

```
000.001      1 PUBLIC EQU 1 NOT PUBLIC
              3 *** INITIALIZE - INITIALIZE DISK.
              4 *
              5 * J.G. LETWIN, 10/77
              6 *
              7 * COPYRIGHT 10/77, HEATH CO.
              8 * COPYRIGHT 79/04, HEATH CO.
              9 *
             10 * G. Chandler, 78/10 Maintenance release
             11 * 79/04 Renamed INIT from INIT17 for H89
             12 * compatibility reasons.
             13 *
             14 *
             15 *
             16 *** INITIALIZE INITIALIZES A NEW DISK VOLUME.
             17 *
             18 * ACTIONS TAKEN:
             19 *
             20 * 1) REQUEST MEDIA INSERTION
             21 * 2) CHECK FOR PROPER HOLE PATTERN
             22 * 3) READ LABEL AND TYPE INFORMATION
             23 * 4) ASK FOR # OF TRACKS
             24 * 5) ASK FOR DISK SERIAL NUMBER AND LABEL
             25 * 6) ZERO AND INITIALIZE TRACKS
             26 * 7) ASK FOR LIST OF BAD SECTORS
             27 * 8) FORMAT DEVICE
             28 * 9) PROMPT RESTORATION OF SYSTEM DEVICE
             29 *
             30 *
000.303      31 MI.JMP EQU 3030
              32 *
000.002      33 SPB EQU 2 2 SECTORS PER GROUP
001.220      34 VOLSIZE EQU 400 400 SECTORS PER VOLUME
              35 *
000.000      36 XTEXT HTR
```

39X \*\* MTR - PAM/8 EQUIVALENCES.

40X \*  
41X \* THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO  
42X \* MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.

44X \*\* IO PORTS

45X  
000.360 46X IP.PAD EQU 3600 PAD INPUT PORT  
000.360 47X OP.CTL EQU 3600 CONTROL OUTPUT PORT  
000.360 48X OP.DIG EQU 3600 DIGIT SELECT OUTPUT PORT  
000.361 49X OP.SEG EQU 3610 SEGMENT SELECT OUTPUT PORT

51X \*\* FRONT PANEL CONTROL BITS.

52X  
000.020 53X CB.SSI EQU 00010000B SINGLE STEP INTERRUPT  
000.040 54X CB.MTL EQU 00100000B MONITOR LIGHT  
000.100 55X CB.CLI EQU 01000000B CLOCK INTERRUPT ENABLE  
000.200 56X CB.SPK EQU 10000000B SPEAKER ENABLE

58X \*\* MONITOR MODE FLAGS.

59X  
000.000 60X DM.MR EQU 0 MEMORY READ  
000.001 61X DM.MW EQU 1 MEMORY WRITE  
000.002 62X DM.RR EQU 2 REGISTER READ  
000.003 63X DM.RW EQU 3 REGISTER WRITE

65X \*\* USER OPTION BITS.

66X \*  
67X \* THESE BITS ARE SET IN CELL .MFLAG.  
68X  
000.200 69X UD.HLT EQU 10000000B DISABLE HALT PROCESSING  
000.100 70X UD.NFR EQU CB.CLI NO REFRESH OF FRONT PANEL  
000.002 71X UD.IDU EQU 00000010B DISABLE DISPLAY UPDATE  
000.001 72X UD.CLK EQU 00000001B ALLOW PRIVATE INTERRUPT PROCESSING

74X \*\* MONITOR IDENTIFICATION FLAGS

75X \*  
76X \* THESE BYTES IDENTIFY THE ROM MONITOR.  
77X \* THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT  
78X \*  
000.021 79X M.PAMB EQU 0210 'LXI' INSTRUCTION AT 000.000 IN PAM-8  
000.303 80X M.FDX EQU 3030 'JMP' INSTRUCTION AT 000.000 IN FOX ROM

ENTRY

	82X **	ROUTINE ENTRY POINTS.		
	83X *			
	84X			
000.000	85X	.IDENT	EQU	0000A IDENTIFICATION LOCATION
000.053	86X	.DLY	EQU	0053A DELAY
001.267	87X	.LOAD	EQU	1267A TAPE LOAD
001.374	88X	.DUMP	EQU	1374A TAPE DUMP
002.136	89X	.ALARM	EQU	2136A ALARM ROUTINE
002.140	90X	.HORN	EQU	2140A HORN
002.172	91X	.CTC	EQU	2172A CHECK TAPE CHECKSUM
002.205	92X	.TPERR	EQU	2205A TAPE ERROR ROUTINE
002.264	93X	.FCHL	EQU	2264A FCHL INSTRUCTION
002.265	94X	.SRS	EQU	2265A SCAN RECORD START
002.325	95X	.RNP	EQU	2325A READ NEXT PAIR
002.331	96X	.RNB	EQU	2331A READ NEXT BYTE
002.347	97X	.CRC	EQU	2347A CRC-16 CALCULATOR
003.017	98X	.WNP	EQU	3017A WRITE NEXT PAIR
003.024	99X	.WNB	EQU	3024A WRITE NEXT BYTE
003.122	100X	.DOB	EQU	3122A DECODE FOR OCTAL DISPLAY
003.260	101X	.RCK	EQU	3260A READ CONSOLE KEYSET
003.356	102X	.DODA	EQU	3356A SEGMENT CODE TABLE

	104X **	RAM CELLS USED BY HBMTX.		
	105X *			
	106X			
040.000	107X	.START	EQU	40000A START DUMP ADDRESS
040.002	108X	.IOWRK	EQU	40002A IN OR OUT INSTRUCTION
040.005	109X	.REGI	EQU	40005A DISPLAYED REGISTER INDEX
040.006	110X	.DSPROT	EQU	40006A PERIOD FLAG BYTE
040.007	111X	.DISPMOD	EQU	40007A DISPLAY MODE
040.010	112X	.MFLAG	EQU	40010A USER OPTION BYTE
040.011	113X	.CTLFLG	EQU	40011A PANEL CONTROL BYTE
040.013	114X	.ALEDS	EQU	40013A ABUSS LEDES
040.021	115X	.BLEDS	EQU	40021A DBUSS LEDES
040.024	116X	.ABUSS	EQU	40024A ABUSS REGISTER
040.027	117X	.CRCSUM	EQU	40027A CRC SUM WORD
040.031	118X	.TPERRX	EQU	40031A TAPE ERROR EXIT VECTOR
040.033	119X	.TICNT	EQU	40033A CLOCK TICK COUNTER
040.035	120X	.REGPTR	EQU	40035A REGISTER POINTER
040.037	121X	.UIVEC	EQU	40037A USER INTERRUPT VECTORS
000.000	122		XTEXT	ROODEF

	124X **	ROODEF - SPECIAL BOOT-HDOS INTERFACE DEFINITIONS.		
	125X			
047.000	126X	SB.ORG	EQU	47000A ORG FOR LOAD OF INITIAL HDOS.SAV
014.000	127X	SB.DVMX	EQU	14000A SIZE OF HOLD AREA FOR SWAPPED USER CODE
	128X *			(=MAX SIZE OF HDOSVOL.SYS)
000.000	129		XTEXT	ECDEF

131X \*\* ERROR CODE DEFINITIONS.

Address	Code	Device	Value	Description
000.000	133X	ORG	0	
000.000	134X	DS	1	NO ERROR #0
000.001	135X EC.EOF	DS	1	END OF FILE
000.002	136X EC.EOM	DS	1	END OF MEDIA
000.003	137X EC.ILC	DS	1	ILLEGAL SYSCALL CODE
000.004	138X EC.CNA	DS	1	CHANNEL NOT AVAILABLE
000.005	139X EC.DNS	DS	1	DEVICE NOT SUITABLE
000.006	140X EC.IDN	DS	1	ILLEGAL DEVICE NAME
000.007	141X EC.IFN	DS	1	ILLEGAL FILE NAME
000.010	142X EC.WRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011	143X EC.FNO	DS	1	CHANNEL NOT OPEN
000.012	144X EC.ILR	DS	1	ILLEGAL REQUEST
000.013	145X EC.FUC	DS	1	FILE USAGE CONFLICT
000.014	146X EC.FNF	DS	1	FILE NAME NOT FOUND
000.015	147X EC.UND	DS	1	UNKNOWN DEVICE
000.016	148X EC.ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017	149X EC.DIF	DS	1	DIRECTORY FULL
000.020	150X EC.IFC	DS	1	ILLEGAL FILE CONTENTS
000.021	151X EC.NEM	DS	1	NOT ENOUGH MEMORY
000.022	152X EC.RF	DS	1	READ FAILURE
000.023	153X EC.WF	DS	1	WRITE FAILURE
000.024	154X EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025	155X EC.WP	DS	1	DISK WRITE PROTECTED
000.026	156X EC.FAP	DS	1	FILE ALREADY PRESENT
000.027	157X EC.DDA	DS	1	DEVICE DRIVER ABORT
000.030	158X EC.FL	DS	1	FILE LOCKED
000.031	159X EC.FAD	DS	1	FILE ALREADY OPEN
000.032	160X EC.IS	DS	1	ILLEGAL SWITCH
000.033	161X EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	162X EC.FNR	DS	1	FILE NAME REQUIRED
000.035	163X EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	164X EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037	165X EC.ILV	DS	1	ILLEGAL VALUE
000.040	166X EC.ILO	DS	1	ILLEGAL OPTION
000.041	167X EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	168X EC.NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	169X EC.FOD	DS	1	FILE OPEN ON DEVICE
000.044	170X EC.NPM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	171X EC.DNI	DS	1	DISK NOT INITIALIZED
000.046	172X EC.DNR	DS	1	DISK IS NOT READABLE
000.047	173X EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050	174X EC.NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051	175X EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	176X EC.IOI	DS	1	ILLEGAL OVERLAY INDEX
000.053	177X EC.OTL	DS	1	OVERLAY TOO LARGE
000.054	178	XTEXT	M17DEF	

```

180X **      H17 CONTROL INFORMATION.
181X
000.177     182X DF.IC  EQU      07FH      DISK CONTROL PORT
183X
000.001     184X DF.HD  EQU      0000001B  HOLE DETECT
000.002     185X DF.TO  EQU      00000010B  TRACK 0 DETECT
000.004     186X DF.WP  EQU      00000100B  WRITE PROTECT
000.010     187X DF.SD  EQU      00001000B  SYNC DETECT
188X
000.001     189X DF.WG  EQU      00000001B  WRITE GATE ENABLE
000.002     190X DF.DS0 EQU      00000010B  DRIVE SELECT 0
000.004     191X DF.DS1 EQU      00000100B  DRIVE SELECT 1
000.010     192X DF.DS2 EQU      00001000B  DRIVE SELECT 2
000.020     193X DF.MO  EQU      00010000B  MOTOR ON (BOTH DRIVES)
000.040     194X DF.DI  EQU      00100000B  DIRECTION (0=OUT)
000.100     195X DF.ST  EQU      01000000B  STEP COMMAND (ACTIVE HIGH)
000.200     196X DF.WR  EQU      10000000B  WRITE ENABLE RAM
197X
198X
199X
200X **      DISK UART PORTS AND CONTROL FLAGS.
201X
000.174     202X UP.DP  EQU      07CH      DATA PORT
000.175     203X UP.FC  EQU      07DH      FILL CHARACTER
000.175     204X UP.ST  EQU      07EH      STATUS FLAGS
000.176     205X UP.SC  EQU      07EH      SYN CHARACTER (OUTPUT)
000.176     206X UP.SR  EQU      07EH      SYNC RESET (INPUT)
207X
000.001     208X UF.RDA  EQU      00000001B  RECEIVE DATA AVAILABLE
000.002     209X UF.ROR  EQU      00000010B  RECEIVER OVERRUN
000.004     210X UF.RPE  EQU      00000100B  RECEIVER PARITY ERROR
000.100     211X UF.FCT  EQU      01000000B  FILL CHAR TRANSMITTED
000.200     212X UF.TBM  EQU      10000000B  TRANSMITTER BUFFER EMPTY
213X
214X
215X
216X **      CHARACTER DEFINITIONS.
217X
000.375     218X C.DSYN EQU      0FDH      PREFIX SYNC CHARACTER
000.054     219      XTEXT  U8251

```

```

222X **      8251 USART BIT DEFINITIONS.
223X *
224X
225X **      PORT ADDRESSES
226X
000.000     227X UDR   EQU    0          DATA REGISTER IS EVEN
000.001     228X USR   EQU    1          STATUS REGISTER IS NEXT
229X
000.372     230X SC.UART EQU    3720        CONSOLE USART ADDRESS (IFF 8251)
231X
232X
233X **      MODE INSTRUCTION CONTROL BITS.
234X
000.100     235X UMI.1B EQU    01000000B      1 STOP BIT
000.200     236X UMI.HB EQU    10000000B      1 1/2 STOP BITS
000.300     237X UMI.2B EQU    11000000B      2 STOP BITS
000.040     238X UMI.PE EQU    00100000B      EVEN PARITY
000.020     239X UMI.PA EQU    00010000B      USE PARITY
000.000     240X UMI.L5 EQU    00000000B      5 BIT CHARACTERS
000.004     241X UMI.L6 EQU    00000100B      6 BIT CHARACTERS
000.010     242X UMI.L7 EQU    00001000B      7 BIT CHARACTERS
000.014     243X UMI.L8 EQU    00001100B      8 BIT CHARACTERS
000.001     244X UMI.1X EQU    00000001B      CLOCK X 1
000.002     245X UMI.16X EQU   00000010B      CLOCK X 16
000.003     246X UMI.64X EQU   00000011B      CLOCK X 64
247X
248X **      COMMAND INSTRUCTION BITS.
249X
000.100     250X UCI.IR EQU    01000000B      INTERNAL RESET
000.040     251X UCI.RO EQU    00100000B      READER-ON CONTROL FLAG
000.020     252X UCI.ER EQU    00010000B      ERROR RESET
000.004     253X UCI.RE EQU    00000100B      RECEIVE ENABLE
000.002     254X UCI.IE EQU    00000010B      ENABLE INTERRUPTS FLAG
000.001     255X UCI.TE EQU    00000001B      TRANSMIT ENABLE
256X
257X **      STATUS READ COMMAND BITS.
258X
000.040     259X USR.FE EQU    00100000B      FRAMING ERROR
000.020     260X USR.DE EQU    00010000B      OVERRUN ERROR
000.010     261X USR.PE EQU    00001000B      PARITY ERROR
000.004     262X USR.TXE EQU    00000100B      TRANSMITTER EMPTY
000.002     263X USR.RXR EQU    00000010B      RECEIVER READY
000.001     264X USR.TXR EQU    00000001B      TRANSMITTER READY
000.054     265      XTEXT    U8250

267X **      8250 UART CONTROL AND BIT DEFINITIONS.
268X
000.350     269X SC.ACE EQU    3500        SYSTEM CONSOLE PORT IF 8250 ACE
000.156     270X AC.DLY EQU    110        220 MIL. SEC. DELAY FOR 8250
271X
000.000     272X UR.RBR EQU    0          RECEIVER BUFFER REGISTER (READ ONLY)
273X
000.000     274X UR.THR EQU    0          TRANSMITTER HOLDING REGISTER (WRITE ONLY)

```

000.000	275X					
	276X	UR.DLL	EQU	0		DIVISOR LATCH (LEAST SIGNIFICANT)
	277X					
000.001	278X	UR.DLM	EQU	1		DIVISOR LATCH (MOST SIGNIFICANT)
	279X					
000.001	280X	UR.IER	EQU	1		INTERRUPT ENABLE REGISTER
000.001	281X	UC.EDA	EQU	00000001B		ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002	282X	UC.TRE	EQU	00000010B		ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004	283X	UC.KSI	EQU	00000100B		ENABLE RECEIVE STATUS INTERRUPT
000.010	284X	UC.MSI	EQU	00001000B		ENABLE MODEM STATUS INTERRUPT
	285X					
000.002	286X	UR.IIR	EQU	2		INTERRUPT IDENTIFICATION REGISTER
000.001	287X	UC.IIP	EQU	00000001B		INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006	288X	UC.IID	EQU	00000110B		INTERRUPT ID
	289X					
000.003	290X	UR.LCR	EQU	3		LINE CONTROL REGISTER
000.000	291X	UC.5BW	EQU	00000000B		5 BIT WORDS
000.001	292X	UC.6BW	EQU	00000001B		6 BIT WORDS
000.002	293X	UC.7BW	EQU	00000010B		7 BIT WORDS
000.003	294X	UC.8BW	EQU	00000011B		8 BIT WORDS
000.004	295X	UC.2SB	EQU	00000100B		TWO STOP BITS SELECTED
000.010	296X	UC.PEN	EQU	00001000B		PARITY COMPUTATION ENABLED
000.020	297X	UC.EPS	EQU	00010000B		EVEN PARITY SELECT
000.040	298X	UC.SKF	EQU	00100000B		STICK PARITY
000.100	299X	UC.SB	EQU	01000000B		SET BREAK
000.200	300X	UC.DLA	EQU	10000000B		DIVISOR LATCH ACCESS
	301X					
000.004	302X	UR.MCR	EQU	4		MODEM CONTROL REGISTER
000.001	303X	UC.DTR	EQU	00000001B		DATA TERMINAL READY
000.002	304X	UC.RTS	EQU	00000010B		REQUEST TO SEND
000.004	305X	UC.DU1	EQU	00000100B		OUT 1
000.010	306X	UC.DU2	EQU	00001000B		OUT 2
000.020	307X	UC.LDD	EQU	00010000B		LOOP
	308X					
000.005	309X	UR.LSR	EQU	5		LINE STATUS REGISTER
000.001	310X	UC.DR	EQU	00000001B		DATA READY
000.002	311X	UC.Over	EQU	00000010B		OVERRUN
000.004	312X	UC.PE	EQU	00000100B		PARITY ERROR
000.010	313X	UC.FE	EQU	00001000B		FRAMING ERROR
000.020	314X	UC.BI	EQU	00010000B		BREAK INTERRUPT
000.040	315X	UC.THE	EQU	00100000B		TRANSMITTER HOLDING REGISTER EMPTY
000.100	316X	UC.TSE	EQU	01000000B		TRANSMITTER SHIFT REGISTER EMPTY
	317X					
000.006	318X	UR.MSR	EQU	6		MODEM STATUS REGISTER
000.001	319X	UC.DCS	EQU	00000001B		DELTA CLEAR TO SEND
000.002	320X	UC.DDR	EQU	00000010B		DELTA DATA SET READY
000.004	321X	UC.TER	EQU	00000100B		TRAILING EDGE OF RING
000.010	322X	UC.DRL	EQU	00001000B		DELTA RECEIVE LINE SIGNAL DETECT
000.020	323X	UC.CTS	EQU	00010000B		CLEAR TO SEND
000.040	324X	UC.DSR	EQU	00100000B		DATA SET READY
000.100	325X	UC.RI	EQU	01000000B		RING INDICATOR
000.200	326X	UC.RLS	EQU	10000000B		RECEIVED LINE SIGNAL DETECT
000.054	327	XTEXT	DDDEF			DEVICE DRIVER CONSTANTS

```

329X **      DEVICE DRIVER COMMUNICATION FLAGS.
330X *
331X
000.000     332X      ORG      0
333X
000.000     334X DC.REA  DS      1      READ
000.001     335X DC.WRI  DS      1      WRITE
000.002     336X DC.RER  DS      1      READ REGARDLESS
000.003     337X DC.OPR  DS      1      OPEN FOR READ
000.004     338X DC.OPW  DS      1      OPEN FOR WRITE
000.005     339X DC.OPU  DS      1      OPEN FOR UPDATE
000.006     340X DC.CLO  DS      1      CLOSE
000.007     341X DC.ABT  DS      1      ABORT
000.010     342X DC.MOU  DS      1      MOUNT DEVICE
000.011     343X DC.LOB  DS      1      LOAD DEVICE DRIVER
000.012     344X DC.MAX  DS      1      MAXIMUM ENTRY INDEX
000.013     345      XTEXT  FILDEF

```

```

347X **      FILDEF - FILE TYPE DEFINITIONS.
348X *
349X *      DB      377Q,FT,XXX
350X
351X
000.000     352X FT.ABS  EQU      0      ABSOLUTE BINARY
000.001     353X FT.PIC  EQU      1      POSITION INDEPENDANT CODE
000.002     354X FT.REL  EQU      2      RELOCATABLE CODE
000.003     355X FT.BAC  EQU      3      COMPILED BASIC CODE
000.013     356      XTEXT  ABSDEF

```

```

358X **      ABS FORMAT EQUIVALENCES.
359X
000.000     360X      ORG      0
361X
000.000     362X ABS.ID  DS      1      377Q = BINARY FILE FLAG
000.001     363X      DS      1      FILE TYPE (FT,ABS)
000.002     364X ABS.LDA  BS      2      LOAD ADDRESS
000.004     365X ABS.LEN  DS      2      LENGTH OF ENTIRE RECORD
000.006     366X ABS.ENT  DS      2      ENTRY POINT
367X
000.010     368X ABS.COD  DS      0      CODE STARTS HERE
000.010     369      XTEXT  DIRDEF

```



```

371X **      DIRECTORY ENTRY FORMAT.
372X
000.000     373X      ORG      0
374X
375X
000.377     376X DF.EMP  EQU      3770      FLAGS ENTRY EMPTY
000.378     377X DF.CLR  EQU      3760      FLAGS ENTRY EMPTY; REST OF DIR ALSO CLEAR
378X
000.000     379X DIR.NAM  DS        8          NAME
000.010     380X DIR.EXT  DS        3          EXTENSION
000.013     381X DIR.PRO  DS        1          PROJECT
000.014     382X DIR.VER  DS        1          VERSION
000.015     383X DIR.IDL  EQU      *          FILE IDENTIFICATION LENGTH
384X
000.015     385X DIR.CLU  DS        1          CLUSTER FACTOR
000.016     386X DIR.FLG  DS        1          FLAGS
000.017     387X          DS        1          RESERVED
000.020     388X DIR.FGN  DS        1          FIRST GROUP NUMBER
000.021     389X DIR.LGN  DS        1          LAST GROUP NUMBER
000.022     390X DIR.LSI  DS        1          LAST SECTOR INDEX (IN LAST GROUP)
000.023     391X DIR.CRD  DS        2          CREATION DATE
000.025     392X DIR.ALD  DS        2          LAST ALTERATION DATE
393X
000.027     394X DIRELEN  EQU      *          DIRECTORY ENTRY LENGTH
000.027     395          XTEXT  DISDEF
  
```

```

397X **      DIRECTORY BLOCK FORMAT.
398X
000.000     399X      ORG      0
400X
000.000     401X DIS.ENT  EQU      *          FIRST ENTRY ADDRESS
000.000     402X          DS      22*DIRELEN  22 DIRECTORY ENTRIES PER BLOCK
001.372     403X          DS        1          0 BYTE = END OF ENTRIES IN THIS BLOCK
404X
001.373     405X      ORG      512-5      AT END OF BLOCK
001.373     406X DIS.ENL  DS        1          LENGTH OF EACH ENTRY (=DIRELEN)
001.374     407X DIS.SEC  DS        2          BLOCK # OF THIS BLOCK;
001.376     408X DIS.LNK  DS        2          BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000     409          XTEXT  DEVDEF
  
```

```

411X **      DEVICE TABLE ENTRIES.
412X
000.000     413X      ORG      0
414X
000.000     415X DEV.NAM  DS        2          DEVICE NAME
000.000     416X DV.EL   EQU      00000000B  END OF DEVICE LIST FLAG
000.001     417X DV.NU   EQU      00000001B  DEVICE ENTRY NOT IN USE
418X
000.002     419X DEV.RES  DS        1          DRIVER RESIDENSE CODE
000.001     420X DR.IM   EQU      00000001B  DRIVER IN MEMORY
000.002     421X DR.PR   EQU      00000010B  DRIVER PERMINANTLY RESIDENT
  
```

```

422X
000.003 423X DEV.JMP DS 1 JMP TO PROCESSOR
000.004 424X DEV.DBA DS 2 DRIVER ADDRESS
000.006 425X DEV.FLG DS 1 FLAG BYTE
000.001 426X DT.DD EQU 00000001B DIRECTORY DEVICE
000.002 427X DT.CR EQU 00000010B CAPABLE OF READ OPERATION
000.004 428X DT.CW EQU 00000100B CAPABLE OF WRITE OPERATION
429X
000.007 430X DEV.SPG DS 1 SECTORS PER GROUP THIS DEVICE
000.010 431X DEV.MUM DS 1 MOUNTED UNIT MASK
000.011 432X DEV.MNU DS 1 MAXIMUM NUMBER OF UNITS
000.012 433X DEV.UNT DS 2 ADDRESS OF UNIT SPECIFIC DATA TABLE
434X
000.014 435X DEV.DVL DS 2 DRIVER BYTE LENGTH
000.016 436X DEV.DVG DS 1 DRIVER ROUTINE GROUP ADDRESS
437X
000.017 438X DEVELEN EQU * DEVICE TABLE ENTRY LENGTH

```

440X \*\* UNIT SPECIFIC DEVICE DATA TABLE ENTRIES

```

441X
000.000 442X ORG 0
443X
000.000 444X UNT.FLG DS 1 UNIT SPECIFIC *DEV.FLG*
000.001 445X UNT.GRT DS 2 ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.003 446X UNT.GTS DS 2 GRT SECTOR NUMBER
000.005 447X UNT.DIS DS 2 DIRECTORY FIRST SECTOR NUMBER
448X
000.007 449X UNT.SIZ EQU * SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.007 450 XTEXT DDFDEF

```

452X \*\* DIRECTORY DEVICE FORMAT DEFINITION.

```

453X *
454X
455X
000.002 456X HOS.SPG EQU 2 2 SECTORS PER GROUP REQUIRED FOR NOW
457X
000.000 458X ORG 0
000.000 459X DDF.B00 DS 9 2K BOOT PROGRAM
000.011 460X DDF.B0L EQU * LENGTH OF BOOT
000.011 461X DDF.LAB DS 1 LABEL SECTOR
000.012 462X DDF.RGT DS 2 RESERVED GROUP TABLE
000.014 463X DDF.USK DS 0 BEGINNING OF OPEN SPACE
000.014 464 XTEXT LABDEF

```

466X \*\* DISK LABEL SECTOR FORMATS.

467X				
000.000	468X	ORG	0	
000.000	469X	LAB.SER DS	1	SERIAL NUMBER OF VOLUME
000.001	470X	LAB.IND DS	2	INITIALIZATION DATE
000.003	471X	LAB.DIS DS	2	SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005	472X	LAB.GRT DS	2	INDEX OF GRT SECTOR
000.007	473X	LAB.SPG DS	1	SECTORS PER GROUP
	474X			
000.000	475X	LAB.DAT EQU	0	DATA VOLUME ONLY
000.001	476X	LAB.SYS EQU	1	SYSTEM VOLUME
000.002	477X	LAB.NOD EQU	2	=> LAB.NOD MEANS VOLUME HAS NO DIRECTORY
	478X			
000.010	479X	LAB.VLT DS	1	VOLUME TYPE
000.011	480X	LAB.VER DS	1	VERSION OF INIT17 THAT INITED DISK
000.012	481X	DS	7	UNUSED
000.021	482X	LAB.LAB DS	60	LABEL
000.074	483X	LAB.LBL EQU	*-LAB.LAB	LABEL LENGTH
000.115	484	XTEXT	DIFDEF	

486X \*\* DIRECTORY FILE FLAGS.

	487X			
000.200	488X	DIF.SYS EQU	10000000B	SYSTEM FILE
000.100	489X	DIF.LOC EQU	01000000B	LOCKED FOR CHANGE
000.040	490X	DIF.WP EQU	00100000B	WRITE PROTECTED
000.020	491X	DIF.CNT EQU	00010000B	CONTIGUOUS FILE
	492X			
000.115	493	XTEXT	NAMDEF	

495X \*\* SYSTEM FILE NAME CONVENTIONS

	496X	*		
	497X	* RGT	.SYS	RESERVED GROUP TABLE (1 SECTOR)
	498X	* GRT	.SYS	GROUP RESERVATION TABLE (1 SECTOR)
	499X	* DIRECT	.SYS	DIRECTORY
	500X	* NOS	.SYS	SYSTEM IMAGE PROGRAM FOR SYSTEM
	501X			
000.115	502	XTEXT	OVLDEF	

504X \*\* OVERLAY TABLE ENTRIES.

	505X			
000.000	506X	DRG	0	
	507X			
000.000	508X	OVL.COD DS	2	FIRST SECTOR OF OVERLAY CODE
000.002	509X	OVL.SIZ DS	2	OVERLAY SIZE
000.004	510X	OVL.ENT DS	2	OVERLAY ENTRY POINT
000.006	511X	OVL.FLB DS	1	OVERLAY FLAG BYTE
000.007	512X	DS	1	DUMMY BYTE TO ROUND TABLE SIZE UP TO 8

```

000.010      513X OVL.ENS EQU      *      OVERLAY ENTRY SIZE
              514X
              515X *      OVERLAY INDICES
              516X
000.000      517X          ORG      0
              518X
000.000      519X OVL0      DS      1
000.001      520X OVL1      DS      1
000.002      521          XTEXT    HOSEQU

              523X **      HDOS SYSTEM EQUIVALENCES.
              524X *
              525X
024.000      526X S.GRT0    EQU      24000A    SYSTEM AREA FOR GRT0
025.000      527X S.GRT1    EQU      25000A    SYSTEM AREA FOR GRT1
026.000      528X S.GRT2    EQU      26000A    SYSTEM AREA FOR GRT2
              529X
030.000      530X ROMBOOT   EQU      30000A    ROM BOOT ENTRY
              531X
040.100      532X          ORG      40100A    FREE SPACE FROM PAM-B
              533X
040.100      534X          DS      8          JUMP TO SYSTEM EXIT
040.110      535X D.CON      DS      16        DISK CONSTANTS
040.130      536X SYDD      EQU      *        SYSTEM DISK ENTRY POINT
040.130      537X D.VEC     DS      24*3     SYSTEM ROM ENTRY VECTORS
040.240      538X D.RAM     DS      31        SYSTEM ROM WORK AREA
040.277      539X S.VAL     DS      36        SYSTEM VALUES
040.343      540X S.INT     DS      115       SYSTEM INTERNAL WORK AREAS
041.124      541X          DS      16
041.144      542X S.SOVR    DS      2        STACK OVERFLOW WARNING
041.150      543X          DS      42200A-*   SYSTEM STACK
001.032      544X STACKL   EQU      *-S.SOVR  STACK SIZE
              545X
042.200      546X STACK     EQU      *        LWA+1 SYSTEM STACK
042.200      547X USERFWA  EQU      *        USER FWA
042.200      548          XTEXT    HOSDEF

              550X **      HOSDEF - DEFINE HOS PARAMETER.
              551X *
              552X
              553X
000.026      554X VERS     EQU      1*16+6    VERSION 1.6
              555X
000.377      556X SYSCALL  EQU      3770     SYSCALL INSTRUCTION
              557X
              558X
000.000      559X          ORG      0
              560X
              561X *      RESIDENT FUNCTIONS
              562X
000.000      563X .EXIT    DS      1        EXIT (MUST BE FIRST)

```

000.001	564X	.SCIN	DS	1	SCIN
000.002	565X	.SCOUT	DS	1	SCOUT
000.003	566X	.PRINT	DS	1	PRINT
000.004	567X	.READ	DS	1	READ
000.005	568X	.WRITE	DS	1	WRITE
000.006	569X	.CONSL	DS	1	SET/CLEAR CONSOLE OPTIONS
000.007	570X	.CLRCD	DS	1	CLEAR CONSOLE BUFFER
000.010	571X	.LOADO	DS	1	LOAD AN OVERLAY
000.011	572X	.VERB	DS	1	RETURN HDOS VERSION NUMBER
000.012	573X	.SYSRES	DS	1	PRECEDING FUNCTIONS ARE RESIDENT
	574X				
	575X				
	576X	*			*HDOSVOL0.SYS* FUNCTIONS
	577X				
000.040	578X		ORG	40A	
	579X				
000.040	580X	.LINK	DS	1	LINK (MUST BE FIRST)
000.041	581X	.CTLG	DS	1	CTL-C
000.042	582X	.OPENR	DS	1	OPENR
000.043	583X	.OPENW	DS	1	OPENW
000.044	584X	.OPENU	DS	1	OPENU
000.045	585X	.OPENC	DS	1	OPENC
000.046	586X	.CLOSE	DS	1	CLOSE
000.047	587X	.POSIT	DS	1	POSITION
000.050	588X	.DELET	DS	1	DELETE
000.051	589X	.RENAM	DS	1	RENAME
000.052	590X	.SETTP	DS	1	SETTOP
000.053	591X	.DECODE	DS	1	NAME DECODE
000.054	592X	.NAME	DS	1	GET FILE NAME FROM CHANNEL
000.055	593X	.CLEAR	DS	1	CLEAR CHAN
000.056	594X	.CLEARA	DS	1	CLEAR ALL CHANS
000.057	595X	.ERROR	DS	1	LOOKUP ERROR
000.060	596X	.CHFLG	DS	1	CHANGE FLAGS
000.061	597X	.DISMT	DS	1	FLAG SYSTEM DISK DISMOUNTED
000.062	598X	.LOADD	DS	1	LOAD DEVICE DRIVER
	599X				
	600X				
	601X	*			*HDOSVOL1.SYS* FUNCTIONS
	602X				
000.200	603X		ORG	2000	
	604X				
000.200	605X	.MOUNT	DS	1	MOUNT (MUST BE FIRST)
000.201	606X	.DMOUN	DS	1	DISMOUNT
000.202	607X	.MONMS	DS	1	MOUNT/NO MESSAGE
000.203	608X	.DMNMS	DS	1	DISMOUNT/NO MESSAGE
000.204	609X	.RESET	DS	1	RESET = DISMOUNT/MOUNT OF UNIT
000.205	610		XTEXT	EDRAM	

```

612X **      EDRAM - DISK RAM WORKAREA DEFINITION.
613X *
614X *      ZEROED UPON BOOTING UP.
615X *
616X *      HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.
617X
618X
040.240      619X      ORG      D.RAM
620X
040.240      621X D.TT      DS      1      TARGET TRACK (CURRENT OPERATION)
040.241      622X D.TS      DS      1      TARGET SECTOR (CURRENT OPERATION)
623X
040.242      624X D.DVCTL DS      1      DEVICE CONTROL BYTE
625X
040.243      626X D.DLYMO DS      1      MOTOR ON DELAY COUNT
040.244      627X D.DLYHS DS      1      HEAD SETTLE DELAY COUNTER
628X
040.245      629X D.TRKPT DS      2      ADDRESS IN D.DRVTB FOR TRACK NUMBER
040.247      630X D.VOLPT DS      2      ADDRESS IN D.DRVTB FOR VOLUME NUMBER
631X
040.251      632X D.DRVTB DS      2*4    TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
633X
040.261      634X D.HECNT DS      1      HARD ERROR COUNT
040.262      635X D.SECNT DS      2      SOFT ERROR COUNT
040.264      636X D.GECNT DS      1      OPERATION ERROR COUNT
637X
638X *      GLOBAL DISK ERROR COUNTERS
639X
040.265      640X D.ERR      DS      0      BEGINNING OF ERROR BLOCK
040.265      641X D.E.MDS DS      1      MISSING DATA SYNC
040.266      642X D.E.HSY DS      1      MISSING HEADER SYNC
040.267      643X D.E.CHK DS      1      DATA CHECKSUM
040.270      644X D.E.HCK DS      1      HEADER CHECKSUM
040.271      645X D.E.VOL DS      1      WRONG VOLUME NUMBER
040.272      646X D.E.TRK DS      1      BAD TRACK SEEK
040.273      647X D.ERRL DS      0      LIMIT OF ERROR COUNTERS
648X
649X *      I/O OPERATION COUNTS
650X
040.273      651X D.OPR      DS      2
040.275      652X D.OPW      DS      2
653X
000.037      654X D.RAML EQU      *-D.RAM
040.277      655      XTEXT EDVEC

657X **      JMP VECTORS FOR ROM CODE
658X *
659X *      SEE DISK ROM FOR ADDRESSES
660X *
661X *      HOSEQU MUST BE ALTERED WHEN THIS TABLE IS ALTERED.
662X
040.130      663X      ORG      D.VEC
664X

```

040.130	665X	D:SYDD	DS	3	JMP	R:SYDD (MUST BE FIRST)
040.133	666X	D:MOUNT	DS	3	JMP	R:MOUNT
040.136	667X	D:XOK	DS	3	JMP	R:XOK
040.141	668X	D:ABORT	DS	3	JMP	R:ABORT
040.144	669X	D:XIT	DS	3	JMP	R:XIT
040.147	670X	D:READ	DS	3	JMP	R:READ
040.152	671X	D:READR	DS	3	JMP	R:READR
040.155	672X	D:WRITE	DS	3	JMP	R:WRITE
040.160	673X	D:CDE	DS	3	JMP	R:CDE
040.163	674X	D:DTS	DS	3	JMP	R:DTS
040.166	675X	D:SDT	DS	3	JMP	R:SDT
040.171	676X	D:MAI	DS	3	JMP	R:MAI
040.174	677X	D:MAO	DS	3	JMP	R:MAO
040.177	678X	D:LPS	DS	3	JMP	R:LPS
040.202	679X	D:RDB	DS	3	JMP	R:RDB
040.205	680X	D:SDP	DS	3	JMP	R:SDP
040.210	681X	D:STS	DS	3	JMP	R:STS
040.213	682X	D:STZ	DS	3	JMP	R:STZ
040.216	683X	D:UDLY	DS	3	JMP	R:UDLY
040.221	684X	D:WSC	DS	3	JMP	R:WSC
040.224	685X	D:WSP	DS	3	JMP	R:WSP
040.227	686X	D:WNB	DS	3	JMP	R:WNB
040.232	687X	D:ERRT	DS	3	JMP	R:ERRT
040.235	688X	D:DLY	DS	3	JMP	R:DLY
040.240	689	XTEXT	ESVAL			

691X \*\* S.VAL - SYSTEM VALUE DEFINITIONS.

692X \*  
 693X \* THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.  
 694X \*  
 695X \* THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.

040.277	696X					
	697X					
	698X	ORG	S.VAL			
	699X					
040.277	700X	S:DATE	DS	9		SYSTEM DATE (IN ASCII)
040.310	701X	S:DATC	DS	2		CODED DATE
040.312	702X	S:TIME	DS	4		TIME FROM MIDNIGHT (IN TIC'S)
040.316	703X	S:HIMEM	DS	2		HARDWARE HIGH MEMORY ADDRESS+1
	704X					
040.320	705X	S:SYSM	DS	2		FWA RESIDENT SYSTEM
	706X					
040.322	707X	S:USRM	DS	2		LWA USER MEMORY
	708X					
040.324	709X	S:OMAX	DS	2		MAX OVERLAY SIZE FOR SYSTEM
	710X					
	711X					

712X \*\* THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL

000.200	714X	CSL:ECH	ERU	10000000B		SUPPRESS ECHO
000.002	715X	CSL:WRP	ERU	00000010B		WRAP LINES AT WIDTH
000.001	716X	CSL:CHR	ERU	00000001B		OPERATE IN CHARACTER MODE
	717X					

```

000.000      718X I.CSLMD EQU      0      S.CSLMD IS FIRST BYTE
040.326      719X S.CSLMD DS      1      CONSOLE MODE
              720X
000.200      721X CTP.BKS EQU     10000000B  TERMINAL PROCESSES BACKSPACES
000.040      722X CTP.MLI EQU     00100000B  MAP LOWER CASE TO UPPER ON INPUT
000.020      723X CTP.MLO EQU     00010000B  MAP LOWER CASE TO UPPER ON OUTPUT
000.010      724X CTP.2SB EQU     00001000B  TERMINAL NEEDS TWO STOP BITS
000.002      725X CTP.BKM EQU     00000010B  MAP BKSP (UPON INPUT) TO RUBOUT
000.001      726X CTP.TAB EQU     00000001B  TERMINAL SUPPORTS TAB CHARACTERS
              727X
000.001      728X I.CONTY EQU      1      S.CONTY IS 2ND BYTE
000.000      729X          ERRNZ  *-S.CSLMD-I.CONTY
040.327      730X S.CONTY DS      1      CONSOLE TYPE FLAGS
000.002      731X I.CUSOR EQU      2      S.CUSOR IS 3RD BYTE
000.000      732X          ERRNZ  *-S.CSLMD-I.CUSOR
040.330      733X S.CUSOR DS      1      CURRENT CURSOR POSITION
000.003      734X I.CONWI EQU      3      S.CONWI IS 4TH BYTE
000.000      735X          ERRNZ  *-S.CSLMD-I.CONWI
040.331      736X S.CONWI DS      1      CONSOLE WIDTH
              737X
000.001      738X CO.FLG EQU     00000001B  CTL-D FLAG
000.200      739X CS.FLG EQU     10000000B  CTL-S FLAG
              740X
000.004      741X I.CONFL EQU      4      S.CONFL IS 5TH BYTE
000.000      742X          ERRNZ  *-S.CSLMD-I.CONFL
040.332      743X S.CONFL DS      1      CONSOLE FLAGS
              744X
040.333      745X S.CAADR DS      2      ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335      746X S.CCTAB DS      6      ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343      747          XTEXT  ESINT

              749X **          S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
              750X *
              751X *          THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
              752X *          MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
              753X
              754X
040.343      755X          ORG      S.INT
              756X
              757X **          CONSOLE STATUS FLAGS
              758X
040.343      759X S.CDB  DS      1      CONSOLE DESCRIPTOR BYTE
000.000      760X CDB.H85 EQU     00000000B
000.001      761X CDB.H84 EQU     00000001B  =0 IF H8-5, =1 IF H8-4
040.344      762X S.BAUD DS      2      [0-14] H8-4 BAUD RATE, =0 IF H8-5
              763X *          [15]   =1 IF BAUD RATE => 2 STOP BITS
              764X
              765X **          TABLE ADDRESS WORDS
              766X
040.346      767X S.DLINK DS      2      ADDRESS OF DATA IN HDOS CODE
040.350      768X S.OFWA DS      2      FWA OVERLAY TABLE
040.352      769X S.CFWA DS      2      FWA CHANNEL TABLE
040.354      770X S.DFWA DS      2      FWA DEVICE TABLE

```



040.356	771X	S.RFWA	DS	2	FWA RESIDENT HDOS CODE
	772X				
	773X	**			DEVICE DRIVER DELAYED LOAD FLAGS
	774X				
040.360	775X	S.DDLDA	DS	2	DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362	776X	S.DDLEN	DS	2	CODE LENGTH IN BYTES
040.364	777X	S.DDGRP	DS	1	GROUP NUMBER FOR DRIVER
040.365	778X		DS	1	HOLD PLACE
	779X	*S.DDSEC	DS	2	SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366	780X	S.DDDTA	DS	2	DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370	781X	S.DDDPC	DS	1	OPEN OPCODE PENDING
	782X				
	783X	**			OVERLAY MANAGEMENT FLAGS
	784X				
000.001	785X	OVL.IN	EQU	00000001B	IN MEMORY
000.002	786X	OVL.RES	EQU	00000010B	PERMINANTLY RESIDENT
000.014	787X	OVL.NUM	EQU	00001100B	OVERLAY NUMBER MASK
000.200	788X	OVL.UCS	EQU	10000000B	USER CODE SWAPPED FOR OVERLAY
	789X				
040.371	790X	S.OVLFL	DS	1	OVERLAY FLAG
040.372	791X	S.UCSF	DS	2	FWA SWAPPED USER CODE
040.374	792X	S.UCSL	DS	2	LENGTH SWAPPED USER CODE
040.376	793X	S.OVLS	DS	2	SIZE OF OVERLAY CODE
041.000	794X	S.OVLE	DS	2	ENTRY POINT OF OVERLAY CODE
	795X				
041.002	796X	S.SSN	DS	2	SWAP AREA SECTOR NUMBER
041.004	797X	S.OSN	DS	2	OVERLAY SECTOR NUMBER
	798X				
	799X	*			SYSCALL PROCESSING WORK AREAS
	800X				
041.006	801X	S.CACC	DS	1	(ACC) UPON SYSCALL
041.007	802X	S.CODE	DS	1	SYSCALL INDEX IN PROGRESS
	803X				
	804X	*			JUMPS TO ROUTINES IN RESIDENT HDOS CODE
	805X				
041.010	806X	S.JUMPS	DS	0	START OF DUMP VECTORS
041.010	807X	S.SDB	DS	3	JUMP TO STAND-IN DEVICE DRIVER
041.013	808X	S.FASER	DS	3	JUMP TO FATERR (FATAL SYSTEM ERROR)
041.016	809X	S.DIREA	DS	3	JUMP TO DIREAD (DISK FILE READ)
041.021	810X	S.FCI	DS	3	JUMP TO FCI (FETCH CHANNEL INFO)
041.024	811X	S.SCI	DS	3	JUMP TO SCI (STORE CHANNEL INFO)
041.027	812X	S.GUP	DS	3	JUMP TO GUP (GET UNIT POINTER)
	813X				
041.032	814X	S.MOUNT	DS	1	<> IF THE SYSTEM DISK IS MOUNTED
041.033	815X	S.DCS	DS	1	DEFAULT CLUSTER SIZE-1
	816X				
041.034	817X	S.BOOTF	DS	1	BOOT FLAGS
000.001	818X	BOOT.P	EQU	00000001B	EXECUTE PROLOGUE UPON BOOTUP
	819X				
	820X	*			STACK VALUE SAVED FOR OVERLAY SYSCALLS
	821X				
041.035	822X	S.OVSTK	DS	2	VALUE OF SP UPON SYSCALLS USING OVERLAY
	823X				
041.037	824X		DS	1	RESERVED

```

826X **      ACTIVE I/O AREA.
827X *
828X *      THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
829X *      CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
830X *      THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
831X *
832X *      NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
833X *      FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
834X *      8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
835X *      COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
836X *      BACKDATED AFTER PROCESSING.
837X *
041.040      838X AIO.VEC DS      3      JUMP INSTRUCTION
041.041      839X AIO.DDA EQU     *-2     DEVICE DRIVER ADDRESS
041.043      840X AIO.FLG DS      1      FLAG BYTE
041.044      841X AIO.GRT DS      2      ADDRESS OF GROUP RESERV TABLE
041.046      842X AIO.SFG DS      1      SECTORS PER GROUP
041.047      843X AIO.CGN DS      1      CURRENT GROUP NUMBER
041.050      844X AIO.CSI DS      1      CURRENT SECTOR INDEX
041.051      845X AIO.LGN DS      1      LAST GROUP NUMBER
041.052      846X AIO.LSI DS      1      LAST SECTOR INDEX
041.053      847X AIO.DTA DS      2      DEVICE TABLE ADDRESS
041.055      848X AIO.DES DS      2      DIRECTORY SECTOR
041.057      849X AIO.DEV DS      2      DEVICE CODE
041.061      850X AIO.UNI DS      1      UNIT NUMBER (0-9)
851X
041.062      852X AIO.DIR DS      DIRELEN  DIRECTORY ENTRY
853X
041.111      854X AIO.CNT DS      1      SECTOR COUNT
041.112      855X AIO.EOM DS      1      END OF MEDIA FLAG
041.113      856X AIO.EOF DS      1      END OF FILE FLAG
041.114      857X AIO.TFP DS      2      TEMP FILE POINTERS
041.116      858X AIO.CHA DS      2      ADDRESS OF CHANNEL BLOCK (IOC.DDA)

041.120      860X S.SCR DS      2      SYSTEM SCRATCH AREA ADDRESS
041.122      861      XTEXT H17ROM

863X **      H17 ROM DEFINITIONS
864X
036.235      865X R.WHD EQU     36235A
036.271      866X R.WNH EQU     36271A
035.303      867X R.DLY EQU     35303A
041.122      868      XTEXT PICDEF

```

870X \*\* PIC FORMAT EQUIVALENCES.

000.000	871X				
	872X	ORG	0		
	873X				
000.000	874X	PIC.ID	DS	1	377Q = BINARY FILE FLAG
000.001	875X		DS	1	FILE TYPE (FT.PIC)
000.002	876X	PIC.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.004	877X	PIC.PTR	DS	2	INDEX OF START OF PIC TABLE
	878X				
000.006	879X	PIC.COD	DS	0	CODE STARTS HERE
000.006	880	XTEXT	ASCII		

882X \*\* ASCII CHARACTER EQUIVALENCES.

	883X				
000.015	884X	CR	EQU	13	CARRIAGE RETURN
000.012	885X	LF	EQU	10	LINE FEED
000.200	886X	NULL	EQU	200Q	PAD CHARACTER
000.000	887X	NUL2	EQU	0	
000.007	888X	BELL	EQU	7	BELL CHARACTER
000.177	889X	RUBOUT	EQU	177Q	
000.010	890X	BKSP	EQU	10Q	CTL-H
000.026	891X	C.SYN	EQU	26Q	SYNC
000.002	892X	C.STX	EQU	2	STX
000.047	893X	QUOTE	EQU	47Q	
000.011	894X	TAB	EQU	11Q	
000.033	895X	ESC	EQU	33Q	
000.012	896X	NL	EQU	12Q	NEW LINE (HDOS SYSTEMS)
000.212	897X	ENL	EQU	NL+200Q	NL + END-OF-LINE-FLAG
000.014	898X	FF	EQU	14Q	FORM FEED
000.001	899X	CTLA	EQU	01Q	CTL-A
000.002	900X	CTLB	EQU	02Q	CTL-B
000.003	901X	CTLC	EQU	03Q	CTL-C
000.004	902X	CTLD	EQU	04Q	CTL-D
000.017	903X	CTL0	EQU	17Q	CTL-0
000.020	904X	CTLP	EQU	20Q	CTL-P
000.021	905X	CTLQ	EQU	21Q	CTL-Q
000.023	906X	CTLS	EQU	23Q	CTL-S
000.032	907X	CTLZ	EQU	32Q	CTL-Z

```

042.170          910      ORG      USERFWA-ABS.COD
          911
          912
042.170  377 000    913      DB      3770,FT,ABS
042.172  200 042    914      DW      USERFWA      LOAD ADDR
042.174  122 022    915      DW      MEML-USERFWA    SIZE
042.176  010 051    916      DW      INIT          ENTRY
          917
          918      LON      C
          919
          920
          921  **      SOBODT - SECTOR 0 BOOT ROUTINE.
          922  *
          923  *      THIS BOOT STARTS AT SECTOR 0 ON EVERY INITIALIZED
          924  *      DISK, AND OCCUPIES THE FIRST 9 SECTORS OF THE DISK.
          925  *
          926  *      IT IS BROUGHT IN BY THE H17 ROM.
          927
          928
          929
042.200  257          930  SOBODT  XRA      A
042.201  323 175      931      OUT     UP,FC      SET FILL CHARACTER = 0
042.203  315 000 046  932      CALL    FCU      FIND CONSOLE USART
042.206  315 155 046  933      CALL    MSD      MOUNT SYSTEM DISK
          934
042.211  061 200 042  935  SOBODTX LXI     SP,STACK
042.214  257          936      XRA      A
042.215  062 062 041  937      STA     AID.DIR      AM WORKING WITH NO FILES YET
042.220  315 055 045  938      CALL    $TYPET
042.223  015 012 012  939      DB     CR,LF,LF,'ACTION? <BOOT>',' '+2000
          940
          941  *      GET REPLY, MAY BE:
          942  *
          943  *      BOOT
          944  *      CHECK
          945  *      HELP
          946
042.245  315 257 044  947      CALL    $ICTT      INPUT TASK TIME
042.250  315 007 050  948      CALL    $MCU      MAP TO UPPER CASE
042.253  376 015      949      CPI     CR
042.255  312 042 043  950      JE      SOBODT2    IS BOOT
042.260  376 102      951      CPI     'B'
042.262  312 042 043  952      JE      SOBODT2    IS BOOT
042.265  376 103      953      CPI     'C'
042.267  312 105 043  954      JE      SOBODT4    IS CHECK
042.272  376 111      955      CPI     'I'
042.274  312 124 043  956      JE      SOBODT5    IS IGNORE
          957
          958  *      ASSUME HELP
          959
042.277  315 055 045  960      CALL    $TYPET
042.302  110 105 114  961      DB     'HELP',0,0
042.310  114 105 107  962      DB     'LEGAL COMMANDS:',0
042.330  102 117 117  963      DB     'BOOT - BOOT HDOS',0
042.353  103 110 105  964      DB     'CHECK - SECTOR CHECKSUMS',0
043.005  110 105 114  965      DB     'HELP - PRINT THIS LIST',0

```

```

000.001          966      IF      PUBLIC
                967      DB      'IGNORE - IGNORE PROLOGUE FILE'
                968      ENDIF
043.036 200      969      DB      2000
043.037 303 211 042 970      JMP     SOB00TX          TRY AGAIN
                971
                972 *      IS ROOT
                973
043.042 315 055 045 974 SOB00T2 CALL  $TYFET
043.045 102 117 117 975      DB      'BOOT',2000
043.052 072 034 041 976      LDA      S.BOOTF
043.055 366 001      977      ORI      BOOT.P          FLAY PROLOGUE EXECUTION UPON BOOTUP
043.057 062 034 041 978      STA      S.BOOTF
043.062 315 155 046 979 SOB00T3 CALL  MSR          MOUNT THIS DISK
043.065 315 151 043 980      CALL  LEP          LOAD AND EXECUTE PROGRAM
043.070 110 104 117 981      DB      'HDDS',0,0,0,0
043.100 123 131 123 982      DB      'SYS',0,0
                983
                984 *      IS CHECK
                985
043.105 315 055 045 986 SOB00T4 CALL  $TYFET
043.110 103 110 105 987      DB      'CHECK',2000
043.114 315 020 050 988      CALL  CDC          COMPUTE DISK CCHECKS
043.121 303 211 042 989      JMP     SOB00TX          TRY AGAIN
                990
                991 *      IS IGNORE
                992
043.124 315 055 045 993 SOB00T5 CALL  $TYFET
043.127 111 107 116 994      DB      'IGNORE',2000
043.136 072 034 041 995      LDA      S.BOOTF
043.141 346 376      996      ANI      3770-BOOT.P
043.143 062 034 041 997      STA      S.BOOTF          TURN OFF PROLOGUE UPON BOOT FLAG
043.146 303 062 043 998      JMP     SOB00T3

```

```

1001 **      LEP IS CALLED TO LOAD AND EXECUTE A DISK FILE.
1002 *
1003 *      THE DISKS DIRECTORY IS SEARCHED FOR THE APPROPRIATE FILE NAME.
1004 *      IF FOUND, IT IS LOADED INTO MEMORY AT SB.ORG AND EXECUTED.
1005 *
1006 *      IF NOT FOUND, TYPE ERROR MESSAGE:
1007 *
1008 *      NEEDED FILE * FNAME * IS MISSING
1009 *
1010 *      AND RETURN TO SOB00T.
1011 *
1012 *      IF ERROR IN READING THE FILE, TYPE
1013 *
1014 *      'DISK READ ERROR IN FILE * FNAME *'
1015 *
1016 *      AND RE-BOOT.
1017 *
1018 *      ENTRY  ((SP)) = FILE NAME
1019 *      EXIT   TO SB.ORG IF LOAD SUCCESSFUL,
1020 *           TO ROMBOOT IF READ ERROR,
1021 *           TO SOB00TX IF FILE MISSING
1022 *      USES   ALL
1023
1024
1025 LEP      POP      D           (DE) = NAME ADDRESS
043.151 321
043.152 001 015 000 1026 LXI      B,DIRIDL
043.155 041 062 041 1027 LXI      H,AIO.DIR
043.160 315 252 030 1028 CALL     $MOVE           MOVE IN NAME
043.163 072 020 051 1029 LDA      BLABEL+LAB.VLT   (A) = VOLUME TYPE
043.166 247
000.000 1030 ANA      A
043.167 312 032 044 1031 ERRNZ   LAB.DAT
043.172 075 1032 JZ       LEP5           IS DATA DISK
000.000 1033 DCR      A
043.173 302 131 044 1034 ERRNZ   LAB.SYS-1
043.176 001 015 000 1035 JNZ     LEP6           IS GARBAGE DISK
043.201 052 013 051 1036 LXI      B,DIRIDL
043.204 315 212 045 1037 MLD     BLABEL+LAB.DIS
043.207 322 262 043 1038 CALL    LHE,,          LOAD ENTRY
1039 JNC     LEP1          FOUND
1040
1041 *      COULDNT FIND IT
1042
043.212 315 055 045 1043 CALL    $TYPET
043.215 007 077 060 1044 DB      BELL,'?00 REQUIRED FILE','+2000
043.240 315 341 044 1045 CALL    TFN           TYPE FILE NAME
043.243 315 055 045 1046 CALL    $TYPET
043.246 040 115 111 1047 DB      ' MISSING',BELL+2000
043.257 303 211 042 1048 JMP     SOB00TX
1049
1050 *      GOT DIRECTORY ENTRY. TRY TO READ IT
1051
043.262 021 016 000 1052 LEP1    LXI      B,DIR.FLG
043.265 031 1053 DAD     D           (HL) = ADDRESS OF FLG
043.266 176 1054 MOV     A,M
043.267 346 020 1055 ANI     BIF.CNT
043.271 312 343 043 1056 JZ      LEP4          NOT CONTIGUOUS

```

```

000.000          1057      ERRNZ  DIR.FGN-DIR.FLG-2
043.274 043      1058      INX    H
043.275 043      1059      INX    H          (HL) = #DIR.FGN
043.276 136      1060      MOV    E,M
043.277 026 000  1061      MVI    D,0          (DE) = GROUP NUMBER
043.301 072 017 051 1062      LDA    BLABEL+LAB.SPG (A) = SECTORS PER GROUP
043.304 315 007 031 1063      CALL   $MUB6        (HL) = SECTOR NUMBER FOR FILE
                    1064
                    1065 *      (HL) = SECTOR NUMBER FOR FILE
                    1066
043.307 001 000 001 1067 LEP3   LXI    B,256
043.312 021 000 047 1068      LXI    D,SB,ORG
043.315 345      1069      PUSH  H
043.316 315 276 045 1070      CALL   READD        READ DISK
043.321 052 002 047 1071      LHLD  SB,ORG+PIC.LEN
043.324 053      1072      DCX   H          (HL) = SECTOR COUNT
043.325 104      1073      MOV    B,H
043.326 016 000  1074      MVI    C,0
043.330 341      1075      POP   H          (HL) = SECTOR NUMBER OF FWA
043.331 043      1076      INX    H          ALREADY READ 1
043.332 021 000 050 1077      LXI    D,SB,ORG+256
043.335 315 276 045 1078      CALL   READD        READ THE REMAINDER
043.340 303 006 047 1079      JMP   SB,ORG+PIC.COD ALL OK, EXECUTE IT
                    1080
                    1081 *      FILE NOT CONTIGUOUS
                    1082
043.343 315 055 045 1083 LEP4   CALL   $TYPET
043.346 000 007 077 1084      DB    0,'BELL,'?00 THIS DISK HAS NOT BEEN PROPERLY SYSGENED.',BELL,2000
044.027 303 131 044 1085      JMP   LEP6
                    1086
                    1087 *      IS DATA DISK. NOT YET SYSGENED
                    1088
044.032 315 055 045 1089 LEP5   CALL   $TYPET
044.035 000 007 077 1090      DB    0,'BELL,'?00 THIS DISK MUST BE SYSGENED BEFORE IT CAN BE USED.',BELL,2000
044.126 303 211 042 1091      JMP   SOB00TX
                    1092
                    1093 *      GARBAGE DISK.
                    1094
044.131 315 055 045 1095 LEP6   CALL   $TYPET
044.134 000 007 077 1096      DB    0,'BELL,'?00 THIS DISK MUST BE INITIALIZED AND THEN SYSGENED'
044.221 000 040 040 1097      DB    0,' BEFORE IT CAN BE USED.',BELL,2000
044.254 303 211 042 1098      JMP   SOB00TX

```

```

044.257          1101          XTEXT  ICTT

1103X **        $ICTT - INPUT FROM CONSOLE TASK TIME.
1104X *
1105X *        $ICTT IS A TASK-TIME CONSOLE INPUT ROUTINE, WHICH
1106X *        PERFORMS SIMPLE SINGLE CHARACTER INPUTS.
1107X *
1108X *        IT IS CALLED DURING BOOT OPERATIONS, AND BY SPECIAL ROUTINES
1109X *        WHICH MAY BE RUNNING IN ENVIRONMENTS WHERE KEYBOARD INTERRUPTS
1110X *        ARE UNDESIRABLE.
1111X *
1112X *        Modified to handle HB-4 ports by G. Chandler, 1-SEP-78
1113X *        This routine assumes that the ports have been previously initialized,
1114X *        and that S.CDB has been previously initialized.
1115X *
1116X *        ENTRY  NONE
1117X *        EXIT  (A) = CHARACTER
1118X *        USES  A,F
1119X *
1120X *
044.257 315 271 044 1121X $ICTT CALL $ICTT.
044.262 332 257 044 1122X      JC   $ICTT
044.265 315 317 044 1123X      CALL $ICTT.,
044.270 311          1124X      RET
1125X *
044.271 072 343 040 1126X $ICTT. LDA  S.CDB
044.274 376 001     1127X      CPI  CDB.HB4
044.276 312 310 044 1128X      JZ   ICTT2          IF HB-4 PORT
1129X *
1130X *        HAVE 8251 FOR CONSOLE
1131X *
044.301 333 373     1132X ICTT1 IN   SC.UART+USR
044.303 346 002     1133X      ANI  USR.RXR
044.305 300          1134X      RNZ          READY
1135X *
044.306 067         1136X      STC          FLAG NOT READY
044.307 311         1137X      RET
1138X *
1139X *        HAVE 8250 PORT FOR CONSOLE
1140X *
044.310 333 355     1141X ICTT2 IN   SC.ACE+UR.LSR
044.312 346 001     1142X      ANI  UC.DR
044.314 300          1143X      RNZ          READY
1144X *
044.315 067         1145X      STC          FLAG NOT READY
044.316 311         1146X      RET
1147X *
044.317 072 343 040 1148X $ICTT. LDA  S.CDB
044.322 376 001     1149X      CPI  CDB.HB4
044.324 312 334 044 1150X      JZ   ICTT3
1151X *
1152X *        HAVE 8251 FOR CONSOLE
1153X *

```



```

044.327 333 372 1154X IN SC:UART+UDR
044.331 346 177 1155X ANI 177Q
044.333 311 1156X RET
1157X
1158X * HAVE 8250 FOR CONSOLE
1159X
044.334 333 350 1160X ICTT3 IN SC:ACE+UR:RBR
044.336 346 177 1161X ANI 177Q
044.340 311 1162X RET

```

```

1164 ** TFN - TYPE FILE NAME.
1165 *
1166 * TFN TYPES THE FILE WHOSE NAME APPEARS IN AIO,XXX
1167 *
1168 * ENTRY NONE
1169 * EXIT NONE
1170 * USES A,F,B,H,L
1171
1172

```

```

044.341 041 062 041 1173 TFN LXI H,AIO:DIR+DIR,NAM
044.344 006 010 1174 MVI B,B
044.348 315 360 044 1175 CALL TFN1 TYPE NAME
044.351 076 056 1176 MVI A,'.'
044.353 315 112 045 1177 CALL $TYPEC.
044.356 006 003 1178 MVI B,3
1179
044.360 176 1180 TFN1 MOV A,M
044.361 247 1181 ANA A
044.362 304 112 045 1182 CNZ $TYPEC.
044.365 043 1183 INX H
044.366 005 1184 DCR B
044.367 302 360 044 1185 JNZ TFN1
044.372 311 1186 RET
044.373 1187 XTEXT BCRC

```

```

1189X ** $BCRC - GENERATE CRC16 ON A BLOCK OF DATA.
1190X *
1191X * *** WARNING ***
1192X *
1193X * THIS CRC-16 IS NOT COMPATIBLE WITH THE ONE
1194X * PRODUCED BY PAM-8, AND THE DECK 'CRC.COM'
1195X *
1196X * ENTRY (BC) = BYTE COUNT
1197X * (HL) = ADDRESS
1198X * (DE) = CRC ACCUMULATOR
1199X * EXIT (HL) = (HL)+(BC)
1200X * (DE) = NEW CRC
1201X * USES ALL
1202X
1203X

```

```

044.373 170      1204X $BCRC  MOV  A>B
044.374 261      1205X      ORA  C
044.375 310      1206X      RZ
044.376 176      1207X      MOV  A>H
044.377 345      1208X      PUSH H
045.000 305      1209X      PUSH B
045.001 253      1210X      XRA  E
045.002 107      1211X      MOV  B>A
045.003 017      1212X      RRC
045.004 017      1213X      RRC
045.005 017      1214X      RRC
045.006 017      1215X      RRC
045.007 117      1216X      MOV  C>A
045.010 250      1217X      XRA  B
045.011 346 360  1218X      ANI  0FOH
045.013 252      1219X      XRA  D
045.014 157      1220X      MOV  L>A
045.015 171      1221X      MOV  A>C
045.016 007      1222X      RLC
045.017 346 037  1223X      ANI  1FH
045.021 255      1224X      XRA  L
045.022 157      1225X      MOV  L>A
045.023 170      1226X      MOV  A>B
045.024 007      1227X      RLC
045.025 346 001  1228X      ANI  1
045.027 252      1229X      XRA  D
045.030 255      1230X      XRA  L
045.031 127      1231X      MOV  D>A
045.032 171      1232X      MOV  A>C
045.033 346 360  1233X      ANI  0FOH
045.035 250      1234X      XRA  B
045.036 137      1235X      MOV  E>A
045.037 171      1236X      MOV  A>C
045.040 250      1237X      XRA  B
045.041 007      1238X      RLC
045.042 346 340  1239X      ANI  0EOH
045.044 253      1240X      XRA  E
045.045 137      1241X      MOV  E>A
045.046 301      1242X      POP  B
045.047 341      1243X      POP  H
045.050 043      1244X      INX  H
045.051 013      1245X      DCX  B
045.052 303 373 044 1246X      JMP  $BCRC
045.055      1247      XTEXT MUB6

```

```

1249X **      $MUB6 - MULTIPLY 8X16 UNSIGNED.
1250X *
1251X *      $MUB6 MULTIPLIES A 16 BIT VALUE BY A 8
1252X *      BIT VALUE.
1253X *
1254X *      ENTRY (A) = MULTIPLIER
1255X *      (DE) = MULTIPLICAND
1256X *      EXIT (HL) = RESULT

```

\*MUB6

```

1257X *      'Z' SET IF NOT OVERFLOW
1258X *      USES      A,F,H,L
1259X
1260X
031.007     1261X *MUB6 EQU      31007A      IN H17 ROM
045.055     1262      XTEXT  TYPET

1264X **      $TYPET - TYPE TEXT.
1265X *
1266X *      $TYPET IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE
1267X *      AT TASK TIME RATHER THAN AT INTERRUPT TIME.
1268X *
1269X *      IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
1270X *      A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE OF THE MESSAGE.
1271X *
1272X *      This routine modified to accomodate H8-4 ports by G.Chandler, 1-SEP-78.
1273X *      This routine assumes that the ports have been previously initialized,
1274X *      and that S.CDB has been previously initialized.
1275X *
1276X *      ENTRY      (RET) = TEXT
1277X *      EXIT      TO (RET+LENGTH)
1278X *      USES      A,F
1279X
1280X
045.055     343     1281X $TYPET XTHL      (HL) = TEXT ADDRESS
045.056     315 063 045 1282X      CALL      $TYPET.      TYPE IT
045.061     343     1283X      XTHL
045.062     311     1284X      RET
1285X
045.063     176     1286X $TYPET. MOV      A,M
045.064     346 177     1287X      ANI      1770
045.066     304 112 045 1288X      CNZ      $TYPEC.      IF NOT CRLF
045.071     247     1289X      ANA      A
045.072     314 103 045 1290X      CZ      $TYPET1      IS CRLF
045.075     276     1291X      CMP      H
045.076     043     1292X      INX      H
045.077     300     1293X      RNE
045.100     303 063 045 1294X      JMP      $TYPET.      WAS 200 BIT SET
1295X
1296X *      TYPE CRLF
1297X
045.103     315 055 045 1298X $TYPET1 CALL      $TYPET
045.106     015 212     1299X      DB      CR,LF+2000
045.110     257     1300X      XRA      A      RESTORE (A)
045.111     311     1301X      RET
  
```

```

1303X **      $TYPEC. - TYPE SINGLE CHARACTER.
1304X *
1305X *      IF CR, PADD WITH 4 ZERO BYTES
1306X *
1307X *      ENTRY (A) = CHARACTER
1308X *      EXIT (A) = CHARACTER
1309X *      USES A,F
1310X
1311X
045,112 365 1312X $TYPEC PUSH PSW SAVE CHAR
045,113 072 343 040 1313X LDA S,CDB
045,114 376 001 1314X CPI CDB,H84
045,120 312 140 045 1315X JZ TYPEC2 IF H8-4 PORT
1316X
1317X *      HAVE 8251 PORT FOR CONSOLE
1318X
045,123 333 373 1319X TYPEC1 IN SC,UART+USR
045,125 346 001 1320X ANI USR,TXR
045,127 312 123 045 1321X JZ TYPEC1 NOT READY
045,132 361 1322X POP PSW
045,133 323 372 1323X OUT SC,UART+UDR
045,135 303 152 045 1324X JMP TYPEC3
1325X
1326X *      HAVE 8250 PORT FOR CONSOLE
1327X
045,140 333 355 1328X TYPEC2 IN SC,ACE+UR,LSR
045,142 346 040 1329X ANI UC,THE
045,144 312 140 045 1330X JZ TYPEC2 NOT READY
045,147 361 1331X POP PSW
045,150 323 350 1332X OUT SC,ACE+UR,THR
1333X
045,152 376 015 1334X TYPEC3 CPI CR
045,154 300 1335X RNE NOT CR
1336X
1337X *      IS CR, PADD 4 TIMES
1338X
045,155 076 004 1339X MVI A,4
045,157 365 1340X TYPEC4 PUSH PSW
045,160 257 1341X XRA A
045,161 315 112 045 1342X CALL $TYPEC.
045,164 361 1343X POP PSW
045,165 075 1344X DCR A
045,166 302 157 045 1345X JNZ TYPEC4
045,171 076 015 1346X MVI A,CR
045,173 311 1347X RET
045,174 1348X XTEXT MOVE

```

```

1350X **      *MOVE - MOVE DATA
1351X *
1352X *      *MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
1353X *      IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
1354X *      FIRST TO LAST.
1355X *
1356X *      IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
1357X *      LAST TO FIRST.
1358X *
1359X *      THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
1360X *
1361X *      ENTRY  (BC) = COUNT
1362X *            (DE) = FROM
1363X *            (HL) = TO
1364X *      EXIT  MOVED
1365X *            (DE) = ADDRESS OF NEXT FROM BYTE
1366X *            (HL) = ADDRESS OF NEXT *TO* BYTE
1367X *      'C' CLEAR
1368X *      USES  ALL
1369X
1370X
030.252      1371X *MOVE EQU 30252A      IN H17 ROM
045.174      1372      XTEXT COMP
  
```

```

1374X **      *COMP - COMPARE TWO CHARACTER STRINGS.
1375X *
1376X *      *COMP COMPARES TWO BYTE STRINGS.
1377X *
1378X *      ENTRY  (C) = COMPARE COUNT
1379X *            (DE) = FWA OF STRING #1
1380X *            (HL) = FWA OF STRING #2
1381X *      EXIT  'Z' CLEAR, IS MIS-MATCH
1382X *            (C) = LENGTH REMAINING
1383X *            (DE) = ADDRESS OF MISMATCH IN STRING#1
1384X *            (HL) = ADDRESS OF MISMATCH IN STRING #2
1385X *      'C' SET, HAVE MATCH
1386X *            (C) = 0
1387X *            (DE) = (DE) + (0C)
1388X *            (HL) = (HL) + (0C)
1389X *      USES  A,F,C,D,E,H,L
1390X
1391X
030.060      1392X *COMP EQU 30040A      IN H17 ROM
045.174      1393      XTEXT DADA2
  
```

```

1395X **      $DADA. - ADD (0,A) TO (H,L)
1396X *
1397X *      ENTRY  NONE
1398X *      EXIT   (HL) = (HL) + (0A)
1399X *      USES   A,F,H,L
1400X
1401X
030.101      1402X $DADA. EQU 30101A      IN H17 ROM
1403

1405 **      CSC - COMPUTE SECTOR CRC.
1406 *
1407 *      CSC IS CALLED TO COMPUTE THE CRC OVER A SECTOR'S WORTH OF
1408 *      DATA.
1409 *
1410 *      ENTRY  (HL) = CURRENT CRC VALUE
1411 *            BUFF = BUFFER WITH SECTORS OF DATA
1412 *            (A) = INDEX (0 TO 9) OF SECTOR IN BUFF
1413 *      EXIT  (HL) = UPDATED CRC VALUE
1414 *      USES  ALL
1415
045.174 353      1416
045.175 041 010 052 1417 CSC XCHG      (DE) = CRC VALUE
045.200 204      1418 LXI      H,BUFF
045.201 147      1419 ADD      H
045.202 001 000 001 1420 MOV      H,A      (HL) = ADDRESS OF SECTOR
045.205 315 373 044 1421 LXI      B,256      (BC) = COUNT
045.210 353      1422 CALL   $BCRC      BLOCK CRC IT
045.211 311      1423 XCHG      (HL) = RESULTANT CRC
1424 RET

1426 **      LDE - LOCATE DIRECTORY ENTRY.
1427 *
1428 *      LDE LOCATES A DIRECTORY ENTRY CORRESPONDING TO THE AIO.DIR ENTRY.
1429 *
1430 *      ENTRY  (BC) = NUMBER OF CHARACTERS TO MATCH ON
1431 *      EXIT  'C' CLEAR IF FOUND
1432 *            AIO.DES SETUP
1433 *            (HL) = ADDRESS OF DIRECTORY ENTRY IN BUFF
1434 *            'C' SET IF NOT FOUND
1435 *            (A) = CODE
1436 *      USES  ALL
1437
1438
1439
1440 **      ENTRY FOR (HL) = SECTOR NUMBER TO START WITH
1441
045.212 305      1442 LDE..  PUSH  B      SAVE COUNT
045.213 001 000 002 1443 LXI      B,512
045.216 021 010 052 1444 LXI      D,BUFF

```

INIT - INITIALIZE DISK  
SUBROUTINES

HEATH HBASH V1.4 01/20/78

PAGE 31

LDE

16:01:22 16-MAY-80

```

045.221 042 055 041 1445 SHLD AIO.DES ASSUME WILL FIND IN THIS BLOCK
045.224 315 276 045 1446 CALL READD READ FRM DEVICE
045.227 301 1447 POP B RESTORE (BC)
1448
1449 * SCAN SECTOR FOR INFO
1450
045.230 041 010 052 1451 LXI H,DIS.ENT+BUFF
1452
1453 * COMPARE
1454
045.233 021 062 041 1455 LDE3 LXI D,AIO.DIR
045.236 305 1456 PUSH B SAVE COPY OF (BC)
045.237 345 1457 PUSH H SAVE ADDRESS
045.240 315 060 030 1458 CALL $COMP COMPARE
045.243 341 1459 POP H
045.244 301 1460 POP B (BC) = COMPARE COUNT
045.245 310 1461 RE GOT MATCH
045.246 021 027 000 1462 LXI D,DIRELEN MISSED, SCAN TO NEXT ENTRY
045.251 031 1463 DAD D
045.252 176 1464 MOV A,M
045.253 247 1465 ANA A
045.254 302 233 045 1466 JNZ LDE3 MORE IN SECTOR
1467
1468 * DIDNT FIND IT IN THIS SECTOR, TRY NEXT
1469
045.257 052 006 054 1470 LHLD DIS.LNK+BUFF
045.262 042 055 041 1471 SHLD AIO.DES SET POSSIBLE SECTOR INDEX
045.265 174 1472 MOV A,H
045.266 265 1473 ORA L
045.267 302 212 045 1474 JNZ LDE, HAVE MORE SECTORS
045.272 076 014 1475 MVI A,EC.FNF FILE NOT FOUND
045.274 067 1476 STC
045.275 311 1477 RET

```

```

1479 ** READD - READ DISK.
1480 *
1481 * READD CALLS THE SYSTEM DEVICE DRIVER FOR A
1482 * READ OPERATION.
1483 *
1484 * IF AN ERROR OCCURS, A MESSAGE IS PRINTED, AND THE
1485 * ROOT OPERATION RESTARTS.
1486 *
1487 * ENTRY REGISTERS SET FOR READ
1488 * EXIT FROM SYDD
1489 * USES ALL
1490
1491
045.276 076 000 1492 READD MVI A,DC.REA
045.300 315 130 040 1493 CALL SYDD ISSUE READ
045.303 320 1494 RNC ALL OK
1495
1496 * READ ERROR
1497

```

```

045.304 315 055 045 1498 READERR CALL $TYPET
045.307 000 000 007 1499 DB 0,0,BELL,'?00 DISK READ ERROR DURING BOOT.',0
045.353 007 040 102 1500 DB BELL,' BOOT RESTARTED.',2000
045.375 303 000 030 1501 JMP ROMBOOT
    
```

```

1503 ** FCU - FIND CONSOLE USART.
1504 *
1505 * FCU FINDS AND CONFIGURES THE CONSOLE USART.
1506 *
1507 * ENTRY NONE
1508 * EXIT NONE
1509 * USES A,F,(BC),(HL)
1510
    
```

```

1511
046.000 257 1512 FCU XRA A
046.001 323 351 1513 OUT SC,ACE+UR,IER OFF INTERRUPTS
046.003 323 373 1514 OUT SC,UART+USR OFF INTERRUPTS
1515
    
```

```

1516 * SEE IF WE HAVE AN 8250
1517
046.005 076 003 1518 MVI A,UC,8BW
046.007 323 353 1519 OUT SC,ACE+UR,LCR
046.011 333 353 1520 IN SC,ACE+UR,LCR
046.013 376 003 1521 CPI UC,8BW SEE IF UNCHANGED
046.015 076 000 1522 MVI A,CDB,H85
046.017 041 000 000 1523 LXI H,0
046.022 302 050 046 1524 JNE FCU1 IS 8251
046.025 315 324 046 1525 CALL ABR AUTO SET BAUD RATE
046.030 174 1526 MOV A,H
046.031 346 200 1527 ANI 10000000B
046.033 312 046 046 1528 JZ FCU0
046.036 072 327 040 1529 LDA S,CONTY
046.041 366 010 1530 ORI CTP,2SB
046.043 062 327 040 1531 STA S,CONTY SET TWO STOP BITS
046.046 076 001 1532 FCU0 MVI A,CDB,H84
1533
    
```

```

1534 * HAVE TYPE AND BAUDRATE.
1535 * (A) = S,CDB VALUE
1536 * (HL) = BAUD RATE (0 IF 8251)
1537
    
```

```

046.050 042 344 040 1538 FCU1 SHLD S,BAUD
046.053 062 343 040 1539 STA S,CDB
046.056 315 252 047 1540 CALL SCU
046.061 311 1541 RET
1542
    
```



```

1544 ** TTDD - TYPE DECIMAL DIGITS.
1545 *
1546 * TTDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.
1547 *
1548 * ENTRY (D,E) = VALUE
1549 * (A) = DIGIT COUNT
1550 * EXIT VALUE TYPED.
1551 * USES A,B,C,F
1552
1553
046.062 076 005 1554 TTDD. MVI A,5
046.064 345 1555 TTDD PUSH H
046.065 365 1556 TTDD1 PUSH PSW
046.066 041 131 046 1557 LXI H,TTDDA-2
046.071 007 1558 RLC (A) = DIGIT NUMBER*2
046.072 315 101 030 1559 CALL $DADA.
046.075 176 1560 MOV A,H
046.076 043 1561 INX H
046.077 146 1562 MOV H,H
046.100 157 1563 MOV L,A (HL) = MULTIPLE OF 10
046.101 353 1564 XCHG (DE) = DIVISOR, (HL) = VALUE
046.102 076 377 1565 MVI A,377D
046.104 031 1566 TTDD2 DAD D
046.105 074 1567 INR A
046.106 332 104 046 1568 JC TTDD2 IF MORE TO GO
046.111 306 060 1569 ADI '0'
046.113 315 112 045 1570 CALL $TYPEC. TYPE DIGIT
046.116 175 1571 MOV A,L
046.117 223 1572 SUB E
046.120 137 1573 MOV E,A REMOVE EXTRA SUBTRACTION
046.121 174 1574 MOV A,H
046.122 232 1575 SBB D
046.123 127 1576 MOV D,A
046.124 361 1577 POP PSW
046.125 075 1578 DCR A
046.126 302 065 046 1579 JNZ TTDD1 IF MORE DIGITS
046.131 341 1580 POP H
046.132 311 1581 RET EXIT
1582
046.133 1583 TTDDA EQU *
046.133 377 377 1584 DW -1
046.135 366 377 1585 DW -10
046.137 234 377 1586 DW -100
046.141 030 374 1587 DW -1000
046.143 360 330 1588 DW -10000
  
```

```

1590 ** TTDDCR - TYPE DECIMAL DIGITS, THEN CRLF.
1591 *
1592 * ENTRY SAME AS TTDD
1593 * EXIT SAME AS TD
1594 * USES SAME AS TTDD
1595
1596
  
```

```

046.145 315 064 046 1597 TTDDCR CALL TTDD
046.150 315 055 045 1598 CALL $TYPET
046.153 200 1599 DB 200Q
046.154 311 1600 RET

1602 ** MSD - MOUNT SYSTEM DISK.
1603 *
1604 * MSD MOUNTS THE SYSTEM DISK.
1605 *
1606 * 1) ABORT DRIVER
1607 * 2) READ BLABEL RECORD
1608 * 3) SET VOLUME NUMBER FOR DRIVER
1609 * 4) INITIALIZE DEVICE TABLE
1610 * 5) BUILD CRT
1611
1612
046.155 076 007 1613 MSD MVI A,DC,ABT
046.157 315 130 040 1614 CALL SYDD ABORT DRIVER
046.162 001 000 001 1615 LXI B,256
046.165 021 010 051 1616 LXI D,BLABEL
046.170 041 011 000 1617 LXI H,DDF.LAB
046.173 076 002 1618 MVI A,DC,RRR READ REGARDLESS
046.175 315 130 040 1619 CALL SYDD
046.200 334 304 045 1620 CC READERR BAD ERROR
1621
1622 * CALL DEVICE MOUNT ROUTINE
1623
046.203 072 010 051 1624 LBA BLABEL+LAB.SER
046.206 157 1625 MOV L,A
046.207 046 000 1626 MVI H,0 (HL) = SERIAL NUMBER
046.211 076 010 1627 MVI A,DC,MOU
046.213 315 130 040 1628 CALL SYDD MOUNT UNIT
046.216 334 304 045 1629 CC READERR BAD ERROR
046.221 311 1630 RET
1631
046.222 000 000 1632 CRCSUM DW 0 CRCSUM WORKAREA
1633
1634
046.224 1635 DB 64 PATCH AREA
1636
1637 ** ALL CODE FOLLOWING MAY BE OVERLAID BY THE HDOS.SYS
1638 * PROGRAM DURING BOOT.
1639
377.324 1640 ERRFL *-SB.ORG MUST BE BEFORE ORG ADDRESS
1641
046.324 1642 XTEXT ABR INCLUDE HERE TO BE OVERLAID

```

```

1644X **   ABR - AUTO BAUD RATE SELECTION.
1645X *
1646X *   ABR READS CHARACTERS FROM THE SYSTEM CONSOLE ACE UNTIL
1647X *   THE CURRENT BAUD RATE IS DETERMINED.
1648X *
1649X *   ENTRY  NONE
1650X *   EXIT   (HL) = BAUD RATE DIVISOR
1651X *   ACE SETUP WITH BAUD RATE, NO INTERRUPTS
1652X *   USES   ALL
1653X
1654X
046.324   1655X ABR   EQU   *
1656X
1657X *   INITIALIZE LED DISPLAY FOR PROMPT
1658X
046.324   072 010 040 1659X ABR0.1 LDA   .MFLAG           /79.01.GC/
046.327   365                1660X   PUSH  FSW           /79.01.GC/
046.330   366 002                1661X   ORI   UO,DDU       /79.01.GC/
046.332   082 010 040 1662X   STA   .MFLAG           /79.01.GC/
046.335   001 011 000 1663X   LXI   B,9
046.340   021 155 047 1664X   LXI   D,ABR.A
046.343   041 013 040 1665X   LXI   H,.ALEDS
046.346   315 252 030 1666X   CALL  $MOVE
046.351   021 013 040 1667X   LXI   D,.ALEDS
046.354   076 144                1668X   MVI   A,100
046.356   315 140 002 1669X   CALL  .HORN
1670X
046.361   041 166 047 1671X   LXI   H,TABLE
1672X
046.364   257                1673X ABR0.3 XRA   A           /79.01.GC/
046.365   323 351                1674X   OUT  SC.ACE+UR.IER  /79.01.GC/
046.367   076 020                1675X   MVI  A,UC,LOD       /79.01.GC/
046.371   323 354                1676X   OUT  SC.ACE+UR.MCR  SET LOOP BACK /79.01.GC/
046.373   076 200                1677X   MVI  A,UC,DLA
046.375   323 353                1678X   OUT  SC.ACE+UR.LCR  LINE CONTROL ACCESS
046.377   176                1679X   MOV  A,M
047.000   043                1680X   INX  H
047.001   323 350                1681X   OUT  SC.ACE+UR.DLL  DIVISOR LEAST SIGNIFICANT
047.003   176                1682X   MOV  A,M
047.004   346 177                1683X   ANI  1770          CLEAR STOP BITS FLAG
047.006   323 351                1684X   OUT  SC.ACE+UR.DLM  DIVISOR MOST SIGNIFICANT
047.010   276                1685X   CMP  M             SEE IF 2 STOP BITS
047.011   043                1686X   INX  H
047.012   076 003                1687X   MVI  A,UC,8BW      ASSUME 8 BIT WORDS, 1 STOP
047.014   312 021 047 1688X   JE   ABR0.5
047.017   076 007                1689X   MVI  A,UC,8BW+UC.2SB SET 2 STOP BITS
047.021   323 353                1690X ABR0.5 OUT  SC.ACE+UR.LCR  LINE CONTROL ACCESS
047.023   076 156                1691X   MVI  A,AC,DLY
047.025   315 053 000 1692X   CALL .DLY          WAIT FOR B250 TO SETTLE /79.01.GC/
047.030   333 354                1693X   IN   SC.ACE+UR.MCR /79.01.GC/
047.032   346 357                1694X   ANI  3770-UC,LOD  /79.01.GC/
047.034   323 354                1695X   OUT  SC.ACE+UR.MCR TURN OFF LOOP /79.01.GC/
1696X
1697X *   WAIT FOR CHARACTER TO BE HIT
1698X
047.036   333 350                1699X ABR0  IN   SC.ACE+UR.RBR GOBBLE OVERRUN

```

```

047.040 333 355 1700X ABR1 IN SC.ACE+UR.LSR
000.000 1701X ERRNZ UC.DR-2
047.042 037 1702X RAR
047.043 037 1703X RAR
047.044 332 036 047 1704X JC ABR0 OVERRUN
047.047 027 1705X RAL
047.050 027 1706X RAL
047.051 346 015 1707X ANI UC.DR+UC.PE+UC.FE
047.053 312 040 047 1708X JZ ABR1 NOTHING YET
047.056 365 1709X PUSH PSM
047.057 032 1710X LDAX D ECHO ' ' AS ' ' ON LEDS
047.060 346 177 1711X ANI 01111111B TURN ON ' '
047.062 022 1712X STAX D
047.063 023 1713X INX D
047.064 361 1714X POP PSM
047.065 346 010 1715X ANI UC.FE
047.067 302 107 047 1716X JNZ ABR3 USER IS SLOWER THAN THIS
047.072 333 350 1717X IN SC.ACE+UR.RBR GET DATA
047.074 346 177 1718X ANI 1770 TRIM
047.076 376 040 1719X CPI ' '
047.100 312 124 047 1720X JE ABR5
1721X
1722X * USER IS FASTER THAN WE ARE. FOLLOW FASTER LINKAGE
1723X
047.103 156 1724X ABR2 MOV L,M FOLLOW LINK
047.104 303 364 046 1725X JMP ABR0.3 TRY AGAIN
1726X
1727X * USER IS SLOWER THAN WE ARE. READ NEXT CHARACTER
1728X
047.107 076 067 1729X ABR3 MVI A,110/2
047.111 315 053 000 1730X CALL .DLY WAIT FOR THINGS TO SETTLE OUT
047.114 333 350 1731X IN SC.ACE+UR.RBR
047.116 333 355 1732X IN SC.ACE+UR.LSR
047.120 043 1733X INX H
047.121 303 103 047 1734X JMP ABR2
1735X
1736X * FOUND THE BAUD RATE. RETURN WITH ANSWERS
1737X
047.124 021 013 040 1738X ABR5 LXI D,.ALED5 BLANK DISPLAY
047.127 006 011 1739X MVI B,9
047.131 076 377 1740X MVI A,377A
047.133 022 1741X ABR5.1 STAX D
047.134 023 1742X INX D
047.135 005 1743X DCR B
047.136 302 133 047 1744X JNZ ABR5.1
047.141 053 1745X DCX H
047.142 126 1746X MOV B,M
047.143 053 1747X DCX H
047.144 136 1748X MOV E,M
047.145 353 1749X XCHG (HL) = BAUD RATE
047.146 333 350 1750X IN SC.ACE+UR.RBR GOBBLE THE GARBAGE /79.01.GC/
047.150 361 1751X POP PSM /79.01.GC/
047.151 062 010 040 1752X STA .MFLAG /79.01.GC/
047.154 311 1753X RET
1754X
047.155 244 230 220 1755X ABR.A DR 2440,2300,2200,2150,2140,3770,3770,3770,3770 'SPACE' FOR LEDS

```

1756X

1758X \*\* BAUD RATE SELECTION TREE.

```

1759X *
047.166 1760X TABLE DS 0 START OF BAUD TABLE
1761X
047.166 060 000 1762X DW 000060A 2400 BAUD
047.170 172 1763X DB #T9600 USER IS FASTER
047.171 176 1764X DB #T600 USER IS SLOWER
1765X
1766X * 2ND TRY GROUPS
1767X
047.172 014 000 1768X T9600 DW 000014A 9600 BAUD
047.174 202 1769X DB #T19200 USER IS FASTER
047.175 206 1770X DB #T4800 USER IS SLOWER
1771X
047.176 300 000 1772X T600 DW 000300A 600 BAUD
047.200 212 1773X DB #T1200 USER IS FASTER
047.201 216 1774X DB #T300 USER IS SLOWER
1775X
1776X * 3RD TRY GROUPS
1777X
047.202 006 000 1778X T19200 DW 000006A 19200 BAUD
047.204 166 1779X DB #TABLE USER IS FASTER, MUST BE SCREWED UP
047.205 166 1780X DB #TABLE USER IS SLOWER, MUST BE SCREWED UP
1781X
047.206 030 000 1782X T4800 DW 000030A 4800 BAUD
047.210 222 1783X DB #T7200 USER IS FASTER
047.211 226 1784X DB #T3600 USER IS SLOWER
1785X
047.212 140 000 1786X T1200 DW 000140A 1200 BAUD
047.214 232 1787X DB #T1800 USER IS FASTER
047.215 166 1788X DB #TABLE USER IS SLOWER, MUST BE SCREWED UP
1789X
047.216 200 001 1790X T300 DW 001200A 300 BAUD
047.220 166 1791X DB #TABLE USER IS FASTER, MUST BE SCREWED UP
047.221 236 1792X DB #T110 USER IS SLOWER
1793X
1794X * 4TH TRY GROUPS
1795X
047.222 020 000 1796X T7200 DW 000020A 7200 BAUD
047.224 166 1797X DB #TABLE USER IS FASTER, MUST BE SCREWED UP
047.225 166 1798X DB #TABLE USER IS SLOWER, MUST BE SCREWED UP
1799X
047.226 040 000 1800X T3600 DW 000040A 3600 BAUD
047.230 166 1801X DB #TABLE USER IS FASTER, MUST BE SCREWED UP
047.231 166 1802X DB #TABLE USER IS SLOWER, MUST BE SCREWED UP
1803X
047.232 100 000 1804X T1800 DW 000100A 1800 BAUD
047.234 166 1805X DB #TABLE USER IS FASTER, MUST BE SCREWED UP
047.235 166 1806X DB #TABLE USER IS SLOWER, MUST BE SCREWED UP
1807X
047.236 027 204 1808X T110 DW 204027A 110 BAUD
    
```

## SUBROUTINES

## TABLE

16:01:34 16-MAY-80

047.240	242	1809X	DB	#T150	USER IS FASTER
047.241	246	1810X	DB	#T75	USER IS SLOWER
		1811X			
		1812X *	STH	TRY GROUPS	
		1813X			
047.242	000 003	1814X	T150	DW	003000A 150 BAUD
047.244	166	1815X	DB	#TABLE	USER IS FASTER, MUST BE SCREWED UP
047.245	166	1816X	DB	#TABLE	USER IS SLOWER, MUST BE SCREWED UP
		1817X			
047.246	000 006	1818X	T75	DW	006000A 75 BAUD
047.250	166	1819X	DB	#TABLE	USER IS FASTER, MUST BE SCREWED UP
047.251	166	1820X	DB	#TABLE	USER IS SLOWER, MUST BE SCREWED UP
		1821X			
000.047		1822X	SET	*/256	
000.000		1823X	ERRNZ	TABLE/256-	MUST BE IN SAME PAGE
047.252		1824	XTEXT	SCU	

1826X \*\* SCU - SETUP CONSOLE USART.

1827X \*

1828X \* SCU CONFIGURES THE CONSOLE USART.

1829X \*

1830X \* IF 8250

1831X \* THEN PORT = 372-30

1832X \* ELSE PORT = 340-70

1833X \*

1834X \*

1835X \* ENTRY NONE

1836X \* EXIT NONE

1837X \* USES A,F,(BC),(HL)

1838X

1839X

047.252 072 343 040 1840X SCU LDA S,CDB

047.255 376 001 1841X CPI CDB,H84

047.257 312 322 047 1842X JZ SCU1 IF 8250

1843X

1844X \* PRESET 8251

1845X

047.262 076 201 1846X MVI A,2010

047.264 323 373 1847X OUT SC,UART+USR GET USART IN KNOWN STATE

047.266 323 373 1848X OUT SC,UART+USR

047.270 323 373 1849X OUT SC,UART+USR

047.272 323 373 1850X OUT SC,UART+USR

047.274 076 100 1851X MVI A,UCI,IR RESET

047.276 323 373 1852X OUT SC,UART+USR

047.300 072 327 040 1853X LDA S,CONY

047.303 346 010 1854X ANI CTF,25B

000.000 1855X ERRNZ CTF,25B\*16+UMI,1E-UMI,2B

047.305 007 1856X RLC

047.306 007 1857X RLC

047.307 007 1858X RLC

047.310 007 1859X RLC

047.311 366 116 1860X ORI UMI,1B+UMI,1B+UMI,16X

047.313 323 373 1861X OUT SC,UART+USR

```

047.315 076 025 1862X MVI A,UCI.ER+UCI.RE+UCI.TE
047.317 323 373 1863X OUT SC.UART+USR
047.321 311 1864X RET
1865X
1866X * IS 8250
1867X
047.322 333 355 1868X SCU1 IN SC.ACE+UR.LSR /80.01.GC/
047.324 346 100 1869X ANI UC.TSE CHECK FOR SHIFT EMPTY /80.01.GC/
047.326 312 322 047 1870X JZ SCU1 /80.01.GC/
1871X
047.331 257 1872X XRA A
047.332 323 351 1873X OUT SC.ACE+UR.IER TURN OFF ANY INTERRUPTS /79.01.GC/
047.334 076 020 1874X MVI A,UC.L00 /79.01.GC/
047.336 323 354 1875X OUT SC.ACE+UR.MCR /79.01.GC/
047.340 052 344 040 1876X LHLB S.BAUD
047.343 076 200 1877X MVI A,UC.DLA
047.345 323 353 1878X OUT SC.ACE+UR.LCR ACCESS DIVISOR LATCHES
047.347 175 1879X MOV A,L
047.350 323 350 1880X OUT SC.ACE+UR.DLL SET LEAST SIGNIFICANT
047.352 174 1881X MOV A,H
047.353 346 177 1882X ANI 1770 TRIM STOP BITS
047.355 323 351 1883X OUT SC.ACE+UR.DLM SET MOST SIGNIFICANT
047.357 072 327 040 1884X LDA S.CONTY
047.362 346 010 1885X ANI CTP.2SB
047.364 017 1886X RRC
000.000 1887X ERRNZ CTP.2SB/2-UC.2SB
000.000 1888X ERRNZ UC.2SB-4 (A) = UC.2SB IF 2 STOP BITS
047.365 346 003 1889X ORI UC.8BW 8 BIT WORDS
047.367 323 353 1890X OUT SC.ACE+UR.LCR
047.371 076 156 1891X MVI A,AC.DLY /79.01.GC/
047.373 315 053 000 1892X CALL .DLY /79.01.GC/
047.374 333 350 1893X IN SC.ACE+UR.RBR GOBBLE ANY TRASH /79.01.GC/
050.000 333 354 1894X IN SC.ACE+UR.MCR /79.01.GC/
050.002 346 357 1895X ANI 3770-UC.L00 /79.01.GC/
050.004 323 354 1896X OUT SC.ACE+UR.MCR /79.01.GC/
050.006 311 1897X RET
050.007 1898X XTEXT MCU INCLUDED HERE TO BE USED AT BOOT-UP
    
```

```

1900X ** MCU - MAP LOWER CASE TO UPPER CASE.
1901X *
1902X * MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
1903X * CASE.
1904X *
1905X * ENTRY (A) = CHARACTER
1906X * EXIT (A) = CHARACTER RESULT
1907X * USES A,F
1908X
1909X
050.007 376 141 1910X $MCU CPI 'a'
050.011 330 1911X RC NOT LOWER CASE
050.012 376 173 1912X CPI 'z'+1
050.014 320 1913X RNC NOT LOWER CASE
050.015 326 040 1914X SUI 'a'-'A'
    
```

050.017 311 1915X RET



```

1918 ** THIS ROUTINE IS OVERLAID BY THE HDOS.SYS PROGRAM DURING BOOT.
1919
1920 ** CDC - COMPUTE DISK CHECKSUMS.
1921 *
1922 * CDC READS EACH DISK SECTOR (EXCEPT FOR TRACK 0) TO SEE IF
1923 * THERE ARE ANY PROBLEMS.
1924 *
1925 * THE CHECKSUM OF EACH SECTOR IS PRINTED, TOGETHER
1926 * WITH THE TOTAL CHECKSUM FOR EACH TRACK, AND THE VOLUME CHECKSUM.
1927 *
1928
1929
050.020 041 000 000 1930 CDC LXI H,0
050.023 042 000 051 1931 SHLD CDCA VOLUME
050.026 042 002 051 1932 SHLD CDCB TRACK
050.031 044 1933 INR H
050.032 042 006 051 1934 SHLD CDCE SET TRACK AND SECTOR NUMBERS
050.035 041 012 000 1935 LXI H,10
050.040 042 004 051 1936 SHLD CDCD SET FIRST SECTOR NUMBER
1937
1938 * READ TRACK
1939
050.043 052 004 051 1940 CDC1 LHL D,CDCD
050.046 021 010 052 1941 LXI D,BUFF
050.051 001 000 012 1942 LXI R,10*256
050.054 076 000 1943 MVI A,DC.REA
050.056 315 130 040 1944 CALL SYDD READ DISK
050.061 322 110 050 1945 JNC CDC2 NO ERROR
1946
1947 * READ ERROR
1948
050.064 315 055 045 1949 CALL $TYPET
050.067 007 077 060 1950 DB BELL,'?00 * ERROR * ',BELL,2000
1951
1952 * CRC SECTOR
1953
050.110 315 055 045 1954 CDC2 CALL $TYPET
050.113 123 105 103 1955 DB 'SECTOR','+2000
050.122 052 004 051 1956 LHL CDCD
050.125 353 1957 XCHG
050.126 076 003 1958 MVI A,3
050.130 315 064 046 1959 CALL TTDD
050.133 315 055 045 1960 CALL $TYPET
050.136 040 075 240 1961 DB '=',''+2000
050.141 052 002 051 1962 LHL CDCB
050.144 072 006 051 1963 LDA CDCE
050.147 315 174 045 1964 CALL CSC COMPUTE TRACK CRC
050.152 042 002 051 1965 SHLD CDCB
050.155 052 000 051 1966 LHL CDCA
050.160 072 006 051 1967 LDA CDCE
050.163 315 174 045 1968 CALL CSC COMPUTE VOLUME CRC
050.166 042 000 051 1969 SHLD CDCA
050.171 041 000 000 1970 LXI H,0
050.174 072 006 051 1971 LDA CDCE
050.177 315 174 045 1972 CALL CSC COMPUTE SECTOR CRC
050.202 353 1973 XCHG

```

```

050.203 076 005      1974      MVI      A,5
050.205 315 145 046 1975      CALL     TTDDCR      TYPE SECTOR CHECKSUM
050.210 052 004 051 1976      LHLD    CDCD        (HL) = SECTOR COUNT
050.213 043          1977      INX     H
050.214 042 004 051 1978      SHLD    CDCD
050.217 041 006 051 1979      LXI    H,CDCD
050.222 315 271 044 1980      CALL     $ICTT,      CHECK FOR CHARACTER
050.225 332 240 050 1981      JC      CDC3        NO CHARACTER WAS HIT
050.230 315 317 044 1982      CALL     $ICTT.,    GET CHARACTER
050.233 376 003      1983      CPI    CTLC
050.235 312 367 050 1984      JE      CDC4        CTL-C HIT
050.240 064          1985      CDC3  INR    M      COUNT SECTOR
050.241 076 012      1986      MVI    A,10
050.243 226          1987      SUB    M
050.244 302 110 050 1988      JNE    CDC2        MORE ON TRACK
1989
1990 *      HAVE COMPLETED TRACK
1991
050.247 167          1992      MOV    M,A
050.250 043          1993      INX    H
050.251 064          1994      INR    M      COUNT TRACK
050.252 315 055 045 1995      CALL     $TYPET
050.255 124 122 101 1996      DB     'TRACK TOTAL ',2000+' '
050.272 052 002 051 1997      LHLD    CDCR
050.275 353          1998      XCHG
050.276 041 000 000 1999      LXI    H,0
050.301 042 002 051 2000      SHLD    CDCR      RESET COUNT
050.304 076 005      2001      MVI    A,5
050.306 315 145 046 2002      CALL     TTDDCR
2003
050.311 052 004 051 2004      LHLD    CDCD
050.314 001 160 376 2005      LXI    B,-400
050.317 011          2006      DAD    B
050.320 322 043 050 2007      JNC    CDC1        MORE TO GO
2008
2009 *      ALL DONE
2010
050.323 315 055 045 2011      CALL     $TYPET
050.326 000 040 126 2012      DB     0,' VOLUME TOTAL CRC = ',2000+' '
050.353 052 000 051 2013      LHLD    CDCA
050.356 353          2014      XCHG
050.357 076 005      2015      MVI    A,5
050.361 315 145 046 2016      CALL     TTDDCR
050.364 303 211 042 2017      JMP     SOB00TX
2018
2019 *      CTL-C STRUCK
2020
050.367 315 055 045 2021      CDC4  CALL     $TYPET
050.372 136 103 200 2022      DB     'C',2000
050.375 303 211 042 2023      JMP     SOB00TX
2024
051.000 000 000      2025      CDCA  DW     0      VOLUME CRC
051.002 000 000      2026      CDCR  DW     0      TRACK CRC
051.004 000 000      2027      CDCD  DW     0      SECTOR NUMBER
051.006 000          2028      CDCE  DB     0      SECTOR NUMBER
051.007 000          2029      DB     0      TRACK NUMBER

```

007.000	2032	SOBOOTL	EQU	*-SOBOOT+255/256*256	SOBOOT LENGTH IN SECTORS *256
	2033				
051.010	2034	BLABEL	EQU	*	LABEL BUFFER STARTS HERE
052.010	2035	BUFF	EQU	BLABEL+256	10 SECTOR BUFFER
064.010	2036	BDDEND	EQU	10*256+BUFF	BUFFER ENDS HERE

```

2039 ***      INIT - MAIN INITIALIZE LOOP.
2040 *
2041
2042          LOF      C          RESTORE LISTING CONTROL
2043
051.010          2044 INIT    EQU      *
051.010 377 011    2045      DB      SYSCALL,VERS
051.012 332 125 051 2046      JC      INIT1    NO ,VERS SYSTEM CALL
051.015 376 026    2047      CFI     VERS
051.017 302 125 051 2048      JNZ     INIT1    VERSIONS DO NOT MATCH
051.022 076 377    2049      MVI     A,3770
051.024 377 046    2050      DB      SYSCALL,CLOSE  CLOSE THE CHANNEL WE CAME IN ON
051.026 257        2051      XRA     A
051.027 062 326 040 2052      STA     S,CSLMD  SETUP CONSOLE MODE
051.032 315 240 062 2053      CALL    *DOS     DISMOUNT OPERATING SYSTEM
051.035 332 136 051 2054      JC      ERROR   ERROR
051.040 315 172 051 2055      CALL    PRS     PRESET SYSTEM, INITIAL MESSAGES
051.043 041 000 072 2056      LXI     H,RMEML
051.046 377 052    2057      DB      SYSCALL,SETTP  SET TOP
051.050 332 136 051 2058      JC      ERROR
051.053 303 064 051 2059      JMP     INIT0    DONT ASK IF MORE FOR THE FIRST TIME
2060
2061 *          RESTART HERE TO INIT NEW DISK
2062
051.056 315 317 052 2063 RESTART CALL  AMW          ASK IF MORE WORK WANTED
051.061 302 133 051 2064      JNZ     EXIT     ALL DONE
051.064 061 200 042 2065 INIT0  LXI     SP,STACK
051.067 257        2066      XRA     A
051.070 062 061 041 2067      STA     AIO,UNI   SELECT UNIT 0
051.073 315 104 053 2068      CALL    RMI     REQUEST MEDIA INSERTION
051.076 076 007    2069      MVI     A,DC,ART
051.100 315 343 063 2070      CALL    SYDD.    RESET DISK
051.103 315 230 054 2071      CALL    AAL     ASK ABOUT LABEL
051.106 315 224 055 2072      CALL    GVI     GET VOLUME ID
051.111 315 101 056 2073      CALL    IDS     INIT DISK SURFACE
051.114 315 346 056 2074      CALL    GBL     GET BAD SECTOR LIST
051.117 315 013 060 2075      CALL    FDU     FORMAT VOLUME
051.122 303 056 051 2076      JMP     RESTART
2077
051.125 076 050    2078 INIT1  MVI     A,EC,NCV  NOT CORRECT VERSION
051.127 067        2079      STC
051.130 303 136 051 2080      JMP     ERROR
2081
2082 *          EXIT
2083
051.133 257        2084 EXIT  XRA     A
051.134 377 000    2085 EXIT.  DB      SYSCALL,EXIT  LET *HDOS* HANDLE THE EXIT PARAMETERS

```

ERROR

```
2087 ** ERROR = ERROR ENCOUNTERED.  
2088  
051.136 365 2089 ERROR PUSH PSW  
051.137 315 075 062 2090 CALL $CCO CLEAR CTL-D  
051.142 315 136 031 2091 CALL $TYPTX  
051.145 012 007 105 2092 DB NL,BELL,'ERROR - ',' '+2000  
051.160 381 2093 POP PSW  
051.161 046 012 2094 MVI H,NL  
051.163 377 057 2095 DB SYSCALL,ERROR  
051.165 076 001 2096 MVI A,1 FLAG ERROR  
051.167 303 134 051 2097 JMP EXIT.
```

```

2101 **      PRS - PRESET PROGRAM.
2102 *
2103 *      PRS PERFORMS ANY INITIALIZATION TASKS, AND INFORMS THE
2104 *      USER OF THE FACTS OF LIFE CONCERNING INITIALIZATION/.
2105 *
2106 *      ENTRY   NONE
2107 *      EXIT    TO SYSTEM IF USER CHICKENS OUT
2108 *             TO CALLER IF OK
2109 *      USES   ALL
2110
2111
2112 PRS      CALL   $CCO          CLEAR CTL-0
051.172 315 075 062 2112
2113      CALL   $TYPTX
051.175 315 136 031 2113
2114      DB     NL,TAB,TAB,TAB,' ','INIT'
051.200 012 011 011 2114
2115      DB     NL,TAB,TAB,TAB,'Version: ','VERS/16+'0',' ','VERS1000011118+'0'
051.214 012 011 011 2115
2116      DB     NL,TAB,TAB,' ','Issue: $50.05.00'
051.235 012 011 011 2116
2117      DB     NL
051.267 012          2117
2118      DB     NL,TAB,'This routine is used to initialize HDOS floppy'
051.270 012 011 124 2118
2119      DB     ' disks.'
051.351 040 144 151 2119
2120      DB     NL,'It is a stand-alone utility, and will destroy any'
051.360 012 111 144 2120
2121      DB     ' files on'
052.046 040 146 151 2121
2122      DB     NL,'the disks it initializes. Do not attempt to use this'
052.057 012 164 150 2122
2123      DB     ' program'
052.145 040 040 160 2123
2124      DB     NL,'until you have studied the appropriate manual.'
052.156 012 165 156 2124
2125      DB     ENL
052.235 212          2125
2126      CALL   $CCO          CLEAR CTL-0
052.236 315 075 062 2126
2127      CALL   $TYPTX
052.241 315 136 031 2127
2128      DB     NL,'Proceed (YES/NO) <NO> ','?' +2000
052.244 012 120 162 2128
2129      CALL   $ITL.
052.274 315 205 063 2129
2130      MOV    A,M
052.277 176          2130
2131      ANA   A
052.300 247          2131
2132      JZ    EXIT          WANTS TO EXIT
052.301 312 133 051 2132
2133      CFI   'N'
052.304 376 116      2133
2134      JE    EXIT          EXIT
052.306 312 133 051 2134
2135      CFI   'Y'
052.311 376 131      2135
2136      JNE   PRS
052.313 302 172 051 2136
2137      RET
052.316 311          2137
  
```

INIT - INITIALIZE DISK  
AMW - ASK FOR MORE WORK

HEATH HBASH V1.4 01/20/78  
16:01:49 16-MAY-80

PAGE 47

```
2141 ** AMW - ASK FOR MORE WORK.
2142 *
2143 * AMW SEES IF THE USER WANTS TO CONTINUE THE PROCESS.
2144 *
2145 * ENTRY NONE
2146 * EXIT 'Z' CLEAR IF NO MORE WORK
2147 * 'Z' SET IF MORE WORK
2148 * USES ALL
2149 *
2150
052.317 315 136 031 2151 AMW CALL $TYPTX
052.322 012 040 104 2152 DB NL,' Disk Initialization complete.',ENL
2153
052.362 315 075 062 2154 AMW1 CALL $CCO CLEAR CTL-D.
052.365 315 136 031 2155 CALL $TYPTX
052.370 012 104 157 2156 DB NL,' Do you have any more disks to initialize (YES/NO) <NO>.',',','+2000.
053.061 315 205 063 2157 CALL $ITL.
053.064 176 2158 MOV A,M (A) = REPLY
053.065 376 131 2159 CPI 'Y'
053.067 310 2160 RE IS YES.
053.070 247 2161 ANA A
053.071 312 101 053 2162 JZ AMW2 IS NO.
053.074 376 116 2163 CPI 'N'
053.076 302 362 052 2164 JNE AMW1 ASK AGAIN
053.101 366 001 2165 AMW2 ORI 1 ANSWER IS NO
053.103 311 2166 RET
```

```

2169 **      RMI - REQUEST MEDIA INSERTION.
2170 *
2171 *      RMI PROMPTS THE USR TO LOAD THE DESIRED MEDIA.
2172 *
2173 *      WHEN IN PLACE, RMI CHECKS THE HOLE TIMINGS
2174 *
2175 *      ENTRY  NONE
2176 *      EXIT   DONE
2177 *      USES   ALL
2178
2179
053.104 315 075 062 2180 RMI  CALL  $CCO
053.107 315 136 031 2181  CALL  $TYPTX
053.112 012 040 111 2182  DB    NL,' Insert the volume you wish to initialize into SY0:!'
053.177 012 040 162 2183  DB    NL,' remember, any data on this volume will be destroyed.'
053.265 012          2184  DB    NL
053.266 012 110 151 2185  DB    NL,' Hit RETURN when ready.'
053.315 012 122 145 2186  DB    NL,'Ready?', 'f2000
2187
053.325 315 213 063 2188  CALL  $ITL
2189
2190 *      GOT HIS OK. CHECK PATTERN.
2191
053.330 076 022      2192  MVI   A,DF,DS0+DF,MO
053.332 323 177      2193  OUT   DF,IC          ON DRIVE
053.334 315 124 054 2194  CALL  W1S          WAIT ONE SECOND
053.337 315 271 036 2195  CALL  R,WNH        WAIT FOR NO HOLE
053.342 315 235 036 2196  CALL  R,WHD        WAIT FOR HOLE DETECT
053.345 315 271 036 2197  CALL  R,WNH        WAIT FOR NO HOLE DETECT
2198
2199 *      CHECK INSERTED MEDIA
2200
053.350 315 136 054 2201  CALL  CIM          CHECK INSERTED MEDIA
053.353 320          2202  RNC          MEDIA IS GOOD
053.354 315 136 054 2203  CALL  CIM          GIVE MEDIA A SECOND CHECK
053.357 320          2204  RNC          WAS GOOD THE SECOND TIME
2205
2206 *      ERROR IN MEDIA FORMAT.
2207
053.360 315 136 031 2208  CALL  $TYPTX
053.363 012 007 040 2209  DB    NL,'BELL,' Wrong type of media, media inserted improperly, or
054.050 012 040 155 2210  DB    NL,' media damaged. Check it and try again.',ENL
054.121 303 104 053 2211  JMP   RMI

2213 **      W1S - WAIT ONE SECOND.
2214 *
2215 *      W1S IS CALLED TO DELAY ONE SECOND.
2216
2217
054.124 076 372      2218  W1S  MVI   A,250
054.126 315 303 035 2219  CALL  R,DLY
054.131 076 372      2220  MVI   A,250
054.133 303 303 035 2221  JMP   R,DLY          DELAY AND EXIT

```



```

2223 **      CIM      - CHECK INSERTED MEDIA
2224 *
2225 *      CIM CHECKS THE INSERTED MEDIA
2226 *
2227 *
2228 *      ENTRY      NONE
2229 *
2230 *      EXIT      (PSW) = 'C' CLEAR IF GOOD MEDIA
2231 *              = 'C' SET   IF BAD MEDIA
2232 *
2233
000.002      2234 CIM.DLY EQU      2          NUMBER OF SECONDS ALLOWED FOR CHECK
000.334      2235 CIM.CNT EQU      300/60*11*CIM.DLY*2  300 RPM, 60 SEC./MIN., 11 HOLES/TRACK
000.320      2236 CIM.MIN EQU      1000*11*CIM.DLY/21/10*2  MIN TOLERANCE COUNT = 21 MIL. SEC. GAP
000.346      2237 CIM.MAX EQU      1000*11*CIM.DLY/19/10*2  MAX TOLERANCE COUNT = 19 MIL. SEC. GAP
2238
054.136 001 000 000 2239 CIM      LXI      B,0          ZERO TRANSITION COUNT
054.141 052 033 040 2240      LHL      .TICCNT        GET CURRENT TIC
054.144 021 350 003 2241      LXI      B,CIM.DLY*1000/2  SET DELAY
054.147 031      2242      DAB      B
054.150 315 224 030 2243      CALL     $CHL          (HL) = -(HL)
054.153 124      2244      MOV      D,H
054.154 135      2245      MOV      E,L          (DE) = -(TARGET STOP TIME)
2246
2247 *      INITIALIZE HOLE DETECT FLAG
2248
054.155 333 177      2249      IN      DP,DC          DISK CONTROL PORT
054.157 346 001      2250      ANI     DF,HD          HOLE DETECT
054.161 062 227 054 2251      STA     CIMA          SET INITIAL VALUE
2252
2253 *      COUNT THE NUMBER OF TRANSITIONS IN CIM.DLY SECONDS
2254
054.164 041 227 054 2255 CIM1    LXI      M,CIMA          M POINTS TO THE FLAG BYTE
054.167 333 177      2256      IN      DP,DC          DISK CONTROL PORT
054.171 346 001      2257      ANI     DF,HD          HOLE DETECT
054.173 276      2258      CMP      M
054.174 312 201 054 2259      JZ      CIM2          NO TRANSITION DETECTED
054.177 003      2260      INX     B          COUNT TRANSITION
054.200 167      2261      MOV     M,A
054.201 052 033 040 2262 CIM2    LHL      .TICCNT
054.204 031      2263      DAB      B
054.205 174      2264      MOV     A,H
054.208 247      2265      ANA     A
054.207 372 164 054 2266      JM      CIM1          IT IS NOT TIME TO STOP YET
2267
2268 *      COMPARE THE COUNT TO EXPECTED VALUES
2269
054.212 170      2270      MOV     A,B
054.213 376 000      2271      CPI     CIM.CNT/256
000.000      2272      SET     CIM.CNT/256
000.000      2273      ERRNZ  CIM.MIN/256-.,
000.000      2274      ERRNZ  CIM.MAX/256-.,
054.215 067      2275      STC          ASSUME BAD VALUE
054.216 300      2276      RNZ          NOT A GOOD VALUE
054.217 171      2277      MOV     A,C
054.220 376 320      2278      CPI     $CIM.MIN

```

INIT - INITIALIZE DISK  
RMI - REQUEST MEDIA INSERTION

CIM  
HEATH HBASH V1.4 01/20/78  
16:01:53 16-MAY-80

054.222	330	2279	RC	
054.223	376 347	2280	CPI	#CIM.MAX+1
054.225	077	2281	CMC	
054.226	311	2282	RET	
		2283		
054.227	000	2284	CIMA	DB 0

LESS THAN THE MINIMUM ACCEPTABLE COUNT VALUE

```

2287 **      AAL - ASK ABOUT LABEL.
2288 *
2289 *      AAL ATTEMPTS TO READ THE VOLUME LABEL RECORD. IF SUCCESSFUL, THE
2290 *      VOLUME # AND LABEL ARE PRINTED. THE USER GETS A CHANCE TO
2291 *      CHICKEN OUT.
2292 *
2293 *      ENTRY  NONE
2294 *      EXIT   NONE
2295 *      USES   ALL
2296 *
2297
054,230 257      2298 AAL  XRA    A
054,231 062 252 040 2299      STA    D,DRVTB+1          CLEAR VOLUME NUMBER
054,234 315 136 031 2300      CALL   $TYPTX
054,237 012 124 150 2301      DB     NL,'The volume now in the drive ...',ENL
054,300 021 000 071 2302      LXI    D,RGTAB          USE RGTAB FOR SCRATCH BUFFER
054,303 001 000 001 2303      LXI    B,1000A
054,306 041 011 000 2304      LXI    H,DFC.LAB
054,311 076 002      2305      MVI    A,DC:RER          READ REGARDLESS
054,313 315 130 040 2306      CALL   SYDD            READ LABEL SECTOR
054,316 322 002 055 2307      JNC    AAL1            OK
2308
2309 *      COULDNT READ IT
2310
054,321 315 136 031 2311      CALL   $TYPTX
054,324 101 160 160 2312      DB     'Apparently has not been initialized before',ENL
054,377 303 066 055 2313      JMP    AAL2
2314
2315 *      COULD READ IT. TYPE DISK NUMBER AND LABEL
2316
055,002 315 136 031 2317 AAL1  CALL   $TYPTX
055,005 151 163 040 2318      DB     'is volume #','+2000
055,021 072 000 071 2319      LDA    RGTAB+LAB.SER
055,024 137      2320      MOV    E,A
055,025 026 000      2321      MVI    D,0
055,027 076 003      2322      MVI    A,3
055,031 315 223 063 2323      CALL   $TDB            TYPE DECIMAL DIGITS
055,034 315 136 031 2324      CALL   $TYPTX
055,037 012 040 114 2325      DB     NL,' Label = ','+2000
055,052 076 074      2326      MVI    A,LAR.LBL
055,054 041 021 071 2327      LXI    H,RGTAB+LAB.LAB
055,057 315 112 063 2328      CALL   $TYPL          TYPE LABEL
055,062 315 136 031 2329      CALL   $TYPTX
055,065 242      2330      DB     '+2000
2331
2332 *      GIVE CHANCE TO CHICKEN OUT
2333
055,066 315 075 062 2334 AAL2  CALL   $CCO            CLEAR CTL-D
055,071 315 136 031 2335      CALL   $TYPTX
055,074 012 124 171 2336      DB     NL,'Type NO to cancel, type YES to erase and initialize the disk. (YES/NO) ','+200
q
055,205 315 205 063 2337      CALL   $ITL
055,210 176      2338      MOV    A,M            (A) = REPLY
055,211 376 116      2339      CPI    'N'
055,213 312 056 051 2340      JE     RESTART
055,216 376 131      2341      CPI    'Y'
055,220 302 066 055 2342      JNE    AAL2

```

INIT - INITIALIZE DISK  
AAL - ASK ABOUT LABEL

HEATH HBASH V1.4 01/20/78  
16:01:54 16-MAY-80

PAGE 52

055.223 311

2343

RET

GOT THE OK

```

2346 **      GVI - GET VOLUME ID.
2347 *
2348 *      GVI GETS A NEW VOLUME SERIAL NUMBER AND AN OPTIONAL VOLUME LABEL.
2349 *
2350 *      ENTRY  NONE
2351 *      EXIT   LABEL+LAB.SER = VOLUME SERIAL #
2352 *           LABEL+LAB.LAB = 60 CHARACTER VOLUME LABEL
2353 *      USES   ALL
2354
2355
055.224      2356 GVI   EQU    *
055.224 315 075 062 2357 CALL  $CCO          CLEAR CTL-0
055.227 315 136 031 2358 CALL  $TYPTX
055.232 012 105 156 2359 DB    NL,'Enter a unique volume serial number from 1 to 255!',' '+2000
055.316 315 134 063 2360 CALL  $IDN          INPUT DECIMAL NUMBER
055.321 332 224 055 2361 JC    GVI           BAD RANGE
055.324 174      2362 MOV   A,H
055.325 267      2363 ORA   A
055.326 302 224 055 2364 JNZ   GVI           TOO LARGE
055.331 265      2365 ORA   L
055.332 312 224 055 2366 JZ    GVI           ZERO ILLEGAL
055.335 062 074 064 2367 STA   LABEL+LAB.SER
2368
2369 *      GET LABEL
2370
055.340 315 075 062 2371 GVI1  CALL  $CCO          CLEAR CTL-0
055.343 315 136 031 2372 CALL  $TYPTX
055.346 105 156 164 2373 DB    'Enter a volume label of 60 characters or less',ENL
056.024 315 213 063 2374 CALL  $ITL
056.027 353      2375 XCHG          (DE) = LINE ADDRESS
056.030 041 115 064 2376 LXI   H,LABEL+LAB.LAB (HL) = TO ADDRESS
2377
2378 *      COPY FROM LINE TO LABEL+LAB.LAB, BLANK FILL OUT TO 60 CHARACTERS
2379
056.033 006 075      2380 MVI   B,61
056.035 032      2381 GVI2  LDAX  D
056.036 167      2382 MOV   M,A          COPY
056.037 023      2383 INX   D
056.040 247      2384 ANA   A
056.041 312 071 056 2385 JZ    GVI3          END OF LINE
056.044 043      2386 INX   H            INCREMENT TO POINTER
056.045 005      2387 DCR   B
056.046 302 035 056 2388 JNZ   GVI2          MORE TO TO
056.051 315 136 031 2389 CALL  $TYPTX
056.054 124 157 157 2390 DB    'Too Long.',ENL
056.066 303 340 055 2391 JMP   GVI1
2392
056.071 066 040      2393 GVI3  MVI   M,' '          BLANK REST OF LABEL
056.073 043      2394 INX   H
056.074 005      2395 DCR   B
056.075 302 071 056 2396 JNZ   GVI3
056.100 311      2397 RET

```

```

2400 **      IDS - INITIALIZE DISK SURFACE.
2401 *
2402 *      IDS IS CALLED TO INITIALIZE THE DISK SURFACE WITH RECORD LABELS,
2403 *      AND A TEST PATTERN.
2404 *
2405 *      ENTRY  MAXTRK = MAX TRACK NUMBER
2406 *      EXIT   NONE
2407 *      USES  ALL
2408
2409
056.101 076 007 2410 IDS MVI A,DC,ABT
056.103 315 343 063 2411 CALL SYDD, RESET DEVICE
056.106 257 2412 XRA A
056.107 062 172 056 2413 STA IDSA SET VOLUME NUMBER
056.112 323 175 2414 OUT UP,FC SET FILL CHARACTER
056.114 107 2415 MOV B,A (B) = TRACK NUMBER
056.115 117 2416 MOV C,A (C) = SECTOR NUMBER
2417
2418 *      ERASE THIS TRACK
2419
056.116 373 2420 IDS1 EI
056.117 076 022 2421 MVI A,DF,DSO+DF,MO
056.121 323 177 2422 OUT DP,DC START THINGS
056.123 062 242 040 2423 STA D,IVCTL SET DEVICE CONTROL
056.126 076 043 2424 MVI A,70/2
056.130 315 303 035 2425 CALL R,DLY WAIT HEAD SETTLE
056.133 076 023 2426 MVI A,DF,DSO+DF,MO+BF,WG
056.135 323 177 2427 OUT DP,DC ON WRITE GATE
056.137 076 175 2428 MVI A,250/2
056.141 315 303 035 2429 CALL R,DLY LET TRACK ERASE
056.144 315 034 064 2430 CALL W,H WAIT INDEX HOLE
056.147 363 2431 DI DISABLE INTERRUPTS
2432
2433 *      WRITE 10 SECTOR HEADERS ON THIS TRACK
2434
056.150 076 377 2435 IDS2 MVI A,3770
056.152 062 243 040 2436 STA D,DLYMO LEAVE MOTOR ON
056.155 315 235 036 2437 CALL R,WHD WAIT HOLE DETECT
056.160 076 001 2438 MVI A,1
056.162 141 2439 MOV H,C SAVE (C) IN H
056.163 016 012 2440 MVI C,10
056.165 315 224 040 2441 CALL D,WSP WRITE SYNC PATTERN
056.170 114 2442 MOV C,H RESTORE (C)
056.171 076 000 2443 MVI A,0
056.172 2444 IDSA EQU *-1 TRACK SERIAL
056.173 315 227 040 2445 CALL D,WNB WRITE BYTE
056.176 170 2446 MOV A,B (A) = TRACK
056.177 315 227 040 2447 CALL D,WNB
056.202 171 2448 MOV A,C (A) = SECTOR
056.203 315 227 040 2449 CALL D,WNB
056.206 172 2450 MOV A,D
056.207 315 227 040 2451 CALL D,WNB WRITE CHECK BYTE
056.212 257 2452 XRA A
056.213 315 227 040 2453 CALL D,WNB WRITE BYTE
056.216 076 001 2454 MVI A,1
056.220 141 2455 MOV H,C SAVE (C) IN H

```

056.221	016 012	2456	MVI	C,10	
056.223	315 224 040	2457	CALL	D,WSP	WRITE SYNC PATTERN
056.226	114	2458	MOV	C,H	RESTORE 'C'
056.227	046 200	2459	MVI	H,128	
056.231	076 107	2460	MVI	A,'G'	IDS3
056.233	315 227 040	2461	CALL	D,WNB	WRITE TEST PATTERN
056.236	076 114	2462	MVI	A,'L'	
056.240	315 227 040	2463	CALL	D,WNB	WRITE
056.243	045	2464	DCR	H	
056.244	302 231 056	2465	JNZ	IDS3	WRITE 256
056.247	172	2466	MOV	A,D	
056.250	315 227 040	2467	CALL	D,WNB	
056.253	315 227 040	2468	CALL	D,WNB	
056.256	315 227 040	2469	CALL	D,WNB	WRITE CHECKSUM, LET TUNNEL GAP PASS
056.261	072 242 040	2470	LDA	D,IVCTL	
056.264	323 177	2471	OUT	DP,DC	OFF WRITE GATE
056.266	014	2472	INR	C	COUNT SECTOR
056.267	076 012	2473	MVI	A,10	
056.271	221	2474	SUB	C	
056.272	302 150 056	2475	JNE	IDS2	NOT NEW TRACK
056.275	373	2476	EI		RESTORE INTERRUPTS
056.276	004	2477	INR	B	COUNT NEW TRACK
056.277	117	2478	MOV	C,A	ZERO SECTOR
056.300	072 074 064	2479	LDA	LABEL+LAB,SER	
056.303	062 172 056	2480	STA	IDS4	USE VOLUME NUMBER FOR OTHER TRACKS
056.306	072 321 064	2481	LDA	MAXTRK	
056.311	270	2482	CMP	B	
056.312	312 323 056	2483	JE	IDS4	ALL DONE
056.315	315 171 040	2484	CALL	D,MAI	MOVE ARM IN
056.320	303 116 056	2485	JMP	IDS1	WRITE NEXT TRACK
		2486			
		2487	*	ALL DONE, RESET DISK ARM.	
		2488			
056.323	076 007	2489	IDS4	MVI	A,DC,ABT
056.325	315 343 063	2490	CALL	SYDD,	ABORT DISK
		2491			
		2492	*	WRITE DUMY BOOTSTRAP	
		2493			
056.330	001 000 007	2494	LXI	B,SOBOOTL	
056.333	021 200 042	2495	LXI	D,SOBOOT	
056.336	041 000 000	2496	LXI	H,0	
056.341	076 001	2497	MVI	A,DC,WRI	
056.343	303 343 063	2498	JMP	SYDD,	WRITE BOOT

```

2501 **      GBL - GET BAD SECTOR LIST.
2502 *
2503 *      GBL GETS A LIST OF BAD SECTORS, AND FLAGS THE BYTES IN THE RGTAB.
2504 *
2505 *      ENTRY  NONE
2506 *      EXIT  RGTAB SETUP
2507 *      USES  ALL
2508
2509
2510 GBL      EQU      *
056.346    041 000 071 2511 LXI      H,RGTAB
056.351    076 310 2512 MVI      A,400/SFG      (A) = # OF EXISTING GROUPS
056.353    066 001 2513 GBL1     MVI      M,1      FLAG UNUSED
056.355    043 2514 INX      H
056.356    075 2515 DCR      A
056.357    302 353 056 2516 JNZ      GBL1     ZERO TABLE
000.310    2517 SET      400/SPG
056.362    076 070 2518 MVI      A,256-      (A) = REMAINDER
000.070    2519 ERRZR   256-      REQUIRE SOME
056.364    066 377 2520 GBL1.5  MVI      M,3770
056.366    043 2521 INX      H
056.367    075 2522 DCR      A
056.370    302 364 056 2523 JNZ      GBL1.5   MORE TO GO
2524
056.373    247 2525 GBL2     ANA      A      CLEAR CARRY
056.374    322 117 057 2526 JNC      GBL4     DONT ALLOW THIS QUESTION
056.377    315 075 062 2527 CALL     $CC0     CLEAR CTL-0
057.002    315 136 031 2528 CALL     $TYPTX
057.005    012 122 105 2529 DB      NL,'RESTRICT TO 35 TRACKS (YES/NO) <NO> ?','+2000
057.054    315 213 063 2530 CALL     $ITL
057.057    176 2531 MOV      A,M
057.060    247 2532 ANA      A
057.061    312 117 057 2533 JZ       GBL4     IS DEFAULT <NO>
057.064    376 116 2534 CPI      'N'
057.066    312 117 057 2535 JE       GBL4     IS NO
057.071    376 015 2536 CPI      CR
057.073    312 117 057 2537 JE       GBL4     IS NO
057.076    376 131 2538 CPI      'Y'
057.100    302 373 056 2539 JNE      GBL2     TRY AGAIN
2540
2541 *      RESTRICT TO 35. FLAG LAST 5 BAD
2542
057.103    041 257 071 2543 LXI      H,35*10/2+RGTAB
057.106    076 031 2544 MVI      A,5*10/2
057.110    066 377 2545 GBL3     MVI      M,-1
057.112    043 2546 INX      H
057.113    075 2547 DCR      A
057.114    302 110 057 2548 JNZ      GBL3     FLAG BAD
2549
2550 *      GET BAD SECTOR LIST
2551
057.117    315 075 062 2552 GBL4     CALL     $CC0     CLEAR CTL-0
057.122    315 136 031 2553 CALL     $TYPTX
057.125    012 105 156 2554 DB      NL,'Enter the numbers of the bad sectors one at a time. Hit RETURN'
057.224    012 141 146 2555 DB      NL,'after each entry, and when finished.',ENL
057.272    315 075 062 2556 GBL5     CALL     $CC0     CLEAR CTL-0

```



057.275	315	136	031	2557	CALL	\$TYPTX	
057.300	123	145	143	2558	DB	'Sector?',' '+2000	
057.310	315	134	063	2559	CALL	\$IDN	INPUT DECIMAL NUMBER
057.313	353			2560	XCHG		
057.314	041	160	376	2561	LXI	H,-400	
057.317	031			2562	DAD	D	
057.320	322	361	057	2563	JNC	GBL7	OK NUMBER
057.323	315	136	031	2564	CALL	\$TYPTX	
057.326	040	040	111	2565	DB	'Illegal Sector Number',ENL	
057.356	303	272	057	2566	JMP	GBL5	
				2567			
057.361	172			2568	GBL7	MOV	A,D
057.362	263			2569	DRA	E	
057.363	310			2570	RZ		CARRIAGE RETURN
057.364	041	365	377	2571	LXI	H,-11	
057.367	031			2572	DAD	D	
057.370	322	323	057	2573	JNC	GBL6	BAD NUMBER
				2574			
				2575	*	FLAG SECTOR BAD	
				2576			
057.373	172			2577	MOV	A,D	
057.374	247			2578	ANA	A	
057.375	037			2579	RAR		
057.376	147			2580	MOV	H,A	
057.377	173			2581	MOV	A,E	
060.000	037			2582	RAR		
060.001	157			2583	MOV	L,A	DIVIDE BY 2
060.002	021	000	071	2584	LXI	D,RGTAB	
060.005	031			2585	DAD	D	
060.006	066	377		2586	MVI	M,-1	FLAG BAD
060.010	303	272	057	2587	JMP	GBL5	GET MORE

```

2590 **      FOV - FORMAT VOLUME.
2591 *
2592 *      FOV FORMATS THE NEW VOLUME.
2593 *
2594 *      IT ALLOCATES SPACE FOR THE DIRECTORY, BUILDS THE
2595 *      DIRECTORY STRUCTURE, AND BUILDS THE
2596 *      RGT AND THE GRT.
2597 *
060.013     2598
2599 FOV     EQU      *
2600
2601 *      FLAG (IN THE RGT) THE LOW SECTORS RESERVED
2602 *      FOR THE SYSTEM.
2603
060.013 041 002 071 2604     LXI      H,RGTAB+2     SKIP GROUPS 0 AND 1, WHICH ARE SPECIAL
377.374     2605     ERRPL     SPG*3-DDF,RGT     MUST BE AT LEAST 3 SPECIAL GROUPS
060.016 076 003     2606     MVI      A,DDF,RGT+SPG-1/SPG-2
060.020 066 377     2607     MVI      M,377H     FLAG RESERVED
060.022 043     2608     INX      H
060.023 075     2609     DCR      A
060.024 302 020 060 2610     JNZ      FOV1
060.027 041 000 000 2611     LXI      H,0
060.032 042 000 071 2612     SHLD     RGTAB     FIRST TWO BLOCKS ARE SPECIAL
2613
060.035 315 147 060 2614     CALL     ADB     ASSIGN DIRECTORY BLOCKS
2615
2616 *      WRITE THE RGT
2617
060.040 072 074 064 2618     LDA      LABEL+LAB.SER
060.043 062 252 040 2619     STA     D,DRVTB+1
060.046 001 000 001 2620     LXI      B,256
060.051 021 000 071 2621     LXI      D,RGTAB
060.054 041 012 000 2622     LXI      H,DDF,RGT
060.057 076 001     2623     MVI      A,DC,WRI
060.061 315 343 063 2624     CALL     SYDD,     WRITE IT
2625
2626
2627 *      WRITE THE LABEL SECTOR
2628
060.064 052 310 040 2629     LHL     S,DATC
060.067 042 075 064 2630     SHLD     LABEL+LAB.IND
060.072 257     2631     XRA      A
060.073 062 252 040 2632     STA     D,DRVTB+1     CLEAR VOLUME NUMBER
060.076 001 000 001 2633     LXI      B,256
060.101 021 074 064 2634     LXI      D,LABEL
060.104 041 011 000 2635     LXI      H,DDF,LAB
060.107 076 001     2636     MVI      A,DC,WRI
060.111 315 343 063 2637     CALL     SYDD,     WRITE LABEL
2638
2639 *      WRITE THE GRT
2640
060.114 315 324 060 2641     CALL     BGT     BUILD THE GRT FIRST
060.117 072 074 064 2642     LDA      LABEL+LAB.SER
060.122 062 252 040 2643     STA     D,DRVTB+1
060.125 001 000 001 2644     LXI      B,256
060.130 021 000 070 2645     LXI      D,GRTAB

```

.....  
INIT - INITIALIZE DISK  
FDV - FORMAT VOLUME  
.....

HEATH HBASM V1.4 01/20/78 PAGE 59  
14:02:05 14-MAY-80  
.....

.....  
060.133 052 101 064 2646 LMLD LABEL+LAB.GRT (HL) = GRT SECTOR INDEX  
060.136 076 001 2647 MVI A,DC,MRI  
060.140 315 343 063 2648 CALL SYDD. WRITE IT  
2649  
2650 \* INITIALIZE THE DIRECTORY  
2651  
060.143 315 053 061 2652 CALL IDD INITIALIZE DEVICE DIRECTORY  
060.146 311 2653 RET  
.....

```

2657 **      ADB - ASSIGN DIRECTORY BLOCKS.
2658 *
2659 *      ADB LOCATES 20 CONTIGUOUS SECTORS TO HOLD THE DIRECTORY
2660 *      AND THE GRT TABLE.
2661 *
2662 *      ENTRY  NONE
2663 *      EXIT   GRIBLK = BLOCK INDEX OF GRT
2664 *            DIRBLK = BLOCK INDEX OF DIRECTORY FIRST BLOCK
2665 *            LABEL+LAB.GRT = GRT SECTOR INDEX
2666 *            LABEL+LAB.DIS = DIRECTORY START SECTOR INDEX (NOT 1ST DIR SECTOR!)
2667 *      USES   ALL
2668 *
2669
060.147      2670 ADB  EQU    *
2671
2672 *      FIND 10 CONTIGUOUS FREE BLOCKS
2673
000.000      2674      ERRNZ  SPG-2      CODE ASSUMES 2
2675
060.147 041 156 071 2676      LXI    H,RGTAB+110      START LOOKING 1/3 THROUGH
060.152 006 012      2677 ADB1  MVI    B,10      NEED 10 BLOCKS
060.154 176          2678 ADB2  MOV    A,M      (A) = RESERVATION BYTE
060.155 043          2679      INX    H
060.156 075          2680      DCR    A
060.157 302 171 060 2681      JNZ    ADB3      RESERVED
060.162 005          2682      DCR    B      COUNT GOT ONE
060.163 302 154 060 2683      JNZ    ADB2      NEED MORE
060.166 303 267 060 2684      JMP    ADB4      GOT OUR 10 BLOCKS
2685
2686 *      RAN INTO A BAD BLOCK. START LOOKING OVER AGAIN
2687
060.171 076 271      2688 ADB3  MVI    A,VOLSIZ/SPG-15+RGTAB
060.173 225          2689      SUB    L
060.174 362 152 060 2690      JP     ADB1      STILL ENOUGH FOR A CHANCE
060.177 306 017      2691      ADI    15
060.201 372 152 060 2692      JM     ADB1      STILL ENOUGH FOR A CHANCE
060.204 315 136 031 2693      CALL  $TYPTX      CANT GET 10 GOOD ONES IN A ROW!
060.207 007 040 126 2694      DB    BELL,' Volume too decrepid for use. Try another.',BELL,ENL
060.264 303 056 051 2695      JMP    RESTART
2696
2697 *      GOT THE TRACKS. (HL) = INDEX OF FIRST GRT+11
2698
060.267 053          2699 ADB4  DCX    H      (HL) = INDEX IF LAST GRT
060.270 175          2700      MOV    A,L
060.271 062 071 064 2701      STA    GRIBLK      USE THIS BLOCK FOR GRT
060.274 345          2702      PUSH  H      SAVE INDEX
060.275 046 000      2703      MVI    H,0
000.000          2704      ERRNZ  SPG-2
060.277 051          2705      DAD    H      (HL) = SECTOR INDEX
060.300 042 101 064 2706      SHLD  LABEL+LAB.GRT  SET GRT ADDRESS
060.303 341          2707      POP    H
060.304 001 367 377 2708      LXI    B,-9
060.307 011          2709      DAD    B      (HL) = BLOCK # FOR BEGINNING OF DIRECTORY
2710
2711 *      SETUP POINTERS FOR LABEL AND INIT PROGRAM.
2712

```

ADB

```

060.310 175      2713      MOV      A,L      (A) = DIRECTORY BLOCK INDEX
060.311 062 072 064 2714      STA      DIRBLK
000.000      2715      ERRNZ   DIRSTRT-1  ASSUME DIRECTORY STARTS ON 2ND BLOCK
060.314 054      2716      INR      L
060.315 046 000 2717      MVI      M,0
060.317 051      2718      DAD      H      (HL) = SECTOR INDEX OF START OF DIRECTORY
060.320 042 077 064 2719      SHLD   LABEL+LAB.DIS
060.323 311      2720      RET

2722 **      BGT - BUILD GRT TABLE.
2723 *
2724 *      BGT BUILDS THE GRT FROM THE RGT, ENTERING BLOCK CHAINS
2725 *      FOR THE DIRECTORY, THE RGT AND THE GRT FILES.
2726 *
2727 *      FIRST, THE BLOCKS USED IN THE RGT, GRT, AND DIRECTORY
2728 *      FILES ARE LINKED UP.
2729 *
2730 *      THEN, ALL FREE BLOCKS LEFT (INDEX = 1) ARE LINKED INTO
2731 *      THE FREE CHAIN.
2732 *
2733 *      ENTRY  NONE
2734 *      EXIT  NONE
2735 *      USES  ALL
2736 *
2737 *
060.324 001 000 001 2738 BGT  LXI      B,256
060.327 021 000 071 2739      LXI      D,RGTAB
060.332 041 000 070 2740      LXI      H,GRTAB
060.335 315 252 030 2741      CALL   $MOVE      COPY RGTAB INTO GRTAB
060.340 257      2742      XRA      A
060.341 062 005 070 2743      STA      BDF,RGT/SPG+GRTAB  RESERVE THE RGT BLOCK (FOR THE RGT FILE)
2744 *
2745 *      LINK THE DIRECTORY BLOCKS TOGETHER, IN THE ORDER
2746 *
2747 *      23 67      DIRECTORY
2748 *      01 45 89
2749 *      23' 67'
2750 *      01' 45'
2751 *
2752 *      89'      GRT
2753 *
000.001      2754 DIRSTRT EQU 1      START DIRECTORY WITH 2ND BLOCK (BLOCKS 23)
2755 *
000.000      2756 ERRNZ   SPG-2  ASSUME SPG
060.344 041 000 070 2757      LXI      H,GRTAB
060.347 072 072 064 2758      LDA      DIRBLK      (A) = DIRECTORY INDEX OF 01
060.352 306 001 2759      ADI      DIRSTRT      (A) = INDEX OF 23
060.354 157      2760      MOV      L,A
060.355 306 002 2761      ADI      2
060.357 167      2762      MOV      M,A      23 -> 67
060.360 157      2763      MOV      L,A
060.361 328 003 2764      SUI      3
060.363 167      2765      MOV      M,A      67 -> 01

```

BGT

```

060.364 157 2766 MOV L,A
060.365 306 002 2767 ADI 2
060.367 167 2768 MOV M,A 01 -> 45
060.370 157 2769 MOV L,A
060.371 306 002 2770 ADI 2
060.373 167 2771 MOV M,A 45 -> 89
060.374 157 2772 MOV L,A
060.375 306 002 2773 ADI 2
060.377 167 2774 MOV M,A 89 -> 23'
061.000 157 2775 MOV L,A
061.001 306 002 2776 ADI 2
061.003 167 2777 MOV M,A 23' -> 67'
061.004 157 2778 MOV L,A
061.005 326 003 2779 SUI 3
061.007 167 2780 MOV M,A 67' -> 01'
061.010 157 2781 MOV L,A
061.011 306 002 2782 ADI 2
061.013 167 2783 MOV M,A 01' -> 45'
061.014 157 2784 MOV L,A
061.015 066 000 2785 MVI M,0 45' -> 00
061.017 062 073 064 2786 STA DIRBLK SET DIRECTORY LAST BLOCK
061.022 072 071 064 2787 LDA GRIBLK
061.025 157 2788 MOV L,A
061.026 066 000 2789 MVI M,0 GRT BLOCK CHAINS TO NONE
2790
2791 * CHAIN ALL UNUSED (001) BLOCKS TOGETHER.
2792 *
2793 * PUT START POINTER IN BLOCK 00, LAST BLOCK POINTS TO 00
2794
061.030 016 000 2795 MVI C,0 (C) = NEXT FREE BLOCK
061.032 041 377 070 2796 LXI M,GRTAB+255 WORK FROM THE BACK
061.035 076 001 2797 BGT2 MVI A,1
061.037 276 2798 CMF M
061.040 302 045 061 2799 JNE BGT3 NOT FREE
061.043 161 2800 MOV M,C LINK ON FRONT OF CHAIN
061.044 115 2801 MOV C,L IS NOW FRONT OF CHAIN
061.045 055 2802 BGT3 ICR L
061.046 302 035 061 2803 JNZ BGT2 MORE TO GO
061.051 161 2804 MOV M,C SET POINTER IN GROUP 00
061.052 311 2805 RET
  
```

```

2807 ** IDD - INITIALIZE DEVICE DIRECTORY.
2808 *
2809 * IDD INITIALIZES THE DEVICE DIRECTORY BY WRITING THE DIRECTORY
2810 * BLOCK 1 CLEAR, BLOCKS 3 - N EMPTY, AND BLOCK 2 CONTAINING
2811 * THE FILES
2812 *
2813 * RGT .SYS
2814 * GRT .SYS
2815 * DIRECT .SYS
2816 *
2817 * THE DIRECTORY'S BLOCKS ARE INTERNALLY LINKED TO FOLLOW THEIR POSITION
2818 * IN THE GRT.
  
```

```

2819 *
2820 *      ENTRY  NONE
2821 *      EXIT  NONE
2822 *      USES  ALL
2823
2824
061.053 076 002 2825 YDD MVI A,2
061.055 062 042 062 2826 STA IDDB SET COUNT FOR SECOND DIRECTORY BLOCK
061.060 006 377 2827 MVI B,DF,EMP SET ENTRIES EMPTY
061.062 315 043 062 2828 CALL PDS PRESET DIRECTORY SECTOR
2829
2830 *      WRITE DIRECTORY BLOCKS, IN ORDER
2831
061.065 076 027 2832 MVI A,DIRELEN
061.067 062 015 067 2833 STA IDDA+DIS,ENL SET DIRECTORY ENTRY LENGTH
061.072 052 077 064 2834 LHLD LABEL+LAB,DIS (HL) = INDEX OF FIRST BLOCK
061.075 042 016 067 2835 SHLD DIS,SEC+IDDA SET IN DIRECTORY ENTRY
000.000 2836 ERRNZ SPG-2 ASSUME =2
061.100 174 2837 MOV A,H
061.101 037 2838 RAR
061.102 175 2839 MOV A,L
061.103 037 2840 RAR (A) = GROUP INDEX OF THIS DIRECTORY BLOCK
061.104 157 2841 MOV L,A
061.105 046 070 2842 MVI H,GRTAB/256
061.107 156 2843 MOV L,M (L) = INDEX OF FOLLOWING ENTRY
061.110 046 000 2844 MVI H,0
061.112 051 2845 DAD H (HL) = SECTOR INDEX OF FOLLOWING ENTRY
061.113 042 020 067 2846 SHLD DIS,LNK+IDDA SET IN BLOCK
061.116 041 042 062 2847 LXI H,IDD4
061.121 065 2848 DCR M COUNT SECTOR
061.122 302 257 061 2849 JNZ IDDA4 IS NOT SECTOR 2
2850
2851 *      IS SECTOR 2. ADD SPECIAL FILES TO THE END OF IT.
2852
061.125 052 310 040 2853 LHLD S,DATC
061.130 042 331 061 2854 SHLD IDDC2 SET CREATION AND ALTERATION DATES FOR ALL
061.133 042 333 061 2855 SHLD IDDC2+2
061.136 042 360 061 2856 SHLD IDDC4
061.141 042 362 061 2857 SHLD IDDC4+2
061.144 042 007 062 2858 SHLD IDDC6
061.147 042 011 062 2859 SHLD IDDC6+2
2860
2861 *      SET DIRECTORY POINTERS TO GRT BLOCK AND DIRECTORY
2862
061.152 072 071 064 2863 LDA GRTBLK
061.155 062 355 061 2864 STA IDDC3 SET GRT BLOCK
061.160 062 356 061 2865 STA IDDC3+1 ONLY ONE BLOCK IN FILE
061.163 076 001 2866 MVI A,1
061.165 062 357 061 2867 STA IDDC3+2 ONLY ONE SECTOR IN FILE
061.170 052 077 064 2868 LHLD LABEL+LAB,DIS
061.173 174 2869 MOV A,H
061.174 037 2870 RAR
061.175 175 2871 MOV A,L
061.176 037 2872 RAR (A) = GROUP NUMBER OF DIRECTORY START
061.177 062 004 062 2873 STA IDDC5
061.202 072 073 064 2874 LDA DIRBLK

```

ID0

```

061.205 062 005 062 2875 STA IDDC5+1 SET GROUP NUMBER OF LAST BLOCK
061.210 076 002 2876 MVI A,2
061.212 062 006 062 2877 STA IDDC5+2 SET LAST SECTOR INDEX
2878
061.215 001 134 000 2879 LXI B, IDDC1
061.220 021 306 061 2880 LXI D, IDDC
061.223 041 260 066 2881 LXI H, IDDB+IDDBL-IDDC1
061.226 315 252 030 2882 CALL $MOVE MOVE ENTRIES INTO BLOCK
061.231 001 000 002 2883 LXI B, 512
061.234 021 022 065 2884 LXI D, IDDA
061.237 052 016 067 2885 LHLD DIS.SEC+IDDA
061.242 076 001 2886 MVI A, DC.WRI
061.244 315 343 063 2887 CALL SYDD, WRITE BLOCK AND EXIT
2888
2889 * HAVE WRITTEN 1ST AND 2ND DIRECTORY BLOCKS. FLAG
2890 * ALL REMAINING BLOCKS EMPTY
2891
061.247 006 376 2892 MVI B, DF.CLR FLAG REST OF DIRECTORY EMPTY
061.251 315 043 062 2893 CALL PDS PRESET DIRECTORY SECTOR
061.254 303 275 061 2894 JMP IDDC5 SKIP RE-WRITING 2ND SECTOR
2895
2896 * WRITE BLOCK TO DISK
2897
061.257 001 000 002 2898 IDDC4 LXI B, 512
061.262 021 022 065 2899 LXI D, IDDA
061.265 052 016 067 2900 LHLD DIS.SEC+IDDA
061.270 076 001 2901 MVI A, DC.WRI
061.272 315 343 063 2902 CALL SYDD, WRITE BLOCK
061.275 052 020 067 2903 IDDC5 LHLD DIS.LNK+IDDA
061.300 174 2904 MOV A, H
061.301 265 2905 ORA L
061.302 302 075 061 2906 JNZ IDDC2 NOT LAST ONE, DO SOME MORE
061.305 311 2907 RET
2908
2909
2910 ** DIRECTORY ENTRIES FOR SPECIAL FILES
2911
061.306 2912 IDDC DS 0
000.000 2913 ERRNZ *-IDDC-DIR.NAM
061.306 122 107 124 2914 DB 'RGT',0,0,0,0,0 RGT.SYS
000.000 2915 ERRNZ *-IDDC-DIR.EXT
061.316 123 131 123 2916 DB 'SYS'
000.000 2917 ERRNZ *-IDDC-DIR.PRO
061.321 000.000 2918 DB 0,0 VERSION AND PPN
000.000 2919 ERRNZ *-IDDC-DIR.CLU
061.323 000 2920 DB 0 CLUSTER
000.000 2921 ERRNZ *-IDDC-DIR.FLG
061.324 360 2922 DB DIF.SYS+DIF.LOC+DIF.CNT+DIF.WP SET UNCHANGABLY WRITE-PROTECTED
061.325 000 2923 DB 0 UNUSED
000.000 2924 ERRNZ *-IDDC-DIR.FGN
061.326 005 2925 DB DDF.RGT/SPG FIRST GROUP
000.000 2926 ERRNZ *-IDDC-DIR.LGN
061.327 005 2927 DB DDF.RGT/SPG LAST GROUP
000.000 2928 ERRNZ *-IDDC-DIR.LSI
061.330 001 2929 DB 1 SECTOR INDEX OF EOF
000.000 2930 ERRNZ *-IDDC-DIR.CRD

```



```

2931 IDDC2 DS 4 CREATION AND ALTERATION DATE
2932
061.335 107 122 124 2933 DB 'GRT',0,0,0,0,0 GRT.SYS
061.345 123 131 123 2934 DB 'SYS'
061.350 000 000 2935 DB 0,0 PFN, VERSION
061.352 000 2936 DB 0 CLUSTER
061.353 360 2937 DB DIF.SYS+DIF.LOC+DIF.CNT+DIF.WP UNCHANGABLY WRITE-PROTECT
061.354 000 2938 DB 0 UNUSED
061.355 2939 IDDC3 DS 3 FIRST GROUP, LAST GROUP, LAST SECTOR
061.360 2940 IDDC4 DS 4 CREATION AND ALTERATION DATE
2941
061.364 104 111 122 2942 DB 'DIRECT',0,0 DIRECT.SYS
061.374 123 131 123 2943 DB 'SYS'
061.377 000 000 2944 DB 0,0 PFN, VERSION
062.001 000 2945 DB 0 CLUSTER
062.002 340 2946 DB DIF.SYS+DIF.LOC+DIF.WP UNCHANGABLY WRITE PROTECT
062.003 000 2947 DB 0 UNUSED
062.004 2948 IDDC5 DS 3 FIRST GROUP, LAST GROUP, LAST SECTOR
062.007 2949 IDDC6 DS 4 CREATION AND ALTERATION DATE
062.013 376 2950 DB DF.CLR LAST SPOT IN 2ND BLOCK IS EMPTY
062.014 107 101 103 2951 DB 'GAC / HEATH CO.'
062.033 2952 DS DIRELEN-1-15 REST OF ENTRY MEANINGLESS
2953
000.134 2954 IDDCI EQU *-IDDC
000.000 2955 ERRNZ 4*DIRELEN-IDDCI SHOULD BE FOUR ENTRYS
2956
062.042 000 2957 IDDB DB 0 DIRECTORY BLOCK COUNTER

```

```

2959 ** PDS - PRESET DIRECTORY SECTOR.
2960 *
2961 * PDS BUILDS A DIRECTORY BLOCK INTO *IDDB* FULL OF DIRECTORY
2962 * ENTRIES IN THE FORMAT:
2963 *
2964 * FLAG,0,0, . . . ,0
2965 *
2966 * WHERE FLAG = SOME SUPPLIED VALUE.
2967 *
2968 * ENTRY (B) = FLAG BYTE
2969 * EXIT NONE
2970 * USES ALL
2971
062.043 041 022 065 2972
2973 PDS LXI H,IDDB
062.046 021 372 001 2974 LXI D,IDDBL
062.051 160 2975 PDS1 MOV M,B SET FIRST BYTE
062.052 043 2976 INX H
062.053 033 2977 DCX D
2978
2979 * NOW ZERO REST OF ENTRY
2980
062.054 016 026 2981 MVI C,DIRELEN-1
062.056 066 000 2982 PDS2 MVI M,0 ZERO IT
062.060 043 2983 INX H

```

062.061	033		2984	DCX	D	COUNT BYTE FROM DIRECTORY BLOCK
062.062	015		2985	DCR	C	COUNT BYTE FROM DIRECTORY ENTRY
062.063	302	056 062	2986	JNZ	PDS2	
062.066	172		2987	MOV	A,D	
062.067	263		2988	ORA	E	
062.070	302	051 062	2989	JNZ	PDS1	
062.073	167		2990	MOV	M,A	ZERO BYTE FOLLOWS ENTRYS
062.074	311		2991	RET		EXIT

062.075

2994 XTEXT SAVALL

2996X \*\* \*RSTALL - RESTORE ALL REGISTERS.  
2997X \*  
2998X \* \*RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND  
2999X \* RETURNS TO THE PREVIOUS CALLER.  
3000X \*  
3001X \* ENTRY (SP) = PSW  
3002X \* (SP+2) = BC  
3003X \* (SP+4) = DE  
3004X \* (SP+6) = HL  
3005X \* (SP+8) = RET  
3006X \* EXIT TO \*RET\*, REGISTERS RESTORED  
3007X \* USES ALL  
3008X  
3009X

031.047

3010X \*RSTALL EQU 31047A IN H17 ROM

3012X \*\* \*SAVALL - SAVE ALL REGISTERS ON STACK.  
3013X \*  
3014X \* \*SAVALL SAVES ALL THE REGISTERS ON THE STACK.  
3015X \*  
3016X \* ENTRY NONE  
3017X \* EXIT (SP) = PSW  
3018X \* (SP+2) = BC  
3019X \* (SP+4) = DE  
3020X \* (SP+6) = HL  
3021X \* USES H,L  
3022X  
3023X

031.054

3024X \*SAVALL EQU 31054A IN H17 ROM

062.075

3025 XTEXT CCO

3027X \*\* \*CCO - CLEAR CONTROL-0  
3028X \*  
3029X \* \*CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-0 CHARACTER.  
3030X \*  
3031X \* ENTRY NONE  
3032X \* EXIT NONE  
3033X \* USES NONE  
3034X  
3035X

062.075 315 054 031  
062.100 076 004  
062.102 001 001 000  
062.105 377 006  
062.107 303 047 031  
062.112

3036X \*CCO CALL \*SAVALL SAVE REGISTERS  
3037X MVI A,I,CONFL  
3038X LXI B,CO,FLG CLEAR CO,FLG  
3039X DB SYSCALL,CONSL  
3040X JMP \*RSTALL RESTORE REGISTERS AND RETURN  
3041 XTEXT CDEHL

```

3043X **      $CDEHL - COMPARE (DE) TO (HL)
3044X *
3045X *      $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
3046X *
3047X *      ENTRY  NONE
3048X *      EXIT   'Z' SET IF (DE) = (HL)
3049X *      USES   A,F
3050X
3051X
030.214      3052X $CDEHL EQU   30216A      IN H17 ROM
062.112      3053      XTEXT  CHL

3055X **      $CHL - COMPLEMENT (HL).
3056X *
3057X *      (HL) = -(HL)          TWO'S COMPLEMENT
3058X *
3059X *      ENTRY  NONE
3060X *      EXIT   NONE
3061X *      USES   A,F,H,L
3062X
3063X
030.224      3064X $CHL  EQU   30224A      IN H17 ROM
062.112      3065      XTEXT  DTB

3067X **      $DTB - DELETE TRAILING BLANKS.
3068X *
3069X *      $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
3070X *
3071X *      ENTRY  (HL) = LINE FWA
3072X *      EXIT   (A) = LENGTH OF RESULT (INCLUDING 00 TERMINATOR BYTE)
3073X *      USES   A,F
3074X
3075X
062.112  325      3076X $DTB  PUSH  D          SAVE (DE)
062.113  124      3077X      MOV   D,H
062.114  135      3078X      MOV   E,L      (DE) = FWA
062.115  033      3079X      DCX   D          (DE) = FWA-1
062.116  176      3080X $DTB1  MOV   A,H
062.117  043      3081X      INX   H
062.120  247      3082X      ANA   A          FIND END OF LINE
062.121  302  116  062  3083X      JNZ   $DTB1
062.124  053      3084X      DCX   H          (HL) = ADDRESS OF TERMINATING ZERO BYTE
3085X
3086X *      GOT END OF LINE. DELETE TRAILING BLANKS
3087X
062.125  053      3088X $DTB2  DCX   H          BACKUP ONE CHARACTER
062.126  315  216  030  3089X      CALL $CDEHL
062.131  312  142  062  3090X      JE    $DTB3      GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
062.134  176      3091X      MOV   A,H
062.135  376  040      3092X      CPI

```

```

062.137 312 125 062 3093X      JE      *DTB2      GOT BLANK
3094X
3095X *      HAVE TRIMED LINE. COMPUTE LENGTH
3096X
062.142 043      3097X *DTB3  INX      H
062.143 066 000 3098X      MVI      M,0      TERMINATE LINE
062.145 175      3099X      MOV      A,L
062.146 223      3100X      SUB      E      (A) = LENGTH +1 (FOR OO BYTE)
062.147 353      3101X      XCHG
062.150 043      3102X      INX      H      (HL) = LINE FWA
062.151 321      3103X      POP      D      RESTORE (DE)
062.152 311      3104X      RET
062.153      3105      XTEXT  MLU

```

```

3107X **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
3108X *
3109X *      MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
3110X *
3111X *      ENTRY (HL) = LINE FWA
3112X *      EXIT  NONE
3113X *      USES  NONE
3114X
3115X

```

```

062.153 365      3116X *MLU  PUSH  PSW      SAVE (PSW)
062.154 345      3117X      PUSH  H      SAVE FWA
062.155 053      3118X      DCX  H      ANTICIPATE INX H
062.156 043      3119X *MLU1 INX  H
062.157 176      3120X      MOV  A,M      (A)= CHARACTER
062.160 315 007 050 3121X      CALL *MCU      MAP CHAR TO UPPER
062.163 167      3122X      MOV  M,A
062.164 247      3123X      ANA  A
062.165 302 156 062 3124X      JNZ  *MLU1      MORE TO GO
062.170 341      3125X      POP  H      RESTORE (HL)
062.171 361      3126X      POP  PSW      RESTORE (PSW)
062.172 311      3127X      RET
062.173      3128      XTEXT  RTL

```

```

3130X **      $RTL - READ TEXT LINE.
3131X *
3132X *      $RTL READS A LINE FROM THE TERMINAL.
3133X *
3134X *      CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
3135X *      CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
3136X *      $RTL RETURNS.
3137X *
3138X *      ENTRY (HL) = BUFFER FWA
3139X *      EXIT  'C' CLEAR IF OK
3140X *      DATA IN BUFFER
3141X *      (A) = TEXT LENGTH
3142X *      'C' SET IF CTL-D STRUCK

```

\*RTL

16:02:33 16-MAY-80

```

3143X *      USES      A,F
3144X
3145X
062.173 315 202 062 3146X $RTL, CALL $RTL      $RTL IN UPPER CASE
062.176 330          3147X          RC          CTL-D
062.177 303 153 062 3148X          JMP $HLU      MAP LINE TO UPPER CASE
3149X
062.202          3150X $RTL EQU *
062.202 345          3151X          PUSH H          SAVE FWA
062.203 315 054 063 3152X $RTL1 CALL $RCHAR
062.206 376 004          3153X          CPI          CTLD
062.210 312 235 062 3154X          JE          $RTL2      CTL-D STRUCK
062.213 167          3155X          MOV M,A
062.214 043          3156X          INX H
062.215 376 012          3157X          CPI NL
062.217 302 203 062 3158X          JNE $RTL1
062.222 053          3159X          DCX H
062.223 066 000          3160X          MVI M,0
062.225 043          3161X          INX H
3162X
3163X *      ALL DONE, COMPUTE LENGTH
3164X
062.226 353          3165X          XCHG          (DE) = LWA+1
062.227 343          3166X          XTHL          (HL) = FWA
062.230 173          3167X          MOV A,E
062.231 225          3168X          SUB L          (A) = LENGTH
062.232 247          3169X          ANA A          CLEAR CARRY
062.233 321          3170X          POP D          RESTORE (DE)
062.234 311          3171X          RET
3172X
3173X *      CTL-D STRUCK
3174X
062.235 341          3175X $RTL2 POP H          (HL) = FWA
062.236 067          3176X          STC
062.237 311          3177X          RET
062.240          3178X          XTEXT UDD

```

```

3180X **      $UDD - UNPACK DECIMAL DIGITS.
3181X *
3182X *      UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
3183X *      DECIMAL DIGITS, THE RESULT IS ZERO FILLED.
3184X *
3185X *      ENTRY (B,C) = ADDRESS VALUE
3186X *      (A) = DIGIT COUNT
3187X *      (H,L) = MEMORY ADDRESS
3188X *      EXIT (HL) = (HL) + (A)
3189X *      USES ALL
3190X
3191X
031.157 3192X $UDD EQU 31157A IN H17 ROM
062.240 3193X XTEXT HLIHL

```

```

3195X ** *HLIHL - LOAD HL INDIRECT THROUGH HL.
3196X *
3197X * (HL) = ((HL))
3198X *
3199X * ENTRY NONE
3200X * EXIT NONE
3201X * USES A,H,L
3202X
030.211 3203X *HLIHL EQU 30211A IN H17 KON
062.240 3204 XTEXT DOS

3206X ** *DOS - DISMOUNT OPERATING SYSTEM.
3207X *
3208X * *DOS DISMOUNTS SY2, SY1 (IF MOUNTED), AND SY0. /79.11.6C/
3209X *
3210X * THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
3211X * SYSTEM IS NOTIFIED.
3212X *
3213X *
3214X * ENTRY NONE
3215X *
3216X * EXIT (PSW) = 'C' CLEAR IF NO ERROR
3217X * 'C' SET IF ERROR
3218X * (A) = ERROR CODE
3219X *
3220X * USES ALL
3221X *
3222X
062.240 315 136 031 3223X *DOS CALL $TYPTX
062.243 012 007 104 3224X DB NL,BELL,'Dismounting All Disks:',NL,ENL
3225X
062.275 076 000 3226X MVI A,DVLO
062.277 377 010 3227X DB SYSCALL,.LOAD0
062.301 330 3228X RC
062.302 076 001 3229X MVI A,DVL1
062.304 377 010 3230X DB SYSCALL,.LOAD0
062.306 330 3231X RC
3232X
062.307 041 047 063 3233X LXI H,DOSC
062.312 315 025 063 3234X CALL DOS.
062.315 330 3235X RC
062.316 041 042 063 3236X LXI H,DOSR
062.321 315 025 063 3237X CALL DOS.
062.324 330 3238X RC FATAL ERROR
062.325 041 035 063 3239X LXI H,DOSA
062.330 315 025 063 3240X CALL DOS.
062.333 330 3241X RC
3242X
062.334 315 136 031 3243X CALL $TYPTX
062.337 012 122 145 3244X DB NL,'Remove the Disk(s). Hit RETURN when ready:','+2000
063.013 315 054 063 3245X IOS1 CALL $RCHAR READ CHARACTER
063.016 376 012 3246X CPI NL
063.020 302 013 063 3247X JNE DOS1

```

```

063.023 247          3248X      ANA      A          CLEAR CARRY
063.024 311          3249X      RET
                   3250X
                   3251X *      DISMOUNT A DEVICE WITHOUT REGARD TO WHETHER MOUNTED OR NOT
                   3252X
063.025 377 201      3253X  DOS.    DB      SYSCALL, DMOUN
063.027 320          3254X      RNC
063.030 376 042      3255X      CPI      EC.NVM      NO VOLUME MOUNTED ERROR NOT CONSIDERED FATAL
063.032 310          3256X      RZ              NOT FATAL, CARRY NOW CLEAR
063.033 067          3257X      STC              FLAG FATAL ERROR
063.034 311          3258X      RET
                   3259X
063.035 123 131 060  3260X  DOSA    DB      'SY01',0
063.042 123 131 061  3261X  DOSB    DB      'SY11',0
063.047 123 131 062  3262X  DOSC    DB      'SY21',0
063.054          3263      XTEXT  WER

```

```

3265X **          $WER - WRITE ENABLE RAM.
3266X *
3267X *          $WER IS CALLED TO ENABLE WRITING TO THE H17 CONTROLLER'S
3268X *          RAM AREA.
3269X *
3270X *          ENTRY  NONE
3271X *          EXIT   NONE
3272X *          USES   NONE
3273X
3274X
031.241          3275X $WER  EQU    31241A      IN H17 ROM

```

```

3277X **          $WDR - WRITE DISABLE RAM.
3278X *
3279X *          $WDR IS CALLED TO DISABLE WRITING TO THE H17 CONTROLLER'S
3280X *          RAM AREA.
3281X *
3282X *          ENTRY  NONE
3283X *          EXIT   NONE
3284X *          USES   NONE
3285X
3286X
031.222          3287X $WDR  EQU    31222A      IN H17 ROM
063.054          3288      XTEXT  RCHAR

```



```

3290X ** $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
3291X *
3292X * ENTRY NONE
3293X * EXIT (A) = CHARACTER
3294X * USES A,F
3295X
3296X
063.054 377 001 3297X *RCHAR DB SYSCALL, SCIN
063.056 332 054 063 3298X JC $RCHAR NOT READY
063.061 311 3299X RET
3300X
063.062 377 002 3301X *WCHAR DB SYSCALL, SCOUT
063.064 311 3302X RET
063.065 3303 XTEXT TYPCH

```

```

3305X ** $TYPCH - TYPE SINGLE CHARACTER.
3306X *
3307X * ENTRY (RET) = CHARACTER
3308X * EXIT TO (RET)+1
3309X * (A) = CHARACTER TYPED
3310X
3311X
063.065 343 3312X $TYPCH XTHL (HL) = RETURN ADDRESS
063.066 176 3313X MOV A,M (A) = CHARACTER
063.067 043 3314X INX H
063.070 343 3315X XTHL RESTORE ADVANCED EXIT ADDRESS
3316X
3317X ** $TYPC. - TYPE SINGLE CHARACTER.
3318X *
3319X * ENTRY (A) = CHARACTER
3320X * EXIT TO (RET)
3321X
063.071 377 002 3322X $TYPC. DB SYSCALL, SCOUT
063.073 311 3323X RET
063.074 3324 XTEXT TYPT2

```

```

3326X ** $TYPTX - TYPE TEXT.
3327X *
3328X * $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
3329X *
3330X * IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.
3331X * A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.
3332X *
3333X * ENTRY (RET) = TEXT
3334X * EXIT TO (RET+LENGTH)
3335X * USES A,F
3336X
3337X
031.138 3338X $TYPTX EQU 31136A IN M17 ROM
3339X

```

```

031.144      3340X $TYPTX, EQU 31144A      IN H17 ROM
000.001      3341 $CMP$ EQU 1              NO COMPRESSED TEXT
063.074      3342      XTEXT  TYPLN

3344X **      $TYPLN - TYPE LINE.
3345X *
3346X *      $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
3347X *      TAKEN AS CRLF (WITH THE PROPER PADDING)
3348X *
3349X *      CALL $TYPLN
3350X *      DB N              BYTE COUNT OF FOLLOWING MESSAGE
3351X *      DB 'N-CHARACTER MESSAGE'
3352X *
3353X *      ENTRY (RET) = TEXT COUNT
3354X *      (RET)+1 - (RET)+N = TEXT
3355X *      EXIT TO (RET)+N+1
3356X *      USES A,F
3357X *
3358X
3359X
063.074 343 3360X $TYPLN, XTHL (H,L) = COUNT ADDRESS
063.075 176 3361X MOV A,M (A) = COUNT
063.076 043 3362X INX H (H,L) = TEXT ADDRESS
063.077 345 3363X PUSH H SAVE TEXT FWA
063.100 315 072 030 3364X CALL $DADA CALCULATE RETURN ADDRESS
063.103 343 3365X XTHL (HL) = TEXT ADDR
063.104 315 112 063 3366X CALL $TYPL, OUTPUT LINE
063.107 341 3367X POP H (HL) = RETURN ADDRESS
063.110 343 3368X XTHL RESTORE (HL), SET RETURN ADDRESS
063.111 311 3369X RET
3370X
3371X **      $TYPL, - TYPE LINE.
3372X *
3373X *      ENTRY (HL) = ADDRESS
3374X *      (A) = COUNT
3375X *      EXIT NONE
3376X *      USES A,F,H,L
3377X
063.112 3378X $TYPL, EQU *
063.112 247 3379X ANA A
063.113 310 3380X RZ NOTHING TO TYPE
063.114 345 3381X PUSH PSW SAVE COUNT
063.115 176 3382X MOV A,M (A) = CHARACTER
063.116 043 3383X INX H
063.117 247 3384X ANA A
000.001 3385X IF $CMP$ IF HAVE COMPRESSED SPACES
3386X JM TPL2 IS COMPRESSED SPACE
3387X ENDIF
063.120 314 304 063 3388X CZ $CRLF
063.123 315 071 063 3389X CALL $TYPC, TYPE CHARACTER
063.126 361 3390X TPL1 POP PSW
063.127 075 3391X DCR A
063.130 302 112 063 3392X JNZ $TYPL,

```

```

063.133 311 3393X RET
000.001 3394X IF $CMP$ IF COMPRESSED TEXT
3395X
3396X * HAVE COMPRESSED SPACE.
3397X
3398X TPL2 DCR A
3399X CP $TYPCH TYPE '00' IF CHARACTER WAS '2000'
3400X DB 0
3401X ANA A SET CODES
3402X TPL3 JF TPL1 ALL EXPANDED
3403X PUSH PSW SAVE COUNT
3404X CALL $TYPCH
3405X DB
3406X FOP PSW
3407X DCR A
3408X JMP TPL3
3409X ENDIF
063.134 3410 XTEXT IDN

```

```

3412X ** $IDN - INPUT DECIMAL NUMBER.
3413X *
3414X * $IDN IS CALLED TO INPUT A DECIMAL NUMBER FROM THE CONSOLE.
3415X *
3416X * AN ENTIRE LINE IS ACCEPTED, AND THEN THE NUMBER IS DECODED.
3417X *
3418X * RUBOUTS AND BACKSPACES MAY BE USED DURING ENTRY.
3419X *
3420X * ENTRY NONE
3421X * EXIT 'C' CLEAR IF OK
3422X * (HL) = NUMBER
3423X * 'C' SET IF SOME ERROR
3424X * USES ALL
3425X
3426X

```

```

063.134 041 022 067 3427X $IDN LXI H,ITLA
063.137 315 202 062 3428X CALL $RTL READ LINE
063.142 303 145 063 3429X JMP $PDD PACK DECIMAL DIGITS
063.145 3430 XTEXT PDD

```

```

3432X ** $PDD - PACK DECIMAL DIGITS.
3433X *
3434X * $PDD PACKS A STRING OF DECIMAL DIGITS INTO A DECIMAL INTEGER.
3435X *
3436X * THE CHARACTERS MUST BE IN MEMORY, AND BE IMMEDIATELY FOLLOWED BY A
3437X * 00 BYTE.
3438X *
3439X * ENTRY (HL) = ADDRESS OF CHARACTERS
3440X * EXIT 'C' CLEAR IF OK
3441X * (HL) = NUMBER
3442X * 'C' SET IF ERROR

```

\$PDD

```

3443X *      USES      A,F,D,E,H,L
3444X
3445X
063.145 353 3446X $PDD XCHG      (DE) = TEXT ADDRESS
063.146 041 000 000 3447X LXI      H,0      (HL) = ACCUM
3448X
063.151 032 3449X $PDD1 LDAX     D
063.152 023 3450X      INX      D      ADVANCE ADDRESS
063.153 247 3451X      ANA      A
063.154 310 3452X      RZ              ALL DONE
063.155 326 060 3453X      SUI      '0'
063.157 330 3454X      RC              TOO SMALL
063.160 376 012 3455X      CFI      10
063.162 077 3456X      CMC
063.163 330 3457X      RC              TOO SMALL
063.164 325 3458X      PUSH     D      SAVE (DE)
063.165 353 3459X      XCHG
063.166 315 324 030 3460X      CALL     $MU10
063.171 321 3461X      POP      D
063.172 330 3462X      RC              OVERFLOW
063.173 205 3463X      ADD      L
063.174 157 3464X      MOV      L,A
063.175 076 000 3465X      MVI      A,0
063.177 214 3466X      ADC      H
063.200 147 3467X      MOV      H,A
063.201 322 151 063 3468X      JNC     $PDD1      NOT OVERFLOW
063.204 311 3469X      RET
063.205 3470      XTEXT    MU10
    
```

```

3472X **      $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
3473X *
3474X *      (HL) = (DE)*10
3475X *
3476X *      ENTRY   (DE) = MULTIPLIER
3477X *      EXIT    'C' CLEAR IF OK
3478X *      (HL) = PRODUCT
3479X *      'C' SET IF ERROR
3480X *      USES   D,E,H,L,F
3481X
3482X
030.324 3483X $MU10 EQU     30324A      IN H17 ROM
063.205 3484      XTEXT    ITL
    
```

```

3486X **      $ITL - INPUT TEXT LINE.
3487X *
3488X *      $ITL INPUTS A LINE FROM THE TERMINAL.
3489X *
3490X *      CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
3491X *      CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
3492X *      $ITL RETURNS.
    
```

\*ITL

```

3493X *
3494X *
3495X *   ENTRY  NONE
3496X *   EXIT   (HL) = #ITLA
3497X *   USES  (A) = TEXT LENGTH
3498X *
3499X *
063.205 315 213 063 3500X $ITL. CALL  $ITL      INPUT LINE IN UPPER CASE
063.210 303 153 062 3501X   JMP    $MLU      MAP LINE TO UPPER
3502X
063.213 041 022 067 3503X $ITL. LXI   H,ITLA
063.216 303 202 062 3504X   JMP    $RTL      READ TEXT LINE
063.221 3505   XTEXT  TDD

```

```

3507X **   $TDD - TYPE DECIMAL DIGITS.
3508X *
3509X *   $TDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.
3510X *
3511X *   ENTRY  (D,E) = VALUE
3512X *   (A) = DIGIT COUNT
3513X *   EXIT  VALUE TYPED.
3514X *   USES  A,B,C,F
3515X
3516X
063.221 076 005 3517X $TDD. MVI   A,5
063.223 345 3518X $TDD. PUSH  H
063.224 365 3519X TDD1  PUSH  PSW
063.225 041 270 063 3520X   LXI   H,TDDA-2
063.230 007 3521X   RLC
063.231 315 101 030 3522X   CALL  $BADA.      (A) = DIGIT NUMBER*2
063.234 176 3523X   MOV   A,M
063.235 043 3524X   INX   H
063.236 146 3525X   MOV   H,M
063.237 157 3526X   MOV   L,A      (HL) = MULTIPLE OF 10
063.240 353 3527X   XCHG      (DE) = DEVISOR, (HL) = VALUE
063.241 076 377 3528X   MVI   A,377R
063.243 031 3529X TDD2  DAD   D
063.244 074 3530X   INR   A
063.245 332 243 063 3531X   JC    TDD2      IF MORE TO GO
063.250 306 060 3532X   ADI   '0'
063.252 315 071 063 3533X   CALL  $TYPC.      TYPE DIGIT
063.253 175 3534X   MOV   A,L
063.256 223 3535X   SUB   E
063.257 137 3536X   MOV   E,A      REMOVE EXTRA SUBTRACTION
063.260 174 3537X   MOV   A,H
063.261 232 3538X   SBB   D
063.262 127 3539X   MOV   D,A
063.263 361 3540X   POP   PSW
063.264 075 3541X   DCR   A
063.265 302 224 063 3542X   JNZ  TDD1      IF MORE DIGITS
063.270 341 3543X   POP   H
063.271 311 3544X   RET
3545X

```

063.272		3546X	TDDA	EQU	*
063.272	377 377	3547X		DW	-1
063.274	366 377	3548X		DW	-10
063.276	234 377	3549X		DW	-100
063.300	030 374	3550X		DW	-1000
063.302	360 330	3551X		DW	-10000
063.304		3552		XTEXT	DADA

3554X	**	\$DADA - PERFORM (H,L) = (H,L) + (O,A)
3555X	*	
3556X	*	ENTRY (H,L) = BEFORE VALUE
3557X	*	(A) = BEFORE VALUE
3558X	*	EXIT (H,L) = (H,L) + (O,A)
3559X	*	'C' SET IF OVERFLOW
3560X	*	USES F,H,L
3561X		

030.072		3562X			
063.304		3563X	\$DADA	EQU	30072A IN H17 ROM
		3564		XTEXT	CRLF

3566X	**	\$CRLF - TYPE CARRIAGE RETURN/ LINE FEED
3567X	*	
3568X	*	\$CRLF IS USED TO GENERATE PADDED CRLF'S.
3569X	*	
3570X	*	ENTRY NONE
3571X	*	EXIT (A) = 0
3572X	*	USES A,F
3573X		
3574X		

063.304	076 012	3575X	\$CRLF	HVI	A,NL
063.306	377 002	3576X		DB	SYSCALL, SCOUT
063.310	257	3577X		XRA	A
063.311	311	3578X		RET	

```

3582 **      CGT - COUNT GAP TIMING.
3583 *
3584 *      CGT COMPUTES THE GAP TIMING BY COUNTING THE NUMBER OF
3585 *      TWO MILLISECOND DELAY INTERVALS IN THE INTER-RECORD GAP.
3586 *      THE TIME FOR THE CURRENT HOLE TO PASS IS INCLUDED IN
3587 *      THE COUNT.
3588 *
3589 *      ENTRY  DISK OVER BEGINNING OF HOLE
3590 *      EXIT   (HL) = 'COUNT'
3591 *      USES   (FSW),(DE),(HL)
3592 *
3593 *
063.312 052 033 040 3594 CGT  LHL0  .TICCNT
3595
063.315 333 177 3596 CGT1  IN    DF,DC
000.000 3597      ERRNZ DF,HD-1
063.317 037 3598      RAR
063.320 332 315 063 3599      JC    CGT1          HOLE PRESENT
3600
063.323 315 224 030 3601      CALL  $CHL
063.326 124 3602      MOV   D,H          SAVE (-START) TIC COUNT
063.327 135 3603      MOV   E,L          WHILE WE HAVE THE TIME!
3604
063.330 333 177 3605 CGT2  IN    DF,DC
000.000 3606      ERRNZ DF,HD-1
063.332 037 3607      RAR
063.333 322 330 063 3608      JNC  CGT2          COUNT HOLE NOT PRESENT
3609
063.336 052 033 040 3610      LHL0  .TICCNT          GET CURRENT TIC COUNT
063.341 031 3611      DAD   D          DELAY = CURRENT + (-START)
063.342 311 3612      RET

```

  

```

3614 **      SYDD, - CALL SYSTEM DEVICE DRIVER, WITH ERROR DETECTION.
3615 *
3616 *      SYDD, PASSES CALLS TO SYDD, CHECKING FOR RETURNED ERRORS.
3617 *
3618 *      ENTRY  FOR SYDD
3619 *      EXIT   FROM SYDD
3620 *      USES   AS SYDD
3621 *
3622 *
063.343 315 130 040 3623 SYDD. CALL  SYDD
063.346 320 3624      RNC          ALL OK
063.347 315 136 031 3625      CALL  $TYPTX
063.352 007 040 103 3626      DB   BELL, ' CANNOT READ/WRITE THIS DISK ! FATAL ERROR !',BELL,ENL
064.031 303 133 051 3627      JMP   EXIT          EXIT

```

```

3629 **      WIH - WAIT INDEX HOLE.
3630 *
3631 *      WIH WAITS UNTIL THE INDEX HOLE HAS PASSED THE SENSOR.
3632 *
3633 *      NOTE:  THIS ROUTINE ASSUMES THAT THE DRIVE WILL NOT BE THAT FAR
3634 *            OFF IN TOLERANCE AS IT DOES NOT EVEN CHECK THE HIGH ORDER
3635 *            BYTE OF THE COUNTY, WHICH SHOULD BE ZERO!
3636 *
3637 *
3638 *      ENTRY  DISK MOVING
3639 *      EXIT   INDEX HOLE JUST PAST (DISK IN INTERHOLE GAP)
3640 *      USES   A,F,H,L
3641
3642
064.034 325 3643 WIH  PUSH  D
064.035 315 271 036 3644 CALL  R,WNH          WAIT NO HOLE
064.040 315 235 036 3645 WIH1 CALL  R,WHD          WAIT HOLE DETECT
064.043 315 312 063 3646 CALL  CGT            COMPUTE GAP TIME
064.046 175 3647 MOV   A,L
064.047 376 006 3648 CPI   1000/100/2+1
064.051 332 040 064 3649 JC    WIH1          DONT HAVE FULL SECTOR TIMING
3650
3651 *      HAVE FULL SECTOR TIMING. WAIT FOR SHORT GAP NOW
3652
064.054 315 312 063 3653 WIH2 CALL  CGT
064.057 175 3654 MOV   A,L
064.060 376 006 3655 CPI   1000/100/2+1
064.062 322 054 064 3656 JNC  WIH2          NOT YET
064.065 321 3657 POP  D
064.066 303 271 036 3658 JMP  R,WNH          GOT INDEX HOLE, WAIT FOR PASS AND RETURN

```



```

3661
3662
064.071 000 3663 GRTBLK DB 0 BLOCK NUMBER CONTAINING GRT
064.072 000 3664 DIRBLK DB 0 BLOCK NUMBER OF FIRST DIRECTORY BLOCK
064.073 000 3665 DIRLBLK DB 0 BLOCK NUMBER OF LAST DIRECTORY BLOCK
3666
3667
3668
3669
3670 ** DISK LABEL
3671
064.074 3672 LABEL DS 0
000.000 3673 ERRNZ *-LABEL-LAB.SER SERIAL NUMBER
064.074 000 3674 DB 0
000.000 3675 ERRNZ *-LABEL-LAB.IND CREATION DATE
064.075 000 000 3676 DW 0
000.000 3677 ERRNZ *-LABEL-LAB.DIS INDEX OF FIRST DIRECTORY SECTOR
064.077 000 000 3678 DW 0
000.000 3679 ERRNZ *-LABEL-LAB.GRT GRT TABLE INDEX
064.101 000 000 3680 DW 0
000.000 3681 ERRNZ *-LABEL-LAB.SPG TWO SECTORS PER GROUP
064.103 002 3682 DB 2
000.000 3683 ERRNZ *-LABEL-LAB.VLT VOLUME TYPE
064.104 000 3684 DB LAB.DAT
000.000 3685 ERRNZ *-LABEL-LAB.VER VERSION OF INIT17 TO INIT DISK
064.105 026 3686 DB VERS
3687
064.106 000 000 000 3688 DB 0,0,0,0,0,0,0,0 UNUSED
000.000 3689 ERRNZ *-LABEL-LAB.LAB
064.115 3690 DS 60 LABEL
064.211 000 000 3691 DB 0,0 END OF LABEL
064.213 040 040 040 3692 DB '
064.220 015 012 3693 DB CR,LF
064.222 123 131 123 3694 DB 'SYSTEM COPYRIGHT HEATH CO., 10/1977, 79/4'
064.273 015 012 3695 DB CR,LF
064.275 040 102 131 3696 DB ' BY JGL, 10/1977'
064.315 057 147 143 3697 DB '/sc'
064.321 3698 DS 0 END OF LABEL
3699
064.321 050 3700 MAXTRK DB 40 MAX # OF TRACKS
3701
064.322 3702 MEML EQU * MAX MEMORY FOR LOAD
3703
3704
064.322 3705 PATCH DS 64 PATCH AREA
3707
3708 ** DIRECTORY SECTOR
3709
065.022 3710 IDDA DS 0 DIRECTORY SECTOR FWA
3711
065.022 3712 IDDB DS 22*DIRELEN SPACE FOR REST OF BLOCK
001.372 3713 IDDBL EQU *-IDDB
067.014 3714 DS 1 SPACE FOR 00 BYTE
000.000 3715 ERRNZ *-IDDA-DIS.ENL
067.015 3716 DS 1 LENGTH OF ENTRIES (DIRELEN)

```

```
000.000 3717 ERRNZ *-IDDA-DIS.SEC
067.016 3718 DS 2 SECTOR NUMBER OF BEGINNING OF THIS BLOCK
000.000 3719 ERRNZ *-IDDA-DIS.LNK
067.020 3720 DS 2 SECTOR NUMBER OF BEGINNING OF NEXT BLOCK
000.000 3721 ERRNZ *-IDDA-512 SHOULD FILL BLOCK
3722
3723 ** RGTAB AND GRTAB MUST BE ON EVEN PAGE BOUNDARYS.
067.022 3724 LINE DS 81
000.120 3725 LINEL EQU *-LINE-1
067.022 3726 ITLA EQU LINE USE LINE FOR BUFFER
3727
067.143 3728 DS **+255/256*256-*
3729
070.000 3730 GRTAB DS 256 GRT TABLE
071.000 3731 RGTAB DS 256 RGT TABLE
3732
072.000 3733 RMEML DS 0 RUNNING MEMORY MAX LEN
3734
072.000 3735
3736 END
ASSEMBLY COMPLETE
3736 STATEMENTS
0 ERRORS DETECTED
9558 BYTES FREE
```



INIT - INITIALIZE DISK  
CROSS REFERENCE TABLE

XREF V1.1  
PAGE 84

.ALED5	040013	114E	1665	1667	1738
.CHFLG	000060	596L			
.CLEAR	000055	593L			
.CLEARA	000056	594L			
.CLOSE	000046	586L	2050		
.CLRCO	000007	570L			
.CONSL	000004	569L	3039		
.CRC	002347	97E			
.CRCSUM	040027	117E			
.CTC	002172	91E			
.CTL	000041	581L			
.CTLFLG	040011	113E			
.DECODE	000053	591L			
.DELET	000050	588L			
.DISMT	000061	597L			
.DLED5	040021	115E			
.DLY	000053	86E	1692	1730	1892
.DMNMS	000203	608L			
.DMOUN	000201	606L	3253		
.DOD	003122	100E			
.DODA	003356	102E			
.DSPMOD	040007	111E			
.DSFROT	040006	110E			
.DUMP	001374	88E			
.ERRDR	000057	595L	2095		
.EXIT	000000	563L	2085		
.HORN	002140	90E	1669		
.IDENT	000000	85E			
.IQWRK	040002	108E			
.LINK	000040	580L			
.LOAD	001267	87E			
.LOADD	000062	598L			
.LOADO	000010	571L	3227	3230	
.MFLAG	040010	112E	1659	1662	1752
.MDNMS	000202	607L			
.MOUNT	000200	605L			
.NAME	000054	592L			
.OPENC	000045	585L			
.OPENR	000042	582L			
.OPENU	000044	584L			
.OPENW	000043	583L			
.PCHL	002264	93E			
.POSIT	000047	587L			
.PRINT	000003	566L			
.RCK	003260	101E			
.READ	000004	567L			
.REGI	040005	109E			
.REGPTR	040035	120E			
.RENAM	000051	589L			
.RESET	000204	609L			
.RNB	002331	96E			
.RNP	002325	95E			
.SCIN	000001	564L	3297		
.SCOUT	000002	565L	3301	3322	3576
.SETTP	000052	590L	2057		
.SRS	002265	94E			
.START	040000	107E			
.SYSRES	000012	573L			







## INIT - INITIALIZE DISK

XREF V1.1

## CROSS REFERENCE TABLE

PAGE 88

DC,REA	000000	334L	1492	1943					
DC,RER	000002	336L	1618	2305					
DC,WRI	000001	335L	2497	2623	2636	2647	2886	2901	
DDF,BOL	000011	460E							
DDF,BDD	000000	459L							
DDF,LAB	000011	461L	1617	2304	2635				
DDF,SGT	000012	462L	2605	2606	2622	2743	2925	2927	
DDF,USR	000014	463L							
DEV,DIA	000004	424L							
DEV,DVG	000016	436L							
DEV,RVL	000014	435L							
DEV,FLG	000006	425L							
DEV,JMP	000003	423L							
DEV,MNU	000011	432L							
DEV,MUM	000010	431L							
DEV,NAM	000000	415L							
DEV,RES	000002	419L							
DEV,SPG	000007	430L							
DEV,UNT	000012	433L							
DEVELEN	000017	438E							
DF,CLR	000376	377E	2892	2950					
DF,DI	000040	194E							
DF,DS0	000002	190E	2192	2421	2426				
DF,DS1	000004	191E							
DF,DS2	000010	192E							
DF,EMP	000377	376E	2827						
DF,HD	000001	184E	2250	2257	3597	3606			
DF,MO	000020	193E	2192	2421	2426				
DF,SD	000010	187E							
DF,ST	000100	195E							
DF,TO	000002	185E							
DF,WG	000001	189E	2426						
DF,WF	000004	186E							
DF,WR	000200	196E							
DIF,CNT	000020	491E	1055	2922	2937				
DIF,LOC	000100	489E	2922	2937	2946				
DIF,SYS	000200	488E	2922	2937	2946				
DIF,WP	000040	490E	2922	2937	2946				
DIR,ALI	000025	392L							
DIR,CLU	000015	385L	2919						
DIR,CSD	000023	391L	2930						
DIR,EXT	000010	380L	2915						
DIR,FAN	000020	388L	1057	2924					
DIR,FLG	000016	386L	1052	1057	2921				
DIR,LGN	000021	389L	2926						
DIR,LSI	000022	390L	2928						
DIR,NAM	000000	379L	1173	2913					
DIR,PRD	000013	381L	2917						
DIR,VER	000014	382L							
DIRBLK	064072	2714	2758	3664L					
DIRELEN	000027	394E	402	852	1462	2832	2952	2955	2981
DIRIDL	000015	383E	1026	1036					
DIRLBLK	064073	2786	2874	3665L					
DIRSTRT	000001	2715	2754E	2759					
DIS,ENL	001373	406L	2833	3715					
DIS,ENT	000000	401E	1451						
DIS,LNK	001374	408L	1470	2846	2903	3719			
DIS,SEC	001374	407L	2835	2885	2900	3717			















INIT - INITIALIZE DISK  
CROSS REFERENCE TABLE

XREF V1.1  
PAGE 95

UCI.RD	000040	251E																		
UCI.TE	000001	255E	1862																	
UDR	000000	227E	1154	1323																
UF.FCT	000100	211E																		
UF.RDA	000001	208E																		
UF.RDR	000002	209E																		
UF.RPE	000004	210E																		
UF.TRM	000200	212E																		
UMI.16X	000002	245E	1860																	
UMI.1B	000100	235E	1855	1860																
UMI.1X	000001	244E																		
UMI.2B	000300	237E	1855																	
UMI.44X	000003	246E																		
UMI.HB	000200	236E																		
UMI.LS	000000	240E																		
UMI.L6	000004	241E																		
UMI.L7	000010	242E																		
UMI.L8	000014	243E	1860																	
UMI.PA	000020	239E																		
UMI.PE	000040	238E																		
UNT.IIS	000005	447L																		
UNT.FLG	000000	444L																		
UNT.GRI	000001	445L																		
UNT.GTS	000003	446L																		
UNT.SIZ	000007	449E																		
UD.CLK	000001	72E																		
UD.DBU	000002	71E	1661																	
UD.HLT	000200	69E																		
UD.NFR	000100	70E																		
UP.DP	000174	202E																		
UP.FC	000175	203E	931	2414																
UP.SC	000176	205E																		
UP.SR	000176	206E																		
UP.ST	000175	204E																		
UR.DLL	000000	276E	1681	1880																
UR.DLM	000001	278E	1684	1883																
UR.IER	000001	280E	1513	1674	1873															
UR.IIR	000002	286E																		
UR.LCR	000003	290E	1519	1520	1678	1690	1878	1890												
UR.LSR	000005	309E	1141	1328	1700	1732	1868													
UR.MCR	000004	302E	1676	1693	1695	1875	1894	1896												
UR.MSR	000006	318E																		
UR.RBR	000000	272E	1160	1699	1717	1731	1750	1893												
UR.THR	000000	274E	1332																	
USERFWA	042200	547E	910	914	915															
USR	000001	228E	1132	1319	1514	1847	1848	1849	1850	1852	1861	1863								
USR.FE	000040	259E																		
USR.OE	000020	260E																		
USR.PE	000010	261E																		
USR.RXR	000002	263E	1133																	
USR.TXE	000004	262E																		
USR.TXR	000001	264E	1320																	
VERS	000026	554E	2047	2115	2115	3686														
VOLSIZ	001220	34E	2688																	
WIS	054124	2194	2218L																	
WIH	064034	2430	3643L																	
WIH1	064040	3645L	3649																	
WIH2	064054	3653L	3656																	

15924 BYTES FREE

