASSPROP.s`.

**B.123 TVCRESTARTS**1958  $\langle \text{src}/\text{TVCRESTARTS}.s \text{ 1958} \rangle \equiv$ 

```

Copyright: Public domain.
Filename: TVCRESTARTS.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 956-960
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
2009-05-20 RSB Corrections: TCF -> BZF in one place.
2009-05-21 RSB In PHSCHK2, CS TVCPHASE corrected to
CCS TVCPHASE and CCS 4 corrected to CCS A.
Page 924 corrected to 961. CORCOPY +2
corrected to CORCOPY +1.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 956
NAME...TVCRESTART PACKAGE, CONSISTING OF REDOTVC, ENABL1, 2, CMDSOUT, PHSCHK2, ET
LOG SECTION...TVCRESTARTS SUBROUTINE...DAPCSM
MODIFIED BY SCHLUNDT 21 OCTOBER 1968
#
FUNCTIONAL DESCRIPTION....
#
*RESTART-PROCESS THE TVC DAPS, INCLUDING PITCHDAP, YAWDAP,
TVCEXECUTIVE, ROLLDAP, TVCINIT4, TVCDAPON, AND CSM/.M V46 SWTCHOVR.

```

#  
# \*TVC RESTARTS DESERVE SPECIAL CONSIDERATION IN SEVERAL AREAS.  
# RESTART DOWN-TIME IS IMPORTANT BECAUSE OF THE TRANSIENTS INTRODUCED  
# BY THE THRUST VECTOR RETURN TO THE ACTUATOR MECHANICAL NULLS  
# FOLLOWING TVC- AND OPTICS-ERROR-COUNTER-DISENABLES (CHANNEL 12).  
# TVC USES A MIXTURE OF WAITLIST, T5, T6, AND JOB CALLS. THERE IS  
# FILTER MEMORY (UP TO 6TH ORDER) TO BE PROTECTED IF WILD TRANSIENTS  
# ARE TO BE AVOIDED. COUNTERS ARE INVOLVED FOR ONE-SHOT  
# CORRECTIONS AND GAIN UPDATES. THE GIMBAL TRIM ESTIMATORS AND THE  
# BODY AXIS ATTITUDE ERROR INTEGRATORS INVOLVE DIGITAL SUMMATION.  
# DIGITAL DIFFERENTIATORS ARE INVOLVED IN THE BODY AXIS RATE ESTIMA-  
# TIONS AND IN THE OUTPUTTING OF ACTUATOR COMMANDS. THERE IS AN  
# OFFSET-TRACKER-FILTER TO PROTECT. ETC., ETC.  
#  
# \*THOSE QUANTITIES WHICH MUST BE PROTECTED ARE STORED IN TEMPORARY  
# REGISTERS AS THEY ARE COMPUTED, FOR UPDATING THE REAL REGISTERS  
# DURING COPYCYCLES.  
#  
# \*THE SEVERAL COPYCYCLES ARE EACH PROTECTED BY PHASE POINTS AT THEIR  
# BEGINNING AND AT THEIR TERMINATION. THE PHASE POINTS ARE SIMPLY  
# "INCR" INSTRUCTIONS, EITHER "INCR TVCEXPHS" FOR COPYCYCLES  
# IN THE TVCEXECUTIVE, OR "INCR TVCPHASE" FOR THE PITCH AND YAW  
# COPYCYCLES. INDEXING ON EACH OF THESE POINTERS THEN PERMITS A  
# RETURN TO THE APPROPRIATE RESTART POINTS.  
#  
# \*IF A RESTART OCCURS DURING EITHER COPYCYCLE, THAT COPYCYCLE IS  
# COMPLETED. THEN THE NORMAL TVCINIT4...DAPINIT...PITCHDAP STARTUP  
# SEQUENCE IS CALLED UPON TO GET THINGS GOING AGAIN.  
#  
# \*TVC-ENABLE AND OPTICS-ERROR-COUNTER ENABLE MUST BE SET ASAP  
# (ALLOWING FOR PROCEDURAL DELAYS). THEN THE ENGINES ARE COMMANDED  
# TO THE P,YACTOFF TRIM VALUES. THE DAPS ARE THEN READY TO GO ON THE  
# AIR, WITH THE REGULAR STARTUP SEQUENCE, EITHER AT MRCLEAN FOR A  
# COMPLETE INITIALIZATION OR AT TVCINIT4 FOR A PARTIAL INITIALIZATION.  
#  
# \*FOR RESTARTS PRIOR TO THE SETTING OF THE T5 BITS AT DOTVCON THE  
# PRE40.6 SECTION OF S40.6 TAKES CARE OF RE-ESTABLISHING TRIMS.  
#  
# \*IF A RESTART OCCURS DURING THE TVCEXEC...TVCEXFIN SEQUENCE THE  
# COMPUTATIONS WILL BE COMPLETED, STARTING AT THE APPROPRIATE RESTART  
# POINT, AFTER THE DAPS ARE READY TO GO ON THE AIR.  
#  
# \*IF A RESTART OCCURS PRIOR TO TVCINIT4 (TVCPHAS = -1) E.G. DURING  
# THE EARLY DAP INITIALIZATION PHASE, THE DAP STARTUP SEQUENCE IS  
# ENTERED AT MRCLEAN FOR A FULL INITIALIZATION.  
#

```

*FOR RESTARTS DURING CSM/LM V46 SWITCH-VER, TVCPHASE IS SET TO -2.
AND THE RESTART LOGIC GOES BACK TO REDO SWITCH-OVER (AFTER THE
NORMAL DAP RESTART SEQUENCE IS FOLLOWED.)
#
*RESTARTS ARE NOT CRITICAL TO THE ROLL DAP PERFORMANCES HENCE THE
ROLL DAP IS MERELY RESTARTED.
#
*RESTARTS DURING A STROKE TEST (STROKER IS NON-ZERO) WILL CAUSE THE
Page 957
STROKE TEST TO BE TERMINATED. A NEW V68 ENTRY WILL BE REQUIRED
TO GET IT GOING AGAIN (NO AUTOMATIC RESTART).
#
*REDOTVC IS REACHED FOLLOWING ANY RESTART WHICH FINDS THE T5 BITS
(BITS 15,14 OF FLAGWRD6) SET FOR TVC. DOTVCON TVCPHASE = -1
AND TVC EXPHS = 0 JUST BEFORE SETTING THESE BITS, JUST BEFORE
MAKING THE T5 CALL TO TVCDAPON. ON A NORMAL SHUTDOWN DOTVCRCS
CALLS RCSDAPON, WHICH RESETS THE T5 BIT FOR RCS
#
CALLING SEQUENCE....T5, IN PARTICULAR BY ELRSKIP OF FRESH START/RESTART
#
NORMAL EXIT MODES....RESUME, NOQRSM, POSTJUMP (TO TVCINIT4 OR MRCLEAN)
#
ALARM OR ABORT EXIT MODES....NONE
#
SUBROUTINES CALLED....
#
*PCOPY+1, YCOPY+1 (PITCH AND YAW COPYCYCLES)
*ENABLE1,2, CMDSOUT (RE-ESTABLISH ACTUATOR TRIMS)
*MRCLEAN OR TVCINIT4 (TVCDAP INITIALIZATIONS)
*SWITCHOVR +5 (CSM/LM V46 SWITCH-OVER)
*EXRSTRT AND TVCEXECUTIVE PHASE POINTS 1 THRU 6
*WAITLIST, IBNKCALL, POSTJUMP, ISWCALL
#
OTHER INTERFACES....DOTVCON AND RCSDAPON (T5 BITS), ELRSKIP (CALLS IT)
#
ERASABLE ININITIALIZATION REQUIRED....
#
*T5 BITS (1,0), TVCPHASE (-2,-1,0,1,2,3), TVCEXPHS (1 THRU 6)
*TVC DAP VARIABLES
*OPERATIONS PERFORMED BY REDOTVC ARE BASED ON THE ASSUMPTION THAT
THE TVC DAPS ARE RUNNING NORMALLY
#
OUTPUT....
#
*PITCH AND YAW TVC DAP COPYCYCLES COMPLETED IF INTERRUPTED
*TVCEXECUTIVE COMPLETED IF INTERRUPTED

```



July 12, 2016

1961

```
*STROKE TEST TERMINATED IF INTERRUPTED
*CSM/LM V46 SWITCH-OVER REPEATED IF INTERRUPTED
*ACTUATOR TRIMS RE-ESTABLISHED (ACTUATORS BACK ON THE AIR)
*TVC DAP INITIALIZATION AS REQUIRED
*ALL TVC DAP OPERATIONS ON THE AIR
#
DEBRIS...TVC TEMPORARIES IN EBANK6
```

#Page 958

```

BANK 16
SETLOC DAPROLL
BANK
EBANK= TVCPHASE
COUNT* $$/RSRT

REDOTVC LXCH BANKRUPT # TVC RESTART PACKAGE
 EXTEND
 QXCH QRUPRT # ("TCR" IN "FINCOPY")

EXECPHS CCS TVCEXPHS # CHECK TVCEXECUTIVE PHASE
 TCF +2 # MUST RESTART TVCEXECUTIVE
 TCF TVCDAPHS # NO NEED TO RESTART TVCEXECUTIVE

 CAF NINE # 9CS DELAY TO FORCE EXRSTRT TO OCCUR
 TC WAITLIST # BEFORE PITCHDAP, AFTER CMDSOUT
 EBANK= TVCEXPHS
 2CADR EXRSTRT

TVCDAPHS CS OCT37776 # CHECK BITS 15 AND 1 OF TVCPHASE TO SEE
 MASK TVCPHASE # DAP RESTART LOCATION (-1,1,2,3)
 CCS A #
 TCF FINCOPY # FINISH THE COPYCYCLE FIRST
 TCF ENABL1 # JUST PREPARE THE OUTCOUNTERS AND GO

 CS TVCPHASE # TEST FOR TVCPHASE = -2
 MASK BIT2 # (THIS INDICATES RESTART OCCURRED
 EXTEND # DURING CSM/LM V46 SWITCH-OVER)
 BZF TRIM/CMD # NO. TVCPHASE = -1. RSTRT WAS IN TVCINIT

ENABL1 CAF BIT8 # TVC ENABLE, FOLLOWED BY 40 MS (MIN) WAIT
 AD BIT11 # SET BIT FOR OPTICS-DAC-ENABLE ALSO
 EXTEND # (ENABL1 ENTERED FROM TVCDAPHS / FINCOPY)
 WOR CHAN12
 CAF TVCADDR # WAIT, CALLING ENABL2 (BBCON THERE)
 TS T5LOC
```

1962

July 12, 2016

|            |        |             |   |                                         |
|------------|--------|-------------|---|-----------------------------------------|
|            | CAF    | TVCADDR +4  | # | 60 MS (TVCEXADR)                        |
|            | TS     | TIME5       |   |                                         |
|            | TCF    | RESUME      |   |                                         |
| ENABL2     | LXCH   | BANKRUPT    | # | CONTINUE PREPARATION OF OUTCOUNTERS     |
|            | CAF    | BIT2        | # | OPTICS ERROR CNTR ENABLE, 4MS MIN WAIT  |
|            | EXTEND |             |   |                                         |
|            | WOR    | CHAN12      |   |                                         |
| # Page 959 | CAF    | TVCADDR +2  | # | WAIT, CALLING CMDSOUT (BBCON THERE)     |
|            | TS     | T5LOC       |   |                                         |
|            | CAF    | OCT37776    | # | 20MS                                    |
|            | TS     | TIME5       |   |                                         |
|            | TCF    | NOQRSM      |   |                                         |
| CMDSOUT    | LXCH   | BANKRUPT    | # | CONTINUE PREPARATION OF OUTCOUNTERS     |
|            | EXTEND |             |   |                                         |
|            | QXCH   | QRUP        |   |                                         |
|            | CS     | ZERO        | # | MOST RECENT ACTUATOR COMMANDS           |
|            | AD     | PCMD        | # | (AVOID +0)                              |
|            | TS     | TVCPITCH    |   |                                         |
|            | CS     | ZERO        |   |                                         |
|            | AD     | YCMT        |   |                                         |
|            | TS     | TVCYAW      |   |                                         |
|            | CAF    | PRI06       | # | RELEASE THE COUNTERS (BITS 11,12)       |
|            | EXTEND |             |   |                                         |
|            | WOR    | CHAN14      |   |                                         |
| PHSCHK2    | CCS    | TVCPHASE    | # | CHECK TVCPHASE AGAIN                    |
|            | TCF    | CHKSTRK     |   |                                         |
|            | TCF    | CHKSTRK     |   |                                         |
|            | CCS    | A           | # | A CONTAINS THE DIMINISHED ABSOLUTE OF   |
|            | TC     | +3          | # | TVCPHASE (-2 BECOMES +1. -1 BECOMES +0) |
|            | TC     | POSTJUMP    | # | REPEAT TVC INITIALIZATION               |
|            | CADR   | MRCLEAN     | # | (DO NOT RETURN)                         |
| +3         | TC     | IBNKCALL    | # | REPEAT CSM/LM V46 SWITCH-OVER           |
|            | CADR   | SWICHOVR +5 | # | (RETURN TO CHECK FOR STROKE TEST)       |
| CHKSTRK    | CCS    | STROKER     | # | CHECK FOR STROKE TEST IN PROGRESS       |

July 12, 2016

1963

|            |        |            |                                            |
|------------|--------|------------|--------------------------------------------|
|            | TCF    | TSTINITJ   | # YES, KILL IT                             |
|            | TCF    | +2         | # NO, PROCEED                              |
|            | TCF    | TSTINITJ   | # YES, KILL IT                             |
| +4         | TC     | POSTJUMP   | # IF POSITIVE OR ZERO, RESTART AT          |
|            | CADR   | TVCINIT4   | # TVCINIT4 (ZEROS TVCPHASE, AND            |
|            |        |            | # CALLS TVC DAPS VIA DAPINIT)              |
| FINCOPY    | INDEX  | TVCPHASE   | # PICK UP THE APPROPRIATE COPYCYCLE        |
|            | CAF    | TVCCADR    |                                            |
|            | TCR    | ISWCALL    | # RE-ENTER THE COPYCYCLE, RETURN AT END    |
|            | TCF    | ENABL1     | # NOW PREPARE THE OUTCOUNTERS              |
| TRIM/CMD   | EXTEND |            | # TVCDAPON INITIALIZATION NOT COMPLETED,   |
| # Page 960 |        |            |                                            |
|            | DCA    | PACTOFF    | # EG. P,YCMD MAY NOT BE SET. SET...        |
|            | DXCH   | PCMD       |                                            |
|            | TCF    | ENABL1     | # NOW PREPARE THE OUTCOUNTERS              |
| TSTINITJ   | CAF    | ZERO       | # DISABLE STROKE TEST (-0 SHOWS PRIOR V68) |
|            | TS     | STROKER    | # (+0 MEANS NEW V68 REQUIRED FOR STARTUP)  |
|            | TCF    | CHKSTRK +4 |                                            |
| EXRSTRT    | INDEX  | TVCEXPHS   | # TVCEXECUTIVE RESTARTS...GO TO            |
|            | CAF    | TVCEXADR   | # APPROPRIATE RESTART POINT                |
|            | INDEX  | A          |                                            |
|            | TCF    | 0          |                                            |

# Page 961

# TVC RESTART TABLES.... ORDER IS REQUIRED. HI-ORDER WORDS ONLY, OF 2CADRS, SINCE BBCON IS ALP

|          |        |             |                                           |
|----------|--------|-------------|-------------------------------------------|
| TVCADDR  | =      | TVCCADR     | # TABLE OF CADRS, UNUSED LOCS FOR GENADRS |
| TVCCADR  | GENADR | ENABL2      | # (FOR T5 CALL, UNUSED TABLE LOC)         |
| +1       | CADR   | PCOPY +1    | # PITCH COPYCYCLE                         |
| +2       | GENADR | CMDSOUT     | # (FOR T5 CALL, UNUSED TABLE LOC)         |
| +3       | CADR   | YCOPY +1    | # YAW COPYCYCLE                           |
| TVCEXADR | OCT    | 37772       | # (UNUSED TABLE LOC, FILL WITH 60MS, T5)  |
| +1       | GENADR | EXECCOPY +1 | # TVCEXECUTIVE RESTART POINTS (ORDERED)   |
| +2       | GENADR | 1SHOTCHK    |                                           |
| +3       | GENADR | TEMPSET     |                                           |
| +4       | GENADR | CORSETUP    |                                           |
| +5       | GENADR | CORCOPY +1  |                                           |
| +6       | GENADR | CNTRCOPY    |                                           |

This code is written to file src/TVCRESTARTS.s.

1964

July 12, 2016

## B.124 TVCROLLDAP

```
1964 <src/TVCROLLDAP.s 1964>≡
Copyright: Public domain.
Filename: TVCROLLDAP.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 984-998
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 984
PROGRAM NAME....TVC ROLL AUTOPILOT
LOG SECTION....TVCROLLDAP SUBROUTINE....DAPCSM
MOD BY SCHLUNDT 21 OCTOBER 1968
#
FUNCTIONAL DESCRIPTION....
#
*AN ADAPTATION OF THE LEM P-AXIS CONTROLLER
*MAINTAIN OGA WITHIN 5 DEG DEADBND OF OGAD, WHERE OGAD = OGA AS SEEN
BY IGNOVER (P40)
*MAINTAIN OGA RATE LESS THAN 0.1 DEG/SEC LIMIT CYCLE RATE
*SWITCHING LOGIC IN PHASE PLANE.... SEE GSOP CHAPTER 3
*USES T6 CLOCK TO TIME JET FIRINGS.
*MAXIMUM JET FIRING TIME = 2.56 SECONDS, LIMITED TO 2.5 IF GREATER
```

July 12, 2016

1965

```
*MINIMUM JET FIRING TIME = 15 MS
*JET PAIRS FIRE ALTERNATELY
*AT LEAST 1/2 SECOND DELAY BEFORE A NEW JET PAIR IS FIRED
*JET FIRINGS MAY NOT BE EXTENDED, ONLY SHORTENED, WHEN RE-EVALUATION
OF A JET FIRING TIME IS MADE ON A LATER PASS
#
CALLING SEQUENCE....
#
*ROLLDAP CALL VIA WAITLIST, IN PARTICULAR BY TVCEXEC (EVERY 1/2 SEC)
WITH A 3CS DELAY TO ALLOW FREE TIME FOR OTHER RUPTS (DWNRPRT, ETC.)
#
NORMAL EXIT MODES.... ENDOFJOB
#
ALARM OR ABORT EXIT MODES.... NONE
#
SUBROUTINES CALLED.....NONE
#
OTHER INTERFACES....
#
*TVCEXEC SETS UP ROLLDAP TASK EVERY 1/2 SECOND AND UPDATES 1/CONACC
EVERY 10 SECONDS (VIA MASSPROP AND S40.15)
*RESTARTS SUSPEND ROLL DAP COMPUTATIONS UNTIL THE NEXT 1/2 SEC
SAMPLE PERIOD. (THE PART OF TVCEXECUTIVE THAT CALLS ROLL DAP IS
NOT RESTARTED.) THE OGAD FROM IGNITION IS MAINTAINED.
#
ERASABLE INITIALIZATION REQUIRED
#
*1/CONACC (S40.15)
*OGAD (CDUX, AT IGNITION)
*OGANOW (CDUX AT TVCINIT4 AND TVCEXECUTIVE)
*OGAPAST (OGANOW AT TVCEXECUTIVE)
*ROLLFIRE = TEMREG = ROLLWORD = 0 (MRCLEAN LOOP IN TVCDAPON)
#
OUTPUT....
#
*ROLL JET PAIR FIRINGS
#
Page 985
DEBRIS.... MISCELLANEOUS, SHAREABLE WITH RCS/ENTRY, IN EBANK6 ONLY
#
Page 986
SOME NOTES ON THE ROLL AUTOPILOT, AND IN PARTICULAR, ON ITS SWITCHING
LOGIC. SEE SECTION THREE OF THE GSOP (SUNDISK/COLOSSUS) FOR DETAILS.
#
SWITCHING LOGIC IN THE PHASE PLANE....
#
```

July 12, 2016

```

OGARATE
*
*
* * * * * * * * * * * * * * * *
* (REGION 1, SEE TEXT BELOW)
*
*
* * * * * * * * * * * * * * * * ...PARABOLA (SWITCHING = CONTROL)
*
* .
* *
* (FIRE NEG ROLL JETS)
* *
(-DB,+LMCRATE)...*. *
*
* *
* OGAERROR

* * (-AK, OGAERR)
* * (REGION 6-PRIME)
* * (SEE TEXT BELOW)
* *
* * ...STRAIGHT LINE
(FIRE POS ROLL JETS) * * * *
* (COAST) *
* * * * * * * * * * *
* * -MINLIM
* *
*
* * * * * * * * * * * * * *
* * -MAXLIM
*
#
SWITCHING PARABOLAS ARE CONTROL PARABOLAS, THUS REQUIRING KNOWLEDGE OF
CONTROL ACCELERATION CONACC, OR ITS RECIPROCAL, 1/CONACC, THE TVC
ROLL DAP GAIN (SEE TVCEXECUTIVE VARIABLE GAIN PACKAGE). JET
FIRING TIME IS SIMPLY THAT REQUIRED TO ACHIEVE THE DESIRED OGARATE,
SUBJECT TO TEH LIMITATIONS DISCUSSED UNDER FUNCTIONAL DESCRIPTION,
ABOVE.
#
THE THREE CONTROL REGIONS (+, -, AND ZERO TORQUE) ARE COMPRISED OF
TWELVE SUBSET REGIONS (1...6, AND THE CORRESPONDING 1-PRIME...
5-PRIME) SEE SECTION 3 OF THE GSOP (SUNDISK OR COLOSSUS)
Page 987
#
GIVEN THE OPERATING POINT NOT IN THE COAST REGION, THE DESIRED OGARATE
IS AT THE POINT OF PENETRATION OF THE COAST REGION BY THE CONTROL

```

```

PARABOLA WHICH PASSES THROUGH THE OPERATING POINT. FOR REGION 3
DESIRED OGARATE IS SIMPLY +-MAXLIM. FOR REGIONS 1 OR 6 THE SOLUTION
TO A QUADRATIC IS REQUIRED (THE PENETRATION IS ALONG THE STRAIGHT
LINE OR MINLIM BOUNDARY SWITCH LINES). AN APPROXIMATION IS MADE
INSTEAD. CONSIDER AN OPERATING POINT IN REGION 6'. PASS A TANGENT TO
THE CONTROL PARABOLA THROUGH THE OPERATING POINT, AND FIND ITS
INTERSECTION WITH THE STRAIGHT LINE SECTION OF THE SWITCH CURVE...
THE INTERSECTION DEFINES THE DESIRED OGARATE. IF THE OPERATING POINT IS
CLOSE TO THE SWITCH LINE, THE APPROXIMATION IS QUITE GOOD (INDEED
THE APPROXIMATE AND QUADRATIC SOLUTIONS CONVERGE IN THE LIMIT AS
THE SWITCH LINE IS APPROACHED). IF THE OPERATING POINT IS NOT CLOSE
TO THE SWITCH LINE, THE APPROXIMATE SOLUTION GIVES VALID TREND
INFORMATION (DIRECTION OF DESIRED OGARATE) AT LEAST. THE
RE-EVALUATION OF DESIRED OGARATE IN SUBSEQUENT ROLL DAP PASSES (1/2
SECOND INTERVALS) WILL BENEFIT FROM THE CONVERGENT NATURE OF THE
APPROXIMATION.
#
FOR LARGE OGAERROR THE TANGENT INTERSECTS +-MINLIM SWITCH BOUNDARY BEFORE
INTERSECTING THE STRAIGHT LINE SWITCH. HOWEVER THE MINLIM IS
IGNORED IN COMPUTING THE FIRING TIME, SO THAT THE EXTENSION (INTO
THE COAST REGION) OF THE STRAIGHT LINE SWITCH IS WHAT IS FIRED TO.
IF THE ROLL DAP FINDS ITSELF IN THE COAST REGION BEFORE REACHING
THE DESIRED INTERSECTION (I.E., IN THE REGION BETWEEN THE MINLIM
AND THE STRAIGHT LINE SWITCH) IT WILL EXHIBIT NORMAL COAST-REGION
BEHAVIOR AND TURN OFF THE JETS. THE PURPOSE OF THIS FIRING POLICY
IS TO MAINTAIN STATIC ROLL STABILITY IN THE EVENT OF A JET
FAILED-ON.
#
WHEN THE OPERATING POINT IS IN REGION 1 THE SAME APPROXIMATION IS
MADE, BUT AT AN ARTIFICIALLY-CREATED OR DUMMY OPERATING POINT,
DEFINED BY: OGAERROR = INTERSECTION OF CONTROL PARABOLA AND
OGAERROR AXIS, OGARATE = +-LMCRATE WHERE SIGN IS OPPOSITE THAT OF
REAL OPERATING POINT RATE. WHEN THE OPERATING POINT HAS PASSED
FROM REGION 1 TO REGION 6', THE DUMMY POINT IS NO LONGER REQUIRED,
AND THE SOLUTION REVERTS TO THAT OF A REGULAR REGION 6' POINT.
#
EQUATION FOR SWITCHING PARABOLA (SEE FIGURE ABOVE)....
#

$$SOGAERROR = (DB - (SOGARATE)^2 / (CONACC/2)) \operatorname{SGN}(SOGARATE)$$

#
EQUATION FOR SWITCHING STRAIGHT LINE SEGMENT....
#

$$SOGARATE = -(-SLOPE)(SOGAERROR) - \operatorname{SGN}(SOGARATE) \operatorname{INTERCEP}$$

#
WHERE $\operatorname{INTERCEP} = DB(-SLOPE) - \operatorname{LMCRATE}$
Page 988

```

```

#
EQUATION FOR INTERSECTION, CONTROL PARABOLA, AND STRAIGHT SWITCH LINE....
#
DOGADOT = NUM/DEN, WHERE
#
NUM = (-SLOPE)(OGARATE) (1/CONACC)
+SGN(DELOGA)(-SLOPE)(OGAERROR - SGN(DELOGA)(DB))
+LMCRATE
#
DEN = (-SLOPE)(LMCRATE)(1/CONACC) = SGN(DELOGA)
#
DELOGA = OGAERROR - (DB - (OGADOT) (1/CONACC)/2)SGN(OGARATE)
#
FOR REGIONS 6 AND 6-PRIME, USE ACTUAL OPERATING POINT (OGA, OGARATE)
FOR OGAERROR AND OGARATE IN THE INTERSECTION EQUATIONS ABOVE.
FOR REGIONS 1 AND 1-PRIME USE DUMMY OPERATING POINT FOR OGAERROR
AND OGARATE, WHERE THE DUMMY POINT IS GIVEN BY....
#
OGAERROR = DELOGA + DB SGN(OGARATE)
#
OGARATE = -LMCRATE SGN(OGARATE)
#
NOTE, OGAERROR = OGA - OGAD USES DUMMY REGISTER OGA IN ROLL DAP CODING
ALSO, AT POINT WHERE DOGADOT IS COMPUTED, REGISTER DELOGA IS USED
AS A DUMMY REGISTER FOR THE OGAERROR IN THE NUM EQUATION ABOVE.
Page 989

ROLLDAP CODING....

 SETLOC DAPROLL
 BANK
 EBANK= OGANOW
 COUNT* $$/ROLL
ROLLDAP CAE OGANOW # OGA RATE ESTIMATOR...SIMPLE FIRST-ORDER
 EXTEND # DIFFERENCE (SAMPLE TIME = 1/2 SEC)
 MSU OGAPAST
 EXTEND
 MP BIT5
 LXCH A
 TS OGARATE # SC.AT B-4 REV/SEC

COMPUTATIONS WHICH FOLLOW USE OGA FOR OGAERR (SAME REGISTER)
EXAMINE DURATION OF LAST ROLL FIRING IF JETS ARE NOW ON.

DURATION CA ROLLFIRE # SAME SGN AS PRESENT TORQ,MAGN=POSMAX
 EXTEND

```



July 12, 2016

1969

```

 BZF +2 # ROLL JETS ARE NOW OFF.
 TCF ROLLOGIC # ENTER LOGIC, JETS NOW ON.

 CAE TEMREG # EXAMINE LAST FIRING INTERVAL
 EXTEND # IF POSITIVE, DON'T FIRE
 BZF ROLLOGIC # ENTER LOGIC, JETS NOW OFF.

 CAF ZERO # JETS HAVE NOT BEEN OFF FOR 1/2 SEC. WAIT
 TS TEMREG # RESET TEMREG
WAIT1/2 TCF TASKOVER # EXIT ROLL DAP

COMPUTE DB-(1/2 CONACC) (OGARATE)SQ (1/2 IN THE SCALING)

ROLLOGIC CS OGARATE # SCALED AT 2(-4) REV/SEC
 EXTEND
 MP 1/CONACC # SCALED AT 2(+9) SEC SQ /REV
 EXTEND
 MP OGARATE
 AD DB # SCALED AT 2(+0) REV
 TS TEMREG # QUANTITY SCALED AT 2(+0) REV.

GET SIGN OF OGARATE

 CA OGARATE
 EXTEND
 BZMF +3 # LET SGN(0) BE NEGATIVE
 CA BIT1
 TCF +2
 CS BIT1
 TS SGNRT # + OR - 2(-14)

Page 990
CALCULATE DISTANCE FROM SWITCH PARABOLA,DELOGA
 EXTEND
 MP TEMREG # SGN(OGARATE) TEMREG NOW IN L
 CS L
 AD OGA # SCALED AT 2(+0) REV
DELOGAC TS DELOGA # SC.AT B+0 REV, PLUS TO RIGHT OF C-PARAB

EXAMINE SGN(DELOGA) AND CREATE CA OR CS INSTR. DEPENDING UPON SIGN.

 EXTEND
 BZMF +3
 CAF PRI030 # = CA (30000)
 TCF +2
 CAF BIT15 # = CS (40000)
```

1970

July 12, 2016

```

 TS I

 INDEX I # TSET ON I SGN(OGARATE)
 0 SGNRT # CA OR CS
 COM
 EXTEND
REG1TST BZMF ROLLON # IF REGION 1 (DELOGA OGARATE SAME SIGN)

NO JET FIRE YET. TEST FOR MAX OGARATE.

 INDEX I
 0 OGARATE # CA OR CS...BOTH MUST BE NEG. HERE
 TS IOGARATE # I.E., I OGARATE
 AD MAXLIM # SCALED AT 2(-4) REV/SEC
 EXTEND
REG3TST BZMF RATELIM # IF REGION 3 (RATES TOO HIGH, FIRE JETS)

COMPUTATION OF I((-SLOPE)OGA + OGARATE) - INTERCEPT: NOTE THAT STR. LINE
SWITCH SLOPE IS (SLOPE) DEG/SEC/DEG, A NEG. QUANTITY

 CA OGARATE
 EXTEND
 MP BIT14
 TS TEMREG
 CA OGA
 EXTEND
 MP -SLOPE
 DDOUBL
 DDOUBL
 DDOUBL # (OGA ERROR MUST BE LESS THAN +-225 DEG)
 AD TEMREG

 INDEX I
 0 A # I((-SLOPE)OGA+OGARATE) AT 2(-3)REV/SEC
 COM
Page 991
 AD INTERCEP # SCALED AT 2(-3) REV.
 COM
 EXTEND
REG2TST BZMF NOROLL # IP REGION 2 (COAST SIDE OF STRT LINE)

CHECK TO SEE IF OGARATE IS ABOVE MINLIM

 CA IOGARATE # ALWAYS NEGATIVE
 AD MINLIM # SCALED AT 2(-4) REV/SEC.
 EXTEND
```

July 12, 2016

1971

REG4TST            BZMF        NOROLL            # IF REGION 4 (COAST SIDE OF MINLIM)

# ALL AREAS CHECKED EXCEPT LAST AREA...NO FIRE IN THIS SMALL SEGMENT

INDEX    I  
O        OGA  
COM  
AD       DB  
COM  
EXTEND

REG5TST            BZMF        NOROLL            # IF REGION 5 (COAST SIDE OF DB)

# JETS MUST FIRE NOW. OGARATE IS NEG. (OR VICE VERSA). USE DIRECT STR. LINE.  
# DELOGA AND DELOGART ARE USED AS DUMMY VARIABLES IN THE SOLUTION OF A  
# STRAIGHT LINE APPROXIMATION TO A QUADRATIC SOLUTION OF THE INTERSECTION  
# OF THE CONTROL PARABOLA AND THE STRAIGHT-LINE SWITCH LINE. THE STRAIGHT  
# LINE IS THE TANGENT TO THE CONTROL PARABOLA AT THE OPERATING POINT. (FOR  
# OPERATING POINTS IN REGIONS 6 AND 6')

REGION6            CAE        OGA            # USE ACTUAL OPERATING POINT FOR TANGENT  
TS            DELOGA            # ACTUAL STATE  
CA            OGARATE  
TS            DELOGART            # ACTUAL STATE, I.E., DEL OGARATE  
TCF            ONROLL

# JETS ALSO FIRE FROM HERE EXCEPT OGARATE IS POS (VICE VERSA), USE INDIRECT  
# STRAIGHT LINE ESTABLISHED BY TANGENT TO A CONTROL PARABOLA AT ((DELOGA  
# + DB SGN(DELOGA) ), -LMCRATE SGN(DELOGA) )        (THIS IS THE DUMMY  
# OPERATING POINT FOR OPERATING POINTS IN REGIONS 1 AND 1')

ROLLON            INDEX    I  
O            DB  
ADS          DELOGA            # DELOGA WAS DIST. FROM SWITCH PARABOLA  
  
CS          LMCRATE            # LIMIT CYCLE RATE AT 2(-4) REV/SEC  
INDEX       I  
O            A  
TS          DELOGART            # EVALUATE STATE FOR INDIRECT LINE.

# Page 992

# SOLVE STRAIGHT LINES SIMULTANEOUSLY TO OBTAIN DESIRED OGARATE.

ONROLL            EXTEND            # DELOGART IN ACC. ON ARRIVAL  
MP            1/CONACC  
DOUBLE  
EXTEND

|            |        |          |                                           |
|------------|--------|----------|-------------------------------------------|
|            | MP     | -SLOPE   |                                           |
|            | TS     | TEMREG   | # 2(-SLOPE)RATE /CONACC                   |
|            | EXTEND |          |                                           |
|            | MP     | DELOGART |                                           |
|            | TS     | DELOGART | # 2(-SLOPE)(RATESQ)/CONACC                |
|            | CS     | BIT11    |                                           |
|            | INDEX  | I        |                                           |
|            | O      | A        |                                           |
| RATEDEN    | ADS    | TEMREG   | # DENOMINATOR COMPLETED                   |
|            | INDEX  | I        |                                           |
|            | O      | DELOGA   |                                           |
|            | COM    |          |                                           |
|            | AD     | DB       |                                           |
|            | COM    |          |                                           |
|            | EXTEND |          |                                           |
|            | MP     | -SLOPE   |                                           |
|            | ADS    | DELOGART |                                           |
|            | CA     | LMCRATE  |                                           |
|            | EXTEND |          |                                           |
|            | MP     | BIT11    |                                           |
| RATENUM    | AD     | DELOGART | # NUMERATOR COMPLETED                     |
|            | XCH    | L        | # PLACE NUMERATOR IN L FOR OVERFL. CHECK  |
|            | CA     | ZERO     |                                           |
|            | EXTEND |          |                                           |
|            | DV     | TEMREG   | # OVERFLOW, IF ANYTHING, NOW APPEARS IN A |
|            | EXTEND |          |                                           |
|            | BZF    | DVOK     | # NO OVERFLOW...(O,L)/TEMREG = O,L        |
| MINLIMAP   | CCS    | A        |                                           |
|            | CAF    | POSMAX   | # POSITIVE OVERFLOW                       |
|            | TCF    | ROLLSET  |                                           |
|            | CS     | POSMAX   | # NEGATIVE OVERFLOW                       |
|            | TCF    | ROLLSET  |                                           |
| DVOK       | LXCH   | A        | # PUT NUMERATOR BACK INTO A, O INTO L     |
|            | EXTEND |          |                                           |
|            | DV     | TEMREG   | # RESULT OF DIVISION IS DESIRED OGARATE   |
|            | TCF    | ROLLSET  | # (SCALED AT B-4 REV/SEC)                 |
| RATELIM    | CS     | MAXLIM   |                                           |
|            | INDEX  | I        |                                           |
| # Page 993 | O      | A        | # IF I = CA, DESIRED RATE IS -MAXLIM      |

July 12, 2016

1973

# COMPUTE JET FIRE TIME, BASED ON DESIRED RATE MINUS PRESENT RATE

```
ROLLSET TS TEMREG # STORE DESIRED OGARATE (SCALED B-4)
EXTEND
SU OGARATE # RATE DIFF. SCALED AT 2(-4) REV/SEC
TS TEMREG # OVERFLOW PROTECT
TCF +3 # " "
INDEX A # " "
CS LIMITS # " "
EXTEND
MP T6SCALE # T6SCALE = 8/10.24
EXTEND
MP 1/CONACC # SCALED AT B+9 SECSQ/REV (MAX < .60)
DDOUBL
DDOUBL
TS TEMREG # OVERFLOW PROTECT
TCF +3 # " "
INDEX A # " "
CS LIMITS # " "
TS TEMREG # JET FIRE TIME AT 625 MICROSEC/BIT
EXTEND
BZF NOROLL
```

# JET FIRE TIME IS NZ, TEST FOR JETS NOW ON.

```
CAE TEMREG # DESIRED CHANGE IN OGARATE
EXTEND
MP ROLLFIRE # (SGN OF TORQUE: ZERO IF JETS NOW OFF)
CCS A
TCF MOREROLL # CONTINUE FIRING WITH PRESENT POLARITY
TCF NEWROLL # START NEW FIRING NOW, PLUS
TCF NOROLL # TERMINATE OLD FIRING, NEW SIGN REQUESTED
TCF NEWROLL # START NEW FIRING NOW, MINUS
```

# CONTINUE PRESENT FIRING

```
MOREROLL CAF ZERO
TS I # USE TEMP. AS MOREROLL SWITCH
TCF MAXTFIRE
```

# START NEW FIRING BUT CHECK IF GREATER THAN MIN FIRE TIME.

```
NEWROLL CCS TEMREG # CALL THIS T6FIRE
AD ONE
TCF +2
AD ONE
```

1974

July 12, 2016

```

Page 994
COM
AD TMINFIRE # -MAG(T6FIRE)
 # TMINFIRE-MAG(T6FIRE)

COM
EXTEND
MINTST BZMF NOROLL # IF NOT GREATER THAN TMINFIRE (NEW FIRE)

PROCEED WITH NEW FIRING BUT NOT LONGER THAN TMAXFIRE

MAXTFIRE CA TEMREG
EXTEND
MP 1/TMXFIR # I.E., 1/TMAXFIRE
EXTEND
MAXTST BZF NOMXFIRE # IF LESS THAN TMAXFIRE

CCS A
CAF TMAXFIRE # USE MAXIMUM
TCF +2
CS TMAXFIRE # USE MAXIMUM
TS TEMREG

SET UP SIGN OF REQUIRED TORQUE

NOMXFIRE CCS TEMREG # FOR TORQUE SIGN
CA POSMAX # POSITIVE TORQUE REQUIRED
TCF +2
CA NEGMAX # NEGATIVE TORQUE REQUIRED
TS ROLLFIRE # SET ROLLFIRE FOR + OR - TORQUE

COM
EXTEND
BZMF +3 # POSITIVE TORQUE REQUIRED
CS TEMREG
TS TEMREG

FIRELOOK CA I # IS IT MOREROLL
EXTEND
BZF FIREPLUG # YES
TCF JETROLL # MAG(T6FIRE) NOW IN TEMREG

FIREPLUG CAE TIME6 # CHECK FOR EXTENDED FIRING
EXTEND
SU TEMREG
EXTEND
EXTENTST BZMF TASKOVER # IF EXTENSION WANTED, DON'T, EXIT ROLL DAP
TCF JETROLL

```

July 12, 2016

1975

```
NOROLL CS ZERO # COAST...(NEG ZERO FOR TIME6)
 TS ROLLFIRE # NOTE, JETS CAN FIRE NEXT PASS
 TS TEMREG

JETROLL EXTEND
 DCA NOROL1T6

Page 995

 DXCH T6LOC
 CA TEMREG # ENTER JET FIRING TIME
 TS TIME6

 CA I # I=0 IF MOREROLL, KEEP SAME JETS ON
 EXTEND
SAMEJETS BZF TASKOVER # IF JETS ON KEEP SAME JETS. EXIT ROLL DAP

 CCS ROLLFIRE
 TCF +TORQUE
 TCF T6ENABL
 TCF -TORQUE
 TCF T6ENABL

PROCEED WITH + TORQUE

+TORQUE CA ROLLWORD # WHAT WAS THE LAST +TORQUE COMBINATION
 MASK BIT1 # WAS IT NO.9-11
 EXTEND
 BZF NO.9-11 # NOT 9-11, SO USE IT THIS TIME

NO.13-15 CS BIT1
 MASK ROLLWORD
 TS ROLLWORD # CHANGE BIT 1 TO ZERO
 CAF +ROLL2
 EXTEND
 WRITE CHAN6
 TCF T6ENABL

NO.9-11 CAF BIT1 # 1ST + JETS TO FIRE (MRCLEAN OS ROLLWORD)
 ADS ROLLWORD # CHANGE BIT 1 TO ONE
 CAF +ROLL1
 EXTEND
 WRITE CHAN6
 TCF T6ENABL

-TORQUE CA ROLLWORD # WHAT WAS LAST -TORQUE COMBINATION
 MASK BIT2 # WAS IT NO.12-10
```

1976

July 12, 2016

|                                        |        |               |                                            |
|----------------------------------------|--------|---------------|--------------------------------------------|
|                                        | EXTEND |               |                                            |
|                                        | BZF    | NO.12-10      | # NOT 12-10, SO USE IT THIS TIME           |
| NO.16-14                               | CS     | BIT2          |                                            |
|                                        | MASK   | ROLLWORD      |                                            |
|                                        | TS     | ROLLWORD      | # CHANGE BIT 2 TO ZERO                     |
|                                        | CAF    | -ROLL2        |                                            |
|                                        | EXTEND |               |                                            |
|                                        | WRITE  | CHAN6         |                                            |
|                                        | TCF    | T6ENABL       |                                            |
| NO.12-10                               | CAF    | BIT2          | # 1ST -JETS TO FIRE (MRCLEAN OS ROLLWORD)  |
| # Page 996                             | ADS    | ROLLWORD      | # CHANGE BIT 2 TO ONE                      |
|                                        | CAF    | -ROLL1        |                                            |
|                                        | EXTEND |               |                                            |
|                                        | WRITE  | CHAN6         |                                            |
| T6ENABL                                | CAF    | BIT15         |                                            |
|                                        | EXTEND |               |                                            |
|                                        | WOR    | CHAN13        |                                            |
| RDAPEND                                | TCF    | TASKOVER      | # EXIT ROLL DAP                            |
| # Page 997                             |        |               |                                            |
| # THIS T6 TASK SHUTS OFF ALL ROLL JETS |        |               |                                            |
| NOROLL1                                | LXCH   | BANKRUPT      | # SHUT OFF ALL (ROLL) JETS, (A T6 TASK     |
|                                        | CAF    | ZERO          | # CALLED BY "JETROLL")                     |
|                                        | TS     | ROLLFIRE      | # ZERO INDICATES JETS NOW OFF              |
|                                        | EXTEND |               |                                            |
| KILLJETS                               | WRITE  | CHAN6         |                                            |
|                                        | TCF    | NOQRSM        |                                            |
| # Page 998                             |        |               |                                            |
| # CONSTANTS FOR ROLL AUTOPILOT....     |        |               |                                            |
|                                        | EBANK= | BZERO         |                                            |
| NOROL1T6                               | 2CADR  | NOROLL1       |                                            |
| DB                                     | DEC    | .01388889     | # DEAD BAND (5 DEG), SC.AT B+0 REV         |
| -SLOPE                                 | DEC    | 0.2           | # -SWITCHLINE SLOPE(0.2 PER SEC) SC.AT B+0 |
|                                        |        |               | # PER SEC                                  |
| LMCRATE                                | DEC    | .00027778 B+4 | # LIMIT CYCLE RATE (0.1 DEG/SEC) SC.AT     |
|                                        |        |               | # B-4 REV/SEC                              |
| INTERCEP                               | DEC    | .0025 B+3     | # DB(-SLOPE) - LMCRATE, SC.AT B-3 REV/SC   |



July 12, 2016

1977

```
MINLIM DEC .00277778 B+4 # RATELIM,MIN (1DEG/SEC), SC.AT B-4 REV/SC

1/MINLIM DEC 360 B-18 # RECIPROCAL THEREOF, SHIFTED 14 RIGHT

MAXLIM DEC .01388889 B+4 # RATELIM,MAX (5DEG/SEC), SC.AT B-4 REV/SC

The following two were B+4.---RSB 2009.
TMINFIRE DEC 1.5 B-10 # 15 MS. (14MIN), SC.AT 16 BITS/CS

TMAXFIRE DEC 250 B-10 # 2.5 SEC, SC.AT 16 BITS/CS

1/TMXFIR = BIT3 # RECIPROCAL THEREOF, SHIFTED 14 RIGHT,
 # ROUNDS TO OCT00004, SO ALLOWS 2.56
 # SEC FIRINGS BEFORE APPLYING LIMIT
T6SCALE = PRI031 # (B+3) (16 BITS/CS) (100CS/SEC)

+ROLL1 = FIVE # ONBITS FOR JETS 9 AND 11
+ROLL2 = OCT120 # ONBITS FOR JETS 13 AND 15
-ROLL1 = TEN # ONBITS FOR JETS 12 AND 10
-ROLL2 OCT 240 # ONBITS FOR JETS 16 AND 14
```

This code is written to file `src/TVCROLLDAP.s`.

1978

July 12, 2016

## B.125 TVCSTROKETEST

```
1978 <src/TVCSTROKETEST.s 1978>≡
Copyright: Public domain.
Filename: TVCSTROKETEST.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 979-983
Mod history: 2009-05-13 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 979
NAME STROKE TEST PACKAGE (INCLUDING INITIALIZATION PACKAGE)
LOG SECTION...TVCSTROKETEST SUBROUTINE...DAPCSM
MODIFIED BY SCHLUNDT 21 OCTOBER 1968
#
FUNCTIONAL DESCRIPTION....
STROKE TEST PACKAGE GENERATES A WAVEFORM DESIGNED TO EXCITE BENDING
STRKTSTI (STROKE TEST INITIALIZATION) IS CALLED AS A JOB BY VB68.
IT INITIALIZES ALL ERASABLES REQD FOR A STROKE TEST, AND
THEN TESTS FOR CSM/LM (BIT 13 OF DAPDATR1). IF CSM/LM,
IN EITHER HIGH OR LOW-BANDWIDTH MODE, THE TEST IS STARTED
IMMEDIATELY. IF NOT CSM/LM, PROGRAM EXITS WITH NO ACTION.
HACK (STROKE TEST) GENERATES THE WAVEFORM BY DUMPING PULSE BURSTS
OF PROPER SIGN AND IN PROPER SEQUENCE DIRECTLY INTO
```

```

TVCPITCH, WORKING IN CONJUNCTION WITH BOTH PITCH AND YAW
TVC DAPS, WITH INTERMEDIATE WAITLIST CALLS. NOTE, HOWEVER
THAT THE STROKE TEST IS PERFORMED ONLY IN THE PITCH AXIS.
AN EXAMPLE WAVEFORM IS GIVEN BELOW, TO DEMONSTRATE STROKE-
TEST PARAMETER SELECTION.
RESTARTS CAUSE TEST TO BE TERMINATED. ANOTHER V68 REQD IF TEST
IS TO BE RE-RUN.
PULSE BURST SIZE IS PAD-LOADED (ESTROKER) SO THAT AMPLITUDE OF
WAVEFORM CAN BE CHANGED. THERE ARE TEN PULSE BURSTS IN
THE HALF-AMPLITUDE OF THE FIRST FREQUENCY SET IN THE
STANDARD WAVEFORM. AMPLITUDE IS 10(ESTROKER)(1/42.15),
NOMINALLY 50/42.15 = 1.185 DEG
#
CALLING SEQUENCE....
EXTENDED VERB 68 SETS UP STRKTSTI JOB
PITCH AND YAW TVCDAPS, FINDING STROKER NON-ZERO, DO A "TC HACK"
AN INTERNALLY-GENERATED WAITLIST CALL ENTERS AT "HACKWLST"
#
NORMAL EXIT MODES....
TC BUNKER ("Q" IF ENTRY FROM DAP, "TCTSKOVR" IF FROM WAITLIST) LIST
#
SUBROUTINES CALLED....
WAITLIST
#
ALARM OR ABORT EXIT MODES....
NONE
#
ERASABLE INITIALIZATION REQUIRED....
ESTROKER (PAD-LOAD)
STROKER, CADDY, REVS, CARD, N
#
OUTPUT....
STRKTSTI...INITIALIZATION FOR STROKE TEST
HACK, HACKWLST...PULSE BURSTS INTO TVCPITCH VIA "ADS"
RESETS STROKER = +0 WHEN TEST COMPLETED
#
DEBRIS....
N = CADDY = +0, CARD = -0, REVS = -1
BUNKER
Page 980
#
EXAMPLE STROKE TEST WAVE FORM, DEMONSTRATING PARAMETER SELECTION
#
NOTE....THIS IS NOT THE OFFICIAL WAVEFORM....
#
**
**

```

July 12, 2016

```
STROKE TEST INITIALIZATION PACKAGE (AS A JOB, FROM VERB 68)
```

July 12, 2016

1981

```
BANK 17
SETLOC DAPS2
BANK

COUNT* $$/STRK
EBANK= CADDY

STRKTSTI TCR TSTINIT # STROKE TEST INITIALIZATION PKG (CALLED
 # AS A JOB BY VERB68)

STRKCHK INHINT

CAE DAPDATR1 # CHECK FOR CSM/LM CONFIGURATION
MASK BIT14
EXTEND
BZF +3

CAE ESTROKER # BEGIN ON NEXT DAP PASS (PITCH OR YAW)
TS STROKER # (STROKING DONE IN PITCH ONLY, HOWEVER)

TCF ENDOFJOB

TSTINIT CS FCADDY # NORMAL ENTRY FROM STRKTSTI
TS CADDY
TS N # NOTE SGN CHNG FCADDY(+) TO CADDY(-)

CAF FREVS
TS REVS

CS FCARD # NOTE SGN CHNG FCARD(+) TO CARD(-)
TS CARD

TC Q # RETURN TO STRKTSTI+1 (OR CHKSTRK+2 OR +4)

Page 982
THE OFFICIAL STROKE TEST WAVEFORM (3 JAN, 1967) CONSISTS OF FOUR STROKE SETS, AS FOLLOWS....
#
SET 1...10 BURSTS IN 1/2 AMP, 4 REVERSALS
SET 2... 6 BURSTS IN 1/2 AMP, 6 REVERSALS
SET 3... 5 BURSTS IN 1/2 AMP, 10 REVERSALS
SET 4... 4 BURSTS IN 1/2 AMP, 14 REVERSALS
#
THE PULSE BURST SIZE (ESTROKER) IS PAD-LOADED (5 BITS AS OF 3 JAN, 1967)
THE REMAINING WAVEFORM-GENERATING PARAMETERS ARE AS FOLLOWS....

FCADDY DEC 10 # NO. PULSE BURSTS IN 1/2 AMP, SET1..(+10)
```

1982

July 12, 2016

|                                  |        |          |                                            |
|----------------------------------|--------|----------|--------------------------------------------|
| FREVS                            | DEC    | 3        | # NO. REVERSALS MINUS 1, SET1.....( 3)     |
| FCARD                            | DEC    | 4        | # NO. STROKE SETS.....(+ 4)                |
| FCARD1                           | DEC    | 5        | # NO. REVERSALS MINUS 1, SET2.....( 5)     |
| FCARD2                           | DEC    | 9        | # 3.....( 9)                               |
| FCARD3                           | DEC    | 13       | # 4.....( 13)                              |
| FCARD4                           | DEC    | 6        | # NO. PULSE BURSTS IN 1/2 AMP, SET2..(+ 6) |
| FCARD5                           | DEC    | 5        | # 3..(+ 5)                                 |
| FCARD6                           | DEC    | 4        | # 4..(+ 4)                                 |
| 20MS                             | =      | BIT2     |                                            |
| # STROKE TEST PACKAGE PROPER.... |        |          |                                            |
|                                  | EBANK= | BUNKER   |                                            |
| HACK                             | EXTEND |          | # ENTRY (IN T5 RUPT) FROM TVCDAPS          |
|                                  | QXCH   | BUNKER   | # SAVE Q FOR DAP RETURN                    |
|                                  | CAF    | 20MS     | # 2DAPSx2(PASSES/DAP)x2(CS/PASS)=8CS=TVCDT |
|                                  | TC     | WAITLIST |                                            |
|                                  | EBANK= | BUNKER   |                                            |
|                                  | 2CADR  | HACKWLST |                                            |
|                                  | TCF    | +3       |                                            |
| HACKWLST                         | CAF    | TCTSKOVR | # ENTRY FROM WAITLIST                      |
|                                  | TS     | BUNKER   | # BUNKER IS TC TASKOVER                    |
|                                  | CA     | STROKER  | # STROKE                                   |
|                                  | ADS    | TVCPITCH |                                            |
|                                  | CAF    | BIT11    | # RELEASE THE ERROR COUNTERS               |
|                                  | EXTEND |          |                                            |
|                                  | WOR    | CHAN14   |                                            |
|                                  | INCR   | CADDY    | # COUNT DOWN THE NO. BURSTS, THIS SLOPE    |
| # Page 982                       | CS     | CADDY    |                                            |
|                                  | EXTEND |          |                                            |
|                                  | BZMF   | +2       |                                            |
|                                  | TC     | BUNKER   | # EXIT, WHILE ON A SLOPE                   |
|                                  | CCS    | REVS     |                                            |

July 12, 2016

1983

|          |        |          |                                           |
|----------|--------|----------|-------------------------------------------|
|          | TCF    | REVUP    | # POSITIVE REVS                           |
|          | TCF    | REVUP +4 | # FINAL REVERSAL, THE SET                 |
|          | INCR   | CARD     | # NEGATIVE REVS SET LAST PASS, READY FOR  |
|          | CS     | CARD     | # THE NEXT SET. CHECK IF NO MORE SETS     |
|          | EXTEND |          |                                           |
|          | BZF    | STROKILL | # ALL SETS COMPLETED                      |
|          | INDEX  | CARD     |                                           |
|          | CAF    | FCARD +4 | # PICK UP NO. REVERSALS (-), NEXT SET     |
|          | TS     | REVS     | # REINITIALIZE                            |
|          | INDEX  | CARD     |                                           |
|          | CS     | FCARD +7 | # PICK UP NO. BURSTS IN 1/2AMP, NEXT SET  |
|          | TS     | N        | # REINITIALIZE                            |
|          | TS     | CADDY    |                                           |
|          | TC     | BUNKER   | # EXIT, AT END OF SET                     |
| STROKILL | TS     | STROKER  | # RESET (TO +0) TO END TEST               |
|          | TC     | BUNKER   | # EXIT, STROKE TEST FINIS                 |
| REVUP    | TS     | REVS     | # ALL REVERSALS EXCEPT LAST OF SET        |
|          | CA     | N        |                                           |
|          | DOUBLE |          | # 2 x 1/2AMP                              |
|          | TCF    | +4       |                                           |
| +4       | CS     | ONE      | # FINAL REVERSAL, THIS SET                |
|          | TS     | REVS     | # PREPARE TO BRANCH TO NEW BURST          |
|          | CA     | N        | # JUST RETURN TO ZERO, FINAL SLOPE OF SET |
|          | TS     | CADDY    | # CADUP                                   |
|          | CS     | STROKER  | # CHANGE SIGN OF SLOPE                    |
|          | TS     | STROKER  |                                           |
|          | TC     | BUNKER   | # EXIT AT A REVERSAL (SLOPE CHANGE)       |

This code is written to file `src/TVCSTROKETEST.s`.

## B.126 UPDATE PROGRAM

```

1984 <src/UPDATE-PROGRAM.s 1984>≡
 # Copyright: Public domain.
 # Filename: UPDATE_PROGRAM.agc
 # Purpose: Part of the source code for Comanche, build 055. It
 # is part of the source code for the Command Module's
 # (CM) Apollo Guidance Computer (AGC), Apollo 11.
 # Assembler: yaYUL
 # Reference: pp. 1497-1507
 # Contact: Ron Burkey <info@sandroid.org>
 # Website: http://www.ibiblio.org/apollo.
 # Mod history: 2009-05-07 RSB Adapted from Colossus249/UPDATE_PROGRAM.agc
 # and page images. Corrected various typos
 # in the transcription of program comments,
 # and these should be back-ported to
 # Colossus249.
 #
 # The contents of the "Comanche055" files, in general, are transcribed
 # from scanned documents.
 #
 # Assemble revision 055 of AGC program Comanche by NASA
 # 2021113-051. April 1, 1969.
 #
 # This AGC program shall also be referred to as Colossus 2A
 #
 # Prepared by
 #
 # Massachussets Institute of Technology
 # 75 Cambridge Parkway
 # Cambridge, Massachusetts
 #
 # under NASA contract NAS 9-4065.
 #
 # Refer directly to the online document mentioned above for further
 # information. Please report any errors to info@sandroid.org.

 # Page 1497
 # PROGRAM NAME: P27
 # WRITTEN BY: KILROY/ DE WOLF
 #
 # MOD NO: 0
 # MOD BY: KILROY
 # DATE: 01DEC67
 #
 # LOG SECTION: UPDATE PROGRAM.
 #

```



July 12, 2016

1985

```
FUNCT. DESCR.: P27 (THE UPDATE PROGRAM) PROCESSES COMMANDS AND DATA
INSERTIONS REQUESTED BY THE GROUND VIA UPLINK.
THE P27 PROGRAM WILL ACCEPT UPDATES
ONLY DURING P00 FOR THE LM, AND ONLY DURING P00,
P02, AND FRESH START FOR THE CSM.
#
CALLING SEQ: PROGRAM IS INITIATED BY UPLINK ENTRY OF VERBS 70, 71, 72, AND 73.
#
SUBROUTINES: TESTXACT, NEWMODEX, NEWMODEX +3, GOXDSPF, BANKCALL, FINDVAC, INTPRET, I
INTWAKEU, ENDEXT, POSTJUMP, FALTON, NEWPHASE, PHASCHNG
#
NORMAL EXIT: TC ENDEXT
#
ALARM/ABORT: TC FALTON FOLLOWED BY TC ENDEXT
#
RESTARTS: P27 IS RESTART PROTECTED IN TWO WAYS ...
1. PRIOR TO VERIFLAG INVERSION (WHICH IS CAUSED BY THE GROUND/ASTR
DATA BY SENDING A V33E WHEN V21N02 IS FLASHING)---
NO PROTECTION EXCEPT PRE-P27 MODE IS RESTROED, COAST + ALIGN DO
ACTIVITY LIGHT IS TURNED OFF. (JUST AS IF A V34E WAS SENT DURI
V70,V71,V72, OR V73 WILL HAVE TO BE COMPLETELY RESENT BY USER.
2. AFTER VERIFLAG INVERSION (WHEN UPDATE OF THE SPECIFIED ERASABLE
PROTECTED AGAINST RESTARTS.
#
DEBRIS: UPBUFF (20D) TEMP STORAGE FOR ADDRESSES AND CONTENTS.
UPVERB (1) VERB NUMBER MINUS 70D (E.G., FOR V72, UPVERB = 72D - 70
UPOLDMOD(1) FOR MAJOR MODE INTERRUPTED BY P27.
COMPNUMB(1) TOTAL NUMBER OF COMPONENTS TO BE TRANSMITTED.
UPCOUNT (1) ACTUAL NUMBER OF COMPONENTS RECEIVED.
UPTEMP (1) SCRATCH, BUT USUALLY CONTAINS COMPONENT NUMBER TO BE CH
#
INPUT:
#
ENTRY DESCRIPTION
V70EXXXXXEXXXXXE (LIFTOFF TIME INCREMENT) DOUBLE PRECISION OCTAL TIME INCREMENT,
IS ADDED TO TEPHEM, SUBTRACTED FROM AGC CLOCK(TIME2,TIME1), SUB
VECTOR TIME(TETCSM) AND SUBTRACTED FROM LEM STATE VECTOR TIME(T
THE DP OCTAL TIME INCREMENT IS SCALED AT 2(28).
Page 1498
V71EIIIEAAAAE (CONTIGUOUS BLOCK UPDATE) II-2 OCTAL COMPONENTS, XXXXX,
XXXXXE ARE LOADED INTO ERASABLE STARTING AT ECADR, AAAA.
XXXXXE IT IS .GE. 3 .AND. .LE. 20D.,
AND (AAAA + II -3) DOES NOT PRODUCE AN ADDRESS IN THE
9 NEXT BANK
.
V72EIIIE SCALING IS SAME AS INTERNAL REGISTERS.
(SCATTER UPDATE) (II-1)/2 OCTAL COMPONENTS, XXXXX, ARE
```

```

AAAAEXXXXXE LOADED INTO ERASABLE LOCATIONS, AAAA.
AAAAEXXXXXE II IS .GE. 3 .AND. .LE. 19D, AND MUST BE ODD.
. SCALING IS SAME AS INTERNAL REGISTERS.
#
V73EXXXXXEXXXXXE (OCTAL CLOCK INCREMENT) DOUBLE PRECISION OCTAL TIME
INCREMENT XXXXX XXXXX, IS ADDED TO THE AGC CLOCK, IN
CENTISECONDS SCALED AT (2)28).
THIS LOAD IS THE OCTAL EQUIVALENT OF V55.
#
OUTPUT: IN ADDITION TO THE ABOVE REGISTER LOADS, ALL UPDATES
COMPLEMENT BIT3 OF FLAGWORD7.
#
ADDITIONAL NOTES: VERB 71, JUST DEFINED ABOVE WILL BE USED TO PERFORM BUT NOT I
1. CSM/LM STATE VECTOR UPDATE
2. REFSMMAT UPDATE
#
THE FOLLOWING COMMENTS DELINEATE EACH SPECIAL UPDATE ---
#
1. CSM/LM STATE VECTOR UPDATE (ALL DATA ENTRIES IN OCTAL)
#
ENTRIES: DATA DEFINITION: SCALING
V71E CONTIGUOUS BLOCK UPDATE VERB
21E NUMBER OF COMPONENTS FOR STATE VECTOR UPDATE
AAAAE ECADR OF 'UPSVFLAG'
XXXXXE STATE VECTOR IDENTIFIER: 00001 FOR CSM, 77776 FOR LEN
00002 FOR CSM, 77775 FOR LEN
#
XXXXXEXXXXXE X POSITION
XXXXXEXXXXXE Y POSITION
XXXXXEXXXXXE Z POSITION
XXXXXEXXXXXE X VELOCITY
XXXXXEXXXXXE Y VELOCITY
XXXXXEXXXXXE Z VELOCITY
XXXXXEXXXXXE TIME FROM AGC CLOCK ZERO
V33E VERB 33 TO SIGNAL THAT THE STATE VECTOR IS READY TO P
#
2. REFSMMAT (ALL DATA ENTRIES IN OCTAL)
#
ENTRIES DATA DEFINITIONS SCALING
#
Page 1499
V71E CONTIGUOUS BLOCK UPDATE VERB
24E NUMBER OF COMPONENTS FOR REFSMMAT UPDATE
AAAAE ECADR OF 'REFSMMAT'
XXXXXEXXXXXE ROW 1 COLUMN 1 2(-1)
XXXXXEXXXXXE ROW 1 COLUMN 2 2(-1)
XXXXXEXXXXXE ROW 1 COLUMN 3 2(-1)
XXXXXEXXXXXE ROW 2 COLUMN 1 2(-1)
XXXXXEXXXXXE ROW 2 COLUMN 2 2(-1)
XXXXXEXXXXXE ROW 2 COLUMN 3 2(-1)

```

July 12, 2016

1987

|   |              |                                                       |       |
|---|--------------|-------------------------------------------------------|-------|
| # | XXXXXEXXXXXE | ROW 3 COLUMN 1                                        | 2(-1) |
| # | XXXXXEXXXXXE | ROW 3 COLUMN 2                                        | 2(-1) |
| # | XXXXXEXXXXXE | ROW 3 COLUMN 3                                        | 2(-1) |
| # | V33E         | VERB 33 TO SIGNAL THAT REFSMMAT IS READY TO BE STORED |       |

BANK 07  
SETLOC EXTVERBS  
BANK

EBANK= TEPHEM

|             |        |             |                                                                                        |
|-------------|--------|-------------|----------------------------------------------------------------------------------------|
| V70UPDAT    | COUNT* | \$\$/P27    |                                                                                        |
|             | CAF    | UP70        | # COMES HERE ON V70E                                                                   |
|             | TCF    | V73UPDAT +1 |                                                                                        |
| V71UPDAT    | CAF    | UP71        | # COMES HERE ON V71E                                                                   |
|             | TCF    | V73UPDAT +1 |                                                                                        |
| V72UPDAT    | CAF    | UP72        | # COMES HERE ON V72E                                                                   |
|             | TCF    | V73UPDAT +1 |                                                                                        |
| V73UPDAT    | CAF    | UP73        | # COMES HERE ON V73E                                                                   |
| +1          | TS     | UPVERBSV    | # SAVE UPVERB UNTIL IT'S OK TO ENTER P27                                               |
|             | TC     | TESTXACT    | # GRAB DISPLAY IF AVAILABLE, OTHERWISE<br># TURN *OPERATOR ERROR* ON AND TERMINATE JOB |
|             | CA     | MODREG      | # CHECK IF UPDATE ALLOWED                                                              |
|             | EXTEND |             | # FIRST CHECK FOR MODREG = +0, -0                                                      |
|             | BZF    | +2          | # (+0 = P00, -0 = FRESHSTART)                                                          |
|             | TC     | CKMDMORE    | # NOW CHECK FOR PROGRAM WHICH CAN BE<br># INTERRUPTED BY P27.                          |
|             | CAE    | MODREG      | # UPDATE ALLOWED                                                                       |
|             | TS     | UPOLDMOD    | # SAVE CURRENT MAJOR MODE                                                              |
| # Page 1500 | CAE    | UPVERBSV    | # SET UPVERB TO INDICATE TO P27                                                        |
|             | TS     | UPVERB      | # WHICH EXTENDED VERB CALLED IT.                                                       |
|             | CAF    | ONE         |                                                                                        |
|             | TS     | UPCOUNT     | # INITIALIZE UPCOUNT TO 1                                                              |
|             | TC     | POSTJUMP    | # LEAVE EXTENDED VERB BANK AND                                                         |
|             | CADR   | UPPART2     | # GO TO UPDATE PROGRAM (P27) BANK.                                                     |

1988

July 12, 2016

|             |        |             |                                            |
|-------------|--------|-------------|--------------------------------------------|
| CKMDMORE    | CS     | FLAGWRD5    |                                            |
|             | MASK   | BIT8        | # CHECK IF COMPUTER IS LGC                 |
|             | CCS    | A           | # IS COMPUER LGC OR AGC                    |
| UPERLEM     | TCF    | UPERROR     | # ERROR: IT'S THE LEM + MODE IS NOT P00.   |
|             | CS     | TWO         |                                            |
|             | MASK   | MODREG      |                                            |
|             | CCS    | A           |                                            |
| UPERCMC     | TCF    | UPERROR     | # ERROR: IT'S THE CMC AND MODE IS NOT      |
|             |        |             | # P00 OR P02.                              |
|             | TC     | Q           | # ALLOW UPDATE TO PROCEED.                 |
| UPERROR     | TC     | POSTJUMP    | # TURN ON 'OPERATOR ERROR' LIGHT           |
|             | CADR   | UPERROUT +2 | # GO TO COMMON UPDATE PROGRAM EXIT         |
|             | SBANK= | LOWSUPER    |                                            |
| UP70        | EQUALS | ZERO        |                                            |
| UP71        | EQUALS | ONE         |                                            |
| UP72        | EQUALS | TWO         |                                            |
| UP73        | EQUALS | THREE       |                                            |
|             | BANK   | 04          |                                            |
|             | SETLOC | UPDATE2     |                                            |
|             | BANK   |             |                                            |
|             | COUNT* | \$\$/P27    |                                            |
| UPPART2     | EQUALS |             | # UPDATE PROGRAM -- PART 2                 |
|             | TC     | PHASCHNG    | # SET RESTART GROUP 6 TO RESTORE OLD MODE  |
|             | OCT    | 07026       | # AND DOWNLIST AND EXIT IF RESTART OCCURS. |
|             | OCT    | 30000       | # PRIORITY SAME AS CHRPRIO                 |
|             | EBANK= | UPBUFF      |                                            |
|             | 2CADR  | UPOUT +1    |                                            |
|             | CAF    | ONE         |                                            |
|             | TS     | DNLSTCOD    | # DOWNLIST                                 |
|             | TC     | NEWMODEX    | # SET MAJOR MODE = 27                      |
| # Page 1501 | DEC    | 27          |                                            |
|             | INDEX  | UPVERB      | # BRANCH DEPENDING ON WHETHER THE UPDATE   |
|             | TCF    | +1          | # VERB REQUIRES A FIXED OR VARIABLE NUMBER |
|             | TCF    | +3          | # V70 FIXED (OF COMPONENTS)                |
|             | TCF    | OHWELL1     | # V71 VARIABLE -- GO GET NO. OF COMPONENTS |
|             | TCF    | OHWELL1     | # V72 VARIABLE -- GO GET NO. OF COMPONENTS |

July 12, 2016

1989

```

 CA TWO # V73 (AND V70) FIXED
 TS COMPNUMB # SET NUMBER OF COMPONENTS TO 2.
 TCF OHWELL2 # GO GET THE TWO UPDATE COMPONENTS

OHWELL1 CAF ADUPBUFF # * REQUEST USER TO SEND NUMBER *
 TS MPAC +2 # * OF COMPONENTS PARAMETER(II) *
 +2 CAF UPLOADNV # (CKV432 RETURNS HERE IF V32 ENCOUNTERED)
 TC BANKCALL # DISPLAY A FLASHING V21N01
 CADR GOXDSPF # TO REQUEST II.
 TCF UPOUT4 # V32 TERMINATE UPDATE (P27) RETURN
 TCF OHWELL1 +2
 TC CK4V32 # DATA OR V32 RETURN
 CS BIT2
 AD UPBUFF # IS II (NUMBER OF COMPONENTS PARAMETER)
 EXTEND # .GE. 3 AND .LE. 20D.
 BZMF OHWELL1 +2
 CS UPBUFF
 AD UP21
 EXTEND
 BZMF OHWELL1 +2
 CAE UPBUFF
 TS COMPNUMB # SAVE II IN COMPNUMB

UPBUFF LOADING SEQUENCE

OHWELL2 INCR UPCOUNT # INCREMENT COUNT OF COMPONENTS RECEIVED.
 CAF ADUPBFM1 # CALCULATE LOCATION (ECADR) IN UPBUFF
 AD UPCOUNT # WHERE NEXT COMPONENT SHOULD BE STORED
 +2 TS MPAC +2 # PLACE ECADR INTO R3.
 +3 CAF UPLOADNV # (CK4V32 RETURNS HERE IF V32 ENCOUNTERED)
 TC BANKCALL # DISPLAY A FLASHING V21N01
 CADR GOXDSPF # TO REQUEST DATA.
 TCF UPOUT4 # V34 TERMINATE UPDATE (P27) RETURN.
 TCF OHWELL2 +3 # V33 PROCEED RETURN
 TC CK4V32 # DATA OR V32 RETURN
 CS UPCOUNT # HAVE WE FINISHED RECEIVING ALL
 AD COMPNUMB # THE DATA WE EXPECTED.
 EXTEND
 BZMF UPVERIFY # YES -- GO TO VERIFICATION SEQUENCE
 TCF OHWELL2 -1 # NO -- REQUEST ADDITIONAL DATA.

Page 1502
VERIFY SEQUENCE
UPVERIFY CAF ADUPTMP # PLACE ECADR WHERE COMPONENT NO. INDEX
 TS MPAC +2 # IS TO BE STORED INTO R3.
 CAF UPVRFYNV # (CK4V32 RETURNS HERE IF V32 ENCOUNTERED)
```

|                                                    |                 |                                            |
|----------------------------------------------------|-----------------|--------------------------------------------|
| TC                                                 | BANKCALL        | # DISPLAY A FLASHING V21N02 TO REQUEST     |
| CADR                                               | GOXDSPF         | # DATA CORRECTION OR VERIFICATION.         |
| TCF                                                | UPOUT4          | # V34 TERMINATE UPDATE (P27) RETURN        |
| TCF                                                | UPSTORE         | # V33 DATA SENT IS GOOD. GO STORE IT.      |
| TC                                                 | CK4V32          | # COMPONENT NO. INDEX OR V32 RETURN        |
| CA                                                 | UPTMP           | # DOES THE COMPONENT NO. INDEX JUST SENT   |
| EXTEND                                             |                 | # SPECIFY A LEGAL COMPONENT NUMBER?        |
| BZMF                                               | UPVERIFY        | # NO, IT IS NOT POSITIVE NONZERO           |
| CS                                                 | UPTMP           |                                            |
| AD                                                 | COMPNUMB        |                                            |
| AD                                                 | BIT1            |                                            |
| EXTEND                                             |                 |                                            |
| BZMF                                               | UPVERIFY        | # NO                                       |
| CAF                                                | ADUPBFM1        | # YES -- BASED ON THE COMPONENT NO. INDEX  |
| AD                                                 | UPTMP           | # CALCULATE THE ECADR OF LOCATION IN       |
| TCF                                                | OHWELL2 +2      | # UPBUFF WHICH USER WANTS TO CHANGE.       |
| UPOUT4                                             | EQUALS UPOUT +1 | # COMES HERE ON V34 TC TERMINATE UPDATE.   |
| # CHECK FOR VERB 32 SEQUENCE                       |                 |                                            |
| CK4V32                                             | CS MPAC         | # ON DATA RETURN FROM 'GOXDSPF'            |
|                                                    | MASK BIT6       | # ON DATA RETURN FROM "GOXDSP" & THE CON-  |
|                                                    | CCS A           | # TENTS OF MPAC = VERB. SO TEST FOR V32.   |
|                                                    | TC Q            | # IT'S NOT A V32, IT'S DATA. PROCEED.      |
|                                                    | INDEX Q         |                                            |
|                                                    | TC 0 -6         | # V32 ENCOUNTERED -- GO BACK AND GET DATA  |
| ADUPTMP                                            | ADRES UPTMP     | # ADDRESS OF TEMP STORAGE FOR CORRECTIONS  |
| ADUPBUFF                                           | ADRES UPBUFF    | # ADDRESS OF UPDATE DATA STORAGE BUFFER    |
| UPLOADNV                                           | VN 2101         | # VERB 21 NOUN 01                          |
| UPVRFYNV                                           | VN 2102         | # VERB 21 NOUN 02                          |
| UP21                                               | = MD1           | # DEC 21 = MAX NO OF COMPONENTS +1         |
| UPDTPHAS                                           | EQUALS FIVE     |                                            |
| # PRE-STORE AND FAN TO APPROPRIATE BRANCH SEQUENCE |                 |                                            |
| UPSTORE                                            | EQUALS          | # GROUND HAS VERIFIED UPDATE. STORE DATA.  |
|                                                    | INHINT          |                                            |
|                                                    | CAE FLAGWRD7    | # INVERT VERIFLAG (BIT 3 OF FLAGWRD7) TO   |
|                                                    | XCH L           | # INDICATE TO THE GROUND (VIA DOWNLINK)    |
|                                                    | CAF BIT3        | # THAT THE V33 (WHICH THE GROUND SENT TO   |
| # Page 1503                                        | EXTEND          | # VERIFY THE UPDATE) HAS BEEN SUCCESSFULLY |

July 12, 2016

1991

```

RXOR LCHAN # RECEIVED BY THE UPDATE PROGRAM
TS FLAGWRD7

TC PHASCHNG # SET RESTART GROUP 6 TO REDO THE UPDATE
OCT 04026 # DATA STORE IF A RESTART OCCURS.
INHINT # (BECAUSE PHASCHNG DID A RELINT)

CS TWO # GO TO UPFNDVAC IF INSTALL IS REQUIRED.
AD UPVERB # THAT IS, IF IT'S A V70 - V72.
EXTEND # GO TO UPEND73 IF IT'S A V73.
BZMF UPFNDVAC

VERB 73 BRANCH

UPEND73 EXTEND # V73 -- PERFORM DP OCTAL AGC CLOCK INCREMENT
 DCA UPBUFF
 DXCH UPBUFF +8D
 TC TIMEDIDL
 TC FALTON # ERROR -- TURN ON *OPERATOR ERROR* LIGHT
 TC UPOUT +1 # GO TO COMMON UPDATE PROGRAM EXIT

UPFNDVAC CAF CHRPRIO # (USE EXTENDED VERB PRIORITY)
 TC FINDVAC # GET VAC AREA FOR 'CALL INTSTALL'
 EBANK= TEPHEM
 2CADR UPJOB # (NOTE: THIS WILL ALSO SET EBANK FOR
 TC ENDOFJOB # 'TEPHEM' UPDATE BY V70)

UPJOB TC INTERPRET # THIS COULD BE A STATE VECTOR UPDATE -- SO
 CALL INTSTALL # WAIT (PUT JOB TO SLEEP) IF ORBIT INT(OI)
 # IS IN PROGRESS -- OR -- GRAB OI AND RETURN
 # TO UPWAKE IF OI IS NOT IN PROGRESS.

UPWAKE EXIT

 TC PHASCHNG # RESTART PROTECT (GROUP 6)
 OCT 04026

 TC UPFLAG # SET INTEGRATION RESTART BIT
 ADRES REINTFLG
 INHINT
UPPART3 EQUALS

 INDEX UPVERB # BRANCH TO THE APPROPRIATE UPDATE VERB
 TCF +1 # ROUTINE TO ACTUALLY PERFORM THE UPDATE
 TCF UPEND70 # V70
 TCF UPEND71 # V71
```

```

TCF UPEND72 # V72

Page 1504
ROUTINE TO INCREMENT CLOCK (TIME2,TIME1) WITH CONTENTS OF DP WORD AT UPBUFF.
TIMEDIDL EXTEND
 QXCH UTEMP # SAVE Q FOR RETURN
 CAF ZERO # ZERO AND SAVE TIME2,TIME1
 ZL
 DXCH TIME2
 DXCH UPBUFF +18D # STORE IN CASE OF OVERFLOW

 CAF UPDTPHAS # DO
 TS L # A
 COM
 DXCH -PHASE6 # PHASCHNG

TIMEDIDR INHINT
 CAF ZERO
 ZL
 TS MPAC +2 # PICK UP INCRMENTER (AND ZERO
 DXCH UPBUFF +8D # IT IN CASE OF RESTARTS) AND
 DXCH MPAC # STORE IT
 # INTO MPAC FOR TPAGREE.

 EXTEND
 DCA UPBUFF +18D
 DAS MPAC # FORM SUM IN MPAC
 EXTEND
 BZF DELTAOK # TEST FOR OVERFLOW
 CAF ZERO
 DXCH UPBUFF +18D # OVERFLOW, RESTORE OLD VALUE OF CLOCK
 DAS TIME2 # AND TURN ON OPERATOR ERROR

 TC PHASCHNG # RESTART PROTECT (GROUP 6)
 OCT 04026

 TC UTEMP # GO TO ERROR EXIT

DELTAOK TC TPAGREE # FORCE SIGN AGREEMENT
 DXCH MPAC
 DAS TIME2 # INCREMENT TIME2,TIME1

 TC PHASCHNG # RESTART PROTECT (GROUP 6)
 OCT 04026

 INHINT
 INDEX UTEMP # (CODED THIS WAY FOR RESTART PROTECTION)

```



```

TC 1 # NORMAL RETURN

VERB 71 BRANCH

UPEND71 CAE UPBUFF +1 # SET EBANK
 TS EBANK # AND

Page 1505

 MASK LOW8 # CALCULATE
 TS UPTMP # S-REG VALUE OF RECEIVING AREA
 AD NEG3 # IN THE PROCESS OF
 AD COMPNUMB # PERFORMING
 EXTEND # THIS UPDATE
 BZF STORLP71 # WILL WE
 MASK BIT9 # OVERFLOW
 CCS A # INTO THE NEXT EBANK...
 TCF UPERROUT # YES

 CA NEG3 # NO -- CALCULATE NUMBER OF
 AD COMPNUMB # WORDS TO BE STORED MINUS ONE
STORLP71 TS MPAC # SAVE NO. OF WORDS REMAINING MINUS ONE
 INDEX A # TAKE NEXT UPDATE WORD FROM
 CA UPBUFF +2 # UPBUFF AND
 TS L # SAVE IT IN L
 CA MPAC # CALCULATE NEXT
 AD UPTMP # RECEIVING ADDRESS
 INDEX A
 EBANK= 1400
 LXCH 1400 # UPDATE THE REGISTER BY CONTENTS OF L
 EBANK= TEPHEM
 CCS MPAC # ARE THERE ANY WORDS LEFT TO BE STORED
 TCF STORLP71 # YES
 TCF UPOUT # NO -- THEN EXIT UPDATE PROGRAM
ADUPBFM1 ADRES UPBUFF -1 # SAME AS ADUPBUFF BUT LESS 1 (DON'T MOVE)
 TCF UPOUT # NO -- EXIT UPDATE (HERE WHEN COMPNUMB = 3)

VERB 72 BRANCH

UPEND72 CAF BIT1 # HAVE AN ODD NO. OF COMPONENTS
 MASK COMPNUMB # BEEN SENT FOR A V72 UPDATE ...
 CCS A
 TCF +2 # YES
 TCF UPERROUT # ERROR -- SHOULD BE ODD NO. OF COMPONENTS
 CS BIT2
 AD COMPNUMB
LDLOOP72 TS MPAC # NOW PERFORM THE UPDATE
 INDEX A

```

1994

July 12, 2016

```

CAE UPBUFF +1 # PICK UP NEXT UPDATE WORD
LXCH A
CCS MPAC # SET POINTER TO ECADR (MUST BE CCS)
TS MPAC
INDEX A
CAE UPBUFF +1 # PICK UP NEXT ECADR OF REG TO BE UPDATED
TS EBANK # SET EBANK
MASK LOW8 # ISOLATE RELATIVE ADDRESS
INDEX A

Page 1506
EBANK= 1400
LXCH 1400 # UPDATE THE REGISTER BY CONTENTS OF L
EBANK= TEPHEM
CCS MPAC # ARE WE THOROUGH THE V72 UPDATE...
TCF LDLOOP72 # NO

NORMAL FINISH OF P27

UPOUT EQUALS
 TC INTWAKEU # RELEASE GRAB OF ORBITAL INTEGRATION
 +1 CAE UPOLDMOD # RESTORE PRIOR P27 MODE
 TC NEWMODEX +3
 CAF ZERO
 TS DNLSTCOD
 TC UPACTOFF # TURN OFF 'UPLINK ACTIVITY' LIGHT
 EXTEND # KILL GROUP 6
 DCA NEG0
 DXCH -PHASE6

 TC ENDEXT # EXTENDED VERB EXIT

VERB TO BRANCH

UPEND70 EXTEND # V70 DOES THE FOLLOWING WITH DP DELTA
 DCS UPBUFF # TIME IN UPBUFF
 DXCH UPBUFF +8D
 TC TIMEDIDL # DECREMENT AGC CLOCK

 TC UPERROUT # ERROR WHILE DECREMENTING CLOCK -- EXIT

 EBANK= TEPHEM
 EXTEND
 DCS UPBUFF # COPY DECREMENTERS FOR
 DXCH UPBUFF +10D # RESTART PROTECTION
 EXTEND

```

July 12, 2016

1995

```
DCS UPBUFF
DXCH UPBUFF +12D

TC PHASCHNG # RESTART PROTECT (GROUP 6)
OCT 04026

CAF ZERO
ZL
DXCH UPBUFF +10D # DECREMENT CSM STATE VECTOR TIME
DAS TETCSM

CAF ZERO
```

# Page 1507

```
ZL
DXCH UPBUFF +12D # DECREMENT LEM STATE VECTOR TIME
DAS TETLEM
CAF ZERO
ZL
DXCH UPBUFF
DAS TEPHEM +1 # INCREMENT TP TEPHEM
ADS TEPHEM

TC PHASCHNG # RESTART PROTECT (GROUP 6)
OCT 04026

EBANK= UPBUFF

TC UPOUT # GO TO STANDARD UPDATE PROGRAM EXIT
```

# ERROR SEQUENCE

```
UPERROUT TC FALTON # TURN ON *OPERATOR ERROR* LIGHT
 TCF UPOUT # GO TO COMMON UPDATE PROGRAM EXIT

 +2 TC FALTON # TURN ON 'OPERATOR ERROR' LIGHT
 TC UPACTOFF # TURN OFF 'UPLINK ACTIVITY' LIGHT
 TC ENDEXT # EXTENDED VERB EXIT
 # (THE PURPOSE OF UPERROUT +2 EXIT IS
 # TO PROVIDE AN ERROR EXIT WHICH DOES NOT
 # RESET ANY RESTART GROUPS)
```

# 'UPACTOFF' IS A ROUTINE TO TURN OFF UPLINK ACTIVITY LIGHT ON ALL EXITS FROM UPDATE PROGRAM (P

```
UPACTOFF CS BIT3
 EXTEND # TURN OFF UPLINK ACTIVITY LIGHT
```

1996

July 12, 2016

|      |          |                         |
|------|----------|-------------------------|
| WAND | DSALMOUT | # (BIT 3 OF CHANNEL 11) |
| TC   | Q        |                         |

This code is written to file `src/UPDATE-PROGRAM.s`.

July 12, 2016

1997

## B.127 WAITLIST

1997

*<src/WAITLIST.s 1997>*≡

```
Copyright: Public domain.
Filename: WAITLIST.agc
Purpose: Part of the source code for Colossus 2A, AKA Comanche 055.
It is part of the source code for the Command Module's (CM)
Apollo Guidance Computer (AGC), for Apollo 11.
Assembler: yaYUL
Contact: Ron Burkey <info@sandroid.org>.
Website: www.ibiblio.org/apollo.
Pages: 1221-1235
Mod history: 2009-05-14 RSB Adapted from the Colossus249/ file of the
same name, using Comanche055 page images.
#
This source code has been transcribed or otherwise adapted from digitized
images of a hardcopy from the MIT Museum. The digitization was performed
by Paul Fjeld, and arranged for by Deborah Douglas of the Museum. Many
thanks to both. The images (with suitable reduction in storage size and
consequent reduction in image quality as well) are available online at
www.ibiblio.org/apollo. If for some reason you find that the images are
illegible, contact me at info@sandroid.org about getting access to the
(much) higher-quality images which Paul actually created.
#
Notations on the hardcopy document read, in part:
#
Assemble revision 055 of AGC program Comanche by NASA
2021113-051. 10:28 APR. 1, 1969
#
This AGC program shall also be referred to as
Colossus 2A
#
Page 1221
PROGRAM DESCRIPTION DATE -- 10 OCTOBER 1966
MOD NO -- 2 LOG SECTION -- WAITLIST
MOD BY -- MILLER (DTMAX INCREASED TO 162.5 SEC) ASSEMBLY -- SUNBURST REV 5
MOD 3 BY KERNAN (INHINT INSERTED AT WAITLIST) 2/28/68 SKIPPER REV 4
MOD 4 BY KERNAN (TWIDDLE IN 54) 3/28/68 SKIPPER REV 13.
#
FUNCTIONAL DESCRIPTION --
PART OF A SECTION OF PROGRAMS -- WAITLIST, TASKOVER, T3RUPT, USED TO CALL A PROGRAM (CA
WHICH IS TO BEGIN IN C(A) CENTISECONDS. WAITLIST UPDATES TIME3, LST1, AND LST2. THE M
FOLLOW.
#
C(TIME3) = 16384 -(T1-T) CENTISECONDS, (T=PRESENT TIME, T1=TIME FOR TASK1)
#
```

July 12, 2016

[illegible]

July 12, 2016

1999

```
#
OUTPUT --
LST1 AND LST2 UPDATED WITH NEW TASK AND ASSOCIATED TIME.
#
DEBRIS --
CENTRALS -- A,Q,L
OTHER -- WAITEXIT, WAITADR, WAITTEMP, WAITBANK
#
DETAILED ANALYSIS OF TIMING --
CONTROL WILL NOT BE RETURNED TO THE SPECIFIED ADDRESS (2CADR) IN EXACTLY DELTA T CENTIS
THE APPROXIMATE TIME MAY BE CALCULATED AS FOLLOWS:
LET TO = THE TIME OF THE TC WAITLIST
LET TS = TO +147U + COUNTER INCREMENTS (SET UP TIME)
LET X = TS -(100TS)/100 (VARIANCE FROM COUNTERS)
LET Y = LENGTH OF TIME OF INHIBIT INTERRUPT AFTER T3RUPT
LET Z = LENGTH OF TIME TO PROCESS TASKS WHICH ARE DUE THIS T3RUPT BUT DISPATCH
(Z=0, USUALLY).
LET DELTD = THE ACTUAL TIME TAKEN TO GIVE CONTROL TO 2CADR
THEN DELTD = TS+DELTA T -X +Y +Z +1.05MS* +COUNTERS*
*THE TIME TAKEN BY WAITLIST ITSELF AND THE COUNTER TICKING DURING THIS WAITLIST
IN SHORT, THE ACTUAL TIME TO RETURN CONTROL TO A 2CADR IS AUGMENTED BY THE TIME TO SET
INTERRUPT, ALL COUNTERS TICKING, THE T3RUPT PROCESSING TIME, THE WAITLIST PROCESSING TI
OF OTHER TASKS INHIBITING THE INTERRUPT.
```

BLOCK 02

# Page 1223

EBANK= LST1 # TASK LISTS IN SWITCHED E BANK.

COUNT 02/WAIT

TWIDDLE

INHINT

TS L

# SAVE DELAY TIME IN L

CA POSMAX

ADS Q

# CREATING OVERFLOW AND Q-1 IN Q

CA BBANK

EXTEND

ROR SUPERBNK

XCH L

WAITLIST

INHINT

EXTEND

BZMF WATLSTO-

XCH Q

SAVE DELTA T IN Q AND RETURN IN  
WAITEXIT.

TS WAITEXIT

EXTEND

INDEX WAITEXIT

# IF TWIDDLING, THE TS SKIPS TO HERE

2000

July 12, 2016

```

 DCA 0 # PICK UP 2CADR OF TASK.
 TS WAITADR # BBCON WILL REMAIN IN L
DLY2 -1 CAF WAITBB # ENTRY FROM FIXDELAY AND VARDELAY.
 XCH BBANK
 TCF WAIT2

WATLSTO- TC POODOO
 OCT 1204 # WAITLIST CALL WITH ZERO OR NEG DT

RETURN TO CALLER AFTER TASK INSERTION:

LVWTLIST DXCH WAITEXIT
 AD TWO
 DTCB

 EBANK= LST1
WAITBB BBCON WAIT2

RETURN TO CALLER +2 AFTER WAITING DT SPECIFIED AT CALLER +1.

FIXDELAY INDEX Q # BOTH ROUTINES MUST BE CALLED UNDER
 # Was CAF --- RSB 2004.
 CA 0 # WAITLIST CONTROL AND TERMINATE THE TASK
 INCR Q # IN WHICH THEY WERE CALLED.

RETURN TO CALLER +1 AFTER WAITING THE DT AS ARRIVING IN A.

VARDELAY XCH Q # DT TO Q. TASK ADRES TO WAITADR.
 TS WAITADR
 CA BBANK # BBANK IS SAVED DURING DELAY.
 EXTEND

Page 1224
 ROR SUPERBNK # ADD SBANK TO BBCON.
 TS L
 CAF DELAYEX
 TS WAITEXIT # GO TO TASKOVER AFTER TASK ENTRY.
 TCF DLY2

DELAYEX TCF TASKOVER -2 # RETURNS TO TASKOVER.

Page 1225
ENDTASK MUST ENTERED IN FIXED-FIXED SO IT IS DISTINGUISHABLE BY ITS ADRES ALONE.

 EBANK= LST1
ENDTASK -2CADR SVCT3
```



July 12, 2016

2001

```
SVCT3 CCS FLAGWRD2 # DRIFT FLAG
 TCF TASKOVER
 TCF TASKOVER
 TCF +1

 CCS IMUCADR # DON'T DO NBDONLY IF SOMEONE ELSE IS IN
 TCF SVCT3X
 TCF +3
 TCF SVCT3X
 TCF SVCT3X
+3 CAF PRI035 # COMPENSATE FOR NBD COEFFICIENTS ONLY.
 TC NOVAC # ENABLE EVERY 81.93 SECONDS
EBANK= NBDX
2CADR NBDONLY

 TCF TASKOVER

SETLOC FFTAG6
BANK

SVCT3X TC FIXDELAY # DELAY MAX OF 2 TIMES FOR IMU ZERO
 DEC 500
 TC SVCT3

Page 1226
BEGIN TASK INSERTION.

 BANK 01
 COUNT 01/WAIT

WAIT2 TS WAITBANK # BBANK OF CALLING PROGRAM.
 CS TIME3
 AD BIT8 # BIT 8 = OCT 200
 CCS A # TEST 200 - C(TIME3). IF POSITIVE,
 # IT MEANS THAT TIME3 OVERFLOW HAS OCCURRED PRIOR TO CS
 # C(TIME3) = T - T1, INSTEAD OF 1.0 - (T1 - T). THE FC
 # ORDERS SET C(A) = TD - T1 + 1 IN EITHER CASE.

 AD OCT40001 # OVERFLOW HAS OCCURRED. SET C(A) =
 CS A # T - T1 + 1.0 - 201

NORMAL CASE (C(A) NNZ) YIELDS SAME C(A): -(-(1.0-(T1-T)) + 200) - 1

 AD OCT40201
 AD Q # RESULT = TD - T1 + 1.
```

2002

July 12, 2016

```
CCS A # TEST TD - T1 +1.

AD LST1 # IF TD - T1 POS, GO TO WTLST5 WITH
TCF WTLST5 # C(A) = (TD - T1) + C(LST1) = TD-T2+1

NOOP
CS Q
```

```
NOTE THAT THIS PROGRAM SECTION IS NEVER ENTERED WHEN T-T1 G/E -1,
SINCE TD-T1+1 = (TD-T) + (T-T1+1), AND DELTA T = TD-T G/E +1. (G/E
SYMBOL MEANS GREATER THAN OR EQUAL TO). THUS THERE NEED BE NO CON-
CERN OVER A PREVIOUS OR IMMINENT OVEFLOW OF TIME3 HERE.
```

```
AD POS1/2 # WHEN TD IS NEXT, FORM QUANTITY
AD POS1/2 # 1.0 - DELTA T = 1.0 - (TD - T)
XCH TIME3
AD NEGMAX
AD Q # 1.0 - DELTAT T NOW COMPLETE.
EXTEND
QXCH 7 # ZERO INDEX Q.
 # (ZQ)
```

# Page 1227

WTLST4

```
XCH LST1
XCH LST1 +1
XCH LST1 +2
XCH LST1 +3
XCH LST1 +4
XCH LST1 +5
XCH LST1 +6
XCH LST1 +7
```

```
CA WAITADR # (MINOR PART OF TASK CADR HAS BEEN IN L.)
INDEX Q
TCF +1
```

```
DXCH LST2
DXCH LST2 +2
DXCH LST2 +4
DXCH LST2 +6
DXCH LST2 +8D
DXCH LST2 +10D # AT END, CHECK THAT C(LST2 +10) IS STD
DXCH LST2 +12D
DXCH LST2 +14D
DXCH LST2 +16D
AD ENDTASK # END ITEM, AS CHECK FOR EXCEEDING
 # THE LENGTH OF THE LIST.
```

July 12, 2016

2003

|             |        |          |                                          |
|-------------|--------|----------|------------------------------------------|
|             | EXTEND |          | # DUMMY TASK ADRES SHOULD BE IN FIXED-   |
|             | BZF    | LVWTLIST | # FIXED SO ITS ADRES ALONE DISTINGUISHES |
|             | TCF    | WTABORT  | # IT.                                    |
| # Page 1228 |        |          |                                          |
| WTLST5      | CCS    | A        | # TEST TD - T2 + 1                       |
|             | AD     | LST1 +1  |                                          |
|             | TCF    | +4       |                                          |
|             | AD     | ONE      |                                          |
|             | TC     | WTLST2   |                                          |
|             | OCT    | 1        |                                          |
| +4          | CCS    | A        | # TEST TD - T3 + 1                       |
|             | AD     | LST1 +2  |                                          |
|             | TCF    | +4       |                                          |
|             | AD     | ONE      |                                          |
|             | TC     | WTLST2   |                                          |
|             | OCT    | 2        |                                          |
| +4          | CCS    | A        | # TEST TD - T4 + 1                       |
|             | AD     | LST1 +3  |                                          |
|             | TCF    | +4       |                                          |
|             | AD     | ONE      |                                          |
|             | TC     | WTLST2   |                                          |
|             | OCT    | 3        |                                          |
| +4          | CCS    | A        | # TEST TD - T5 + 1                       |
|             | AD     | LST1 +4  |                                          |
|             | TCF    | +4       |                                          |
|             | AD     | ONE      |                                          |
|             | TC     | WTLST2   |                                          |
|             | OCT    | 4        |                                          |
| +4          | CCS    | A        | # TEST TD - T6 + 1                       |
|             | AD     | LST1 +5  |                                          |
|             | TCF    | +4       |                                          |
|             | AD     | ONE      |                                          |
|             | TC     | WTLST2   |                                          |
|             | OCT    | 5        |                                          |
| +4          | CCS    | A        | # TEST TD - T7 + 1                       |
|             | AD     | LST1 +6  |                                          |
|             | TCF    | +4       |                                          |
|             | AD     | ONE      |                                          |
|             | TC     | WTLST2   |                                          |
|             | OCT    | 6        |                                          |

2004

July 12, 2016

# Page 1229

```
+4 CCS A
 AD LST1 +7
 TCF +4
 AD ONE
 TC WTLST2
 OCT 7

 +4 CCS A
WTABORT TC BAILOUT # NO ROOM IN THE INN
 OCT 1203

 AD ONE
 TC WTLST2
 OCT 10

OCT40201 OCT 40201
```

# Page 1230

```
THE ENTRY TC WTLST2 JUST PRECEDING OCT N IS FOR T LE TD LE T -1.
N N+1
#
(LE MEANS LESS THAN OR EQUAL TO). AT ENTRY, C(A) = -(TD - T + 1)
N+1
#
THE LST1 ENTRY-(T -T +1) IS TO BE REPLACED BY -(TD - T + 1), AND
N+1 N N
#
THE ENTRY-(T - TD + 1) IS TO BE INSERTED IMMEDIATELY FOLLOWING.
N+1
```

```
WTLST2 TS WAITTEMP # C(A) = -(TD - T + 1)
 INDEX Q
 # Was CAF --- RSB 2004.
 CA 0
 TS Q # INDEX VALUE INTO Q.

 CAF ONE
 AD WAITTEMP
 INDEX Q # C(A) = -(TD - T) + 1.
 ADS LST1 -1 # N

 CS WAITTEMP
 INDEX Q
 TCF WTLST4
```

July 12, 2016

2005

# Page 1231

# ENTERS HERE ON T3 RUPT TO DISPATCH WAITLISTED TASK.

```
T3RUPT EXTEND
 ROR SUPERBNK # READ CURRENT SUPERBANK VALUE AND
 TS BANKRUPT # SAVE WITH E AND F BANK VALUES.
 EXTEND
 QXCH QRUPT

T3RUPT2 CAF NEG1/2 # DISPATCH WAITLIST TASK.
 XCH LST1 +7
 XCH LST1 +6
 XCH LST1 +5
 XCH LST1 +4 # 1. MOVE UP LST1 CONTENTS, ENTERING
 XCH LST1 +3 # A VALUE OF 1/2 +1 AT THE BOTTOM
 XCH LST1 +2 # FOR T6-T5, CORRESPONDING TO THE
 XCH LST1 +1 # INTERVAL 81.91 SEC FOR ENDTASK.
 XCH LST1
 AD POSMAX # 2. SET T3 = 1.0 - T2 - T USING LIST 1.
 ADS TIME3 # SO T3 WON'T TICK DURING UPDATE.
 TS RUPTAGN
 CS ZERO
 TS RUPTAGN # SETS RUPTAGN TO +1 ON OVERFLOW.

 EXTEND # DISPATCH TASK.
 DCS ENDTASK
 DXCH LST2 +16D
 DXCH LST2 +14D
 DXCH LST2 +12D
 DXCH LST2 +10D
 DXCH LST2 +8D
 DXCH LST2 +6
 DXCH LST2 +4
 DXCH LST2 +2
 DXCH LST2
 XCH L
 EXTEND
 WRITE SUPERBNK # SET SUPERBANK FROM BBCON OF 2CADR
 XCH L # RESTORE TO L FOR DXCH Z.
 DTCB
```

# Page 1232

# RETURN, AFTER EXECUTION OF T3 OVERFLOW TASK:

2006

July 12, 2016

```

 BLOCK 02
 COUNT 02/WAIT

TASKOVER CCS RUPTAGN # IF +1 RETURN TO T3RUPT, IF -0 RESUME.
 CAF WAITBB
 TS BBANK
 TCF T3RUPT2 # DISPATCH NEXT TASK IF IT WAS DUE.

 CA BANKRUPT
 EXTEND
 WRITE SUPERBNK # RESTORE SUPERBANK BEFORE RESUME IS DONE

RESUME EXTEND
 QXCH QRUPT
NOQRSM CA BANKRUPT
 XCH BBANK
NOQBRSM DXCH ARUPT
 RELINT
 RESUME

Page 1233
LONGCALL
PROGRAM DESCRIPTION DATE -- 17 MARCH 1967
PROGRAM WRITTEN BY W.H.VANDEVER LOG SECTION WAITLIST
MOD BY -- R. MELANSON TO ADD DOCUMENTATION ASSEMBLY SUNDISK REV. 100
#
FUNCTIONAL DESCRIPTION --
LONGCALL IS CALLED WITH THE DELTA TIME ARRIVING IN A,L SCALED AS TIME2,TIME1
IMMEDIATELY FOLLOWING THE TC LONGCALL. FOR EXAMPLE, IT MIGHT BE DONE AS FOL
A DP REGISTER CONTAINING A DELTA TIME AND WHERE TASKTODO IS THE NAME OF THE I
START.
#
CALLING SEQUENCE --
EXTEND
DCA TIMELOC
TC LONGCALL
2CADR TASKTODO
#
NORMAL EXIT MODE --
1) TC WAITLIST
2) DTCB (TC L+3 OF CALLING ROUTINE 1ST PASS THRU LONGCYCL)
3) DTCB (TO TASKOVER ON SUBSEQUENT PASSES THRU LONGCYCL)
#
ALARM OR ABORT EXIT MODE --
NONE
#
```

July 12, 2016

2007

```
OUTPUT --
LONGTIME AND LONGTIME+1 = DELTA TIME
LONGEXIT AND LONGEXIT+1 = RETURN 2CADR
LONGCADR AND LONGCADR+1 = TASK 2CADR
A = SINGLE PRECISION TIME FOR WAITLIST
#
ERASABLE INITIALIZATION --
A = MOST SIGNIFICANT PART OF DELTA TIME
L = LEAST SIGNIFICANT PART OF DELTA TIME
Q = ADDRESS OF 2CADR TASK VALUE
#
DEBRIS --
A,Q,L
LONGCADR AND LONGCADR+1
LONGEXIT AND LONGEXIT+1
LONGTIME AND LONGTIME+1
#
*** THE FOLLOWING IS TO BE IN FIXED-FIXED AND UNSWITCHED ERASIBLE **

 BLOCK 02
 EBANK= LST1
LONGCALL DXCH LONGTIME # OBTAIN THE DELTA TIME

 EXTEND # OBTAIN THE 2CADR
Page 1234
 NDX Q
 DCA 0
 DXCH LONGCADR

 EXTEND # NO GO TO THE APPROPRIATE SWITCHED BANK
 DCA LGCL2CDR # FOR THE REST OF LONGCALL
 DTCB

 EBANK= LST1
LGCL2CDR 2CADR LNGCALL2

*** THE FOLLOWING MAY BE IN A SWITCHED BANK, INCLUDING ITS ERASABLE ***

 BANK 01
 COUNT 01/WAIT

LNGCALL2 LXCH LONGEXIT +1 # SAVE THE CORRECT BB FOR RETURN
 CA TWO # OBTAIN THE RETURN ADDRESS
 ADS Q
 TS LONGEXIT
```

2008

July 12, 2016

# \*\*\* WAITLIST TASK LONGCYCL \*\*\*

```
LONGCYCL EXTEND # CAN WE SUCCESSFULLY TAKE ABOUT 1.25
 DCS DPBIT14 # MINUTES OFF OF LONGTIME
 DAS LONGTIME

 CCS LONGTIME +1 # THE REASONING BEHIND THIS PART IS
 TCF MUCHTIME # INVOLVED, TAKING INTO ACCOUNT THAT THE
 # WORDS MAY NOT BE SIGNED CORRECTED (DP
 # BASIC INSTRUCTIONS
 # DO NOT SIGN CORRECT) AND THAT WE SUBTRAC-
 # TED BIT14 (1 OVER HALF THE POS. VALUE
 # REPRESENTABLE IN SINGLE WORD)
 # CAN'T GET HERE *****

 NOOP
 TCF +1
 CCS LONGTIME
 TCF MUCHTIME
DPBIT14 OCT 00000
 OCT 20000

LASTTIME CA BIT14 # LONGCALL
 ADS LONGTIME +1 # GET BACK THE CORRECT DELTA TFOR WAITLIST
 TC WAITLIST
 EBANK= LST1
 2CADR GETCADR # THE ENTRY TO OUR LONGCADR

LONGRTRN CA TSKOVCDR # SET IT UP SO THAT ONLY THE FIRST EXIT IS
Page 1235 DXCH LONGEXIT # TO THE CALLER OF LONGCALL
 DTCB # THE REST ARE TO TASKOVER

MUCHTIME CA BIT14 # WE HAVE OVER OUR ABOUT 1.25 MINUTES
 TC WAITLIST # SO SET UP FOR ANOTHER CYCLE THROUGH HERE
 EBANK= LST1
 2CADR LONGCYCL

 TCF LONGRTRN # NOW EXIT PROPERLY

*** WAITLIST TASK GETCADR ***

GETCADR DXCH LONGCADR # GET THE LONGCALL THAT WE WISHED TO START
 DTCB # AND TRANSFER CONTROL TO IT

TSKOVCDR GENADR TASKOVER
```



July 12, 2016

2009

This code is written to file `src/WAITLIST.s`.

2010

July 12, 2016

## Appendix C

# Miscellaneous Files

2011

*<check.sh 2011>≡*

```
orig=AGC_BLOCK_TWO_SELF_CHECK.s; new=AGC-BLOCK-TWO-SELF--CHECK.s; echo "XX $orig XX"; diff orig
orig=AGC_BLOCK_TWO_SELF-CHECK.s; new=AGC-BLOCK-TWO-SELF-CHECK.s; echo "XX $orig XX"; diff orig
orig=AGS_INITIALIZATION.s; new=AGS-INITIALIZATION.s; echo "XX $orig XX"; diff origsrc/$orig src
orig=ALARM_AND_ABORT.s; new=ALARM-AND-ABORT.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=ANGLFIND.s; new=ANGLFIND.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=AOSTASK_AND_AOSJOB.s; new=AOSTASK-AND-AOSJOB.s; echo "XX $orig XX"; diff origsrc/$orig src
orig=AOTMARK.s; new=AOTMARK.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=ASCENT_GUIDANCE.s; new=ASCENT-GUIDANCE.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=ASSEMBLY_AND_OPERATION_INFORMATION.s; new=ASSEMBLY-AND-OPERATION-INFORMATION.s; echo "XX $
orig=ATTITUDE_MANEUVER_ROUTINE.s; new=ATTITUDE-MANEUVER-ROUTINE.s; echo "XX $orig XX"; diff ori
orig=AUTOMATIC_MANEUVERS.s; new=AUTOMATIC-MANEUVERS.s; echo "XX $orig XX"; diff origsrc/$orig s
orig=BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.s; new=BURN-BABY-BURN--MASTER-IGNITION-ROUTINE.s;
orig=CM_BODY_ATTITUDE.s; new=CM-BODY-ATTITUDE.s; echo "XX $orig XX"; diff origsrc/$orig src/$ne
orig=CM_ENTRY_DIGITAL_AUTOPILOT.s; new=CM-ENTRY-DIGITAL-AUTOPILOT.s; echo "XX $orig XX"; diff o
orig=CONIC_SUBROUTINES.s; new=CONIC-SUBROUTINES.s; echo "XX $orig XX"; diff origsrc/$orig src/$
orig=CONTRACT_AND_APPROVALS.s; new=CONTRACT-AND-APPROVALS.s; echo "XX $orig XX"; diff origsrc/$
orig=CONTROLLED_CONSTANTS.s; new=CONTROLLED-CONSTANTS.s; echo "XX $orig XX"; diff origsrc/$orig
orig=CSM_GEOMETRY.s; new=CSM-GEOMETRY.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=DAPIDLER_PROGRAM.s; new=DAPIDLER-PROGRAM.s; echo "XX $orig XX"; diff origsrc/$orig src/$ne
orig=DAP_INTERFACE_SUBROUTINES.s; new=DAP-INTERFACE-SUBROUTINES.s; echo "XX $orig XX"; diff ori
orig=DISPLAY_INTERFACE_ROUTINES.s; new=DISPLAY-INTERFACE-ROUTINES.s; echo "XX $orig XX"; diff o
orig=DOWNLINK_LISTS.s; new=DOWNLINK-LISTS.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=DOWN_TELEMETRY_PROGRAM.s; new=DOWN--TELEMETRY-PROGRAM.s; echo "XX $orig XX"; diff origsrc/
orig=DOWN-TELEMETRY_PROGRAM.s; new=DOWN-TELEMETRY-PROGRAM.s; echo "XX $orig XX"; diff origsrc/$
orig=ENTRY_LEXICON.s; new=ENTRY-LEXICON.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=ERASABLE_ASSIGNMENTS.s; new=ERASABLE-ASSIGNMENTS.s; echo "XX $orig XX"; diff origsrc/$orig
orig=EXECUTIVE.s; new=EXECUTIVE.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=EXTENDED_VERBS.s; new=EXTENDED-VERBS.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=FINDCDUW--GUIDAP_INTERFACE.s; new=FINDCDUW--GUIDAP-INTERFACE.s; echo "XX $orig XX"; diff o
orig=FIXED_FIXED_CONSTANT_POOL.s; new=FIXED-FIXED-CONSTANT-POOL.s; echo "XX $orig XX"; diff ori
```

```

orig=FLAGWORD_ASSIGNMENTS.s; new=FLAGWORD-ASSIGNMENTS.s; echo "XX $orig XX"; diff orig
orig=FRESH_START_AND_RESTART.s; new=FRESH-START-AND-RESTART.s; echo "XX $orig XX"; d
orig=GIMBAL_LOCK_AVOIDANCE.s; new=GIMBAL-LOCK-AVOIDANCE.s; echo "XX $orig XX"; diff
orig=GROUND_TRACKING_DETERMINATION_PROGRAM.s; new=GROUND-TRACKING-DETERMINATION-PROG
orig=HeaderTemplate.s; new=HeaderTemplate.s; echo "XX $orig XX"; diff origsrc/$orig s
orig=IMU_CALIBRATION_AND_ALIGNMENT.s; new=IMU-CALIBRATION-AND-ALIGNMENT.s; echo "XX s
orig=IMU_COMPENSATION_PACKAGE.s; new=IMU-COMPENSATION-PACKAGE.s; echo "XX $orig XX";
orig=IMU_MODE_SWITCHING_ROUTINES.s; new=IMU-MODE-SWITCHING-ROUTINES.s; echo "XX $orig
orig=IMU_PERFORMANCE_TEST_2.s; new=IMU-PERFORMANCE-TEST-2.s; echo "XX $orig XX"; diff
orig=IMU_PERFORMANCE_TESTS_4.s; new=IMU-PERFORMANCE-TESTS-4.s; echo "XX $orig XX"; d
orig=INFLIGHT_ALIGNMENT_ROUTINES.s; new=INFLIGHT-ALIGNMENT-ROUTINES.s; echo "XX $orig
orig=INPUT_OUTPUT_CHANNEL_BIT_DESCRIPTIONS.s; new=INPUT-OUTPUT-CHANNEL-BIT-DESRIPTION
orig=INTEGRATION_INITIALIZATION.s; new=INTEGRATION-INITIALIZATION.s; echo "XX $orig X
orig=INTER-BANK_COMMUNICATION.s; new=INTER-BANK-COMMUNICATION.s; echo "XX $orig XX";
orig=INTERPRETER.s; new=INTERPRETER.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=INTERPRETIVE_CONSTANT.s; new=INTERPRETIVE-CONSTANT.s; echo "XX $orig XX"; diff
orig=INTERPRETIVE_CONSTANTS.s; new=INTERPRETIVE-CONSTANTS.s; echo "XX $orig XX"; diff
orig=INTERRUPT_LEAD_INS.s; new=INTERRUPT-LEAD-INS.s; echo "XX $orig XX"; diff origsrc
orig=JET_SELECTION_LOGIC.s; new=JET-SELECTION-LOGIC.s; echo "XX $orig XX"; diff orig
orig=KALCMANU_STEERING.s; new=KALCMANU-STEERING.s; echo "XX $orig XX"; diff origsrc/s
orig=KALMAN_FILTER.s; new=KALMAN-FILTER.s; echo "XX $orig XX"; diff origsrc/$orig src
orig=KEYRUPT_UPRUPT.s; new=KEYRUPT-UPRUPT.s; echo "XX $orig XX"; diff origsrc/$orig s
orig=LAMBERT_AIMPOINT_GUIDANCE.s; new=LAMBERT-AIMPOINT-GUIDANCE.s; echo "XX $orig XX";
orig=LANDING_ANALOG_DISPLAYS.s; new=LANDING-ANALOG-DISPLAYS.s; echo "XX $orig XX"; d
orig=LATITUDE_LONGITUDE_SUBROUTINES.s; new=LATITUDE-LONGITUDE-SUBROUTINES.s; echo "XX
orig=LEM_GEOMETRY.s; new=LEM-GEOMETRY.s; echo "XX $orig XX"; diff origsrc/$orig src/$
orig=LUNAR_AND_SOLAR_EPHEMERIDES_SUBROUTINES.s; new=LUNAR-AND-SOLAR-EPHEMERIDES-SUBRO
orig=LUNAR_LANDING_GUIDANCE_EQUATIONS.s; new=LUNAR-LANDING-GUIDANCE-EQUATIONS.s; echo
orig=LUNAR_LANDMARK_SELECTION_FOR_CM.s; new=LUNAR-LANDMARK-SELECTION-FOR-CM.s; echo '
orig=MAIN.s; new=MAIN.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=MEASUREMENT_INCORPORATION.s; new=MEASUREMENT-INCORPORATION.s; echo "XX $orig XX
orig=MYSUBS.s; new=MYSUBS.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=ORBITAL_INTEGRATION.s; new=ORBITAL-INTEGRATION.s; echo "XX $orig XX"; diff orig
orig=P11.s; new=P11.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P12.s; new=P12.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P20-P25.s; new=P20-P25.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P30_P37.s; new=P30--P37.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P30-P37.s; new=P30-P37.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P32-P33_P72-P73.s; new=P32-P33-P72-P73.s; echo "XX $orig XX"; diff origsrc/$orig
orig=P32-P35_P72-P75.s; new=P32-P35-P72-P75.s; echo "XX $orig XX"; diff origsrc/$orig
orig=P34-35_P74-75.s; new=P34-35-P74-75.s; echo "XX $orig XX"; diff origsrc/$orig src
orig=P37_P70.s; new=P37-P70.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P40-P47.s; new=P40-P47.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P51-P53.s; new=P51-P53.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P61-P67.s; new=P61-P67.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P70-P71.s; new=P70-P71.s; echo "XX $orig XX"; diff origsrc/$orig src/$new

```

July 12, 2016

2013

```
orig=P76.s; new=P76.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=P-AXIS_RCS_AUTOPILOT.s; new=P-AXIS-RCS-AUTOPILOT.s; echo "XX $orig XX"; diff origsrc/$orig
orig=PHASE_TABLE_MAINTENANCE.s; new=PHASE-TABLE-MAINTENANCE.s; echo "XX $orig XX"; diff origsrc/$orig
orig=PINBALL_GAME_BUTTONS_AND_LIGHTS.s; new=PINBALL-GAME-BUTTONS-AND-LIGHTS.s; echo "XX $orig XX"; diff origsrc/$orig
orig=PINBALL_NOUN_TABLES.s; new=PINBALL-NOUN-TABLES.s; echo "XX $orig XX"; diff origsrc/$orig
orig=PLANETARY_INERTIAL_ORIENTATION.s; new=PLANETARY-INERTIAL-ORIENTATION.s; echo "XX $orig XX"; diff origsrc/$orig
orig=POWERED_FLIGHT_SUBROUTINES.s; new=POWERED-FLIGHT-SUBROUTINES.s; echo "XX $orig XX"; diff origsrc/$orig
orig=Q_R-AXIS_RCS_AUTOPILOT.s; new=Q-R-AXIS-RCS-AUTOPILOT.s; echo "XX $orig XX"; diff origsrc/$orig
orig=R30.s; new=R30.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=R31.s; new=R31.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=R60_62.s; new=R60-62.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=R63.s; new=R63.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=RADAR_LEADIN_ROUTINES.s; new=RADAR-LEADIN-ROUTINES.s; echo "XX $orig XX"; diff origsrc/$orig
orig=RCS-CSM_DAP_EXECUTIVE_PROGRAMS.s; new=RCS-CSM-DAP-EXECUTIVE-PROGRAMS.s; echo "XX $orig XX"; diff origsrc/$orig
orig=RCS-CSM_DIGITAL_AUTOPILOT.s; new=RCS-CSM-DIGITAL-AUTOPILOT.s; echo "XX $orig XX"; diff origsrc/$orig
orig=RCS_FAILURE_MONITOR.s; new=RCS-FAILURE-MONITOR.s; echo "XX $orig XX"; diff origsrc/$orig
orig=README.md; new=README.md; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=REENTRY_CONTROL.s; new=REENTRY-CONTROL.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=RESTARTS_ROUTINE.s; new=RESTARTS-ROUTINE.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=RESTART_TABLES.s; new=RESTART-TABLES.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=RT8_OP_CODES.s; new=RT8-OP-CODES.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=RTB_OP_CODES.s; new=RTB-OP-CODES.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=S-BAND_ANTENNA_FOR_CM.s; new=S-BAND-ANTENNA-FOR-CM.s; echo "XX $orig XX"; diff origsrc/$orig
orig=S-BAND_ANTENNA_FOR_LM.s; new=S-BAND-ANTENNA-FOR-LM.s; echo "XX $orig XX"; diff origsrc/$orig
orig=SERVICER207.s; new=SERVICER207.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=SERVICE_ROUTINES.s; new=SERVICE-ROUTINES.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=SERVICER.s; new=SERVICER.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=SINGLE_PRECISION_SUBROUTINES.s; new=SINGLE-PRECISION-SUBROUTINES.s; echo "XX $orig XX"; diff origsrc/$orig
orig=SPS_BACK-UP_RCS_CONTROL.s; new=SPS-BACK-UP-RCS-CONTROL.s; echo "XX $orig XX"; diff origsrc/$orig
orig=STABLE_ORBIT.s; new=STABLE-ORBIT.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=STAR_TABLES.s; new=STAR-TABLES.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=SXTMARK.s; new=SXTMARK.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=SYSTEM_TEST_STANDARD_LEAD_INS.s; new=SYSTEM-TEST-STANDARD-LEAD-INS.s; echo "XX $orig XX"; diff origsrc/$orig
orig=T4RUPT_PROGRAM.s; new=T4RUPT-PROGRAM.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=T6-RUPT_PROGRAMS.s; new=T6-RUPT-PROGRAMS.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TAGS_FOR_RELATIVE_SETLOC.s; new=TAGS-FOR-RELATIVE-SETLOC.s; echo "XX $orig XX"; diff origsrc/$orig
orig=THE_LUNAR_LANDING.s; new=THE-LUNAR-LANDING.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=THROTTLE_CONTROL_ROUTINES.s; new=THROTTLE-CONTROL-ROUTINES.s; echo "XX $orig XX"; diff origsrc/$orig
orig=TIME_OF_FREE_FALL.s; new=TIME-OF-FREE-FALL.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TJET_LAW.s; new=TJET-LAW.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TPI_SEARCH.s; new=TPI-SEARCH.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TRIM_GIMBAL_CNTRL_SYSTEM.s; new=TRIM-GIMBAL-CNTRL-SYSTEM.s; echo "XX $orig XX"; diff origsrc/$orig
orig=TVCDAPS.s; new=TVCDAPS.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TVCEXECUTIVE.s; new=TVCEXECUTIVE.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TVCINITIALIZE.s; new=TVCINITIALIZE.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TVCMASSPROP.s; new=TVCMASSPROP.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
```

2014

July 12, 2016

```
orig=TVCRESTARTS.s; new=TVCRESTARTS.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TVCROLLDAP.s; new=TVCROLLDAP.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=TVCSTROKETEST.s; new=TVCSTROKETEST.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=UPDATE_PROGRAM.s; new=UPDATE-PROGRAM.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
orig=WAITLIST.s; new=WAITLIST.s; echo "XX $orig XX"; diff origsrc/$orig src/$new
```

This code is written to file `check.sh`.

## Appendix D

# Notes, Bibliography and Indexes

### D.1 Chunks

*<agc block two self check 58e>*  
*<alarm and abort 59a>*  
*<anglfind code 50g>*  
*<Apollo-11.s 15a>*  
*<assembly and operation information 17>*  
*<automatic maneuvers 55c>*  
*<check.sh 2011>*  
*<chieftan routines 56a>*  
*<cm body attitude 53g>*  
*<cm entry digital autopilot 55f>*  
*<comaid routines 49a>*  
*<comekiss routines 51a>*  
*<comerase routine 48c>*  
*<conic subroutines 57i>*  
*<contract and approvals 15c>*  
*<csn geometry 50c>*  
*<display interface routines 58j>*  
*<down-telemetry program 56b>*  
*<downlink lists 49d>*  
*<entry lexicon 53e>*  
*<erasable assignments 48d>*  
*<executive code 57d>*  
*<extended verbs 50a>*  
*<fixed fixed constant pool 57a>*  
*<fresh start and restart 49e>*  
*<gimbal lock avoidance 50h>*

<ground tracking determination program 51b>  
 <imu calibration and alignment 50k>  
 <imu compensation package 50d>  
 <imu mode switching routines 58h>  
 <inflight alignment routines 58a>  
 <integration initialization 57j>  
 <inter-bank communication 56c>  
 <interpreter code 56d>  
 <interpretive constants 57b>  
 <interrupt lead ins 49b>  
 <jet selection logic 55e>  
 <kalcmanu steering 50i>  
 <keyrupt uprupt 58i>  
 <latitude longitude subroutines 57f>  
 <lunar and solar ephemerides subroutines 53b>  
 <lunar landmark selection for cm 53j>  
 <main headers 15b>  
 <measurement incorporation 57h>  
 <mysubs code 55a>  
 <orbital integration 57k>  
 <p11 code 52b>  
 <p20 p25 code 52d>  
 <p30 p37 code 52e>  
 <p32-p33 p72-p73 code (never defined)>  
 <p34-35 p74-75 code 51c>  
 <p37 p70 code 53h>  
 <p40-p47 code 52f>  
 <p51-p53 code 53a>  
 <p61-p67 code 53c>  
 <p76 code 51e>  
 <Page 1 16>  
 <Page 10 26b>  
 <Page 11 27>  
 <Page 12 29>  
 <Page 13 30>  
 <Page 14 31>  
 <Page 15 32>  
 <Page 16 34>  
 <Page 17 36>  
 <Page 18 37>  
 <Page 19 38>  
 <Page 2 18>  
 <Page 20 40>  
 <Page 21 42>  
 <Page 22 43>  
 <Page 23 44>



<Page 24 46>  
 <Page 25 47>  
 <Page 26 48a>  
 <Page 3 20>  
 <Page 4 21>  
 <Page 5 22>  
 <Page 6 23>  
 <Page 7 24>  
 <Page 8 25>  
 <Page 9 26a>  
 <phase table maintenance 58f>  
 <pinball game buttons and lights 50e>  
 <pinball noun tables 50b>  
 <planetary inertial orientation 57g>  
 <powered flight subroutines 58b>  
 <r30 code 51f>  
 <r31 code 51d>  
 <r60-62 code 50f>  
 <rcs-csm dap executive programs 55d>  
 <rcs-csm digital autopilot 55b>  
 <reentry control 53f>  
 <restart tables 49f>  
 <restarts routine 58g>  
 <rtb op codes 59c>  
 <s-band antenna for cm 53i>  
 <sctmark code 49g>  
 <service routines 58k>  
 <servicer207 code 53d>  
 <single precision subroutines 57c>  
 <src/AGC-BLOCK-TWO-SELF-CHECK.s 68>  
 <src/AGC-BLOCK-TWO-SELF-CHECK.s 80>  
 <src/AGS-INITIALIZATION.s 92>  
 <src/ALARM-AND-ABORT.s 98>  
 <src/ANGLFIND.s 104>  
 <src/AOSTASK-AND-AOSJOB.s 118>  
 <src/AOTMARK.s 142>  
 <src/ASCENT-GUIDANCE.s 158>  
 <src/ASSEMBLY-AND-OPERATION-INFORMATION.s 173>  
 <src/ATTITUDE-MANEUVER-ROUTINE.s 176>  
 <src/AUTOMATIC-MANEUVERS.s 199>  
 <src/BURN-BABY-BURN-MASTER-IGNITION-ROUTINE.s 210>  
 <src/CM-BODY-ATTITUDE.s 234>  
 <src/CM-ENTRY-DIGITAL-AUTOPILOT.s 241>  
 <src/CONIC-SUBROUTINES.s 269>  
 <src/CONTRACT-AND-APPROVALS.s 311>  
 <src/CONTROLLED-CONSTANTS.s 312>

<src/CSM-GEOMETRY.s 325>  
 <src/DAP-INTERFACE-SUBROUTINES.s 346>  
 <src/DAPIDLER-PROGRAM.s 335>  
 <src/DISPLAY-INTERFACE-ROUTINES.s 350>  
 <src/DOWN-TELEMETRY-PROGRAM.s 392>  
 <src/DOWN-TELEMETRY-PROGRAM.s 402>  
 <src/DOWNLINK-LISTS.s 383>  
 <src/ENTRY-LEXICON.s 413>  
 <src/ERASABLE-ASSIGNMENTS.s 422>  
 <src/EXECUTIVE.s 505>  
 <src/EXTENDED-VERBS.s 516>  
 <src/FINDCDUW-GUIDAP-INTERFACE.s 546>  
 <src/FIXED-FIXED-CONSTANT-POOL.s 563>  
 <src/FLAGWORD-ASSIGNMENTS.s 569>  
 <src/FRESH-START-AND-RESTART.s 597>  
 <src/GIMBAL-LOCK-AVOIDANCE.s 630>  
 <src/GROUND-TRACKING-DETERMINATION-PROGRAM.s 633>  
 <src/HeaderTemplate.s 67>  
 <src/IMU-CALIBRATION-AND-ALIGNMENT.s 638>  
 <src/IMU-COMPENSATION-PACKAGE.s 669>  
 <src/IMU-MODE-SWITCHING-ROUTINES.s 678>  
 <src/IMU-PERFORMANCE-TEST-2.s 702>  
 <src/IMU-PERFORMANCE-TESTS-4.s 712>  
 <src/INFLIGHT-ALIGNMENT-ROUTINES.s 720>  
 <src/INPUT-OUTPUT-CHANNEL-BIT-DESCRIPTIONS.s 727>  
 <src/INTEGRATION-INITIALIZATION.s 732>  
 <src/INTER-BANK-COMMUNICATION.s 758>  
 <src/INTERPRETER.s 763>  
 <src/INTERPRETIVE-CONSTANT.s 830>  
 <src/INTERPRETIVE-CONSTANTS.s 832>  
 <src/INTERRUPT-LEAD-INS.s 834>  
 <src/JET-SELECTION-LOGIC.s 837>  
 <src/KALCMANU-STEERING.s 858>  
 <src/KALMAN-FILTER.s 864>  
 <src/KEYRUPT-UPRUPT.s 867>  
 <src/LAMBERT-AIMPOINT-GUIDANCE.s 871>  
 <src/LANDING-ANALOG-DISPLAYS.s 875>  
 <src/LATITUDE-LONGITUDE-SUBROUTINES.s 887>  
 <src/LEM-GEOMETRY.s 894>  
 <src/LUNAR-AND-SOLAR-EPHEMERIDES-SUBROUTINES.s 899>  
 <src/LUNAR-LANDING-GUIDANCE-EQUATIONS.s 904>  
 <src/LUNAR-LANDMARK-SELECTION-FOR-CM.s 937>  
 <src/MAIN.s 938>  
 <src/MEASUREMENT-INCORPORATION.s 942>  
 <src/MYSUBS.s 954>  
 <src/ORBITAL-INTEGRATION.s 957>

*<src/P-AXIS-RCS-AUTOPILOT.s 1387>*  
*<src/P11.s 978>*  
*<src/P12.s 999>*  
*<src/P20-P25.s 1005>*  
*<src/P30-P37.s 1082>*  
*<src/P30-P37.s 1087>*  
*<src/P32-P33-P72-P73.s 1101>*  
*<src/P32-P35-P72-P75.s 1132>*  
*<src/P34-35-P74-75.s 1163>*  
*<src/P37-P70.s 1201>*  
*<src/P40-P47.s 1244>*  
*<src/P51-P53.s 1297>*  
*<src/P61-P67.s 1346>*  
*<src/P70-P71.s 1373>*  
*<src/P76.s 1383>*  
*<src/PHASE-TABLE-MAINTENANCE.s 1411>*  
*<src/PINBALL-GAME-BUTTONS-AND-LIGHTS.s 1420>*  
*<src/PINBALL-NOUN-TABLES.s 1503>*  
*<src/PLANETARY-INERTIAL-ORIENTATION.s 1522>*  
*<src/POWERED-FLIGHT-SUBROUTINES.s 1531>*  
*<src/Q-R-AXIS-RCS-AUTOPILOT.s 1539>*  
*<src/R30.s 1558>*  
*<src/R31.s 1569>*  
*<src/R60-62.s 1576>*  
*<src/R63.s 1585>*  
*<src/RADAR-LEADIN-ROUTINES.s 1589>*  
*<src/RCS-CSM-DAP-EXECUTIVE-PROGRAMS.s 1592>*  
*<src/RCS-CSM-DIGITAL-AUTOPILOT.s 1594>*  
*<src/RCS-FAILURE-MONITOR.s 1616>*  
*<src/README.md 65>*  
*<src/REENTRY-CONTROL.s 1620>*  
*<src/RESTART-TABLES.s 1664>*  
*<src/RESTARTS-ROUTINE.s 1656>*  
*<src/RT8-OP-CODES.s 1677>*  
*<src/RTB-OP-CODES.s 1685>*  
*<src/S-BAND-ANTENNA-FOR-CM.s 1691>*  
*<src/S-BAND-ANTENNA-FOR-LM.s 1694>*  
*<src/SERVICE-ROUTINES.s 1717>*  
*<src/SERVICER.s 1723>*  
*<src/SERVICER207.s 1699>*  
*<src/SINGLE-PRECISION-SUBROUTINES.s 1761>*  
*<src/SPS-BACK-UP-RCS-CONTROL.s 1763>*  
*<src/STABLE-ORBIT.s 1768>*  
*<src/STAR-TABLES.s 1778>*  
*<src/SXTMARK.s 1783>*  
*<src/SYSTEM-TEST-STANDARD-LEAD-INS.s 1799>*

<src/T4RUPT-PROGRAM.s 1803>  
 <src/T6-RUPT-PROGRAMS.s 1835>  
 <src/TAGS-FOR-RELATIVE-SETLOC.s 1839>  
 <src/THE-LUNAR-LANDING.s 1849>  
 <src/THROTTLE-CONTROL-ROUTINES.s 1857>  
 <src/TIME-OF-FREE-FALL.s 1862>  
 <src/TJET-LAW.s 1878>  
 <src/TPI-SEARCH.s 1890>  
 <src/TRIM-GIMBAL-CNTROL-SYSTEM.s 1903>  
 <src/TVCDAPS.s 1917>  
 <src/TVCEXECUTIVE.s 1935>  
 <src/TVCINITIALIZE.s 1942>  
 <src/TVCMASSPROP.s 1952>  
 <src/TVCRESTARTS.s 1958>  
 <src/TVCROLLDAP.s 1964>  
 <src/TVCSTROKETEST.s 1978>  
 <src/UPDATE-PROGRAM.s 1984>  
 <src/WAITLIST.s 1997>  
 <st4rupt program 49c>  
 <stable orbit 51g>  
 <stags for relative setloc 48b>  
 <star tables 58d>  
 <system test standard lead ins 50j>  
 <time of free fall 58c>  
 <tpi search 52c>  
 <trouble routines 52a>  
 <tvcdaps code 54f>  
 <tvcdaps routines 54a>  
 <tvcexecutive code 54c>  
 <tvcinitialize code 54b>  
 <tvcmassprop code 54d>  
 <tvcrestarts code 54e>  
 <tvcrolldap code 54h>  
 <tvcstroketest code 54g>  
 <update program 59b>  
 <waitlist code 57e>

## D.2 Index