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

Edited by Gary Young

proudindiv@gmail.com

August 2, 2016

Contents

Chapter 1

Introduction and Notes

#140 Apollo-11 sources as Literate Programs Jul 16 chrislgarry/Apollo-11

I'm literating Apollo-11 for documentation, chunk and page descriptions and indexing, and memory label indexing. So far I've been working on this early experiment for a week, and the structure and contents will change drastically as my ideas evolve.

I've started a typeset book for the Apollo-11 source in my clone at https://github.com/proudindiv/Apollo-11. The original sources were not "files" in any sense, but about 45 boxes of hollerith punch cards (2000 cards to a box) on a large cart, with the listings produced by the assembler which have been imaged. So I'm looking at the code as virtual card decks.

The main output is currently in my Apollo-11.pdf, which is better downloaded and examined because the github pdf viewer doesn't show the massive internal linkages.

I just wanted to make this early announcement, so others were aware of and could think about it.

Thanks,

Gary Young

aka proudindiv

Sat Jul 16, 2016

So far, I'm breaking the code into chunks, organizing and structuring the chunks, indexing the memory labels and all references to them by chunks, making notes and comments, and adding the table of contents, bookmarks, chunk index, and general and label indices. I do the markup and noweb and LaTeX do the type-

setting.

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 originally takes about five seconds to typeset this book. It takes longer as I create complex indexing and add structure to the listing, but still less than a minute. 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.
- I think the biggest difficulty in recreating the card decks is going to be white space. In creating the files from the page listing images, tabs were used. The original programmers in the 1960's used card punches to create the input to the assembler. I think the assembler only accepted fixed format cards and that's why if you were to scan the listings there probably aren't any errors in the spacing on the cards. Those card punch machines had a method of setting tab stops and there was a key to advance the current card being punched to the next tab stop. Characters being typed or duped then continued from that location. There was no tab character punched into any column of the card. Similarly, the advancement to the next page was controlled by the assembler program. The programmer could force a short page by inserting a control pseudo-op into the second field of the card, something like "newpage", but I don't remember what it was. There was also an "end" pseudo-op in the last card. Some pseudo-ops weren't printed by the assembler.

- The line numbers on the left side of the listings also seem to indicate other pseudo-ops, such as begin comment block and end comment block. We used columns 72-80 for a card sequence number, but they were hard to keep consistent because of card insertions and replacements.

> proudindiv:
>
> Did you notice Comanche055.binsource and Luminary099.binsource in the rburkey2005/virtualagc repo? I'm thinking of running their assembler on the sources and comparing, or writing my own simple assembler. Do you think we should bring those files into chrislgarry/Apollo-11?

> wopian:
>
> I would disagree with adding those files.
>
> This repo is a historical archive of the Apollo 11 source code, VirtualAGC was made to compile the Apollo 11 code (and will be pulling the .agc files from here once they've all been checked)
>
> Scroll down on #38

> proudindiv:
>
> But with that argument, the historical Apollo 11 source code is defined by the jpg images linked to in the chrislgarry/master README.md file. The added 51 files from the VirtualAGC are required if you are going to match the images. They are also required if you are going to compile the source code and fly the Apollo-11 to the moon. There are issues with the encoding used in the VirtualAGC transcriptions representing the original source code, but adding the 51 files doesn't change that issue.
>
> I'm sorry, I went back and re-read your comment, and realize that you were talking about the two .binsource files. Those files are a modern creation from the original sources, and as such can be excluded from the historical archive. They are better placed in a modern analysis of the historical code, and I agree with you on excluding them.
>
> But with your argument, we need to remove the file "HeaderTemplate.agc" and discuss the "MAIN.agc" files and split-

> ting the source lines (cards) into 174 files (plus two MAIN.agc files) instead of only 2 files representing the original card decks. See VirtualAGC's Programmer's Manual for what I am talking about.
>
> Thanks for the link to #38, I hadn't seen that but had inferred the information in the discussion. I'm a long retired programmer and was programming on similar style programs in the later 1960's. I became interested in the Apollo-11 code when it was announced because it brings back lots of memories. Reading the code is very intimate because I can see what the original programmers were thinking. I started my Literate version of the code to give me a structure to reverse engineer in modern terms what was in it. We'll see how long I work with it and keep up the effort.
>
> Gary Young aka proudindiv

> chrislgarry:
>
> @wopian yep that's my stance as well. @proudindiv yes I did plan on doing something with main.agc since its purpose is to be a wrapper to organize all of the assembly files for compliation, but it slipped my mind. We could simply just remove it if there is no original logic in there. I haven't looked yet. I also agree with splitting into the original number of files so we don't have multiple original assembly files in one digital assembly file here. And removal of header template if it's not original.

#38:

> ohommes:
>
> It is crucial to split out the code for the Command Module and the Lunar Module. I have worked closely with Ron Burkey and many others in the world to transcribe the source code for the 40th anniversary of the lunar landing for Apollo 11 over a period of many weeks. Without thinking to much about the source code build names; for the Command Module (CM) the names are starting with a "C" (e.g. Colossus238 and Comanche055) and the Lunar Module (LM) started with "L" (e.g. Luminary).

There are however many exceptions to this Like Artemis and Sundance, Sundial and Sunburst but having the code split is a must. There are MANY difference crucial for the landing versus the orbit only and reentry for the CM. The original source listings that we made available on sourcecode.google.com and at biblio had these significant distinctions. Combining them in one location does not even make them compile anymore with yaYUL from Ron Burkey. Hope to see this fixed soon.

ohommes:

For sure I support that setup by original build-name. I worked on the transcription painstakingly with many others to support the 40th anniversary of the Moon Landings.

In 2009, the year of the 40th anniversary of Apollo 11, a size-able group of Apollo enthusiast from all across the world transcribed both Comanche055 and Luminary099 from pictures taken of each page.

This process started by Ron Burkey, who travelled to make photographs of each page of the code on site since we could not take the actual hardcopy. Then a process started where each individual of this project checked out a block of pages and transcribed the assemble code with comments. The images contained both the binary code and the assembly mnemonics with comments.

Once transcribed the process of verification started:

- The first pass was to compile the code with yaYUL from Ron Burkey. The binary code was then verified agains the binary code on the printout.
- The second stage was visual re-inspection for the comments.
- The third stage was the verification of the memory bank which was done by running the actual code in yaAGC and by running the original Checksum verification program with Verb 91 (V91E) the checksum (including the so called bugger word) if correct is the same value as the bank number. I think it would be interesting for people to know these verification steps and how this process took many weeks with the deadline of the 40th anniversay of Apollo 11.

> chrislgarry:
>
> @ohommes @corvuscrypto first, thanks for sharing your thoughts and the effort you guys put. Very much appreciated. I created this repo two years ago when I saw the original Apollo 11 source online bundled with Virtual AGC. I wanted the original Apollo 11 source just for viewing, so I extracted it and uploaded it to a repo for myself. Recently, somebody found my repo and shared it online, and here we are. Although Virtual AGC is awesome, I never was and am still not interested in Virtual AGC here in this repo. I am interested in the original Apollo 11 source only. I didn't know about the Virtual AGC repo until the other day when Ron shared the link with me and candidly stated "It probably would have helped if I had actually included this information on my website rather than keeping it to myself. :-)." This repo was meant to be for my own interest, but here we are, and now I am doing my best to maintain it as a repo for simply viewing the code. The community has already cleaned up a bunch of typos from digitization, and I suspect a lot more work in that area. We may get it to the point where it is truly reflective of the hard copies, which is my goal. The rekindled interest is tremendous, and I want to keep that going. Such a historical artifact as this deserves it's own repo without all the extras to build it in a VM, and I'm very appreciative of all the hard work that went into digitization, and is still going into it right now.

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
- Programmer's Manual AGC Assembly Language

- Skylab quick-reference card
- Space Guidance Analysis (SGA) memos
- Space-Related Applications of Forth

1.2 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)

Line 66 of Luminary099/AOTMARK.agc has junk control characters.

12 ⟨*tmpline line 66 of AOTMARK.agc* 12⟩≡

```
                TS      MARKSTAT        # STORE VAC ADR IN LOW 9 OF MARKSTAT
```

In Comanche055/REENTRY_CONTROL.agc there should be a new line "# Page 882 (page is empty)" added to the end.

This comment has been added recently on page CM0785:

```
##      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
##      <a href="http://nassp.sourceforge.net/wiki/Lunar_Ephemeris_Polynomials">
##      <b>Orbiter</b> NASSP wiki</a>, as well as information about calculation
##      of the <a href="http://nassp.sourceforge.net/wiki/Solar_Ephemeris">
##      solar ephemerides</a>.
```

This comment has been added recently on page CM0948:

```
1SHOTOK         CAF     BIT13           # CHECK ENGINE-ON BIT, NOT PERMITTING
# 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.
```

Part I

Apollo 11 Computer Topics

Chapter 2

The Physical Level

Chapter 3

The Original Programmer's Level

Chapter 4

The Reference Documentation Level

Chapter 5

The Mission Level

Part II

Suplimentary Material

Appendices

Appendix A

Original Files

A.1 README.md

27 ⟨*src/README.md* 27⟩≡

```
Apollo-11
=========

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

####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`.

A.2 HeaderTemplate

29 ⟨*src/Luminary099/HeaderTemplate.agc* 29⟩≡

```
# 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/Luminary099/HeaderTemplate.agc`.

Appendix B

Miscellaneous Files

B.1 Internal vs External File Names

31 ⟨*src/generic.sh* 31⟩≡

```
sys=Comanche055;

ex=AGC_BLOCK_TWO_SELF-CHECK;                in=CM1394SELF;  eval $cmd
ex=ALARM_AND_ABORT;                         in=CM1493ALARM; eval $cmd
ex=ANGLFIND;                                in=CM0399ANGL;  eval $cmd
ex=ASSEMBLY_AND_OPERATION_INFORMATION;      in=CM0002ASM;   eval $cmd
ex=AUTOMATIC_MANEUVERS;                     in=CM1025MANVR; eval $cmd
ex=CM_BODY_ATTITUDE;                        in=CM0883BODY;  eval $cmd
ex=CM_ENTRY_DIGITAL_AUTOPILOT;              in=CM1063AUTO;  eval $cmd
ex=CONIC_SUBROUTINES;                       in=CM1262CONIC; eval $cmd
ex=CONTRACT_AND_APPROVALS;                  in=CM0037APPRV; eval $cmd
ex=CSM_GEOMETRY;                            in=CM0285GEOM;  eval $cmd
ex=DISPLAY_INTERFACE_ROUTINES;              in=CM1452DISP;  eval $cmd
ex=DOWNLINK_LISTS;                          in=CM0170DOWN;  eval $cmd
ex=DOWN-TELEMETRY_PROGRAM;                  in=CM1093TELEM; eval $cmd
ex=ENTRY_LEXICON;                           in=CM0837LEXI;  eval $cmd
ex=ERASABLE_ASSIGNMENTS;                    in=CM0037ERASE; eval $cmd
ex=EXECUTIVE;                               in=CM1208EXEC;  eval $cmd
ex=EXTENDED_VERBS;                          in=CM0236VERBS; eval $cmd
ex=FIXED_FIXED_CONSTANT_POOL;               in=CM1200FIXED; eval $cmd
ex=FRESH_START_AND_RESTART;                 in=CM0181FRESH; eval $cmd
ex=GIMBAL_LOCK_AVOIDANCE;                   in=CM0412AVOID; eval $cmd
ex=GROUND_TRACKING_DETERMINATION_PROGRAM;   in=CM0456GRND;  eval $cmd
ex=IMU_CALIBRATION_AND_ALIGNMENT;           in=CM0423CALIB; eval $cmd
ex=IMU_COMPENSATION_PACKAGE;                in=CM0297ICOMP; eval $cmd
ex=IMU_MODE_SWITCHING_ROUTINES;             in=CM1420IMODE; eval $cmd
```

```
ex=INFLIGHT_ALIGNMENT_ROUTINES;             in=CM1355INFLT; eval $cmd
ex=INTEGRATION_INITIALIZATION;              in=CM1309INTEG; eval $cmd
ex=INTER-BANK_COMMUNICATION;                in=CM1103BKCOM; eval $cmd
ex=INTERPRETER;                             in=CM1107TRPRT; eval $cmd
ex=INTERPRETIVE_CONSTANTS;                  in=CM1205ICONS; eval $cmd
ex=INTERRUPT_LEAD_INS;                      in=CM0131LEAD;  eval $cmd
ex=JET_SELECTION_LOGIC;                     in=CM1039JET;   eval $cmd
ex=KALCMANU_STEERING;                       in=CM0414STEER; eval $cmd
ex=KEYRUPT_UPRUPT;                          in=CM1449KEYRP; eval $cmd
ex=LATITUDE_LONGITUDE_SUBROUTINES;          in=CM1236LAT;   eval $cmd
ex=LUNAR_AND_SOLAR_EPHEMERIDES_SUBROUTINES; in=CM0785EPHEM; eval $cmd
ex=LUNAR_LANDMARK_SELECTION_FOR_CM;         in=CM0936LAND;  eval $cmd
ex=MAIN;                                    in=CM0000MAIN;  eval $cmd
ex=MEASUREMENT_INCORPORATION;               in=CM1252MEAS;  eval $cmd
ex=MYSUBS;                                  in=CM0999MYSUB; eval $cmd
ex=ORBITAL_INTEGRATION;                     in=CM1334ORBIT; eval $cmd
ex=P11;                                     in=CM0533P11;   eval $cmd
ex=P20-P25;                                 in=CM0562P20;   eval $cmd
ex=P30-P37;                                 in=CM0635P30;   eval $cmd
ex=P32-P33_P72-P73;                         in=CM0649P32;   eval $cmd
ex=P34-35_P74-75;                           in=CM0460P34;   eval $cmd
ex=P37_P70;                                 in=CM0890P37;   eval $cmd
ex=P40-P47;                                 in=CM0684P40;   eval $cmd
ex=P51-P53;                                 in=CM0737P51;   eval $cmd
ex=P61-P67;                                 in=CM0789P61;   eval $cmd
ex=P76;                                     in=CM0511P76;   eval $cmd
ex=PHASE_TABLE_MAINTENANCE;                 in=CM1404PHASE; eval $cmd
ex=PINBALL_GAME_BUTTONS_AND_LIGHTS;         in=CM0307GAME;  eval $cmd
ex=PINBALL_NOUN_TABLES;                     in=CM0268NOUN;  eval $cmd
ex=PLANETARY_INERTIAL_ORIENTATION;          in=CM1243INERT; eval $cmd
ex=POWERED_FLIGHT_SUBROUTINES;              in=CM1365POWER; eval $cmd
ex=R30;                                     in=CM0514R30;   eval $cmd
ex=R31;                                     in=CM0505R31;   eval $cmd
ex=R60_62;                                  in=CM0390R60;   eval $cmd
ex=RCS-CSM_DAP_EXECUTIVE_PROGRAMS;          in=CM1037RCDAP; eval $cmd
ex=RCS-CSM_DIGITAL_AUTOPILOT;               in=CM1002RCSDG; eval $cmd
ex=REENTRY_CONTROL;                         in=CM0844REENT; eval $cmd
ex=RESTARTS_ROUTINE;                        in=CM1414RESTR; eval $cmd
ex=RESTART_TABLES;                          in=CM0211RETAB; eval $cmd
ex=RT8_OP_CODES;                            in=CM1508CODES; eval $cmd
ex=S-BAND_ANTENNA_FOR_CM;                   in=CM0934SBAND; eval $cmd
ex=SERVICER207;                             in=CM0819SV207; eval $cmd
ex=SERVICE_ROUTINES;                        in=CM1485SERV;  eval $cmd
ex=SINGLE_PRECISION_SUBROUTINES;            in=CM1207SINGL; eval $cmd
ex=STABLE_ORBIT;                            in=CM0525STABL; eval $cmd
ex=STAR_TABLES;                             in=CM1389STARS; eval $cmd
```

```
ex=SXTMARK;                                    in=CM0222SXTMK; eval $cmd
ex=SYSTEM_TEST_STANDARD_LEAD_INS;              in=CM0420STEST; eval $cmd
ex=T4RUPT_PROGRAM;                             in=CM0133T4RPT; eval $cmd
ex=TAGS_FOR_RELATIVE_SETLOC;                   in=CM0027TAGS;  eval $cmd
ex=TIME_OF_FREE_FALL;                          in=CM1373FALL;  eval $cmd
ex=TPI_SEARCH;                                 in=CM0551TPI;   eval $cmd
ex=TVCDAPS;                                    in=CM0961DAPS;  eval $cmd
ex=TVCEXECUTIVE;                               in=CM0945TVCX;  eval $cmd
ex=TVCINITIALIZE;                              in=CM0937TVCI;  eval $cmd
ex=TVCMASSPROP;                                in=CM0951TVCM;  eval $cmd
ex=TVCRESTARTS;                                in=CM0956TVCR;  eval $cmd
ex=TVCROLLDAP;                                 in=CM0984TROLL; eval $cmd
ex=TVCSTROKETEST;                              in=CM0949STROK; eval $cmd
ex=UPDATE_PROGRAM;                             in=CM1497UPDT;  eval $cmd
ex=WAITLIST;                                   in=CM1221WAIT;  eval $cmd

sys=Luminary099;

ex=AGC_BLOCK_TWO_SELF_CHECK;                   in=LM1284SELF;  eval $cmd
ex=AGS_INITIALIZATION;                         in=LM0206AGSI;  eval $cmd
ex=ALARM_AND_ABORT;                            in=LM1381ALARM; eval $cmd
ex=AOSTASK_AND_AOSJOB;                         in=LM1485AOST;  eval $cmd
ex=AOTMARK;                                    in=LM0244AOTMK; eval $cmd
ex=ASCENT_GUIDANCE;                            in=LM0843ASCNT; eval $cmd
ex=ASSEMBLY_AND_OPERATION_INFORMATION;         in=LM0002ASM;   eval $cmd
ex=ATTITUDE_MANEUVER_ROUTINE;                  in=LM0342ATTIT; eval $cmd
ex=BURN_BABY_BURN--MASTER_IGNITION_ROUTINE;    in=LM0731BURN;  eval $cmd
ex=CONIC_SUBROUTINES;                          in=LM1159CONIC; eval $cmd
ex=CONTRACT_AND_APPROVALS;                     in=LM0001APPRV; eval $cmd
ex=CONTROLLED_CONSTANTS;                       in=LM0038CONST; eval $cmd
ex=DAPIDLER_PROGRAM;                           in=LM1410IDLER; eval $cmd
ex=DAP_INTERFACE_SUBROUTINES;                  in=LM1406DAP;   eval $cmd
ex=DISPLAY_INTERFACE_ROUTINES;                 in=LM1341DISP;  eval $cmd
ex=DOWNLINK_LISTS;                             in=LM0193DOWN;  eval $cmd
ex=DOWN_TELEMETRY_PROGRAM;                     in=LM0988TELEM; eval $cmd
ex=ERASABLE_ASSIGNMENTS;                       in=LM0090ERASE; eval $cmd
ex=EXECUTIVE;                                  in=LM1103EXEC;  eval $cmd
ex=EXTENDED_VERBS;                             in=LM0262VERBS; eval $cmd
ex=FINDCDUW--GUIDAP_INTERFACE;                 in=LM0908CDUW;  eval $cmd
ex=FIXED_FIXED_CONSTANT_POOL;                  in=LM1095FIXED; eval $cmd
ex=FLAGWORD_ASSIGNMENTS;                       in=LM0061FLAG;  eval $cmd
ex=FRESH_START_AND_RESTART;                    in=LM0211FRESH; eval $cmd
ex=GIMBAL_LOCK_AVOIDANCE;                      in=LM0364AVOID; eval $cmd
ex=GROUND_TRACKING_DETERMINATION_PROGRAM;      in=LM0654GRND;  eval $cmd
ex=IMU_COMPENSATION_PACKAGE;                   in=LM0326ICOMP; eval $cmd
ex=IMU_MODE_SWITCHING_ROUTINES;                in=LM1309IMODE; eval $cmd
```

```
ex=IMU_PERFORMANCE_TEST_2;                  in=LM0373TEST2; eval $cmd
ex=IMU_PERFORMANCE_TESTS_4;                 in=LM0382TEST4; eval $cmd
ex=INFLIGHT_ALIGNMENT_ROUTINES;             in=LM1249INFLT; eval $cmd
ex=INPUT_OUTPUT_CHANNEL_BIT_DESCRIPTIONS;   in=LM0054INOUT; eval $cmd
ex=INTEGRATION_INITIALIZATION;              in=LM1205INTEG; eval $cmd
ex=INTER-BANK_COMMUNICATION;                in=LM0998BKCOM; eval $cmd
ex=INTERPRETER;                             in=LM1002TRPRT; eval $cmd
ex=INTERPRETIVE_CONSTANT;                   in=LM1100ICONS; eval $cmd
ex=INTERRUPT_LEAD_INS;                      in=LM0153LEAD;  eval $cmd
ex=KALCMANU_STEERING;                       in=LM0365STEER; eval $cmd
ex=KALMAN_FILTER;                           in=LM1470KALMN; eval $cmd
ex=KEYRUPT_UPRUPT;                          in=LM1338KEYRP; eval $cmd
ex=LAMBERT_AIMPOINT_GUIDANCE;               in=LM0651LAMBT; eval $cmd
ex=LANDING_ANALOG_DISPLAYS;                 in=LM0898ANALG; eval $cmd
ex=LATITUDE_LONGITUDE_SUBROUTINES;          in=LM1133LAT;   eval $cmd
ex=LEM_GEOMETRY;                            in=LM0320GEOM;  eval $cmd
ex=LUNAR_AND_SOLAR_EPHEMERIDES_SUBROUTINES; in=LM0984EPHEM; eval $cmd
ex=LUNAR_LANDING_GUIDANCE_EQUATIONS;        in=LM0798EQUAT; eval $cmd
ex=MAIN;                                    in=LM0000MAIN;  eval $cmd
ex=MEASUREMENT_INCORPORATION;               in=LM1149MEAS;  eval $cmd
ex=ORBITAL_INTEGRATION;                     in=LM1227ORBIT; eval $cmd
ex=P12;                                     in=LM0838P12;   eval $cmd
ex=P20-P25;                                 in=LM0492P20;   eval $cmd
ex=P30_P37;                                 in=LM0615P30;   eval $cmd
ex=P32-P35_P72-P75;                         in=LM0618P32;   eval $cmd
ex=P34-35_P74-75;                           in=LM0658P34;   eval $cmd
ex=P40-P47;                                 in=LM0752P40;   eval $cmd
ex=P51-P53;                                 in=LM0926P51;   eval $cmd
ex=P70-P71;                                 in=LM0829P70;   eval $cmd
ex=P76;                                     in=LM0709P76;   eval $cmd
ex=P-AXIS_RCS_AUTOPILOT;                    in=LM1421PAXIS; eval $cmd
ex=PHASE_TABLE_MAINTENANCE;                 in=LM1294PHASE; eval $cmd
ex=PINBALL_GAME_BUTTONS_AND_LIGHTS;         in=LM0390GAME;  eval $cmd
ex=PINBALL_NOUN_TABLES;                     in=LM0301NOUN;  eval $cmd
ex=PLANETARY_INERTIAL_ORIENTATION;          in=LM1140INERT; eval $cmd
ex=POWERED_FLIGHT_SUBROUTINES;              in=LM1259POWER; eval $cmd
ex=Q_R-AXIS_RCS_AUTOPILOT;                  in=LM1442QRAXS; eval $cmd
ex=R30;                                     in=LM0712R30;   eval $cmd
ex=R31;                                     in=LM0703R31;   eval $cmd
ex=R60_62;                                  in=LM0472R60;   eval $cmd
ex=R63;                                     in=LM0338R63;   eval $cmd
ex=RADAR_LEADIN_ROUTINES;                   in=LM0490RADAR; eval $cmd
ex=RCS_FAILURE_MONITOR;                     in=LM0190FAIL;  eval $cmd
ex=RESTARTS_ROUTINE;                        in=LM1303RESTR; eval $cmd
ex=RESTART_TABLES;                          in=LM0238RETAB; eval $cmd
ex=RTB_OP_CODES;                            in=LM1397CODES; eval $cmd
```

```
ex=S-BAND_ANTENNA_FOR_LM;                in=LM0486SBAND; eval $cmd
ex=SERVICER;                             in=LM0857SERVR; eval $cmd
ex=SERVICE_ROUTINES;                     in=LM1374SERV;  eval $cmd
ex=SINGLE_PRECISION_SUBROUTINES;         in=LM1102SINGL; eval $cmd
ex=SPS_BACK-UP_RCS_CONTROL;              in=LM1507BKUP;  eval $cmd
ex=STABLE_ORBIT;                         in=LM0723STABL; eval $cmd
ex=SYSTEM_TEST_STANDARD_LEAD_INS;        in=LM0370STEST; eval $cmd
ex=T4RUPT_PROGRAM;                       in=LM0155T4RPT; eval $cmd
ex=T6-RUPT_PROGRAMS;                     in=LM1403T6RPT; eval $cmd
ex=TAGS_FOR_RELATIVE_SETLOC;             in=LM0028TAGS;  eval $cmd
ex=THE_LUNAR_LANDING;                    in=LM0785LANDG; eval $cmd
ex=THROTTLE_CONTROL_ROUTINES;            in=LM0793THROT; eval $cmd
ex=TIME_OF_FREE_FALL;                    in=LM1268FALL;  eval $cmd
ex=TJET_LAW;                             in=LM1460TJET;  eval $cmd
ex=TRIM_GIMBAL_CNTROL_SYSTEM;            in=LM1472TRIM;  eval $cmd
ex=UPDATE_PROGRAM;                       in=LM1386UPDT;  eval $cmd
ex=WAITLIST;                             in=LM1117WAIT;  eval $cmd
```

This code is written to file `src/generic.sh`.

B.2 Check for Changes

35a ⟨*src/check.sh* 35a⟩≡

```
#!/bin/bash

cmd='echo "XX $in XX"; diff origsrc/$sys/$ex.agc src/$sys/$in.agc'
source generic.sh
```

This code is written to file `src/check.sh`.

B.3 Export with Original File Names

35b ⟨*src/export.sh* 35b⟩≡

```
#!/bin/bash

rm -r -f tmp
mkdir -p tmp/{Comanche055,Luminary099}

cmd='echo "XX $in XX"; cp $sys/$in.agc tmp/$sys/$ex.agc'
source generic.sh
```

This code is written to file `src/export.sh`.

B.4 Migrate to Internal File Names

36 ⟨*src/migrate.sh* 36⟩≡

```
#!/bin/bash

cmd='echo "XX $in XX"; git mv $sys/$ex.agc $sys/$in.agc'
source generic.sh
```

This code is written to file `src/migrate.sh`.

Appendix C

Notes, Bibliography and Indexes

C.1 Chunks

⟨*src/check.sh* 35a⟩
⟨*src/export.sh* 35b⟩
⟨*src/generic.sh* 31⟩
⟨*src/Luminary099/HeaderTemplate.agc* 29⟩
⟨*src/migrate.sh* 36⟩
⟨*src/README.md* 27⟩
⟨*tmpline line 66 of AOTMARK.agc* 12⟩

C.2 Index