

13:35:31 24-OCT-80

000.000

```
1  DEBUG  EQU      0          ASSEMBLE FOR DEBUG
3  ***    DDDVD  -  DD: Device Driver
4  *
5  *      DDDVD is the device driver for the DD: device
6  *      the initial incarnation of which is the Remex
7  *      M-47.
8  *
9  *
10 *      Copyright Heath Co, 1980
11 *
12 *      G. C.                80.02.GC
13 *
```

Assembly Constants

13:35:31 24-OCT-80

16 ** Assembly Constants

17 *

18

000.000

19

XTEXT H47PAR

21X ** H47PAR - H47 Parameters

22X *

23X

000.015

24X NSPTS

EQU

13

Sectors/Track Single Density [1-13]

000.032

25X NSPTD

EQU

26

Sectors/Track Double Density [1-26]

26X

000.115

27X NTRK

EQU

76+1

Number of Tracks

[0-76]

000.000	30	XTEXT	ASCII	
	32X **	ASCII CHARACTER EQUIVALENCES.		
	33X			
000.015	34X CR	EQU	13	CARRIAGE RETURN
000.012	35X LF	EQU	10	LINE FEED
000.200	36X NULL	EQU	200Q	PAD CHARACTER
000.000	37X NUL2	EQU	0	
000.007	38X BELL	EQU	7	BELL CHARACTER
000.177	39X RUBOUT	EQU	177Q	
000.010	40X BKSP	EQU	10Q	CTL-H
000.026	41X C.SYN	EQU	26Q	SYNC
000.002	42X C.STX	EQU	2	STX
000.047	43X QUOTE	EQU	47Q	
000.011	44X TAB	EQU	11Q	
000.033	45X ESC	EQU	33Q	
000.012	46X NL	EQU	12Q	NEW LINE (HDS SYSTEMS)
000.212	47X ENL	EQU	NL+200Q	NL + END-OF-LINE-FLAG
000.014	48X FF	EQU	14Q	FORM FEED
000.001	49X CTLA	EQU	01Q	CTL-A
000.002	50X CTLB	EQU	02Q	CTL-B
000.003	51X CTLC	EQU	03Q	CTL-C
000.004	52X CTLD	EQU	04Q	CTL-D
000.017	53X CTLO	EQU	17Q	CTL-O
000.020	54X CTLP	EQU	20Q	CTL-P
000.021	55X CTLQ	EQU	21Q	CTL-Q
000.023	56X CTLS	EQU	23Q	CTL-S
000.032	57X CTLZ	EQU	32Q	CTL-Z
000.000	58	XTEXT	ECDEF	

	60X **	ERROR CODE DEFINITIONS.		
	61X			
000.000	62X	ORG	0	
000.000	63X	DS	1	NO ERROR #0
000.001	64X EC.EOF	DS	1	END OF FILE
000.002	65X EC.EOM	DS	1	END OF MEDIA
000.003	66X EC.ILC	DS	1	ILLEGAL SYSCALL CODE
000.004	67X EC.CNA	DS	1	CHANNEL NOT AVAILABLE
000.005	68X EC.DNS	DS	1	DEVICE NOT SUITABLE
000.006	69X EC.IDN	DS	1	ILLEGAL DEVICE NAME
000.007	70X EC.IFN	DS	1	ILLEGAL FILE NAME
000.010	71X EC.NRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011	72X EC.FNO	DS	1	CHANNEL NOT OPEN
000.012	73X EC.ILR	DS	1	ILLEGAL REQUEST
000.013	74X EC.FUC	DS	1	FILE USAGE CONFLICT
000.014	75X EC.FNF	DS	1	FILE NAME NOT FOUND
000.015	76X EC.UND	DS	1	UNKNOWN DEVICE
000.016	77X EC.ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017	78X EC.DIF	DS	1	DIRECTORY FULL
000.020	79X EC.IFC	DS	1	ILLEGAL FILE CONTENTS
000.021	80X EC.NEM	DS	1	NOT ENOUGH MEMORY
000.022	81X EC.RF	DS	1	READ FAILURE
000.023	82X EC.WF	DS	1	WRITE FAILURE

External Definitions

ECDEF

13:35:34 24-OCT-80

000.024	83X	EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025	84X	EC.WP	DS	1	DISK WRITE PROTECTED
000.026	85X	EC.FAP	DS	1	FILE ALREADY PRESENT
000.027	86X	EC.DDA	DS	1	DEVICE DRIVER ABORT
000.030	87X	EC.FL	DS	1	FILE LOCKED
000.031	88X	EC.FAO	DS	1	FILE ALREADY OPEN
000.032	89X	EC.IS	DS	1	ILLEGAL SWITCH
000.033	90X	EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	91X	EC.FNR	DS	1	FILE NAME REQUIRED
000.035	92X	EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	93X	EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037	94X	EC.ILV	DS	1	ILLEGAL VALUE
000.040	95X	EC.ILO	DS	1	ILLEGAL OPTION
000.041	96X	EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	97X	EC.NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	98X	EC.FOD	DS	1	FILE OPEN ON DEVICE
000.044	99X	EC.NPM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	100X	EC.DNI	DS	1	DISK NOT INITIALIZED
000.046	101X	EC.DNR	DS	1	DISK IS NOT READABLE
000.047	102X	EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050	103X	EC.NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051	104X	EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	105X	EC.IOI	DS	1	ILLEGAL OVERLAY INDEX
000.053	106X	EC.OTL	DS	1	OVERLAY TOO LARGE
000.054	107		XTEXT	DDDEF	

109X ** DEVICE DRIVER COMMUNICATION FLAGS.

110X *

111X

000.000 112X ORG 0

113X

000.000 114X DC.REA DS 1 READ

000.001 115X DC.WRI DS 1 WRITE

000.002 116X DC.RER DS 1 READ REGARDLESS

000.003 117X DC.OPR DS 1 OPEN FOR READ

000.004 118X DC.OPW DS 1 OPEN FOR WRITE

000.005 119X DC.OPU DS 1 OPEN FOR UPDATE

000.006 120X DC.CLO DS 1 CLOSE

000.007 121X DC.ABT DS 1 ABORT

000.010 122X DC.MOU DS 1 MOUNT DEVICE

000.011 123X DC.LOD DS 1 LOAD DEVICE DRIVER

000.012 124X DC.RDY DS 1 Device Ready /80.04.GC/

000.013 125X DC.MAX DS 1 MAXIMUM ENTRY INDEX

000.014 126 XTEXT DDDEF

```

128X **      DIRECTORY DEVICE FORMAT DEFINITION.          /80.09.sc/
129X *
130X *      Modified:          Sep-80
131X *      No longer require 2 sectors per group
132X *      Reserved Group Table dynamically allocated
133X *
134X
000.000      135X      ORG      0
136X
000.000      137X DDF.BOO DS      9      2K BOOT PROGRAM
000.011      138X DDF.BOL EQU      *      LENGTH OF BOOT
000.011      139X DDF.LAB DS      1      LABEL SECTOR
000.012      140X DDF.USR DS      0      BEGINNING OF OPEN SPACE
000.012      141      XTEXT      DEVDEF

```

```

143X **      DEVICE TABLE ENTRIES.
144X
000.000      145X      ORG      0
146X
000.000      147X DEV.NAM DS      2      DEVICE NAME
000.000      148X DV.EL EQU      00000000B      END OF DEVICE LIST FLAG
000.001      149X DV.NU EQU      00000001B      DEVICE ENTRY NOT IN USE
150X
000.002      151X DEV.RES DS      1      DRIVER RESIDENSE CODE
000.001      152X DR.IM EQU      00000001B      DRIVER IN MEMORY
000.002      153X DR.PR EQU      00000010B      DRIVER PERMINANTLY RESIDENT
154X
000.003      155X DEV.JMP DS      1      JMP TO PROCESSOR
000.004      156X DEV.DDA DS      2      DRIVER ADDRESS
000.006      157X DEV.FLG DS      1      FLAG BYTE
000.001      158X DT.DD EQU      00000001B      DIRECTORY DEVICE
000.002      159X DT.CR EQU      00000010B      CAPABLE OF READ OPERATION
000.004      160X DT.CW EQU      00000100B      CAPABLE OF WRITE OPERATION
000.010      161X DT.RN EQU      00001000B      Capable of random access          /80.02.sc/
000.020      162X DT.CH EQU      00010000B      Capable of Character mode          /80.02.sc/
163X
000.007      164X DEV.MUM DS      1      MOUNTED UNIT MASK
000.010      165X DEV.MNU DS      1      MAXIMUM NUMBER OF UNITS
000.011      166X DEV.UNT DS      2      ADDRESS OF UNIT SPECIFIC DATA TABLE
167X
000.013      168X DEV.DVL DS      2      DRIVER BYTE LENGTH
000.015      169X DEV.DVG DS      1      DRIVER ROUTINE GROUP ADDRESS
170X
000.016      171X DEVELEN EQU      *      DEVICE TABLE ENTRY LENGTH

```

External Definitions

UNT.TAB

13:35:36 24-OCT-80

173X ** UNIT SPECIFIC DEVICE DATA TABLE ENTRIES

174X					
000.000	175X	ORG	0		
	176X				
000.000	177X	UNT.FLG DS	1	UNIT SPECIFIC *DEV.FLG*	
000.001	178X	UNT.SPG DS	1	Sectors Per Group	/80.04.GC/
000.002	179X	UNT.GRT DS	2	ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)	
000.004	180X	UNT.GTS DS	2	GRT SECTOR NUMBER	
000.006	181X	UNT.DIS DS	2	DIRECTORY FIRST SECTOR NUMBER	
	182X				
000.010	183X	UNT.SIZ EQU	*	SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT	
000.010	184	XTEXT	PICDEF		

186X ** PIC FORMAT EQUIVALENCES.

	187X				
000.000	188X	ORG	0		
	189X				
000.000	190X	PIC.ID DS	1	377Q = BINARY FILE FLAG	
000.001	191X	DS	1	FILE TYPE (FT.PIC)	
000.002	192X	PIC.LEN DS	2	LENGTH OF ENTIRE RECORD	
000.004	193X	PIC.PTR DS	2	INDEX OF START OF PIC TABLE	
	194X				
000.006	195X	PIC.COD DS	0	CODE STARTS HERE	
000.006	196	XTEXT	DVDDEF		

198X ** DEVICE DRIVER EQUIVALENCES.

	199X				
000.307	200X	DVD.FLV EQU	307Q	DEVICE DRIVER FLAG VALUE	
	201X				
000.006	202X	ORG	PIC.COD	STARTS AT PIC CODE AREA	
	203X				
000.006	204X	DVD.DVD DS	1	MUST BE DVD.FLV, FLAGS TO HDOS AS DRIVER	
000.007	205X	DVD.CAP DS	1	DEVICE CAPABILITY FLAG	
000.010	206X	DVD.MUM DS	1	MOUNTED UNIT MASK	
000.011	207X	DVD.MNU DS	1	MAXIMUM NUMBER OF UNITS	
000.012	208X	DVD.UFL DS	8	UNIT SUB-CAPABILITY FLAGS FOR UNITS 0-7	
000.022	209X	DVD.SET DS	1	= DVD.FLV IFF DRIVER WILL TAKE SET OPTIONS	
000.023	210X	DVD.INP DS	2	Pointer to Init Code	/80.07.sc/
000.025	211X	DS	22	RESERVED, MUST BE 0	/80.07.sc/
000.053	212X	DVD.STE EQU	*	ENTRY FOR 'SET' INVOCATION	
	213X				
002.000	214X	DVD.ENT EQU	2000A	DRIVER ENTRY POINT (MUST BE MULT OF 256)	
000.053	215	XTEXT	DIRDEF		

```

217X **      DIRECTORY ENTRY FORMAT.
218X
000.000      219X      ORG      0
220X
221X
000.377      222X DF.EMP EQU      3770      FLAGS ENTRY EMPTY
000.376      223X DF.CLR EQU      3760      FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
224X
000.000      225X DIR.NAM DS      8      NAME
000.010      226X DIR.EXT DS      3      EXTENSION
000.013      227X DIR.PRO DS      1      PROJECT
000.014      228X DIR.VER DS      1      VERSION
000.015      229X DIRIDL EQU      *      FILE IDENTIFICATION LENGTH
230X
000.015      231X DIR.CLU DS      1      CLUSTER FACTOR
000.016      232X DIR.FLG DS      1      FLAGS
000.017      233X      DS      1      RESERVED
000.020      234X DIR.FGN DS      1      FIRST GROUP NUMBER
000.021      235X DIR.LGN DS      1      LAST GROUP NUMBER
000.022      236X DIR.LSI DS      1      LAST SECTOR INDEX (IN LAST GROUP)
000.023      237X DIR.CRD DS      2      CREATION DATE
000.025      238X DIR.ALD DS      2      LAST ALTERATION DATE
239X
000.027      240X DIRELEN EQU      *      DIRECTORY ENTRY LENGTH
000.027      241      XTEXT      HOSEQU

```

```

243X **      HDOS SYSTEM EQUIVALENCES.
244X *
245X
024.000      246X S.GRT0 EQU      24000A      SYSTEM AREA FOR GRT0
025.000      247X S.GRT1 EQU      25000A      SYSTEM AREA FOR GRT1
026.000      248X S.GRT2 EQU      26000A      SYSTEM AREA FOR GRT2
249X
030.000      250X ROMBOOT EQU      30000A      ROM BOOT ENTRY
251X
040.100      252X      ORG      40100A      FREE SPACE FROM PAM-8
253X
040.100      254X      DS      8      JUMP TO SYSTEM EXIT
040.110      255X D.CON DS      16      DISK CONSTANTS
040.130      256X SYDD EQU      *      SYSTEM DISK ENTRY POINT
040.130      257X D.VEC DS      24*3      SYSTEM ROM ENTRY VECTORS
040.240      258X D.RAM DS      31      SYSTEM ROM WORK AREA
040.277      259X S.VAL DS      36      SYSTEM VALUES
040.343      260X S.INT DS      115      SYSTEM INTERNAL WORK AREAS
041.126      261X      DS      16
041.146      262X S.SOVR DS      2      STACK OVERFLOW WARNING
041.150      263X      DS      42200A-*      SYSTEM STACK
001.032      264X STACKL EQU      *-S.SOVR      STACK SIZE
265X
042.200      266X STACK EQU      *      LWA+1 SYSTEM STACK
042.200      267X USERFWA EQU      *      USER FWA
042.200      268      XTEXT      ESINT

```

```
270X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
271X *
272X *      THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
273X *      MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
274X
275X
040.343      276X      ORG      S.INT
277X
278X **      CONSOLE STATUS FLAGS
279X
040.343      280X S.CDB  DS      1      CONSOLE DESCRIPTOR BYTE
000.000      281X CDB.H85 EQU      00000000B
000.001      282X CDB.H84 EQU      00000001B      =0 IF H8-5, =1 IF H8-4
040.344      283X S.BAUD DS      2      [0-14] H8-4 BAUD RATE, =0 IF H8-5
284X *      [15] =1 IF BAUD RATE => 2 STOP BITS
285X
286X **      TABLE ADDRESS WORDS
287X
040.346      288X S.DLINK DS      2      ADDRESS OF DATA IN HDOS CODE
040.350      289X S.DFWA DS      2      FWA OVERLAY TABLE
040.352      290X S.CFWA DS      2      FWA CHANNEL TABLE
040.354      291X S.DFWA DS      2      FWA DEVICE TABLE
040.356      292X S.RFWA DS      2      FWA RESIDENT HDOS CODE
293X
294X **      DEVICE DRIVER DELAYED LOAD FLAGS
295X
040.360      296X S.DDLDA DS      2      DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362      297X S.DPLEN DS      2      CODE LENGTH IN BYTES
040.364      298X S.DDGRP DS      1      GROUP NUMBER FOR DRIVER
040.365      299X      DS      1      HOLD PLACE
300X *S.DDSEC DS      2      SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366      301X S.DDDTA DS      2      DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370      302X S.DDDPC DS      1      OPEN OP CODE PENDING
303X
304X **      OVERLAY MANAGEMENT FLAGS
305X
000.001      306X OVL.IN  EQU      00000001B      IN MEMORY
000.002      307X OVL.RES EQU      00000010B      PERMINANTLY RESIDENT
000.014      308X OVL.NUM EQU      00001100B      OVERLAY NUMBER MASK
000.200      309X OVL.UCS EQU      10000000B      USER CODE SWAPPED FOR OVERLAY
310X
040.371      311X S.OVLFL DS      1      OVERLAY FLAG
040.372      312X S.UCSF  DS      2      FWA SWAPPED USER CODE
040.374      313X S.UCSL DS      2      LENGTH SWAPPED USER CODE
040.376      314X S.OVLS  DS      2      SIZE OF OVERLAY CODE
041.000      315X S.OVLE  DS      2      ENTRY POINT OF OVERLAY CODE
316X
041.002      317X S.SSN  DS      2      SWAP AREA SECTOR NUMBER
041.004      318X S.OSN  DS      2      OVERLAY SECTOR NUMBER
319X
320X *      SYSCALL PROCESSING WORK AREAS
321X
041.006      322X S.CACC DS      1      (ACC) UPON SYSCALL
041.007      323X S.CODE DS      1      SYSCALL INDEX IN PROGRESS
324X
325X *      JUMPS TO ROUTINES IN RESIDENT HDOS CODE
```


	326X			
041.010	327X S.JUMPS DS	0		START OF DUMP VECTORS
041.010	328X S.SDD DS	3		JUMP TO STAND-IN DEVICE DRIVER
041.013	329X S.FASER DS	3		JUMP TO FATERR (FATAL SYSTEM ERROR)
041.016	330X S.DIREA DS	3		JUMP TO DIREAD (DISK FILE READ)
041.021	331X S.FCI DS	3		JUMP TO FCI (FETCH CHANNEL INFO)
041.024	332X S.SCI DS	3		JUMP TO SCI (STORE CHANNEL INFO)
041.027	333X S.GUP DS	3		JUMP TO GUP (GET UNIT POINTER)
	334X			
041.032	335X S.MOUNT DS	1		<>0 IF THE SYSTEM DISK IS MOUNTED
041.033	336X S.DCS DS	1		DEFAULT CLUSTER SIZE-1
	337X			
041.034	338X S.BOOTF DS	1		BOOT FLAGS
000.001	339X BBOOT.P EQU	00000001B		EXECUTE PROLOGUE UPON BOOTUP
	340X			
	341X *			STACK VALUE SAVED FOR OVERLAY SYSCALLS
	342X			
041.035	343X S.OVSTK DS	2		VALUE OF SP UPON SYSCALLS USING OVERLAY
	344X			
041.037	345X DS	1		RESERVED
	347X **			ACTIVE I/O AREA.
	348X *			
	349X *			THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
	350X *			CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
	351X *			THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
	352X *			
	353X *			NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
	354X *			FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
	355X *			8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
	356X *			COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
	357X *			BACKDATED AFTER PROCESSING.
	358X			
041.040	359X AIO.VEC DS	3		JUMP INSTRUCTION
041.041	360X AIO.DDA EQU	*-2		DEVICE DRIVER ADDRESS
041.043	361X AIO.FLG DS	1		FLAG BYTE
041.044	362X AIO.GRT DS	2		ADDRESS OF GROUP RESERV TABLE
041.046	363X AIO.SPG DS	1		SECTORS PER GROUP
041.047	364X AIO.CGN DS	1		CURRENT GROUP NUMBER
041.050	365X AIO.CSI DS	1		CURRENT SECTOR INDEX
041.051	366X AIO.LGN DS	1		LAST GROUP NUMBER
041.052	367X AIO.LSI DS	1		LAST SECTOR INDEX
041.053	368X AIO.ITA DS	2		DEVICE TABLE ADDRESS
041.055	369X AIO.DES DS	2		DIRECTORY SECTOR
041.057	370X AIO.DEV DS	2		DEVICE CODE
041.061	371X AIO.UNI DS	1		UNIT NUMBER (0-9)
	372X			
041.062	373X AIO.DIR DS	DIRELEN		DIRECTORY ENTRY
	374X			
041.111	375X AIO.CNT DS	1		SECTOR COUNT
041.112	376X AIO.EOM DS	1		END OF MEDIA FLAG
041.113	377X AIO.EOF DS	1		END OF FILE FLAG
041.114	378X AIO.TFP DS	2		TEMP FILE POINTERS

041.118 379X AIO:CHA DS 2 ADDRESS OF CHANNEL BLOCK (IOC:DDA)

041.120 381X S:BDA DS 1 Boot Device Address (Setup by ROM) /80.09.sc/
041.121 382X S:SCR DS 2 SYSTEM SCRATCH AREA ADDRESS
041.123 383 XTEXT SETCAL

385X ** SETCAL - FIXED ADDRESS ROUTINES IN SET

386X *

387X *

THESE VECTORS ARE FIXED ENTRY POINTS INTO THE
SET PROGRAM TO UTILIZED BY DEVICE DRIVERS IN
PROCESSING SET COMMANDS.

388X *

389X *

390X *

391X

042.201 392X ORG USERFWA+1

393X

042.201 394X \$SNA DS 3

395X

042.204 396X \$DCS DS 3

397X

042.207 398X \$CNA DS 3

399X

042.212 400X \$FST DS 3

401X

042.215 402X \$TBLS DS 3

403X

042.220 404X \$WTBLS DS 3

405X

042.223 406X \$LBD DS 3

407X

042.226 408X \$SOP DS 3

409X

042.231 410X \$PBF DS 3

411X

042.234 412X \$PBV DS 3

413X

042.237 414X DS 60 RESERVED

042.333 417 XTEXT H47DEF

419X ** H47DEF - H47 Constant Definitions
420X *

422X * Z-80 INSTRUCTIONS

423X
242.355 424X M.INI EQU 10100010B*256+11101101B INI INSTRUCTION
243.355 425X M.OUTI EQU 10100011B*256+11101101B OUTI INSTRUCTION

427X ** DISK INTERFACE CONSTANTS

428X *
429X
000.170 430X D.STA EQU 170H INTERFACE STATUS PORT
000.171 431X D.DAT EQU D.STA+1 DATA PORT
432X
000.001 433X S.ERR EQU 00000001B ERROR BIT
000.040 434X S.DON EQU 00100000B DONE
000.100 435X S.IEN EQU 01000000B INTERRUPT ENABLE
000.200 436X S.DTR EQU 10000000B DATA TERMINAL REQUEST
437X
000.002 438X S.SW0 EQU 00000010B DIP SWITCH: 0
000.004 439X S.SW1 EQU 00000100B DIP SWITCH: 1
000.010 440X S.SW2 EQU 00001000B DIP SWITCH: 2
000.020 441X S.SW3 EQU 00010000B DIP SWITCH: 3
442X
000.002 443X W.RES EQU 00000010B RESET COMMAND

445X ** STATUS BYTE FLAGS

446X *
447X
000.200 448X SB.UNR EQU 10000000B UNIT NOT READY
000.100 449X SB.WPD EQU 01000000B WRITE PROTECTED DRIVE
000.040 450X SB.DLD EQU 00100000B DELETED DATA
000.020 451X SB.NRF EQU 00010000B NO RECORD FOUND
000.010 452X SB.CRC EQU 00001000B CRC ERROR
000.004 453X SB.LTD EQU 00000100B LATE DATA
000.002 454X SB.ILC EQU 00000010B ILLEGAL COMMAND
000.001 455X SB.BTO EQU 00000001B BAD TRACK OVERFLOW

457X ** AUXILLARY STATUS BYTE FLAGS

458X *

459X

000.100	460X	AS.ODD	EQU	01000000B	TRACK 0 DOUBLE DENSITY
000.040	461X	AS.IDD	EQU	00100000B	TRACK 1-76 DOUBLE DENSITY
000.020	462X	AS.S1A	EQU	00010000B	SIDE 1 AVAILABLE
000.003	463X	AS.SLM	EQU	00000011B	SECTOR LENGTH MASK

465X ** DISK COMMANDS

466X *

467X

000.000	468X		ORG	0	
000.000	469X	DD.BOOT	DS	1	BOOT
000.001	470X	DD.RST	DS	1	READ STATUS
000.002	471X	DD.RAS	DS	1	READ AUX. STATUS
000.003	472X	DD.LSC	DS	1	LOAD SECTOR COUNT
000.004	473X	DD.RAD	DS	1	READ ADDRESS OF LAST SECTOR ACCESSED
000.005	474X	DD.REA	DS	1	READ SECTORS
000.006	475X	DD.WRI	DS	1	WRITE SECTORS
000.007	476X	DD.REAB	DS	1	READ SECTORS BUFFERED
000.010	477X	DD.WRIB	DS	1	WRITE SECTORS BUFFERED
000.011	478X	DD.WRD	DS	1	DD.WRI + DELETED
000.012	479X	DD.WRBD	DS	1	DD.WRIB + DELETED
000.013	480X	DD.CPY	DS	1	COPY
000.014	481X	DD.FRM0	DS	1	FORMAT IBM SD
000.015	482X	DD.FRM1	DS	1	FORMAT SD
000.016	483X	DD.FRM2	DS	1	FORMAT IBM DD
000.017	484X	DD.FRM3	DS	1	FORMAT DD
000.020	485X	DD.RRDY	DS	1	Read Ready (conflict with DD.SPF0)

487X ** Special De-Bug Functions

488X *

489X

000.020	490X		ORG	010H	
000.020	491X	DD.SPF0	DS	1	SPECIAL FUNCTION 0
000.021	492X	DD.SPF1	DS	1	SPECIAL FUNCTION 1
000.022	493X	DD.SPF2	DS	1	SPECIAL FUNCTION 2
000.023	494X	DD.SPF3	DS	1	SPECIAL FUNCTION 3
000.024	495X	DD.SPF4	DS	1	SPECIAL FUNCTION 4
000.025	496X	DD.SPF5	DS	1	SPECIAL FUNCTION 5

498X ** Special Heath Functions

499X *

500X

000.200

501X

ORG

080H

000.200

502X

DD.SDC

DS

1

SET DRIVE CHARACTERISTICS

000.201

503X

DD.ST

DS

1

SEEK TO TRACK

000.202

504X

DD.DS

DS

1

DISK STATUS

000.203

505X

DD.RDL

DS

1

READ LOGICAL

000.204

506X

DD.WTL

DS

1

WRITE LOGICAL

000.205

507X

DD.RDBL

DS

1

READ BUFFERED LOGICAL

000.206

508X

DD.WTBL

DS

1

WRITE BUFFERED LOGICAL

000.207

509X

DD.WTDL

DS

1

WRITE DELETED DATA LOGICAL

000.210

510X

DD.WDLB

DS

1

WRITE BUFFERED DELETED DATA LOGICAL

512X ** Useful Flags

513X *

514X

000.000

515X

UNT.0

EQU

00000000B

Unit: 0

000.040

516X

UNT.1

EQU

00100000B

Unit: 1

000.100

517X

UNT.2

EQU

01000000B

Unit: 2

000.140

518X

UNT.3

EQU

01100000B

Unit: 3

519X

000.140

520X

UNT.M

EQU

UNT.0!UNT.1!UNT.2!UNT.3 Unit Mask

521X

522X

523X

000.000

524X

SID.0

EQU

00000000B

Side: 0

000.200

525X

SID.1

EQU

10000000B

Side: 1

526X

000.200

527X

SID.M

EQU

SID.0!SID.1

Side Mask

528X

529X

530X

000.037

531X

SEC.M

EQU

00011111B

Track Mask

532X

533X

534X

004.000

535X

SSIZ.M

EQU

1024

Maximum Sector Size

536X

537X

538X

*C.128

EQU

128

539X

*C.256

EQU

256

540X

*C.26

EQU

26

```
543 **      Device Unit Definition
544 *
545
546 CODE     PIC
547
000.017    548 DDCAP EQU      DT,CW+DT,CR+DT,DD+DT,RN      Read, Write, Directory, Random
549
000.006 307 550         DB      DDDFLV      DEVICE DRIVER FLAG
000.007 017 551         DB      DDCAP      Device Capabilities
000.010 000 552         DB      00000000B    Mounted Units Mask
000.011 002 553 DDMNU  DB      2          Maximum of 2 Units
000.012 017 554         DB      DDCAP      0: Read, Write, Directory, Random
000.013 017 555         DB      DDCAP      1: Read, Write, Directory, Random
000.014 017 556         DB      DDCAP      2: Read, Write, Directory, Random
000.015 017 557         DB      DDCAP      3: Read, Write, Directory, Random
000.016 000 558         DB      0          4: Ignored
000.017 000 559         DB      0          5: Ignored
000.020 000 560         DB      0          6: Ignored
000.021 000 561         DB      0          7: Ignored
000.022 307 562         DB      DDDFLV      Process Set Commands
563
000.023    564         SET      023Q
000.000    565         ERRNZ   *-
000.023    566         DS      INH,STE-
```

```

569 *** Set Entry Point
570 *
571 * SET commands enter here.
572 *
573 * ENTRY: A = unit number
574 * DE = line pointer
575 *
576 * EXIT: PSW = 'C' clear if no error
577 * 'C' set if error
578 *
579 * USES: ALL
580 *
581 *
000.053 582 SETNTR EQU *
000.000 583 ERRNZ *-DVD,STE
000.053 247 584 ANA A
000.054 302 103 000 585 JNZ SET1 only allow unit 0
586
000.057 102 587 MOV B,D
000.060 113 588 MOV C,E
000.061 021 204 000 589 LXI D,PRCTAB DE = processor table
000.064 041 166 000 590 LXI H,OPTTAB HL = option table
000.067 315 226 042 591 CALL $SOP
000.072 330 592 RC error
593
000.073 315 201 042 594 CALL $SNA
000.076 310 595 RZ at the end of the line
596
000.077 076 040 597 MVI A,EC.ILO must be other trailing Junk
000.101 067 598 STC
000.102 311 599 RET
600
000.103 076 033 601 SET1 MVI A,EC.UUN
000.105 067 602 STC
000.106 311 603 RET

```

Processors

13:35:48 24-OCT-80

606 ** Processors
607 *

609 ** FLAG - Process Flag Options

610 *

611

042.231 612 FLAG EQU \$PBF

614 ** VAL - Process Value Options

615 *

616

042.234 617 VAL EQU \$PBV

619 ** HELP - Process Help Option

620 *

621

000.107 315.136.031 622 HELP CALL \$TYPX

000.112 012 012 123 623 DB NL,NL,'Set Options:',NL,NL

000.132 110.145.154 624 DB 'Help',TAB,TAB,'Type this message',NL

000.001 625 IF 1

626 DB 'Units',TAB,'n',NL

627 ENDIF

000.162 012.212 628 DB NL,ENL

000.164 257 629 XRA A

000.165 311 630 RET

633 *** Tables
634 *

636 ** Option Table
637 *

000.166	203 000	638			
000.170	006	639	OPTTAB	DW	OPTABE End of the option table
		640		DB	6
000.171	110 105 114	641			
000.176	000 000 000	642		DB	'HEL';'P'+2000;HELPI
		643		DB	0,0,0,0,0
		644			
000.001		645		IF	1
		646		DB	'UNIT';'S'+2000;VALI;10;1;4
		647		DW	DDMNU
		648		ENDIF	
		649			
000.203	000	650	OPTABE	DB	0

652 ** Processor Table
653 *

000.204		654			
		655	PRCTAB	DS	0
000.000		656			
000.204	231 042	657	FLAGI	EQU	*-PRCTAB/2
		658		DW	FLAG
		659			
000.001		660	HELPI	EQU	*-PRCTAB/2
000.206	107 000	661		DW	HELP
		662			
000.002		663	VALI	EQU	*-PRCTAB/2
000.210	234 042	664		DW	VAL
		665			
000.212		666	.	SET	0235A-23Q
000.212		667		DS	DDV.ENT-

```
670 **      Driver Dispatch
671 *
672 *
673 *      ENTRY: A      = Driver function code
674 *      DE      = I/O Channel Table Address
675 *      HL      = Device Channel Table Entry Address
676 *
677 *      EXIT:   ???
678 *
679 *      USES:   ALL
680 *
681
002.000 682 DDDVD EQU *
000.000 683 ERRNZ *-DVD.ENT
684
002.000 376 013 685 CPI DC,MAX
002.002 322 023 002 686 JNC DDILRQ      Illegal Request
687
002.005 315 076 031 688 CALL $TBRA      Valid Device Driver request
002.010 689 SET 2010A
000.000 690 ERRNZ *-
691
002.010 055 692 DB DDREAD-*      READ
002.011 057 693 DB DDWRIT-*      WRITE
002.012 053 694 DB DDREAR-*      READ REGARDLESS is illegal
002.013 014 695 DB DDOPNR-*      OPEN FOR READ
002.014 013 696 DB DDOPNW-*      OPEN FOR WRITE
002.015 012 697 DB DDOPNU-*      OPEN FOR UPDATE
002.016 011 698 DB DDCLDS-*      CLOSE
002.017 152 699 DB DDABRT-*      ABORT
700
000.000 701 ERRNZ *-,-DC,MOU
002.020 154 702 DB DDMOUN-*      MOUNT
703
000.000 704 ERRNZ *-,-DC,LOD
002.021 200 705 DB DDLOAD-*      LOAD
706
000.000 707 ERRNZ *-,-DC,RDY
002.022 007 708 DB DDREDY-*      Ready
```

712 ** DDILRQ - Illegal Request
713 *
714 * DDILRQ processes illegal requests by flagging them illegal,
715 * and returning with the error flag set.
716 *
717 *
002.023 076 012 718 DDILRQ MOV A,EC.ILR
002.025 067 719 STC
002.026 311 720 RET

722 ** DDNOP - IGNORE OR 'NOP' PROCESSOR
723 *
724 * DDNOP ignores the specified operation.
725 *
726 *
002.027 727 DDNOP EQU *
728 *
002.027 247 729 ANA A CLEAR CARRY
730 *
002.030 311 731 RET

733 ** Illegal Device Driver Requests
734 *
735 * Treated as NOP for the sake of PIP.
736 *
737 *
002.027 738 DDOPNR EQU DDNOP OPEN FOR READ
739 *
002.027 740 DDOPNW EQU DDNOP OPEN FOR WRITE
741 *
002.027 742 DDOPNU EQU DDNOP OPEN FOR UPDATE
743 *
002.027 744 DDCLDS EQU DDNOP CLOSE

```

748 **      DDREDY - DD: Ready
749 *
750 *      DDREDY waits for the specified unit of a given device
751 *      to go ready
752 *
753 *      ENTRY:  NONE
754 *
755 *      EXIT:   PSW      = 'C' SET   if NOT READY
756 *              'C' CLEAR if   Ready
757 *
758 *      USES:   PSW,BC
759 *
760
002.031 315.250.002 761 DDREDY CALL RREADY      Read Ready
002.034 332.063.002 762 JC DDREDY1    Error trying to check READY
002.037 315.250.002 763 CALL RREADY
002.042 332.063.002 764 JC DDREDY1    Error trying to check READY
765
002.045 117 766 MOV C,A      Save ready bits
002.046 072.041.041 767 LDA AIO,UNI
002.051 107 768 MOV B,A
002.052 257 769 XRA A
002.053 315.220.005 770 CALL BITS
002.056 241 771 ANA C
002.057 067 772 STC
002.060 300 773 RNZ      Specified unit is not ready
774
002.061 247 775 ANA A      Clear Carry
002.062 311 776 RET
777
778 *      Error trying to read READY
779
002.063 067 780 DDREDY1 STC      Error implies device is NOT Ready
002.064 311 781 RET

```

```
785 ** DDREAD - Process Read
786 *
787 * DDREAD reads the specified data from the device.
788 *
789 *
002.065 790 DDREAD EQU *
791
002.065 303 320 002 792 JMP BLKRD CALL BLOCK READ
```

```
795 ** DDREAR - Read Resardless
796 *
797 * DDREAR reads the specified data regardless of ID. Unlike
798 * the read-resardless of the H17, the volume must be pre-
799 * iously mounted as volume 0. This is done to initialize
800 * the volume parameters. Any HDOS code should take this into
801 * account.
802 *
803
002.065 804 DDREAR EQU DDREAD
```

```

808 ** DDWRIT - Process Write
809 *
810 * DDWRIT writes the specified data to the device.
811 *
812 *
002.070 170 813 DDWRIT MOV A,B
002.071 261 814 ORA C
002.072 312 111 002 815 JZ DDWRI1 Special Case 0 byte write for write protect
816 *
817 * Round the write count up
818 *
002.075 345 819 PUSH H
002.076 041 377 000 820 LXI H,255
002.101 011 821 DAD B
002.102 104 822 MOV B,H
002.103 016 000 823 MVI C,0 BC = rounded up count
002.105 341 824 POP H
825 *
002.106 303 274 002 826 JMP BLKWT
827 *
828 * Check the write protect flag
829 *
002.111 315 233 002 830 DDWRI1 CALL FAS
002.114 062 237 005 831 STA AUXSTAT Initialize Aux Status
002.117 076 001 832 MVI A,1
002.121 062 245 005 833 STA STC Set Transfer Count
002.124 257 834 XRA A
002.125 062 246 005 835 STA TRACK Initialize Track
002.130 076 013 836 MVI A,DDF.LAB+1+1
002.132 062 241 005 837 STA SECTOR Initialize to sector after label
838 *
002.135 315 127 003 839 CALL LSC Set Count
002.140 330 840 RC
841 *
842 * Try to Copy sector to itself
843 *
002.141 315 071 003 844 CALL COM
002.144 013 845 DB DD,CPY
002.145 330 846 RC
002.146 315 037 005 847 CALL TRK
002.151 330 848 RC
002.152 315 254 004 849 CALL SUS
002.155 330 850 RC
002.156 315 037 005 851 CALL TRK
002.161 330 852 RC
002.162 315 254 004 853 CALL SUS
002.165 330 854 RC
002.166 303 131 005 855 JMP WDN Wait for DONE, and check for errors

```

Device Processors: Read Regardless / Abort

DDABRT

13:35:51 24-OCT-80

859 ** DDABRT - Process Abort

860 *

861 * DDABRT Process the abort entry by:

862 *

863 * resetting the interface

864 *

865

002.171 303 060 004 866 DDABRT JMP RST Reset the device

Device Processors: Mount / Load

13:35:51 24-OCT-80

```

870 **      DDOUN - Process Mount
871 *
872 *      DDOUN mounts the volume in the specified device.
873 *
874 *      sector size
875 *      number of sides
876 *      write protect
877 *
878 *
879
002.174 315 060 004 880 DDOUN CALL RST      Reset the device
002.177 330      881 RC
002.200 315 021 004 882 CALL RAS      A = Aux. Status byte
002.203 330      883 RC
884
002.204 107      885 MOV B,A
002.205 072 061 041 886 LDA AIO.UNI
002.210 041 247 005 887 LXI H,STATBL
002.213 315 072 030 888 CALL $DADA      HL = Address for status byte
002.216 160      889 MOV M,R      Save byte in table
890
002.217 247      891 ANA A      Clear Carry
002.220 311      892 RET

```

```

894 **      DDLOAD - Process Load
895 *
896 *      DDLOAD is a dummy entry to process any device driver
897 *      load code, i.e. initialization, buffer allocation, etc.
898 *
899
002.221 315 060 004 900 DDLOAD CALL RST
002.224 247      901 ANA A      Ignore any errors
902
002.225 076 170      903 MVI A,D.STA      /2.0b/
002.227 042 120 041 904 STA S.BDA      Stuff this for any boots /2.0b/
002.232 311      905 RET

```


Subroutines

FAS

13:35:52 24-OCT-80

```

909 **      FAS      - Fetch Auxiliary Status
910 *
911 *      FAS fetches the auxiliary status byte for the
912 *      currently selected unit. It is up to the mount
913 *      routine to maintain the table of status bytes.
914 *
915 *      ENTRY:  AIO.UNI = Device Unit
916 *
917 *      EXIT:   A      = Aux. Status byte for this unit
918 *
919 *      USES:   PSW
920 *
921 *
002,233 345      922 FAS      PUSH    H
002,234 072 061 041 923      LDA     AIO.UNI
002,237 041 247 005 924      LXI     H,STATBL
002,242 315 072 030 925      CALL   $DADA      HL = Address of status byte
002,245 176      926      MOV     A,M      A = Status Byte
002,246 341      927      POP     H
002,247 311      928      RET

```

```

930 **      RREADY  - Read Ready
931 *
932 *      RREADY checks the ready status of the sub-system
933 *
934 *      ENTRY:  NONE
935 *
936 *      EXIT:   A      = ready byte
937 *
938 *      USES:   PSW
939 *
940 *
002,250 315 071 003 941 RREADY CALL    COM
002,253 020      942      DB     DD:RRDY
002,254 330      943      RC
944
002,255 315 275 003 945      CALL   PIN
002,260 365      946      PUSH   PSW
002,261 315 131 005 947      CALL   WDN
002,264 332 271 002 948      JC     RREADY1      Error waiting for DONE
002,267 361      949      POP     PSW
002,270 311      950      RET
951
952 *      Ignore data back from ready
953
002,271 063      954 RREADY1 INX     SP
002,272 063      955      INX     SP      Discard PSW
002,273 311      956      RET

```

000.000	959	.BLKW	EQU	0	Leave in WRITE
000.001	960	.SMALL	EQU	1	Use full size routines
002.274	961		XTEXT	H47LIB	

964X *** Assembly Constants

965X *

966X *

967X * .BLKW Used to conditional WRITE operations in and out.

968X *

969X * .SMALL Used to conditional BLK operations out

970X * memory, or minimal run-time.

971X *

000.001

973X

IF .SMALL

974X

ELSE

976X **

BLK - Block

977X *

978X *

979X * BLK repeatedly READ/WRITEs the data until all of the data
980X * is transfered. Data is always transfered so that it will
981X * not wrap over a single side track boundary, so as to avoid
982X * the multiple sector algorithm problem.

983X *

984X *

ENTRY: BC = total count (should be a multiple of 256)

985X *

DE = buffer address

986X *

HL = block number

987X *

988X *

EXIT: FSW = 'C' set if error

989X *

'C' clear if no error

990X *

991X *

USES: ALL

992X *

993X

002.274 076 010

994X

BLKW

MVI A,DD.WRIB

block write

002.276 062 070 003

995X

STA BLKB

002.301 315 116 004

996X

CALL SDE Default error is WRITE

002.304 023

997X

DB EC.WF

002.305 345

998X

PUSH H

002.306 041 202 003

999X

LXI H,OUTB

002.311 042 066 003

1000X

SHLD BLKA set block operation as output

002.314 341

1001X

POP H

002.315 303 341 002

1002X

JMP BLK1

1003X

002.320 076 007

1004X

BLKR

MVI A,DD.REAB

block read

002.322 062 070 003

1005X

STA BLKB

002.325 315 116 004

1006X

CALL SDE Default error is READ

002.330 022

1007X

DB EC.RF

002.331 345

1008X

PUSH H

002.332 041 304 003

1009X

LXI H,PINB

002.335 042 066 003

1010X

SHLD BLKA set block operation as input

002.340 341

1011X

POP H

000.000

1012X

ERRNZ *-BLK1

1013X

H47 Library

BLK

13:35:54 24-OCT-80

```

002.341 315 126 004 1014X BLK1 CALL SDP Set-Up device parameters
002.344 072 036 005 1015X LDA DEFERR
002.347 330 1016X RC Illegal parameters
1017X
002.350 170 1018X BLK2 MOV A,B
002.351 261 1019X ORA C
002.352 310 1020X RZ all finished with the block operation
1021X
002.353 305 1022X PUSH B
002.354 315 001 003 1023X CALL BLK3
002.357 301 1024X POP B
002.360 330 1025X RC Error
1026X
002.361 171 1027X MOV A,C
002.362 225 1028X SUB L
002.363 117 1029X MOV C,A
002.364 170 1030X MOV A,B decrement the bytes read count
002.365 234 1031X SBB H
002.366 107 1032X MOV B,A
1033X
002.367 315 045 005 1034X CALL USN Update sector number
1035X
002.372 315 131 005 1036X CALL WDN
002.375 322 350 002 1037X JNC BLK2 No errors
1038X
003.000 311 1039X RET ERROR waiting for DONE

```

```

1041X ** BLK3
1042X *
1043X * EXIT: HL = bytes actually read
1044X *
1045X
003.001 1046X BLK3 EQU *
1047X
1048X * Compute Transfer Size
1049X
003.001 140 1050X MOV H,B HL = BC ; Initialize byte count
003.002 151 1051X MOV L,C
1052X
003.003 171 1053X MOV A,C
003.004 247 1054X ANA A
003.005 312 011 003 1055X JZ BLK4
003.010 004 1056X INR B round sector count up for partial sector
003.011 1057X BLK4 EQU *
1058X
003.011 170 1059X MOV A,B
003.012 062 245 005 1060X STA STC Initialize sector count
1061X
003.015 345 1062X PUSH H
003.016 072 244 005 1063X LDA SPT A = Sectors per Track
003.021 052 241 005 1064X LHLD SECTOR
003.024 225 1065X SUB L
003.025 074 1066X INR A A = maximum number of sectors left this track

```

```

003.026 270      1067X      CMP      B
003.027 341      1068X      POP      H
003.030 322 041 003 1069X      JNC      BLK5      Can read all that we need
                                1070X
003.033 062 245 005 1071X      STA      STC      Update sector count for end of track
003.036 147      1072X      MOV      H,A
003.037 056 000      1073X      MVI      L,0      HL = count for the rest of this track
003.041      1074X BLK5      EQU      *
                                1075X
003.041 315 127 003 1076X      CALL     LSC      Set the transfer count
003.044 330      1077X      RC              ERROR
                                1078X
003.045 072 070 003 1079X      LDA      BLKB
003.050 315 075 003 1080X      CALL     COM,      command
003.053 330      1081X      RC              ERROR issuing command
                                1082X
003.054 315 037 005 1083X      CALL     TRK      track
003.057 315 254 004 1084X      CALL     SUS      side/unit/sector
003.062 330      1085X      RC              ERROR bit is set
                                1086X
003.063 104      1087X      MOV      B,H
003.064 115      1088X      MOV      C,L      RC = actual byte transfer count
003.065 303 377 377 1089X      JMP      -1      Enter data transfer processor
003.066      1090X BLKA      EQU      *-2
                                1091X
003.070 000      1092X BLKB      DB      0      Transfer Command (Read/Write)
001.000      1093X BLKC      EQU      256

```

```

                                1095X **      COM      = Command
                                1096X *
                                1097X *      COM outputs a command byte
                                1098X *
                                1099X *
                                1100X *      ENTRY:  A      = command
                                1101X *
                                1102X *      EXIT:   PSW      = 'C' SET IF ERROR
                                1103X *              = 'C' CLEAR IF NO ERROR
                                1104X *
                                1105X *      USES:   PSW
                                1106X *
                                1107X *
003.071 343      1108X COM      XTHL
003.072 176      1109X      MOV      A,M      Fetch the command byte
003.073 043      1110X      INX      H
003.074 343      1111X      XTHL
                                1112X
003.075 365      1113X COM,     PUSH     PSW
003.076 315 131 005 1114X      CALL     WDN
003.101 332 114 003 1115X      JC      COM1      ERROR
003.104 361      1116X      POP      PSW
003.105 315 174 003 1117X COM,     CALL     OUT
003.110 315 117 003 1118X      CALL     DLY
003.113 311      1119X      RET

```

```
003.114 063 1120X
003.115 063 1121X COM1 INX SP
003.116 311 1122X INX SP
1123X RET Return with error from WDN

1125X ** DLY - Delay
1126X *
1127X * DLY delays for a short time.
1128X *
1129X * ENTRY: NONE
1130X *
1131X * EXIT: A = 0
1132X *
1133X * USES: PSW
1134X *
1135X
003.117 076 040 1136X DLY MVI A,0400
003.121 247 1137X ANA A F = 'NC'
003.122 075 1138X DLY1 DCR A
003.123 302 122 003 1139X JNZ DLY1 Wait longer
003.126 311 1140X RET

1142X ** LSC - Load Sector Count
1143X *
1144X * LSC loads the sector count for subsequent operations.
1145X *
1146X * ENTRY: AUXSTAT initialized
1147X *
1148X * EXIT: PSW = 'C' CLEAR if NO ERROR
1149X * 'C' SET if ERROR
1150X *
1151X * USES: PSW
1152X *
1153X
003.127 315 071 003 1154X LSC CALL COM
003.132 003 1155X DB DD,LSC
003.133 330 1156X RC Tough luck if COM returns errors
1157X
003.134 257 1158X XRA A High order byte
003.135 315 164 003 1159X CALL OUT
003.140 330 1160X RC
1161X
003.141 072 237 005 1162X LDA AUXSTAT
003.144 346 100 1163X ANI AS,0DD Check for Double Density
003.146 072 245 005 1164X LDA STC
003.151 302 155 003 1165X JNZ LSC1 Double Density
003.154 207 1166X ADD A Double Count for single density 128 byte sect.
003.155 315 164 003 1167X LSC1 CALL OUT Low order byte
003.160 330 1168X RC
1169X
```

```

003.161 303 131 005 1170X      JMP      WDN
................................................................
1172X **      OUT      - Output
1173X *
1174X *      OUT outputs a byte to the port with a *S.DTR* handshake.
1175X *
1176X *
1177X *      ENTRY:  A      = byte
1178X *
1179X *      EXIT:   PSW      = 'C' if ERROR
1180X *                  'NC' if NO Error, no byte output
1181X *
1182X *      USES:   PSW
1183X *
1184X *
003.164 365      1185X OUT      PUSH      PSW
003.165 315 162 005 1186X      CALL      WTR      Wait for DTR
003.170 332 177 003 1187X      JC        OUT0
003.173 361      1188X      POP        PSW
1189X *
003.174 323 171      1190X OUT.    OUT      D.DAT      OUTPUT TO THE DATA PORT
003.176 311      1191X      RET
1192X *
003.177 063      1193X OUT0     INX      SP      Return with error from WTR
003.200 063      1194X      INX      SP
003.201 311      1195X      RET
1196X      ENDIF

```

```

000.000      1198X      IF      .BLKW

```

```

1200X **      OUTB      - Output Block
1201X *
1202X *      OUTB outputs a block. This is one of the more critical
1203X *      routines as far as time goes in transferring data. This
1204X *      routine should be highly tuned.
1205X *
1206X *
1207X *      ENTRY:  BC      = count
1208X *              DE      = buffer address
1209X *
1210X *      EXIT:   PSW      = 'C' clear if NO error
1211X *                  = 'C' set if error
1212X *                  BC = count remaining
1213X *
1214X *      USES:   ALL
1215X *
1216X *

```

```

003.202 170 1217X OUTB MOV A,B
003.203 261 1218X ORA C
003.204 312 131 005 1219X JZ WDN Finished; wait for DONE
1220X
003.207 305 1221X PUSH B
003.210 315 230 003 1222X CALL OUT128
003.213 301 1223X POP B
003.214 330 1224X RC Transfer Error
1225X
003.215 345 1226X PUSH H
003.216 041 200 377 1227X LXI H,-128
003.221 011 1228X DAD B
003.222 104 1229X MOV B,H
003.223 115 1230X MOV C,L
003.224 341 1231X POP H
003.225 303 202 003 1232X JMP OUTB

```

```

1234X ** OUT128 - OUT 128
1235X *
1236X * OUT128 outputs 128 bytes from the data input port.
1237X * The first and last bytes are transferred via hand-
1238X * shake, the rest are transferred as good. The
1239X * reason 128 was chosen, as it is the minimum trans-
1240X * fer size if the H47 code is somehow lost.
1241X *
1242X * NOTE: This code assumes that the H47 accepts
1243X * bytes sufficiently fast in the MAIN loop.
1244X *
1245X * ENTRY: DE = buffer
1246X *
1247X * EXIT: PSW = 'C' CLEAR if NO error
1248X * DE = DE advanced
1249X * 'C' SET if error
1250X * 'A' = Error Code
1251X *
1252X * USES: PSW,BC,DE
1253X *
1254X *

```

```

003.230 315 162 005 1255X OUT128 CALL WTR
003.233 330 1256X RC Synchronization error
1257X
1258X * Output 127 bytes
1259X
003.234 016 177 1260X MVI C,127
003.236 333 170 1261X OUT1 IN D,STA
003.240 346 240 1262X ANI S,DTR+S,DON
003.242 372 253 003 1263X JM OUT2 S,DTR is set, is not done
000.000 1264X ERRNZ S,DTR-200Q
1265X
003.245 312 236 003 1266X JZ OUT1 is not done yet
1267X
003.250 303 323 004 1268X JMP TEB. Generate error; and examine status
1269X

```


OUT128

```

003.253 032 1270X OUT2 LDAX D
003.254 323 171 1271X OUT D,DAT output a byte
003.256 023 1272X INX D
003.257 015 1273X DCR C
003.260 302 236 003 1274X JNZ OUT1
1275X
1276X * Handshake last byte
1277X
003.263 315 162 005 1278X CALL WTR
003.266 330 1279X RC Synchronization error
1280X
003.267 032 1281X LDAX D
003.270 323 171 1282X OUT D,DAT
003.272 023 1283X INX D
003.273 247 1284X ANA A Clear 'C'
003.274 311 1285X RET
1286X ENDIF
000.001 1287X IF .SMALL
1288X ELSE

```

```

1290X ** PIN - Input
1291X *
1292X * PIN inputs a byte from the data data port.
1293X *
1294X *
1295X * ENTRY: NONE
1296X *
1297X * EXIT: PSW = 'C' if ERROR
1298X * A = Error Code
1299X * 'NC' if NO Error
1300X * A = Byte
1301X *
1302X * USES: PSW
1303X *
1304X
003.275 315 162 005 1305X PIN CALL WTR Wait for DTR
003.300 330 1306X RC
1307X
003.301 333 171 1308X PIN IN D,DAT
003.303 311 1309X RET

```

```

1311X ** PINB - Input Block
1312X *
1313X * PINB inputs a block. This is one of the more critical
1314X * routines as far as time goes in transferring data. This
1315X * routine should be highly tuned.
1316X *
1317X *
1318X * ENTRY: BC = count
1319X * DE = buffer address

```

```

1320X *
1321X *      EXIT: PSW      = 'C' clear if NO error
1322X *      = 'C' set   if   error
1323X *      A = Error Code
1324X *      BC = count remaining
1325X *
1326X *      USES:  ALL
1327X *
1328X *
003.304 170      1329X PINB  MOV      A,B
003.305 247      1330X      ANA      A
003.306 312 323 003 1331X      JZ      PINB2      Need less than one sector
1332X *
003.311 305      1333X PINB1  PUSH     B
003.312 315 364 003 1334X      CALL    PIN256      read one sector
003.315 301      1335X      POP      B
003.316 330      1336X      RC          ERROR
1337X *
003.317 005      1338X      DCR      B          Count the bytes read
003.320 302 311 003 1339X      JNZ     PINB1
1340X *
003.323 171      1341X PINB2  MOV      A,C
003.324 247      1342X      ANA      A
003.325 312 131 005 1343X      JZ      WDN          Finished, don't need partial sector
1344X *
1345X *      Read any partial sectors
1346X *
003.330 305      1347X      PUSH     B          C = bytes left to read in partial sector
003.331 315 372 003 1348X      CALL    PIN1.      Read a partial sector
003.334 301      1349X      POP      B
003.335 330      1350X      RC          ERROR
1351X *
003.336 333 170      1352X PINB3  IN      D,STA
003.340 346 240      1353X      ANI     S,DTR+S,DON
003.342 372 353 003 1354X      JM      PINB4          H47 has a byte
000.000      1355X      ERRNZ     S,DTR-2000
1356X *
003.345 312 336 003 1357X      JZ      PINB3          DONE is not set
1358X *
003.350 303 323 004 1359X      JMP     TEB.          Generate error, and look at status
1360X *
003.353 333 171      1361X PINB4  IN      D,DAT          Eat the byte
003.355 014      1362X      INR      C
003.356 302 336 003 1363X      JNZ     PINB3
1364X *
003.361 303 131 005 1365X      JMP     WDN          DONE accepting bytes

```

```

1367X **      PIN256 - PIN 256
1368X *
1369X *      PIN256 inputs 256 bytes from the data input port.
1370X *      S.ITR must be set before any bytes may be transferred.
1371X *      This is one of the more critical routines, and should
1372X *      be highly tuned.
1373X *
1374X *      ENTRY: DE      = buffer
1375X *
1376X *      EXIT:  PSW      = 'C' CLEAR if NO error
1377X *              DE = DE advanced
1378X *              'C' SET if error
1379X *
1380X *      USES:  PSW,BC,DE
1381X *
1382X *
003.364 315 162 005 1383X PIN256 CALL WTR
003.367 330 1384X RC Synchronization ERROR
1385X *
1386X *      Accept 256 bytes
1387X *
003.370 016 000 1388X MVI C,0 Set count to 256
003.372 333 170 1389X PIN1 IN D,STA
003.374 346 240 1390X ANI S.ITR+S.DON
003.376 372 007 004 1391X JM PIN2 H47 has a byte
000.000 1392X ERNZ S.ITR-2000
1393X *
004.001 312 372 003 1394X JZ PIN1 done is not set
1395X *
004.004 303 323 004 1396X JMP TER, Generate Error on pre-mature done
1397X *
004.007 333 171 1398X PIN2 IN D,DAT
004.011 022 1399X STAX D
004.012 023 1400X INX D
004.013 015 1401X DCR C
004.014 302 372 003 1402X JNZ PIN1
1403X *
004.017 247 1404X ANA A Clear 'C'
004.020 311 1405X RET

1407X **      RAS - Read Auxiliary Status
1408X *
1409X *      RAS reads the auxiliary status for the unit specified
1410X *      in AIO.UNI.
1411X *
1412X *      ENTRY: AIO.UNI = Device Unit
1413X *
1414X *      EXIT:  PSW      = 'C' if ERROR
1415X *              A = error code
1416X *              'NC' if NO error
1417X *
1418X *      USES:  PSW,HL,BC
1419X *

```

```

004.021 041 000 000 1420X
004.024 042 241 005 1421X RAS LXI H,0
000.000 1422X SHLD SECTOR Zero initial parameters
000.000 1423X ERRNZ SIDE-SECTOR-1
000.000 1424X ERRNZ SID,0
004.027 315 071 003 1425X
004.032 002 1426X CALL COM Output original command
004.033 330 1427X DB DD,RAS
004.034 315 246 004 1428X RC
004.037 330 1429X CALL SUS.. Unit number
004.040 315 275 003 1430X RC
004.043 330 1431X CALL PIN A = Aux. Status byte
004.043 330 1432X RC
004.044 365 1433X
004.045 315 131 005 1434X PUSH PSW
004.050 332 055 004 1435X CALL WDN Wait for DONE
004.053 361 1436X JC RAS1
004.054 311 1437X POP PSW
004.055 063 1438X RET NO Error, so return with 'NC' and A
004.056 063 1439X
004.057 311 1440X
004.055 063 1441X RAS1 INX SP
004.056 063 1442X INX SP Discard saved A
004.057 311 1443X RET Exit with WDN return values

```

```

1445X ** RST - Reset
1446X *
1447X * RST reset the device.
1448X *
1449X *
1450X * ENTRY: NONE
1451X *
1452X * EXIT: NONE
1453X *
1454X * USES: PSW
1455X *
1456X
004.060 305 1457X RST PUSH B
004.061 315 066 004 1458X CALL RST,
004.064 301 1459X POP B
004.065 311 1460X RET
004.066 076 002 1461X
004.070 323 170 1462X RST, MVI A,W,RES
004.072 315 117 003 1463X OUT D,STA
004.072 315 117 003 1464X CALL DLY
004.072 315 117 003 1465X
004.072 315 117 003 1466X * Wait for DONE
004.072 315 117 003 1467X
004.075 001 000 000 1468X LXI B,RSTA
004.100 013 1469X RST1 DCX B
004.101 170 1470X MOV A,B
004.102 261 1471X ORA C
004.103 312 323 004 1472X JZ IEB, Set error flag

```

```

1473X
004.106 333 170 1474X IN D,STA
004.110 346 040 1475X ANI S,DON
004.112 302 100 004 1476X JNZ RST1 Wait some more
1477X
004.115 311 1478X RET
1479X
000.000 1480X RSTA EQU 0 Time-Out Counter

```

```

1482X ** SDE - Set Default Error
1483X *
1484X * SDE sets the default error to the specified one
1485X *
1486X * ENTRY: (SP) = default error
1487X *
1488X * EXIT: (SP) advanced to the RETURN address
1489X *
1490X * USES: PSW
1491X *
1492X
004.116 343 1493X SDE XTHL
004.117 176 1494X MOV A,M
004.120 043 1495X INX H
004.121 062 036 005 1496X STA DEFERR
004.124 343 1497X XTHL
004.125 311 1498X RET
1499X ENDIF

```

```

1501X ** SDP - Set-up Device Parameters
1502X *
1503X * SDP sets up the device TRACK, SIDE, and SECTOR from the
1504X * sector number.
1505X *
1506X * IF .SMALL, this code assumes that AUXSTAT is initialized.
1507X *
1508X * ENTRY: HL = sector number
1509X *
1510X * EXIT: TRACK, SIDE, and SECTOR initialized for the
1511X * transfer
1512X *
1513X * USES: PSW,HL
1514X *
1515X
004.126 1516X SDP EQU *
004.126 305 1517X PUSH B
004.127 325 1518X PUSH D
1519X
004.130 315 136 004 1520X CALL SDP.
1521X
004.133 321 1522X POP D

```

```

004.134 301      1523X      POP      B
004.135 311      1524X      RET
004.136 104      1525X
004.137 115      1526X SDP.  MOV      B,H
004.140 257      1527X      MOV      C,L      BC = sector number
000.000          1528X      XRA      A
000.000          1529X      ERRNZ   SID.0
004.141 062 242 005 1530X      STA      SIDE      Initialize Side Byte
000.001          1531X
000.001          1532X      IF      .SMALL
000.001          1533X      LDA      AUXSTAT      A = Alternate Status
000.001          1534X      ELSE
004.144 315 233 002 1535X      CALL   FAS      A = Alternate Status
004.147 062 237 005 1536X      STA      AUXSTAT
004.147          1537X      ENDIF
004.152 346 100      1538X
004.154 076 015      1539X      ANI      AS.000      Track 0 is the real clue
004.156 312 162 004 1540X      MVI      A,NSPTS
004.161 207          1541X      JZ      SDP1      Is Single Density
000.000          1542X      ADD      A      A = 2 * A
000.000          1543X      ERRNZ   NSPTS*2-NSPTD
004.162 062 244 005 1544X SDP1 STA      SPT      Save Sectors per Track
004.165 157          1545X
004.166 046 000      1546X      MOV      L,A
004.170 072 237 005 1547X      MVI      H,0      HL = Sectors per Track
004.173 346 020      1548X      LDA      AUXSTAT
004.175 312 201 004 1549X      ANI      AS.S1A
004.200 051          1550X      JZ      SDP2      Only 1 Side
004.201          1551X      DAD      H      HL = 2 * HL
004.201          1552X SDP2 EQU      *
000.001          1553X      IF      .SMALL
000.001          1554X      ELSE
004.201 175          1555X      MOV      A,L
004.202 062 243 005 1556X      STA      SPC      Save sectors per cylinder
004.202          1557X      ENDIF
004.205 353          1558X
004.206 315 106 030 1559X      XCHG
004.206          1560X      CALL   $DU66      DE = Sectors per Cylinder
004.211 175          1561X      HL = BC/DE = Track Number
004.212 062 246 005 1562X      MOV      A,L
004.215 173          1563X      STA      TRACK
000.001          1564X      MOV      A,E      Assume Track is Good (Let H47 fix errors)
000.001          1565X      IF      .SMALL
000.001          1566X      ELSE
004.216 062 240 005 1567X      STA      CSN      Save cylinder sector number
004.216          1568X      ENDIF
004.221 074          1569X      INR      A      Range for sector is [1-NSPTxJ
004.222 062 241 005 1570X      STA      SECTOR
004.225 041 244 005 1571X
004.230 276          1572X      LXI      H,SPT
004.231 310          1573X      CMP      M
004.232 077          1574X      RZ      Is on Side 0
004.233 320          1575X      CMC
004.233          1576X      RNC      Is on Side 0
004.233          1577X
004.234 226          1578X      SUB      M

```

```

004.235 062 241 005 1579X STA SECTOR Compute Real sector number
004.240 076 200 1580X MVI A,SID,1
004.242 062 242 005 1581X STA SIDE Use side 1
004.245 311 1582X RET
000.001 1583X IF .SMALL
1584X ELSE

```

```

1586X ** SUS - Side Unit Sector

```

```

1587X *
1588X * SUS outputs the Side/Unit/Sector byte. It assumes that
1589X * SIDE, AIO,UNI, and SECTOR are already initialized.
1590X *
1591X *

```

```

1592X * NOTE: This code no longer masks the fields
1593X * to insure against overflow. Be careful!!!
1594X *
1595X *

```

```

1596X * ENTRY: SIDE = side
1597X * AIO,UNI = unit number
1598X * SECTOR = sector number
1599X *

```

```

1600X * EXIT: NONE

```

```

1601X *
1602X * USES PSW
1603X *
1604X

```

```

004.246 072 241 005 1605X SUS, LDA SECTOR
004.251 303 300 004 1606X JMP SUS1 Do not map sector number
1607X

```

```

004.254 305 1608X SUS PUSH B
004.255 315 242 004 1609X CALL SUS,
004.260 301 1610X POP B
004.261 311 1611X RET
1612X

```

```

004.262 072 237 005 1613X SUS, LDA AUXSTAT
004.265 346 100 1614X ANI AS.ODD
004.267 072 241 005 1615X LDA SECTOR A = Sector
004.272 302 300 004 1616X JNZ SUS1 Double Density
004.275 075 1617X JCR A
004.276 207 1618X ADD A Map Sector Number
004.277 074 1619X INR A
004.300 107 1620X SUS1 MOV B,A
1621X

```

```

004.301 072 242 005 1622X LDA SIDE
004.304 260 1623X ORA B
004.305 107 1624X MOV B,A Accumulate Side
1625X

```

```

004.306 072 061 041 1626X LDA AIO,UNI
004.311 017 1627X RRC
004.312 017 1628X RRC
004.313 017 1629X RRC
000.000 1630X ERRNZ UNT.M-96
004.314 260 1631X ORA B

```

```
1632X
004.315 303 164 003 1633X      JMP      OUT      OUTPUT THE BYTE

1635X **      YEB      - Test Error Bit
1636X *
1637X *      YEB test for the error bit to be set. This routine assumes
1638X *      that the error bit will already be valid; that is, that the
1639X *      caller has already verified *S.DON*.
1640X *
1641X *      If the error bit is set, a table look-up is performed to find
1642X *      the HDOS error.
1643X *
1644X *
1645X *      ENTRY: NONE
1646X *
1647X *      EXIT: PSW      = 'C' clear if no error
1648X *                  'C' set if error
1649X *
1650X *      USES: PSW
1651X *
1652X
004.320 315 060 004 1653X TEB.. CALL RST      The system needs cleaning up
1654X
004.323 072 036 005 1655X TEB. LDA DEFERR
004.326 315 333 004 1656X CALL TEB      Check for error other than default
004.331 067      1657X STC      Force at least some error flag
004.332 311      1658X RET
1659X
004.333 365      1660X TEB PUSH PSW      Look for error in the status port
004.334 333 170 1661X IN D.STA
004.336 346 040 1662X ANI S.DON
004.340 312 033 005 1663X JZ TEB4      DONE is NOT set
1664X
004.343 333 170 1665X IN D.STA
004.345 346 001 1666X ANI S.ERR
004.347 312 033 005 1667X JZ TEB4      ERROR is NOT set
1668X
1669X *      ERROR is set
1670X
004.352 361      1671X POP PSW      Discard saved PSW
004.353 076 001 1672X MVI A,DD.RST A = Command
004.355 315 105 003 1673X CALL COM. Read status (Know Done is already set)
004.360 332 027 005 1674X JC TEB3      Things are rapidly disintegrating
1675X
1676X *      Input the error byte
1677X
004.363 305      1678X PUSH B
004.364 001 000 000 1679X LXI B,WTRA      Initialize Time-Out Counter
1680X
004.367 013      1681X TEB1 DCX B
004.370 170      1682X MOV A,B
004.371 261      1683X ORA C
004.372 312 026 005 1684X JZ TEB2      Time-Out
```



```

1685X
004.375 333 170 1686X IN D.STA
004.377 346 040 1687X ANI S.DON
005.001 302 026 005 1688X JNZ TEB2 Pre-Mature Done
1689X
005.004 333 170 1690X IN D.STA
005.006 346 200 1691X ANI S.DTR
005.010 312 367 004 1692X JZ TEB1 No DONE yet
1693X
005.013 301 1694X POP B
005.014 315 301 003 1695X CALL PIN. Get the error byte
1696X
1697X * Determine HDOS error
1698X
005.017 346 100 1699X ANI SB.WPD
005.021 076 025 1700X MVI A,EC.WP
005.023 067 1701X STC
005.024 300 1702X RNZ Drive was write-protected
1703X
005.025 305 1704X PUSH B
005.026 301 1705X TEB2 POP B
1706X
1707X * Take the default error
1708X
005.027 072 036 005 1709X TEB3 LDA DEFERR A = default error
005.032 311 1710X RET
1711X
1712X * NO Error
1713X
005.033 361 1714X TEB4 POP PSW Restore A
005.034 247 1715X ANA A Clear Error Flag
005.035 311 1716X RET
1717X
005.036 000 1718X DEFERR DB 0 Default Error for anything but write-protect

```

```

1720X ** TRK - Track
1721X *
1722X * TRK output the track.
1723X *
1724X *
1725X * ENTRY: TRACK = track sought
1726X *
1727X * EXIT: NONE
1728X *
1729X * USES: PSW
1730X *
1731X
005.037 072 246 005 1732X TRK LDA TRACK
005.042 303 164 003 1733X JMP OUT

```

```

1735X **      USN      - Update Sector Number
1736X *
1737X *      USN updates the sector number to the next group. SDP
1738X *      must have been previously called to initialize all of
1739X *      the device parameters.
1740X *
1741X *      ENTRY:  SPC, CSN, and STC initialized.
1742X *
1743X *      EXIT:   Parameters updated
1744X *
1745X *      USES:   PSW,HL
1746X *
1747X *
005.045      1748X USN      EQU      *
1749X
1750X *      Compute new sector number
1751X
005.045 072 240 005 1752X      LDA      CSN
005.050 052 245 005 1753X      LHLD     STC
005.053 205      1754X      ADD      L
005.054 062 240 005 1755X      STA      CSN      Update cylinder sector number
1756X
1757X *      Check for cylinder wrap
1758X
005.057 052 243 005 1759X      LHLD     SPC
005.062 225      1760X      SUB      L
005.063 332 075 005 1761X      JC       USN1      Is no cylinder wrap
005.066 062 240 005 1762X      STA      CSN      Update cylinder number
005.071 041 246 005 1763X      LXI      H,TRACK
005.074 064      1764X      INR      M      Move to the next track
005.075      1765X USN1      EQU      *
1766X
1767X *      Compute Physical sector number
1768X
005.075 257      1769X      XRA      A
000.000      1770X      ERRNZ     SID.0
005.076 062 242 005 1771X      STA      SIDE      Assume side 0
005.101 072 240 005 1772X      LDA      CSN
005.104 074      1773X      INR      A      Range for sector number is [1-NSPTx]
005.105 062 241 005 1774X      STA      SECTOR      Assume on side 0
005.110 041 244 005 1775X      LXI      H,SPT
005.113 276      1776X      CMP      M
005.114 310      1777X      RZ              Is side 0
005.115 077      1778X      CMC
005.116 320      1779X      RNC              Is side 0
1780X
005.117 226      1781X      SUB      M
005.120 062 241 005 1782X      STA      SECTOR      Is Side 1
005.123 076 200      1783X      MVI      A,SID.1
005.125 062 242 005 1784X      STA      SIDE
005.130 311      1785X      RET
1786X      ENDIF

```

```

1788X **      WDN      - Wait for Done
1789X *
1790X *      WDN waits for the done bit to be asserted.
1791X *
1792X *
1793X *      ENTRY:  NONE
1794X *
1795X *      EXIT:   PSW      = 'C' clear if NO errors
1796X *                  = 'C' set   if error bit set
1797X *                  A  = Error Code
1798X *
1799X *      USES:   PSW
1800X *
1801X
005.131 305    1802X WDN      PUSH      B
005.132 315 137 005 1803X      CALL      WDN.
005.135 301    1804X      POP       B
005.136 311    1805X      RET
1806X
005.137 001 000 000 1807X WDN.    LXI      B,WDNA      Initialize Time-Out counter
1808X
005.142 013    1809X WDN1      DCX      B
005.143 170    1810X      MOV      A,B
005.144 261    1811X      QRA      C
005.145 312 320 004 1812X      JZ      TEB..      Time-out ERROR
1813X
005.150 333 170    1814X      IN      D,STA
005.152 346 040    1815X      ANI      S,DON
005.154 312 142 005 1816X      JZ      WDN1      Wait for Done
1817X
005.157 303 333 004 1818X      JMP      TEB      Test error bits
1819X
000.000          1820X WDNA      EQU      0      Time-Out Counter
000.001          1821X      IF      1
1822X WND        SPACE      4,10
1823X **      WND      - Wait for Not Done
1824X *
1825X *      WND waits for the Done bit to be cleared.
1826X *
1827X *
1828X *      ENTRY:  NONE
1829X *
1830X *      EXIT:   NONE
1831X *
1832X *      USES:   PSW
1833X *
1834X
1835X WND        PUSH      B
1836X      CALL      WND.
1837X      POP       B
1838X      RET
1839X
1840X WND.        LXI      B,WDNA
1841X
1842X WND1      DCX      B
1843X      MOV      A,B

```

```

1844X      ORA      C
1845X      JZ       TEB..      Time-Out Error
1846X
1847X      IN       D.STA
1848X      ANI      S.DON
1849X      JNZ      WND1      DONE IS STILL HIGH
1850X
1851X      RET
1852X
1853X WND1 EQU      0      Wait for Not Done Time-Out count
1854X      ENDDIF
000.001 1855X      IF      .SMALL
1856X      ELSE

1858X **      WTR      - Wait for Transfer Request
1859X *
1860X *      WTR waits for a transfer request. It checks for done
1861X *      first, and if it is found flags an error. The code
1862X *      will also time-out waiting for *S.DTR*.
1863X *
1864X *      ENTRY: NONE
1865X *
1866X *      EXIT: PSW      = 'C' CLEAR if NO error
1867X *                  'C' SET  if error
1868X *
1869X *      USES: PSW
1870X *
1871X
005.162 305 1872X WTR      PUSH      B
005.163 315 170 005 1873X      CALL      WTR.
005.166 301 1874X      POP       B
005.167 311 1875X      RET
1876X
005.170 001 000 000 1877X WTR.      LXI      B,WTRA      Initialize Time-Out Counter
1878X
005.173 333 170 1879X WTR1     IN       D.STA
005.175 346 040 1880X      ANI      S.DON
005.177 302 323 004 1881X      JNZ      TEB..      Done means some type of synchronization error
1882X
1883X *      Wait for Handshake on First Byte, Time-Out if not fast enough
1884X
005.202 013 1885X      DCX      B
005.203 170 1886X      MOV     A,B
005.204 261 1887X      ORA      C
005.205 312 320 004 1888X      JZ       TEB..      Time-Out fatal
1889X
005.210 333 170 1890X      IN       D.STA
005.212 346 200 1891X      ANI      S.DTR
005.214 312 173 005 1892X      JZ       WTR1      Wait for Data Transfer Request
1893X
005.217 311 1894X      RET
1895X
000.000 1896X WTRA EQU      0      Time-Out Loop control

```

DDDD - DD: Device Driver

H47 Library

HEATH HBASH V1.4 01/20/78

PAGE 45

WTR

13:36:08 24-OCT-80

1897X

ENDIF

005.220 1900 XTEXT BITS

1902X ** BITS - BIT SET

1903X *

1904X * BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.

1905X *

1906X * ENTRY: A = ORIGINAL A

1907X * B = NUMBER OF BIT TO SET (7=HIGH,...,0=LOW)

1908X *

1909X * EXIT: A = ORIGINAL A WITH BIT(B) SET

1910X *

1911X * USES: PSW

1912X *

1913X *

005.220 305 1914X BITS PUSH B

1915X *

005.221 365 1916X PUSH PSW

005.222 076 200 1917X MVI A,10000000B

005.224 004 1918X INR B

005.225 007 1919X BITS1 RLC

005.226 005 1920X DCR B

005.227 302 225 005 1921X JNZ BITS1

1922X *

005.232 117 1923X MOV C,A

005.233 361 1924X POP PSW

005.234 261 1925X ORA C

1926X *

005.235 301 1927X POP BC

005.236 311 1928X RET

005.237 1929 XTEXT DADA

1931X ** \$DADA - PERFORM (H,L) = (H,L) + (Q,A)

1932X *

1933X * ENTRY (H,L) = BEFORE VALUE

1934X * (A) = BEFORE VALUE

1935X * EXIT (H,L) = (H,L) + (Q,A)

1936X * 'C' SET IF OVERFLOW

1937X * USES F,H,L

1938X *

1939X *

030.072 1940X \$DADA EQU 30072A IN H17 ROM

005.237 1941 XTEXT DU46

Common Decks.....

*DU66

13:36:10 24-OCT-80

1943X ** \$DU66 - UNSIGNED 16 / 16 DIVIDE.

1944X *

1945X * (HL) = (BC)/(DE)

1946X *

1947X * ENTRY (BC), (DE) PRESET

1948X * EXIT (HL) = RESULT

1949X * (DE) = REMAINDER

1950X *

USES ALL

1951X

1952X

030.106

1953X \$DU66 EQU 30106A IN H17 ROM

005.237

1954

XTEXT TBRA

1956X ** \$TBRA - BRANCH RELATIVE THROUGH TABLE.

1957X *

1958X * \$TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE

1959X * JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE

1960X * ADDRESS OF THE BYTE, YIELDING THE PROCESSOR ADDRESS.

1961X *

1962X *

CALL \$TBRA

1963X *

DB LAB1-*

INDEX = 0 FOR LAB1

1964X *

DB LAB2-*

INDEX = 1 FOR LAB2

1965X *

DB LABN-*

INDEX = N-1 FOR LABN

1966X *

1967X *

ENTRY (A) = INDEX

1968X *

(RET) = TABLE FWA

1969X *

EXIT TO COMPUTED ADDRESS

1970X *

USES F,H,L

1971X

1972X

031.076

1973X \$TBRA EQU 31076A IN H17 ROM

000.000

1974

IF

DEBUG

005.237

1975

XTEXT TYPTX

1977X ** \$TYPTX - TYPE TEXT.

1978X *

1979X *

\$TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.

1980X *

1981X *

IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.

1982X *

A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.

1983X *

1984X *

ENTRY (RET) = TEXT

1985X *

EXIT TO (RET+LENGTH)

1986X *

USES A,F

1987X

1988X

031.136

1989X \$TYPTX EQU 31136A IN H17 ROM

1990X

031.144

1991X \$TYPTX EQU 31144A IN H17 ROM

1992

ENDIF

Data

13:36:13 24-OCT-80

		1995	**	Data	
		1996	*		
		1997			
005.237	000	1998	AUXSTAT	DB	0
005.240	000	1999	CSN	DB	0
005.241	000	2000	SECTOR	DB	0
005.242	000	2001	SIDE	DB	0
005.243	000	2002	SFC	DB	0
005.244	000	2003	SPT	DB	0
005.245	000	2004	STC	DB	0
005.246	000	2005	TRACK	DB	0
		2006			
005.247		2007	STATBL	DS	4
		2008			
005.253	000 000	2009	DW	0	
005.255		2010	DS	32	
		2011			
		2012	LON	6	
		2013			
005.315	055 000 062	2014	END		
	000 065 000				
	166 000 206				
	000 003 002				
	032 002 035				
	002 040 002				
	043 002 054				
	002 066 002				
	073 002 107				
	002 112 002				
	115 002 122				
	002 126 002				
	133 002 136				
	002 142 002				
	147 002 153				
	002 157 002				
	163 002 167				
	002 172 002				
	175 002 201				
	002 211 002				
	222 002 240				
	002 251 002				
	256 002 262				
	002 265 002				
	277 002 302				
	002 307 002				
	312 002 316				
	002 323 002				
	326 002 333				
	002 336 002				
	342 002 345				
	002 355 002				
	370 002 373				
	002 376 002				
	006 003 013				
	003 017 003				
	022 003 031				

Data

13:36:13 24-OCT-80

003 034 003
042 003 046
003 051 003
055 003 060
003 077 003
102 003 106
003 111 003
124 003 130
003 136 003
142 003 147
003 152 003
156 003 162
003 166 003
171 003 205
003 211 003
226 003 231
003 243 003
246 003 251
003 261 003
264 003 276
003 307 003
313 003 321
003 326 003
332 003 343
003 346 003
351 003 357
003 362 003
365 003 377
003 002 004
005 004 015
004 025 004
030 004 035
004 041 004
046 004 051
004 062 004
073 004 104
004 113 004
122 004 131
004 142 004
145 004 150
004 157 004
163 004 171
004 176 004
203 004 213
004 217 004
223 004 226
004 236 004
243 004 247
004 252 004
256 004 263
004 270 004
273 004 302
004 316 004
321 004 324
004 327 004
341 004 350

Data

13:36:14 24-OCT-80

004 356 004
361 004 373
004 002 005
011 005 015
005 030 005
040 005 043
005 046 005
051 005 055
005 060 005
064 005 067
005 072 005
077 005 102
005 106 005
111 005 121
005 126 005
133 005 146
005 155 005
160 005 164
005 200 005
206 005 215
005 230 005
000 000

ASSEMBLY COMPLETE

2014 STATEMENTS

0 ERRORS DETECTED

12344 BYTES FREE

PAGE 51

\$CNA	042207	398L											
\$DADA	030072	888	925	1940E									
\$DCS	042204	396L											
\$DU66	030106	1560	1953E										
\$FST	042212	400L											
\$LBD	042223	406L											
\$PBF	042231	410L	612										
\$PBV	042234	412L	617										
\$SNA	042201	394L	594										
\$SOP	042226	408L	591										
\$TBLS	042215	402L											
\$TBRA	031076	688	1973E										
\$TYPTX	031136	622	1989E										
\$TYPTX.	031144	1991E											
\$WTBLS	042220	404L											
.	002010	564S	565	566	666S	667	889S	690	701	704	707		
.BLKW	000000	959E	1198										
.SMALL	000001	960E	973	1287	1532	1553	1565	1583	1855				
AIO.CGN	041047	364L											
AIO.CHA	041116	379L											
AIO.CNT	041111	375L											
AIO.CSI	041050	365L											
AIO.DDA	041041	360E											
AIO.DES	041055	369L											
AIO.DEV	041057	370L											
AIO.DIR	041062	373L											
AIO.DTA	041053	368L											
AIO.EOF	041113	377L											
AIO.EOM	041112	376L											
AIO.FLG	041043	361L											
AIO.GRT	041044	362L											
AIO.LGN	041051	366L											
AIO.LSI	041052	367L											
AIO.SPG	041046	363L											
AIO.TFP	041114	378L											
AIO.UNI	041061	371L	767	886	923	1626							
AIO.VEC	041040	359L											
AS.ODD	000100	460E	1163	1537	1614								
AS.IDD	000040	461E											
AS.SIA	000020	462E	1549										
AS.SLM	000003	463E											
AUXSTAT	005237	831	1162	1536	1548	1613	1998L						
BELL	000007	38E											
BITS	005220	770	1914L										
RITS1	005225	1919L	1921										
EKSP	000010	40E											
BLK1	002341	1002	1012	1014L									
BLK2	002350	1018L	1037										
BLK3	003001	1023	1046E										
BLK4	003011	1055	1057E										
BLK5	003041	1069	1074E										
BLKA	003066	1000	1010	1090E									
BLKB	003070	995	1005	1079	1092L								
BLKC	001000	1093E											
BLKRD	002320	792	1004L										
BLKWT	002274	826	994L										
BOOT.P	000001	339E											
C.STX	000002	42E											

PAGE 52

[illegible]

DD.SPFS	000025	496L					
DD.ST	000201	503L					
DD.WDLB	000210	510L					
DD.WRBD	000012	479L					
DD.WRD	000011	478L					
DD.WRI	000006	475L					
DD.WRIB	000010	477L	994				
DD.WTBL	000206	508L					
DD.WTDL	000207	509L					
DD.WTL	000204	506L					
DDABRT	002171	699	866L				
DDCAP	000017	548E	551	554	555	556	557
DDCLOS	002027	698	744E				
DDDVD	002000	682E					
DDF.BOL	000011	138E					
DDF.BOD	000000	137L					
DDF.LAB	000011	139L	836				
DDF.USR	000012	140L					
DDILRQ	002023	686	718L				
DDLOAD	002221	705	900L				
DDMNU	000011	553L					
DDMOUN	002174	702	880L				
DDNOP	002027	727E	738	740	742	744	
DDOPNR	002027	695	738E				
DDOPNU	002027	697	742E				
DDOPNW	002027	696	740E				
DDREAD	002065	692	790E	804			
DDREAR	002065	694	804E				
DDREDY	002031	708	761L				
DDREDY1	002063	762	764	780L			
DDWRI1	002111	815	830L				
DDWRIT	002070	693	813L				
DEBUG	000000	1E	1974				
DEFERR	005036	1015	1496	1655	1709	1718L	
DEV.DBA	000004	156L					
DEV.DVG	000015	169L					
DEV.DVL	000013	168L					
DEV.FLG	000006	157L					
DEV.JMP	000003	155L					
DEV.MNU	000010	165L					
DEV.MUM	000007	164L					
DEV.NAM	000000	147L					
DEV.RES	000002	151L					
DEV.UNT	000011	166L					
DEVELEN	000016	171E					
DF.CLR	000376	223E					
DF.EMP	000377	222E					
DIR.ALD	000025	238L					
DIR.CLU	000015	231L					
DIR.CRD	000023	237L					
DIR.EXT	000010	226L					
DIR.FGN	000020	234L					
DIR.FLG	000016	232L					
DIR.LGN	000021	235L					
DIR.LSI	000022	236L					
DIR.NAM	000000	225L					
DIR.PRO	000013	227L					
DIR.VER	000014	228L					

CROSS REFERENCE TABLE

DIRELEN	000027	240E	373	
DIRIDL	000015	229E		
DLY	003117	1118	1136L	1464
DLY1	003122	1138L	1139	
DR.IM	000001	152E		
DR.PR	000002	153E		
DT.CH	000020	162E		
DT.CR	000002	159E	548	
DT.CW	000004	160E	548	
DT.DD	000001	158E	548	
DT.RN	000010	161E	548	
DV.EL	000000	148E		
DV.NU	000001	149E		
DVD.CAP	000007	205L		
DVD.DVD	000006	204L		
DVD.ENT	002000	214E	667	683
DVD.INP	000023	210L		
DVD.MNU	000011	207L		
DVD.MUM	000010	206L		
DVD.SET	000022	209L		
DVD.STE	000053	212E	566	583
DVD.UFL	000012	208L		
DVDFLV	000307	200E	550	562
EC.CNA	000004	67L		
EC.DDA	000027	86L		
EC.DIF	000017	78L		
EC.DIW	000035	92L		
EC.DNI	000045	100L		
EC.DNR	000046	101L		
EC.DNS	000005	68L		
EC.DSC	000047	102L		
EC.EOF	000001	64L		
EC.EOM	000002	65L		
EC.FAO	000031	88L		
EC.FAP	000026	85L		
EC.FL	000030	87L		
EC.FNF	000014	75L		
EC.FNO	000011	72L		
EC.FNR	000034	91L		
EC.FOD	000043	98L		
EC.FUC	000013	74L		
EC.ICN	000016	77L		
EC.IDN	000006	69L		
EC.IFC	000020	79L		
EC.IFN	000007	70L		
EC.ILC	000003	66L		
EC.ILO	000040	95L	597	
EC.ILR	000012	73L	718	
EC.ILV	000037	94L		
EC.IOI	000052	105L		
EC.IS	000032	89L		
EC.NCV	000050	103L		
EC.NEM	000021	80L		
EC.NOS	000051	104L		
EC.NPM	000044	99L		
EC.NRD	000010	71L		
EC.NVM	000042	97L		
EC.OTL	000053	106L		

CROSS REFERENCE TABLE

EC.RF	000022	81L	1007						
EC.UNA	000036	93L							
EC.UND	000015	76L							
EC.UUN	000033	90L	601						
EC.VFM	000041	96L							
EC.WF	000023	82L	997						
EC.WP	000025	84L	1700						
EC.WPV	000024	83L							
ENL	000212	47E	628						
ESC	000033	45E							
FAS	002233	830	922L	1535					
FF	000014	48E							
FLAG	042231	612E	658						
FLAGI	000000	657E							
HELP	000107	622L	661						
HELFI	000001	642	660E						
LF	000012	35E							
LSC	003127	839	1076	1154L					
LSC1	003155	1165	1167L						
M.INI	242355	424E							
M.OUTI	243355	425E							
NL	000012	46E	47	623	623	623	623	624	628
NSPTD	000032	25E	1543						
NSPTS	000015	24E	1540	1543					
NTRK	000115	27E							
NUL2	000000	37E							
NULL	000200	36E							
OPTABE	000203	639	650L						
OPTIAR	000166	590	639L						
OUT	003164	1159	1167	1185L	1633	1733			
OUT.	003174	1117	1190L						
OUT0	003177	1187	1193L						
OUT1	003236	1261L	1266	1274					
OUT128	003230	1222	1255L						
OUT2	003253	1263	1270L						
OUTB	003202	999	1217L	1232					
OVL.IN	000001	306E							
OVL.NUM	000014	308E							
OVL.RES	000002	307E							
OVL.UCS	000200	309E							
PIC.COD	000006	195L	202						
PIC.ID	000000	190L							
PIC.LEN	000002	192L							
PIC.PTR	000004	193L							
PIN	003275	945	1305L	1431					
PIN.	003301	1308L	1695						
PIN1.	003372	1348	1389L	1394	1402				
PIN2	004007	1391	1398L						
PIN256	003364	1334	1383L						
PINB	003304	1009	1329L						
PINB1	003311	1333L	1339						
PINB2	003323	1331	1341L						
PINB3	003336	1352L	1357	1363					
PINB4	003353	1354	1361L						
PRCTAR	000204	589	655L	657	660	663			
QUOTE	000047	43E							
RAS	004021	882	1421L						
RAS1	004055	1436	1441L						

PAGE 56

[illegible]

[illegible]

CROSS REFERENCE TABLE

WDN	005131'	855	947	1036	1114	1170	1219	1343	1365	1435	1802L
WDN.	005137'	1803	1807L								
WDN1	005142'	1809L	1816								
WDNA	000000	1807	1820E								
WTR	005162'	1186	1255	1278	1305	1383	1872L				
WTR.	005170'	1873	1877L								
WTR1	005173'	1879L	1892								
WTRA	000000	1679	1877	1896E							

25644 BYTES FREE