

13:57:38 14-MAY-80

000.001

```
2 DEBUG EQU 1           DON'T ASSEMBLE FOR DEBUG
3 *** SYSTEM I/O HANDLER.
4 *
5 * JG LETWIN, 10/77
6 *
7 * COPYRIGHT HEATH COMPANY.
8 *
9 * G. Chandler, 78/10 Maintenance Release.
10 * 79/04/
11 * 79/05 > Release. $50.04.00
12 * 79/06/
13 * 79/10 Release. $50.05.00
14 *
15 *
16 *
17 *** THE SYSTEM I/O HANDLER HANDLES SYSTEM REQUESTS FOR
18 * READS AND WRITES.
19 *
20 * IF A MASS STORAGE DEVICE, THIS DOES THE CORRECT STORAGE
21 * MANAGEMENT. IF A SERIAL DEVICE, THE COMMAND IS PASSED
22 * ONTO THE DEVICE DRIVER.
```

25  
26  
27 \*\* MACHINE INSTRUCTIONS  
28  
000.376 29 MI.CPI EQU 376Q  
000.303 30 MI.JMP EQU 303Q  
000.311 31 MI.RET EQU 311Q  
32  
33  
34 \*\* SYSTEM SYMBOLS  
35  
000.000 36 XTEXT U8250

38X \*\* 8250 UART CONTROL AND BIT DEFINITIONS.  
39X  
000.350 40X SC.ACE EQU 350Q SYSTEM CONSOLE PORT IF 8250 ACE  
000.156 41X AC.DLY EQU 110 220 MIL. SEC. DELAY FOR 8250  
42X  
000.000 43X UR.RBR EQU 0 RECEIVER BUFFER REGISTER (READ ONLY)  
44X  
000.000 45X UR.THR EQU 0 TRANSMITTER HOLDING REGISTER (WRITE ONLY)  
46X  
000.000 47X UR.DLL EQU 0 DIVISOR LATCH (LEAST SIGNIFICANT)  
48X  
000.001 49X UR.DLM EQU 1 DIVISOR LATCH (MOST SIGNIFICANT)  
50X  
000.001 51X UR.IER EQU 1 INTERRUPT ENABLE REGISTER  
000.001 52X UC.EDA EQU 00000001B ENABLE RECEIVED DATA AVAILABLE INTERRUPT  
000.002 53X UC.TRE EQU 00000010B ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT  
000.004 54X UC.RSI EQU 00000100B ENABLE RECEIVE STATUS INTERRUPT  
000.010 55X UC.MSI EQU 00001000B ENABLE MODEM STATUS INTERRUPT  
56X  
000.002 57X UR.IIR EQU 2 INTERRUPT IDENTIFICATION REGISTER  
000.001 58X UC.IIP EQU 00000001B INVERTED INTERRUPT PENDING (0 MEANS PENDING)  
000.006 59X UC.IID EQU 00000110B INTERRUPT ID  
60X  
000.003 61X UR.LCR EQU 3 LINE CONTROL REGISTER  
000.000 62X UC.SBW EQU 00000000B 5 BIT WORDS  
000.001 63X UC.6BW EQU 00000001B 6 BIT WORDS  
000.002 64X UC.7BW EQU 00000010B 7 BIT WORDS  
000.003 65X UC.8BW EQU 00000011B 8 BIT WORDS  
000.004 66X UC.2SB EQU 00000100B TWO STOP BITS SELECTED  
000.010 67X UC.PEN EQU 00001000B PARITY COMPUTATION ENABLED  
000.020 68X UC.EPS EQU 00010000B EVEN PARITY SELECT  
000.040 69X UC.SKP EQU 00100000B STICK PARITY  
000.100 70X UC.SB EQU 01000000B SET BREAK  
000.200 71X UC.DLA EQU 10000000B DIVISOR LATCH ACCESS  
72X  
000.004 73X UR.MCR EQU 4 MODEM CONTROL REGISTER  
000.001 74X UC.DTR EQU 00000001B DATA TERMINAL READY  
000.002 75X UC.RTS EQU 00000010B REQUEST TO SEND  
000.004 76X UC.OU1 EQU 00000100B OUT 1  
000.010 77X UC.OU2 EQU 00001000B OUT 2

000.020	78X UC.LOO	EQU	00010000B	LOOP
	79X			
000.005	80X UC.LSR	EQU	5	LINE STATUS REGISTER
000.001	81X UC.DR	EQU	00000001B	DATA READY
000.002	82X UC.DR	EQU	00000010B	OVERRUN
000.004	83X UC.PE	EQU	00000100B	PARITY ERROR
000.010	84X UC.FE	EQU	00001000B	FRAMING ERROR
000.020	85X UC.BI	EQU	00010000B	BREAK INTERRUPT
000.040	86X UC.THE	EQU	00100000B	TRANSMITTER HOLDING REGISTER EMPTY
000.100	87X UC.TSE	EQU	01000000B	TRANSMITTER SHIFT REGISTER EMPTY
	88X			
000.006	89X UC.MSR	EQU	6	MODEM STATUS REGISTER
000.001	90X UC.DCS	EQU	00000001B	DELTA CLEAR TO SEND
000.002	91X UC.DDR	EQU	00000010B	DELTA DATA SET READY
000.004	92X UC.TER	EQU	00000100B	TRAILING EDGE OF RING
000.010	93X UC.DRL	EQU	00001000B	DELTA RECEIVE LINE SIGNAL DETECT
000.020	94X UC.CTS	EQU	00010000B	CLEAR TO SEND
000.040	95X UC.DSR	EQU	00100000B	DATA SET READY
000.100	96X UC.RI	EQU	01000000B	RING INDICATOR
000.200	97X UC.RLS	EQU	10000000B	RECEIVED LINE SIGNAL DETECT
000.000	98	XTEXT	U8251	

101X \*\* 8251 USART BIT DEFINITIONS.

102X \*

103X

104X \*\* PORT ADDRESSES

105X

000.000	106X UDR	EQU	0	DATA REGISTER IS EVEN
000.001	107X USR	EQU	1	STATUS REGISTER IS NEXT

000.372	109X SCUART	EQU	3720	CONSOLE USART ADDRESS (IFF 8251)
---------	-------------	-----	------	----------------------------------

110X

111X

112X \*\* MODE INSTRUCTION CONTROL BITS.

113X

000.100	114X UMI.1B	EQU	01000000B	1 STOP BIT
000.200	115X UMI.HB	EQU	10000000B	1 1/2 STOP BITS
000.300	116X UMI.2B	EQU	11000000B	2 STOP BITS
000.040	117X UMI.PE	EQU	00100000B	EVEN PARITY
000.020	118X UMI.PA	EQU	00010000B	USE PARITY
000.000	119X UMI.L5	EQU	00000000B	5 BIT CHARACTERS
000.004	120X UMI.L6	EQU	00000100B	6 BIT CHARACTERS
000.010	121X UMI.L7	EQU	00001000B	7 BIT CHARACTERS
000.014	122X UMI.L8	EQU	00001100B	8 BIT CHARACTERS
000.001	123X UMI.1X	EQU	00000001B	CLOCK X 1
000.002	124X UMI.16X	EQU	00000010B	CLOCK X 16
000.003	125X UMI.64X	EQU	00000011B	CLOCK X 64

126X

127X \*\* COMMAND INSTRUCTION BITS.

128X

000.100	129X UCI.IR	EQU	01000000B	INTERNAL RESET
000.040	130X UCI.RD	EQU	00100000B	READER-ON CONTROL FLAG
000.020	131X UCI.ER	EQU	00010000B	ERROR RESET
000.004	132X UCI.RE	EQU	00000100B	RECEIVE ENABLE
000.002	133X UCI.IE	EQU	00000010B	ENABLE INTERRUPTS FLAG
000.001	134X UCI.TE	EQU	00000001B	TRANSMIT ENABLE

135X

136X \*\* STATUS READ COMMAND BITS.

137X

000.040	138X USR.FE	EQU	00100000B	FRAMING ERROR
000.020	139X USR.OE	EQU	00010000B	OVERRUN ERROR
000.010	140X USR.PE	EQU	00001000B	PARITY ERROR
000.004	141X USR.TXE	EQU	00000100B	TRANSMITTER EMPTY
000.002	142X USR.RXR	EQU	00000010B	RECEIVER READY
000.001	143X USR.TXR	EQU	00000001B	TRANSMITTER READY

000.000 144 XTEXT H17DEF

146X \*\* H17 CONTROL INFORMATION.

147X

000.177	148X DF.DC	EQU	07FH	DISK CONTROL PORT
	149X			
000.001	150X DF.HD	EQU	00000001B	HOLE DETECT
000.002	151X DF.T0	EQU	00000010B	TRACK 0 DETECT
000.004	152X DF.WP	EQU	00000100B	WRITE PROTECT
000.010	153X DF.SD	EQU	00001000B	SYNC DETECT

154X

000.001	155X DF.WG	EQU	00000001B	WRITE GATE ENABLE
000.002	156X DF.DSO	EQU	00000010B	DRIVE SELECT 0
000.004	157X DF.DS1	EQU	00000100B	DRIVE SELECT 1
000.010	158X DF.DS2	EQU	00010000B	DRIVE SELECT 2
000.020	159X DF.MO	EQU	00010000B	MOTOR ON (BOTH DRIVES)
000.040	160X DF.DI	EQU	00100000B	DIRECTION (0=OUT)
000.100	161X DF.ST	EQU	01000000B	STEP COMMAND (ACTIVE HIGH)
000.200	162X DF.WR	EQU	10000000B	WRITE ENABLE RAM

163X

164X

165X

166X \*\* DISK, UART, PORTS AND CONTROL FLAGS.

167X

000.174	168X UF.DP	EQU	07CH	DATA PORT
000.175	169X UF.FC	EQU	07DH	FILL CHARACTER
000.175	170X UF.ST	EQU	07DH	STATUS FLAGS
000.176	171X UF.SC	EQU	07EH	SYNC CHARACTER (OUTPUT)
000.176	172X UF.SR	EQU	07EH	SYNC RESET (INPUT)
	173X			
000.001	174X UF.RDA	EQU	00000001B	RECEIVE DATA AVAILABLE
000.002	175X UF.ROR	EQU	00000010B	RECEIVER OVERRUN
000.004	176X UF.RPE	EQU	00000100B	RECEIVER PARITY ERROR
000.100	177X UF.FCT	EQU	01000000B	FILL CHAR TRANSMITTED
000.200	178X UF.TBM	EQU	10000000B	TRANSMITTER BUFFER EMPTY

179X

180X

181X

182X \*\* CHARACTER DEFINITIONS.

183X	184X C.DSYN	EQU	0FDH	PREFIX SYNC CHARACTER
000.375	185	XTEXT	ASCII	

187X \*\* ASCII CHARACTER EQUIVALENCES.

188X

000.015	189X CR	EQU	13	CARRIAGE RETURN
000.012	190X LF	EQU	10	LINE FEED
000.200	191X NULL	EQU	2000	PAD CHARACTER
000.000	192X NUL2	EQU	0	
000.007	193X BELL	EQU	7	BELL CHARACTER
000.177	194X RUBOUT	EQU	177Q	
000.010	195X BKSP	EQU	10Q	CTL-H
000.026	196X C.SYN	EQU	26Q	SYNC
000.002	197X C.STX	EQU	2	STX
000.047	198X QUOTE	EQU	47Q	
000.011	199X TAB	EQU	11Q	
000.033	200X ESC	EQU	33Q	
000.012	201X NL	EQU	120	NEW LINE (HDOS SYSTEMS)
000.212	202X ENL	EQU	NL+200Q	NL + END-OF-LINE-FLAG
000.014	203X FF	EQU	14Q	FORM FEED
000.001	204X CTLA	EQU	01Q	CTL-A
000.002	205X CTLB	EQU	02Q	CTL-B
000.003	206X CTLC	EQU	03Q	CTL-C
000.004	207X CTLD	EQU	04Q	CTL-D
000.017	208X CTL0	EQU	17Q	CTL-O
000.020	209X CTLF	EQU	20Q	CTL-P

8251 USART BIT DEFINITIONS,

ASCII

13:58:01 16-MAY-80

000.021	210X	CTLQ	EQU	21Q	CTL-Q
000.023	211X	CTL\$	EQU	23Q	CTL-S
000.032	212X	CTLZ	EQU	32Q	CTL-Z
000.000	213	XTEXT	MTR		

216X \*\* MTR - PAM/8 EQUIVALENCES.

217X \*

218X \* THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO  
219X \* MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.

221X \*\* IO PORTS

222X

000.360	223X	IP.PAD	EQU	360Q	PAD INPUT PORT
000.360	224X	OP.CTL	EQU	360Q	CONTROL OUTPUT PORT
000.360	225X	OP.DIG	EQU	360Q	DIGIT SELECT OUTPUT PORT
000.361	226X	OP.SEG	EQU	361Q	SEGMENT SELECT OUTPUT PORT

228X \*\* FRONT PANEL CONTROL BITS.

229X

000.020	230X	CB.SSI	EQU	00010000B	SINGLE STEP INTERRUPT
000.040	231X	CB.MTL	EQU	00100000B	MONITOR LIGHT
000.100	232X	CB.CLI	EQU	01000000B	CLOCK INTERRUPT ENABLE
000.200	233X	CB.SPK	EQU	10000000B	SPEAKER ENABLE

235X \*\* MONITOR MODE FLAGS.

236X

000.000	237X	DM.MR	EQU	0	MEMORY READ
000.001	238X	DM.MW	EQU	1	MEMORY WRITE
000.002	239X	DM.RR	EQU	2	REGISTER READ
000.003	240X	DM.RW	EQU	3	REGISTER WRITE

242X \*\* USER OPTION BITS.

243X \*

244X \* THESE BITS ARE SET IN CELL .MFLAG.

245X

000.200	246X	UO.HLT	EQU	10000000B	DISABLE HALT PROCESSING
000.100	247X	UO.NFR	EQU	CB.CLI	NO REFRESH OF FRONT PANEL
000.002	248X	UO.IDU	EQU	00000010B	DISABLE DISPLAY UPDATE
000.001	249X	UO.CLK	EQU	00000001B	ALLOW PRIVATE INTERRUPT PROCESSING

251X \*\* MONITOR IDENTIFICATION FLAGS.

252X \*

253X \* THESE BYTES IDENTIFY THE ROM MONITOR.

254X \* THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT

255X

000.021	256X	M.PAMB	EQU	0210	'LXI' INSTRUCTION AT 000.000 IN PAM-B
000.303	257X	M.FOX	EQU	3030	'JMP' INSTRUCTION AT 000.000 IN FOX ROM

ENTRY 13:58:04 16-MAY-80

## 259X \*\* ROUTINE ENTRY POINTS.

260X \* 261X

000.000	262X	.IDENT	EQU	0000A	IDENTIFICATION LOCATION
000.053	263X	.DLY	EQU	0053A	DELAY
001.267	264X	.LOAD	EQU	1267A	TAPE LOAD
001.374	265X	.DUMP	EQU	1374A	TAPE DUMP
002.136	266X	.ALARM	EQU	2136A	ALARM ROUTINE
002.140	267X	.HORN	EQU	2140A	HORN
002.172	268X	.CTC	EQU	2172A	CHECK TAPE CHECKSUM
002.205	269X	.TPERR	EQU	2205A	TAPE ERROR ROUTINE
002.264	270X	.PCHL	EQU	2264A	PCHL INSTRUCTION
002.265	271X	.SRS	EQU	2265A	SCAN RECORD START
002.325	272X	.RNP	EQU	2325A	READ NEXT PAIR
002.331	273X	.RNB	EQU	2331A	READ NEXT BYTE
002.347	274X	.CRC	EQU	2347A	CRC-16 CALCULATOR
003.017	275X	.WNP	EQU	3017A	WRITE NEXT PAIR
003.024	276X	.WNB	EQU	3024A	WRITE NEXT BYTE
003.122	277X	.DOD	EQU	3122A	DECODE FOR OCTAL DISPLAY
003.260	278X	.RCK	EQU	3260A	READ CONSOLE KEYSET
003.356	279X	.DODA	EQU	3356A	SEGMENT CODE TABLE

## 281X \*\* RAM CELLS USED BY HBMTR.

282X \*

283X

040.000	284X	.START	EQU	40000A	START DUMP ADDRESS
040.002	285X	.IOWRK	EQU	40002A	IN OR OUT INSTRUCTION
040.005	286X	.REGI	EQU	40005A	DISPLAYED REGISTER INDEX
040.006	287X	.DSPROT	EQU	40006A	PERIOD FLAG BYTE
040.007	288X	.DSPMOD	EQU	40007A	DISPLAY MODE
040.010	289X	.MFLAG	EQU	40010A	USER OPTION BYTE
040.011	290X	.CTLFLG	EQU	40011A	PANEL CONTROL BYTE
040.013	291X	.ALEDS	EQU	40013A	ABUSS LEDS
040.021	292X	.DLEDIS	EQU	40021A	DBUSS LEDS
040.024	293X	.ABUSS	EQU	40024A	ABUSS REGISTER
040.027	294X	.CRCSUM	EQU	40027A	CRCSUM WORD
040.031	295X	.TPERRX	EQU	40031A	TAPE ERROR EXIT VECTOR
040.033	296X	.TICCNT	EQU	40033A	CLOCK TICK COUNTER
040.035	297X	.REGPTR	EQU	40035A	REGISTER POINTER
040.037	298X	.UIVEC	EQU	40037A	USER INTERRUPT VECTORS
000.000	299	XTEXT		HDSROM	

## 301X \*\* HDOS H17 ROM ENTRY POINTS.

302X ORG 31253A

303X DWRITE EQU \*

031.253	304X	DS		31256A-31253A
031.256	305X	DREAD	EQU	*
031.256	306X	DS		31275A-31256A
031.275	307X	S.READ	EQU	*
031.275	308X	DS		31321A-31266A
031.330	309X	S.WRITE	EQU	*
031.330	310X	DS		31325A-31311A

031.344	311X	ERR.FNO	EQU	*
031.344	312X		DS	31331A-31325A
031.350	313X	ERR.ILR	EQU	*
031.350	314X		DS	31335A-31331A
031.354	315X	CFF	EQU	*
031.354	316X		DS	31363A-31335A
032.002	317X	ICA	EQU	*
032.002	318X		DS	32114A-31363A
032.133	319X	FFB	EQU	*
032.133	320X		DS	32166A-32114A
032.205	321X	FFL	EQU	*
032.205	322X		DS	32204A-32166A
	323X	*LDD	EQU	*
032.223	324X		DS	32372A-32204A+1
033.012	325X	LDO	EQU	*
033.012	326X		DS	33135A-33002A
033.145	327X	PBI	EQU	*
033.145	328X		DS	33154A-33124A
033.175	329X	REL.	EQU	*
033.175	330X		DS	33156A-33154A
033.177	331X	REL	EQU	*
033.177	332X		DS	33212A-33156A
033.233	333X	TFE	EQU	*
033.233	334X		DS	33232A-33206A
033.257	335X	RUC	EQU	*
033.257	336		XTEXT	.FILDEF

## 338X \*\* FILDEF - FILE TYPE DEFINITIONS.

339X	*		
340X	*	DB	377Q,FT,XXX

341X

342X

000.000	343X	FT.ABS	EQU	0	ABSOLUTE BINARY
000.001	344X	FT.PIC	EQU	1	POSITION INDEPENDANT CODE
000.002	345X	FT.REL	EQU	2	RELOCATABLE CODE
000.003	346X	FT.BAC	EQU	3	COMPILED BASIC CODE
033.257	347		XTEXT	.HOSDEF	

## 349X \*\* .HOSDEF - DEFINE HOS PARAMETER.

350X \*

351X

352X

000.026	353X	VERS	EQU	1*16+6	VERSION 1.6
---------	------	------	-----	--------	-------------

354X

000.377	355X	SYSCALL	EQU	3779	SYSCALL INSTRUCTION
---------	------	---------	-----	------	---------------------

356X

357X

000.000	358X	ORG	0	
---------	------	-----	---	--

359X

360X	*	RESIDENT FUNCTIONS
------	---	--------------------

361X

000.000	362X	.EXIT	DS	1	EXIT (MUST BE FIRST)
000.001	363X	.SCIN	DS	1	SCIN
000.002	364X	.SCOUT	DS	1	SCOUT
000.003	365X	.PRINT	DS	1	PRINT
000.004	366X	.READ	DS	1	READ
000.005	367X	.WRITE	DS	1	WRITE
000.006	368X	.CONSL	DS	1	SET/CLEAR CONSOLE OPTIONS
000.007	369X	.CLRRCO	DS	1	CLEAR CONSOLE BUFFER
000.010	370X	.LOADO	DS	1	LOAD AN OVERLAY
000.011	371X	.VERS	DS	1	RETURN HDOS VERSION NUMBER
000.012	372X	.SYSRES	DS	1	PRECEDING FUNCTIONS ARE RESIDENT
	373X				
	374X				

## 375X \* \*HDOSOVLO.SYS\* FUNCTIONS

376X					
000.040	377X	ORG	40A		
	378X				
000.040	379X	.LINK	DS	1	LINK (MUST BE FIRST)
000.041	380X	.CTL-C	DS	1	CTL-C
000.042	381X	.OPENR	DS	1	OPENR
000.043	382X	.OPENW	DS	1	OPENW
000.044	383X	.OPENU	DS	1	OPENU
000.045	384X	.OPENC	DS	1	OPENC
000.046	385X	.CLOSE	DS	1	CLOSE
000.047	386X	.POSIT	DS	1	POSITION
000.050	387X	.DELETE	DS	1	DELETE
000.051	388X	.RENAM	DS	1	RENAME
000.052	389X	.SETTP	DS	1	SETTOP
000.053	390X	.DECODE	DS	1	NAME DECODE
000.054	391X	.NAME	DS	1	GET FILE NAME FROM CHANNEL
000.055	392X	.CLEAR	DS	1	CLEAR CHAN
000.056	393X	.CLEARA	DS	1	CLEAR ALL CHANS
000.057	394X	.ERROR	DS	1	LOOKUP ERROR
000.060	395X	.CHFLG	DS	1	CHANGE FLAGS
000.061	396X	.DISMT	DS	1	FLAG SYSTEM DISK DISMOUNTED
000.062	397X	.LOADD	DS	1	LOAD DEVICE DRIVER
	398X				

## 399X \* \*HDOSOVLI.SYS\* FUNCTIONS

400X					
000.200	401X				
	402X	ORG	2000		
	403X				
000.200	404X	.MOUNT	DS	1	MOUNT (MUST BE FIRST)
000.201	405X	.DMOUN	DS	1	DISMOUNT
000.202	406X	.MONMS	DS	1	MOUNT/NO MESSAGE
000.203	407X	.DMNMS	DS	1	DISMOUNT/NO MESSAGE
000.204	408X	.RESET	DS	1	RESET = DISMOUNT/MOUNT OF UNIT
000.205	409	XTEXT	OVLDEF		

## 411X \*\* OVERLAY TABLE ENTRYS.

000.000	413X	ORG	0	
	414X			
000.000	415X	OVL.COD DS	2	FIRST SECTOR OF OVERLAY CODE
000.002	416X	OVL.SIZ DS	2	OVERLAY SIZE
000.004	417X	OVL.ENT DS	2	OVERLAY ENTRY POINT
000.006	418X	OVL.FLB DS	1	OVERLAY FLAG BYTE
000.007	419X	DS	1	DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010	420X	OVL.ENS EQU	*	OVERLAY ENTRY SIZE

## 422X \* OVERLAY INDICES

000.000	424X	ORG	0	
	425X			
000.000	426X	OVL0 DS	1	
000.001	427X	OVL1 DS	1	
000.002	428	XTEXT DEVDEF		

## 430X \*\* DEVICE TABLE ENTRYS.

000.000	432X	ORG	0	
	433X			
000.000	434X	DEV.NAM DS	2	DEVICE NAME
000.000	435X	DEV.EL EQU	00000000B	END OF DEVICE LIST FLAG
000.001	436X	DEV.NU EQU	00000001B	DEVICE ENTRY NOT IN USE
000.002	437X			
000.002	438X	DEV.RES DS	1	DRIVER RESIDENCE CODE
000.001	439X	DR.IM EQU	00000001B	DRIVER IN MEMORY
000.002	440X	DR.PR EQU	00000010B	DRIVER PERMINANTLY RESIDENT
000.003	441X			
000.004	442X	DEV.JMP DS	1	JMP TO PROCESSOR
000.004	443X	DEV.DDA DS	2	DRIVER ADDRESS
000.006	444X	DEV.FLG DS	1	FLAG BYTE
000.001	445X	DT.DD EQU	00000001B	DIRECTORY DEVICE
000.002	446X	DT.CR EQU	00000010B	CAPABLE OF READ OPERATION
000.004	447X	DT.CW EQU	00000100B	CAPABLE OF WRITE OPERATION
000.007	448X			
000.010	449X	DEV.SPG DS	1	SECTORS PER GROUP THIS DEVICE
000.011	450X	DEV.MUM DS	1	MOUNTED UNIT MASK
000.012	451X	DEV.MNU DS	1	MAXIMUM NUMBER OF UNITS
000.012	452X	DEV.UNT DS	2	ADDRESS OF UNIT SPECIFIC DATA TABLE
000.014	453X			
000.014	454X	DEV.DVL DS	2	DRIVER BYTE LENGTH
000.016	455X	DEV.RVG DS	1	DRIVER ROUTINE GROUP ADDRESS
000.017	456X			
000.017	457X	DEV.ELEN EQU	*	DEVICE TABLE ENTRY LENGTH

UNT.TAB 13:58:29 16-MAY-80

## 459X \*\* UNIT SPECIFIC DEVICE DATA TABLE ENTRIES

000.000	461X	ORG	0	
	462X			
000.000	463X	UNT.FLG	DS	1 UNIT SPECIFIC *DEV.FLGX
000.001	464X	UNT.GRT	DS	2 ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.003	465X	UNT.GTS	DS	2 GRT SECTOR NUMBER
000.005	466X	UNT.DIS	DS	2 DIRECTORY FIRST SECTOR NUMBER
	467X			
000.007	468X	UNT.SIZ	EQU	*
000.007	469	XTEXT	DIRDEF	SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT

## 471X \*\* DIRECTORY ENTRY FORMAT.

000.000	473X	ORG	0	
	474X			
	475X			
000.377	476X	DF.EMP	EQU	3770 FLAGS ENTRY EMPTY
000.376	477X	DF.CLR	EQU	3760 FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
	478X			
000.000	479X	DIR.NAM	DS	8 NAME
000.010	480X	DIR.EXT	DS	3 EXTENSION
000.013	481X	DIR.PRO	DS	1 PROJECT
000.014	482X	DIR.VER	DS	1 VERSION
000.015	483X	DIRIDL	EQU	*
	484X			
000.015	485X	DIR.CLU	DS	1 CLUSTER FACTOR
000.016	486X	DIR.FLG	DS	1 FLAGS
000.017	487X	DS	1 RESERVED	
000.020	488X	DIR.FGN	DS	1 FIRST GROUP NUMBER
000.021	489X	DIR.LGN	DS	1 LAST GROUP NUMBER
000.022	490X	DIR.LSI	DS	1 LAST SECTOR INDEX (IN LAST GROUP)
000.023	491X	DIR.CRD	DS	2 CREATION DATE
000.025	492X	DIR.ALD	DS	2 LAST ALTERATION DATE
	493X			
000.027	494X	DIRELEN	EQU	*
000.027	495	XTEXT	DISDEF	DIRECTORY ENTRY LENGTH

## 497X \*\* DIRECTORY BLOCK FORMAT.

000.000	499X	ORG	0	
	500X			
000.000	501X	DIS.ENT	EQU	*
000.000	502X	DS	22*DIRELEN	FIRST ENTRY ADDRESS 22 DIRECTORY ENTRYS PER BLOCK
001.372	503X	DS	1	0 BYTE = END OF ENTRYS IN THIS BLOCK
	504X			
001.373	505X	ORG	512-5	AT END OF BLOCK
001.373	506X	DIS.ENL	DS	1 LENGTH OF EACH ENTRY (=DIRELEN)
001.374	507X	DIS.SEC	DS	2 BLOCK # OF THIS BLOCK,
001.376	508X	DIS.LNK	DS	2 BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000	509	XTEXT	I0CDEF	

IOC

## 511X \*\* I/O CHANNEL DEFINITIONS.

000.000	513X	ORG	0	
	514X			
000.000	515X	IOC.LNK	DS	2 ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002	516X	IOC.IDA	DS	2 THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
	517X			
000.004	518X	IOC.FLG	DS	1 FILE TYPE FLAGS
000.001	519X	FT.DD	EQU	00000001B =1 IF DIRECTORY DEVICE
000.002	520X	FT.OR	EQU	00000010B =1 IF OPEN FOR READ
000.004	521X	FT.OW	EQU	00000100B =1 IF OPEN FOR WRITE
000.010	522X	FT.OU	EQU	00001000B =1 IF OPEN FOR UPDATE
000.003	523X	IOC.SRL	EQU	*-IOC.IDA LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
	524X			
000.005	525X	IOC.GRT	DS	2 ADDRESS OF GROUP RESERVATION TABLE
000.007	526X	IOC.SPG	DS	1 SECTORS PER GROUP, THIS DEVICE
000.010	527X	IOC.CGN	DS	1 CURRENT GROUP NUMBER
000.011	528X	IOC.CSI	DS	1 CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012	529X	IOC.LGN	DS	1 LAST GROUP NUMBER
000.013	530X	IOC.LSI	DS	1 LAST SECTOR INDEX (IN LAST GROUP)
000.010	531X	IOC.DRL	EQU	*-IOC.FLG LENGTH OF INFO NORMALLY COPIED BACK TO THE CHANNEL TABLE
	532X	*		
000.014	533X	IOC.DTA	DS	2 DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016	534X	IOC.DES	DS	2 SECTOR NUMBER OF DIRECTORY ENTRY
000.020	535X	IOC.DEV	DS	2 DEVICE CODE
000.022	536X	IOC.UNI	DS	1 UNIT NUMBER (0-9)
000.021	537X	IOC.DIL	EQU	*-IOC.IDA LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
	538X			
000.023	539X	IOC.DIR	DS	DIRELEN DIRECTORY ENTRY
	540X			
000.052	541X	IOCELEN	EQU	*
	542X			IOC ENTRY LENGTH
000.001	543X	IOCCTD	EQU	1 INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052	544	XTEXT	DDDEF	

## 546X.\*\* DEVICE DRIVER COMMUNICATION FLAGS.

	547X	*		
	548X			
000.000	549X	ORG	0	
	550X			
000.000	551X	DC.REA	DS	1 READ
000.001	552X	DC.WRI	DS	1 WRITE
000.002	553X	DC.RER	DS	1 READ REGARDLESS
000.003	554X	DC.DPR	DS	1 OPEN FOR READ
000.004	555X	DC.OPW	DS	1 OPEN FOR WRITE
000.005	556X	DC.OPU	DS	1 OPEN FOR UPDATE
000.006	557X	DC.CLO	DS	1 CLOSE
000.007	558X	DC.ABT	DS	1 ABORT
000.010	559X	DC.MOU	DS	1 MOUNT DEVICE
000.011	560X	DC.LOD	DS	1 LOAD DEVICE DRIVER
000.012	561X	DC.MAX	DS	1 MAXIMUM ENTRY INDEX
000.013	562	XTEXT	ECDEF	

## 564X \*\* ERROR CODE DEFINITIONS.

000.000	565X	ORG	0	
000.000	566X	DS	1	NO ERROR #0
000.001	567X	DS	1	END OF FILE
000.002	568X	EC.EOF	DS	1
000.002	569X	EC.EOM	DS	1
000.003	570X	EC.ILC	DS	1
000.004	571X	EC.CNA	DS	1
000.005	572X	EC.DNS	DS	1
000.006	573X	EC.IDN	DS	1
000.007	574X	EC.IFN	DS	1
000.010	575X	EC.NRD	DS	1
000.011	576X	EC.FNO	DS	1
000.012	577X	EC.ILR	DS	1
000.013	578X	EC.FUC	DS	1
000.014	579X	EC.FNF	DS	1
000.015	580X	EC.UND	DS	1
000.016	581X	EC.ICN	DS	1
000.017	582X	EC.DIF	DS	1
000.020	583X	EC.IFC	DS	1
000.021	584X	EC.NEM	DS	1
000.022	585X	EC.RF	DS	1
000.023	586X	EC.WF	DS	1
000.024	587X	EC.WPV	DS	1
000.025	588X	EC.WP	DS	1
000.026	589X	EC.FAP	DS	1
000.027	590X	EC.IDA	DS	1
000.030	591X	EC.FL	DS	1
000.031	592X	EC.FAO	DS	1
000.032	593X	EC.IS	DS	1
000.033	594X	EC.UUN	DS	1
000.034	595X	EC.FNR	DS	1
000.035	596X	EC.DIW	DS	1
000.036	597X	EC.UNA	DS	1
000.037	598X	EC.ILV	DS	1
000.040	599X	EC.ILO	DS	1
000.041	600X	EC.VFM	DS	1
000.042	601X	EC.NVM	DS	1
000.043	602X	EC.FOD	DS	1
000.044	603X	EC.NPM	DS	1
000.045	604X	EC.DNI	DS	1
000.046	605X	EC.DNR	DS	1
000.047	606X	EC.DSC	DS	1
000.050	607X	EC.NCV	DS	1
000.051	608X	EC.NOS	DS	1
000.052	609X	EC.IOI	DS	1
000.053	610X	EC.OTL	DS	1
000.054	611	XTEXT	DDFDEF	

## 613X \*\* DIRECTORY DEVICE FORMAT DEFINITION.

614X \*

615X

616X

000.002 617X HDOS.PRG EQU 2 2 SECTORS PER GROUP REQUIRED FOR NOW

618X

000.000 619X ORG 0

000.000 620X DDF.BOO DS 9

2K BOOT PROGRAM

000.011 621X DDF.BOL EQU \*

LENGTH OF BOOT

000.011 622X DDF.LAB DS 1

LABEL SECTOR

000.012 623X DDF.RGT DS 2

RESERVED GROUP TABLE

000.014 624X DDF.USR DS 0

BEGINNING OF OPEN SPACE

000.014 625 XTEXT LABDEF

## 627X \*\* DISK LABEL SECTOR FORMATS.

628X

000.000 629X ORG 0

000.000 630X LAB.SER DS 1

SERIAL NUMBER OF VOLUME

000.001 631X LAB.IND DS 2

INITIALIZATION DATE

000.003 632X LAB.DIS DS 2

SECTOR NUMBER OF 1ST DIRECTORY SECTOR

000.005 633X LAB.GRT DS 2

INDEX OF GRT SECTOR

000.007 634X LAB.SPG DS 1

SECTORS PER GROUP

635X

000.000 636X LAB.DAT EQU 0

DATA VOLUME ONLY

000.001 637X LAB.SYS EQU 1

SYSTEM VOLUME

000.002 638X LAB.NOD EQU 2

=&gt; LAB.NOD MEANS VOLUME HAS NO DIRECTORY

639X

000.010 640X LAB.VLT DS 1

VOLUME TYPE

000.011 641X LAB.VER DS 1

VERSION OF INIT17 THAT INITED DISK

000.012 642X DS 7

UNUSED

000.021 643X LAB.LAB DS 60

LABEL

000.074 644X LABLBL EQU \*-LAB,LAB

LABEL LENGTH

000.115 645 XTEXT PICDEF

## 647X \*\* PIC FORMAT EQUIVALENCES.

648X

000.000 649X ORG 0

650X

000.000 651X PIC.ID DS 1

377Q = BINARY FILE FLAG

000.001 652X DS 1

FILE TYPE.(FI,PIC)

000.002 653X PIC.LEN DS 2

LENGTH OF ENTIRE RECORD

000.004 654X PIC.PTR DS 2

INDEX OF START OF PIC TABLE

655X

000.006 656X PIC.COD DS 0

CODE STARTS HERE

000.006 657 XTEXT DVDEF

DUDDEF 13:59:18 16-MAY-80

## 659X \*\* DEVICE DRIVER EQUIVALENCES.

000.307	661X	DUDFLV	EQU	307Q	DEVICE DRIVER FLAG VALUE
	662X				
000.006	663X		ORG	PIC.COD	STARTS AT PIC CODE AREA
	664X				
000.006	665X	DUD.DVD	DS	1	MUST BE DUDFLV, FLAGS TO HDOS AS DRIVER
000.007	666X	DUD.CAP	DS	1	DEVICE CAPABILITY FLAG
000.010	667X	DUD.MUM	DS	1	MOUNTED UNIT MASK
000.011	668X	DUD.MNU	DS	1	MAXIMUM NUMBER OF UNITS
000.012	669X	DUD.UFL	DS	8	UNIT SUB-CAPABILITY FLAGS FOR UNITS 0-7
000.022	670X	DUD.SET	DS	1	= DUDFLV, IFF DRIVER WILL TAKE SET OPTIONS
000.023	671X		DS	24	RESERVED, MUST BE 0
000.053	672X	DUD.STE	EQU	*	ENTRY FOR 'SET' INVOCATION
	673X				
002.000	674X	DUD.ENT	EQU	2000A	DRIVER ENTRY POINT (MUST BE MULT. OF 256)
000.053	675		XTEXT	DIFDEF	

## 677X \*\* DIRECTORY FILE FLAGS.

000.200	678X				
000.100	679X	DIF.SYS	EQU	10000000B	SYSTEM FILE
000.040	680X	DIF.LOC	EQU	01000000B	LOCKED FOR CHANGE
000.020	681X	DIF.WP	EQU	00100000B	WRITE PROTECTED
	682X	DIF.CNT	EQU	00010000B	CONTIGUOUS FILE
000.053	683X				
	684		XTEXT	NAMDEF	

## 686X \*\* SYSTEM FILE NAME CONVENTIONS.

687X *					
688X *	RGT	.	SYS		RESERVED GROUP TABLE (1 SECTOR)
689X *	GRT	.	SYS		GROUP RESERVATION TABLE (1 SECTOR)
690X *	DIRECT	.	SYS		DIRECTORY
691X *	HOS	.	SYS		SYSTEM IMAGE PROGRAM FOR SYSTEM
692X					
000.053	693		XTEXT	MTRDEF	

## 695X \*\* HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.

696X					
000.000	697X		ORG	0	
000.000	698X	M.SYSM	DS	1	SYSCALL ITERATION COUNT
000.001	699X	M.SALO	DS	1	STAND-ALONE FLAG
000.002	700X	M.CSLC	DS	1	LINES IN CONSOLE BUFFER
000.003	701X	M.CPRE	DS	1	CONSOLE PREVIOUS CHARACTER
000.004	702X	M.CRUB	DS	1	CONSOLE CRUB FLAG
000.005	703X	M.CINT	DS	1	CONSOLE INTERRUPT FLAG
000.006	704X	M.CIN	DS	2	CONSOLE CB IN POINTER
000.010	705X	M.COUT	DS	2	CONSOLE CB OUT POINTER

MTRDEF 13:59:37 16-MAY-80

000.012	706X	M.CFWA	DS	2	CONSOLE CB FWA POINTER
000.014	707X	M.CLWA	DS	2	CONSOLE CB LWA POINTER
000.016	708X	M.CDLY	DS	1	CONSOLE PAD CHARACTER COUNT
000.017	709X	M.CDCA	DS	2	ADDRESS OF CHARACTER BEING PADDED
000.021	710	XTEXT	FLTDEF		

## 712X \*\* FLTDEF - DEFAULT SECTOR DEFINITIONS

713X					
000.000	714X	ORG	0		
000.000	715X	FLT.CTY	DS	1	CONSOLE TYPE FLAGS (FOR S.CONTY)
000.001	716X	FLT.CWI	DS	1	CONSOLE WIDTH (FOR S.CONWI)
000.002	717X	FLT.CFC	DS	1	CONSOLE FILL CHARACTERS NEEDED
000.003	718X	FLT.CRF	DS	1	CONSOLE CHARACTER REQUIRING FILL(3770 IF NONE)
000.004	719X	FLT.MNC	DS	1	MAXIMUM NUMBER OF I/O CHANNELS
000.005	720X	FLT.TDT	DS	1	TRACK SEEK DELAY TIME (MS/2)
000.006	721X	FLT.CDB	DS	1	CONSOLE DEFINITION BYTE
000.007	722X	FLT.CBD	DS	2	CONSOLE BAUD RATE
000.011	723X	FLT.BOP	DS	1	BOOTUP FLAGS
000.012	724X	FLT.SAL	DS	1	STAND-ALONE FLAGS. != 0 => CAN GO STAND-ALONE

000.013 727 XTEXT HOSEQU

## 729X \*\* HDDOS SYSTEM EQUIVALENCES.

730X \*

731X

024.000	732X S.GRTO	EQU	24000A	SYSTEM AREA FOR GRTO
025.000	733X S.GRT1	EQU	25000A	SYSTEM AREA FOR GRT1
026.000	734X S.GRT2	EQU	26000A	SYSTEM AREA FOR GRT2
	735X			
030.000	736X ROMBOOT	EQU	30000A	ROM BOOT ENTRY
	737X			
040.100	738X	ORG	40100A	FREE SPACE FROM PAM-8
	739X			
040.100	740X	DS	8	JUMP TO SYSTEM EXIT
040.110	741X D.CON	DS	16	DISK CONSTANTS
040.130	742X SYDD	EQU	*	SYSTEM DISK ENTRY POINT
040.130	743X D.VEC	DS	24*3	SYSTEM ROM ENTRY VECTORS
040.240	744X D.RAM	DS	31	SYSTEM ROM WORK AREA
040.277	745X S.VAL	DS	36	SYSTEM VALUES
040.343	746X S.INT	DS	115	SYSTEM INTERNAL WORK AREAS
041.126	747X	DS	16	
041.146	748X S.SOVR	DS	2	STACK OVERFLOW WARNING
041.150	749X	DS	42200A-*	SYSTEM STACK
001.032	750X STACKL	EQU	*-S.SOVR	STACK SIZE
	751X			
042.200	752X STACK	EQU	*	LWA11 SYSTEM STACK
042.200	753X USERFWA	EQU	*	USER FWA
	754			
042.200	755	XTEXT	EICON	

## 757X \*\* D.CON DETAILED EQUIVALENCES.

758X \*

759X \* HOSEQU MUST BE MODIFIED WHEN THIS TABLE IS MODIFIED.

760X

040.110	761X	ORG	D.CON	
	762X			
040.110	763X D.XITA	DS	2	SEE SYSTEM ROM FOR DESCRIPTION
040.112	764X D.WRITA	DS	1	
040.113	765X D.WRITEB	DS	1	
040.114	766X D.WRITEC	DS	1	
040.115	767X D.MAIA	DS	1	
040.116	768X D.LFSA	DS	1	
040.117	769X D.SDPA	DS	1	
040.120	770X D.SDFB	DS	1	
040.121	771X D.STSA	DS	1	
040.122	772X D.STSB	DS	1	
040.123	773X D.WHDA	DS	1	
040.124	774X D.WNHA	DS	1	
040.125	775X D.WSCA	DS	1	
	776X			

040.126  
040.130777X D.ERTS DS 2  
778 XTEXT EDVEC

TRACK AND SECTOR OF LAST DISK ERRORS

040.130

780X \*\* JMP VECTORS FOR ROM CODE

781X \*

782X \* SEE DISK ROM FOR ADDRESSES

783X \*

784X \* HOSEQU MUST BE ALTERED WHEN THIS TABLE IS ALTERED.

785X

040.130

786X ORG D.VEC

787X

040.130

788X D.SYDD DS 3 JMP R.SYDD (MUST BE FIRST)

040.133

789X D.MOUNT DS 3 JMP R.MOUNT

040.136

790X D.XOK DS 3 JMP R.XOK

040.141

791X D.ABORT DS 3 JMP R.ABORT

040.144

792X D.XIT DS 3 JMP R.XIT

040.147

793X D.READ DS 3 JMP R.READ

040.152

794X D.READR DS 3 JMP R.READR

040.155

795X D.WRITE DS 3 JMP R.WRITE

040.160

796X D.CDE DS 3 JMP R.CDE

040.163

797X D.DTS DS 3 JMP R.DTS

040.166

798X D.SDT DS 3 JMP R.SDT

040.171

799X D.MAI DS 3 JMP R.MAI

040.174

800X D.MAO DS 3 JMP R.MAO

040.177

801X D.LPS DS 3 JMP R.LPS

040.202

802X D.RDB DS 3 JMP R.RDB

040.205

803X D.SDP DS 3 JMP R.SDP

040.210

804X D.STS DS 3 JMP R.STS

040.213

805X D.STZ DS 3 JMP R.STZ

040.216

806X D.UDLY DS 3 JMP R.UDLY

040.221

807X D.WSC DS 3 JMP R.WSC

040.224

808X D.WSP DS 3 JMP R.WSP

040.227

809X D.WNB DS 3 JMP R.WNB

040.232

810X D.ERRT DS 3 JMP R.ERRT

040.235

811X D.DLY DS 3 JMP R.DLY

040.240

812 XTEXT EDRAM

814X \*\* EDRAM - DISK/RAM WORKAREA DEFINITION.

815X \*

816X \* ZEROED UPON BOOTING UP.

817X \*

818X \* HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.

819X

040.240

820X ORG D.RAM

822X

040.240

823X D.TT DS 1 TARGET TRACK (CURRENT OPERATION)

040.241

824X D.TS DS 1 TARGET SECTOR (CURRENT OPERATION)

825X

040.242

826X D.DVCTL DS 1 DEVICE CONTROL BYTE

	827X		
040.243	828X D.DLYMO DS	1	MOTOR ON DELAY COUNT
040.244	829X D.DLYHS DS	1	HEAD SETTLE DELAY COUNTER
	830X		
040.245	831X D.TRKPT DS	2	ADDRESS IN D.DRVTB FOR TRACK NUMBER
040.247	832X D.VOLPT DS	2	ADDRESS IN D.DRVTB FOR VOLUME NUMBER
	833X		
040.251	834X D.DRVTB DS	2*4	TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
	835X		
040.241	836X D.HECNT DS	1	HARD ERROR COUNT
040.262	837X D.SECNT DS	2	SOFT ERROR COUNT
040.264	838X D.OECNT DS	1	OPERATION ERROR COUNT
	839X		
	840X *		GLOBAL DISK ERROR COUNTERS
	841X		
040.265	842X D.ERR DS	0	BEGINNING OF ERROR BLOCK
040.265	843X D.E.MDS DS	1	MISSING DATA SYNC
040.266	844X D.E.HSY DS	1	MISSING HEADER SYNC
040.267	845X D.E.CHK DS	1	DATA CHECKSUM
040.270	846X D.E.HCK DS	1	HEADER CHECKSUM
040.271	847X D.E.VOL DS	1	WRONG VOLUME NUMBER
040.272	848X D.E.TRK DS	1	BAD TRACK SEEK
040.273	849X D.ERRL DS	0	LIMIT OF ERROR COUNTERS
	850X		
	851X *		I/O OPERATION COUNTS
	852X		
040.273	853X D.OFR DS	2	
040.275	854X D.OPW DS	2	
	855X		
000.037	856X D.RAML EQU	*-D.RAM	
040.277	857 XTEXT	ESVAL	

859X \*\* S.VAL - SYSTEM VALUE DEFINITIONS.

860X \*

861X \* THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.

862X \*

863X \* THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.

864X

865X

040.277 866X ORG S.VAL

867X

040.277 868X S.DATE DS 9 SYSTEM DATE (IN ASCII)

040.310 869X S.DATC DS 2 CODED DATE

040.312 870X S.TIME DS 4 TIME FROM MIDNIGHT (IN TICS)

040.316 871X S.HIMEM DS 2 HARIWARE HIGH MEMORY ADDRESS+1

872X

040.320 873X S.SYSM DS 2 FWA RESIDENT SYSTEM

874X

040.322 875X S.USRM DS 2 LWA USER MEMORY

876X

040.324 877X S.OMAX DS 2 MAX OVERLAY SIZE FOR SYSTEM

878X

879X

880X \*\* THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL

000.200	882X CSL.ECH EQU	10000000B	SUPPRESS ECHO
000.002	883X CSL.WRAP EQU	00000010B	WRAP LINES AT WIDTH
000.001	884X CSL.CHR EQU	00000001B	OPERATE IN CHARACTER MODE
	885X		
000.000	886X I.CSLMD EQU	0	S.CSLMD IS FIRST BYTE
040.326	887X S.CSLMD DS	1	CONSOLE MODE
	888X		
000.200	889X CTF.BKS EQU	10000000B	TERMINAL PROCESSES BACKSPACES
000.040	890X CTF.MLI EQU	00100000B	MAP LOWER CASE TO UPPER ON INPUT
000.020	891X CTF.MUD EQU	00010000B	MAP LOWER CASE TO UPPER ON OUTPUT
000.010	892X CTF.2SB EQU	00001000B	TERMINAL NEEDS TWO STOP BITS
000.002	893X CTF.BKM EQU	00000010B	MAP BKSP. (UPON INPUT). TO RUBOUT
000.001	894X CTF.TAB EQU	00000001B	TERMINAL SUPPORTS TAB CHARACTERS
	895X		
000.001	896X I.CONTY EQU	1	S.CONTY IS 2ND BYTE
000.000	897X ERRNZ	*-S.CSLMD-I.CONTY	
040.327	898X S.CONTY DS	1	CONSOLE TYPE FLAGS
000.002	899X I.CUSOR EQU	2	S.CUSOR IS 3RD BYTE
000.000	900X ERRNZ	*-S.CSLMD-I.CUSOR	
040.330	901X S.CUSOR DS	1	CURRENT CURSOR POSITION
000.003	902X I.CONWI EQU	3	S.CONWI IS 4TH BYTE
000.000	903X ERRNZ	*-S.CSLMD-I.CONWI	
040.331	904X S.CONWI DS	1	CONSOLE WIDTH
	905X		
000.001	906X CO.FLG EQU	00000001B	CTL-O FLAG
000.200	907X CS.FLG EQU	10000000B	CTL-S FLAG
	908X		
000.004	909X I.CONFL EQU	4	S.CONFL IS 5TH BYTE
000.000	910X ERRNZ	*-S.CSLMD-I.CONFL	
040.332	911X S.CONFL DS	1	CONSOLE FLAGS
	912X		
040.333	913X S.CAADR DS	2	ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335	914X S.CCTAB DS	6	ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343	915		
	916 XTEXT ESINT		

918X \*\* S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.

919X \*

920X \* THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND

921X \* MUST THEREFORE RESIDE IN FIXED LOW MEMORY.

922X

923X

040.343 924X ORG S.INT

925X

926X \*\* CONSOLE STATUS FLAGS.

927X

040.343 928X S.CDB DS 1 CONSOLE DESCRIPTOR BYTE

000.000 929X CDB.H85 EQU 00000000B

000.001 930X CDB.H84 EQU 00000001B =0 IF H8-5, =1 IF H8-4

040.344 931X S.BAUD DS 2 [0-14] H8-4 BAUD RATE, =0 IF H8-5

932X \* [15] =1 IF BAUD RATE => 2 STOP BITS

..... 933X  
..... 934X \*\* TABLE ADDRESS WORDS  
..... 935X  
040.346 936X S.DLINK DS 2 ADDRESS OF DATA IN HDOS CODE  
040.350 937X S.OFWA DS 2 FWA OVERLAY TABLE  
040.352 938X S.CFWA DS 2 FWA CHANNEL TABLE  
040.354 939X S.IDFWA DS 2 FWA DEVICE TABLE  
040.356 940X S.RFWA DS 2 FWA RESIDENT HDOS CODE  
..... 941X  
..... 942X \*\* DEVICE DRIVER DELAYED LOAD FLAGS  
..... 943X  
040.360 944X S.DDLDA DS 2 DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)  
040.362 945X S.DDLEN DS 2 CODE LENGTH IN BYTES  
040.364 946X S.DDGRP DS 1 GROUP NUMBER FOR DRIVER  
040.365 947X DS 1 HOLD PLACE  
..... 948X \*S.DDSEC DS 2 SECTOR NUMBER FOR DRIVER (\* OBSOLETE ! \*)  
040.366 949X S.DDDTA DS 2 DEVICE'S ADDRESS IN DEVLIST +DEV.RES  
040.370 950X S.DDOPC DS 1 OPEN OPCODE PENDING  
..... 951X  
..... 952X \*\* OVERLAY MANAGEMENT FLAGS  
..... 953X  
000.001 954X OVL.IN ERU 00000001B IN MEMORY  
000.002 955X OVL.RES EQU 00000010B PERMINANTLY RESIDENT  
000.014 956X OVL.NUM ERU 00001100B OVERLAY NUMBER MASK  
000.200 957X OVL.UCS EQU 10000000B USER CODE SWAPPED FOR OVERLAY  
..... 958X  
040.371 959X S.OVFLF DS 1 OVERLAY FLAG  
040.372 960X S.UCSF DS 2 FWA SWAPPED USER CODE  
040.374 961X S.UCSL DS 2 LENGTH SWAPPED USER CODE  
040.376 962X S.OVLS DS 2 SIZE OF OVERLAY CODE  
041.000 963X S.OVLE DS 2 ENTRY POINT OF OVERLAY CODE  
..... 964X  
041.002 965X S.SSN DS 2 SWAP AREA SECTOR NUMBER  
041.004 966X S.OSN DS 2 OVERLAY SECTOR NUMBER  
..... 967X  
..... 968X \* SYSCALL PROCESSING WORK AREAS  
..... 969X  
041.006 970X S.CACC DS 1 (ACC) UPON SYSCALL  
041.007 971X S.CODE DS 1 SYSCALL INDEX IN PROGRESS  
..... 972X  
..... 973X \* JUMPS TO ROUTINES IN RESIDENT HDOS CODE  
..... 974X  
041.010 975X S.JUMPS DS 0 START OF DUMP VECTORS  
041.010 976X S.SDD DS 3 JUMP TO STAND-IN DEVICE DRIVER  
041.013 977X S.FASER DS 3 JUMP TO FATSERR (FATAL SYSTEM ERROR)  
041.016 978X S.DIREA DS 3 JUMP TO DIREAD (DISK FILE READ)  
041.021 979X S.FCI DS 3 JUMP TO FCI (FETCH CHANNEL INFO)  
041.024 980X S.SCI DS 3 JUMP TO SCI (STORE CHANNEL INFO)  
041.027 981X S.GUP DS 3 JUMP TO GUP (GET UNIT POINTER)  
..... 982X  
041.032 983X S.MOUNT DS 1 <>0 IF THE SYSTEM DISK IS MOUNTED  
041.033 984X S.DCS DS 1 DEFAULT CLUSTER SIZE-1  
..... 985X  
041.034 986X S.BOOTF DS 1 BOOT FLAGS  
000.001 987X BOOT.P EQU 00000001B EXECUTE PROLOGUE UPON BOOTUP  
..... 988X

989X \* STACK VALUE SAVED FOR OVERLAY SYSCALLS  
990X  
041.035 991X S.OVSTK DS 2 VALUE OF SP UPON SYSCALLS USING OVERLAY  
992X  
041.037 993X DS 1 RESERVED

995X \*\* ACTIVE I/O AREA.  
996X \*  
997X \* THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION.  
998X \* CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM  
999X \* THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.  
1000X \*  
1001X \* NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY.  
1002X \* FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE  
1003X \* 8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY  
1004X \* COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND  
1005X \* BACKDATED AFTER PROCESSING.  
1006X

041.040 1007X AIO.VEC DS 3 JUMP INSTRUCTION  
041.041 1008X AIO.IDA EQU \*-2 DEVICE DRIVER ADDRESS  
041.043 1009X AIO.FLG DS 1 FLAG BYTE  
041.044 1010X AIO.GRT DS 2 ADDRESS OF GROUP RESERV TABLE  
041.046 1011X AIO.SPG DS 1 SECTORS PER GROUP  
041.047 1012X AIO.CGN DS 1 CURRENT GROUP NUMBER  
041.050 1013X AIO.CSI DS 1 CURRENT SECTOR INDEX  
041.051 1014X AIO.LGN DS 1 LAST GROUP NUMBER  
041.052 1015X AIO.LSI DS 1 LAST SECTOR INDEX  
041.053 1016X AIO.DTA DS 2 DEVICE TABLE ADDRESS  
041.055 1017X AIO.DES DS 2 DIRECTORY SECTOR  
041.057 1018X AIO.DEV DS 2 DEVICE CODE  
041.061 1019X AIO.UNI DS 1 UNIT NUMBER (0-9)  
1020X  
041.062 1021X AIO.DIR DS DIRELEN DIRECTORY ENTRY  
1022X  
041.111 1023X AIO.CNT DS 1 SECTOR COUNT  
041.112 1024X AIO.EOM DS 1 END OF MEDIA FLAG  
041.113 1025X AIO.EOF DS 1 END OF FILE FLAG  
041.114 1026X AIO.TFP DS 2 TEMP FILE POINTERS  
041.116 1027X AIO.CHA DS 2 ADDRESS OF CHANNEL BLOCK (IOC, DDA)

041.120 1029X S.SCR DS 2 SYSTEM SCRATCH AREA ADDRESS  
041.122 1030 XTEXT B00DEF

1032X \*\* BOODEF - SPECIAL BOOT-HDOS INTERFACE DEFINITIONS.

1033X  
047.000 1034X SB.ORG EQU 47000A ORG FOR LOAD OF INITIAL HDOS.SAV  
014.000 1035X SB.QVMX EQU 14000A SIZE OF HOLD AREA FOR SWAPPED USER CODE  
1036X \* (=MAX SIZE OF HDOSQVL.SYS)  
1037  
1038  
1039  
1040 CODE P,SB.ORG POSITION INDEPENDANT CODE  
1041 CODE -R THIS CODE WILL NOT BE RELOCATED  
1042

1046 \*\* TEMP INITIALIZE  
1047  
047.006 303 024 047 1048 HOSBOOT JMP HOSB001 PERFORM BOOT  
1049  
1050 \* DEFAULT VALUES FOR SYSTEM  
1051  
047.011 1052 HOSTAB DS 0 DEFAULT VALUE TABLE  
000.000 1053 ERRNZ \*-HOSTAB-FLT.CTY  
047.011 072 1054 \* 1055 DB CTP.MLO+CTP.MLI+CTP.BKM+CTP.2SB  
000.000 1056 ERRNZ \*-HOSTAB-FLT.CWI  
047.012 120 1057 DB 80 S.CONWI  
000.000 1058 ERRNZ \*-HOSTAB-FLT.CFC  
047.013 004 1059 DB 4 NUMBER OF FILL CHARACTERS  
000.000 1060 ERRNZ \*-HOSTAB-FLT.CRF  
047.014 015 1061 DB CR CHARACTER TO BE FILLED  
000.000 1062 ERRNZ \*-HOSTAB-FLT.MNC  
047.015 006 1063 DB 6 NUMBER OF CHANNELS  
000.000 1064 ERRNZ \*-HOSTAB-FLT.TDT  
047.016 017 1065 DB 30/2 TRACK DELAY  
000.000 1066 ERRNZ \*-HOSTAB-FLT.CDB  
047.017 000 1067 DB CDB.H85 H8-5 CONSOLE  
000.000 1068 ERRNZ \*-HOSTAB-FLT.CBD  
047.020 000 200 1069 DW 200000A BAUD => 2 STOP BITS FOR H8-5  
000.000 1070 ERRNZ \*-HOSTAB-FLT.BOP  
047.022 000 1071 DB 0 BOOT OPTION FLAGS  
000.000 1072 ERRNZ \*-HOSTAB-FLT.SAL  
047.023 000 1073 DB 0 STAND-ALONE OPTION  
1074  
1075 \* END OF DEFAULT TABLE. START OF BOOT CODE  
1076  
047.024 1077 HOSB001 EQU \*  
047.024 061 200 042 1078 LXI SP,STACK SET UP THE NEW STACK  
000.001 1079 IF DEBUG  
1080 LDA 40077A  
1081 ANA A  
1082 JNZ 160000A ENTER HBUG  
1083 ENDIF  
1084  
1085 \* SAVE DATA DETERMINED FROM BOOT  
1086  
047.027 072 343 040 1087 LDA S.CDB  
047.032 062 017 047 1088 STA HOSTAB+FLT.CDB  
047.035 052 344 040 1089 LHLD S.BAUD  
047.040 042 020 047 1090 SHLD HOSTAB+FLT.CBD  
047.043 041 011 047 1091 LXI H,HOSTAB+FLT.CTY  
047.046 072 327 040 1092 LDA S.CONTY  
047.051 346 010 1093 ANI CTP.2SB  
047.053 266 1094 ORA M  
047.054 167 1095 MOV M,A  
047.055 072 034 041 1096 LDA S.BOOTF  
047.060 062 022 047 1097 STA HOSTAB+FLT.BOP..SAVE THE BOOT FLAGS  
1098  
1099 \* BOOT CODE  
1100  
047.063 041 312 040 1101 LXI H,S.TIME

047.066	006 204	1102	MVI	B,AIO.CHA-S.TIME	
047.070	315.212.031	1103	CALL	\$ZERO.	ZERO OUT LOTS OF MEMORY
047.073	315 356 050	1104	CALL	SDV	SETUP DEFAULT SYSTEM VALUES
047.076	315 211.047	1105	CALL	RRH	RELOCATE RESIDENT HIDS CODE
047.101	315 102 050	1106	CALL	SRR	SET UP ROM REPLACEMENTS
047.104	315 133 050	1107	CALL	SLR	SET LOW MEMORY REFERENCES
047.107	315 356 050	1108	CALL	SDV	SETUP DEFAULT SYSTEM VALUES
047.112	315 046 051	1109	CALL	SCD	SETUP CONSOLE DRIVER
047.115	315 316 051	1110	CALL	MSD	MOUNT SYSTEM DISK
047.120	315 242 051	1111	CALL	GVM	GIVE VERSION MESSAGE
047.123	315 167 052	1112	CALL	PGT	PROCESS GRT
047.126	315 320 053	1113	CALL	CDS	CLEAR DIRECTORY SPACES
047.131	315 071 054	1114	CALL	LSD	LOCATE SYSTEM OVERLAYS
047.134	315 077 055	1115	CALL	SDT	SETUP DEVICE TABLES
047.137	315 110 057	1116	CALL	SSD	SET SYSTEM DATE
047.142	072 001	1117	MVI	A,1	
047.144	062 032 041	1118	STA	S.MOUNT	FLAG SYSTEM MOUNTED
047.147	072 034 041	1119	LDA	S.BOOTF	
047.152	346 001	1120	ANI	BOOT.F	
047.154	312.164.047	1121	JZ	HOSB2	IGNORE PROLOGUE FILE
		1122			
047.157	041 170 047	1123	LXI	H,HOSRA	
047.162	377 040	1124	DB	SYSCALL,,LINK	TRY TO LINK TO PROLOGUE
		1125			
047.164	076 001	1126	HOSB2	MVI	A,1 COULDN'T FIND PROFILE, SO TRY NORMAL
047.166	377 000	1127	DB	SYSCALL,,EXIT	EXIT WITH RESET
		1128			
047.170	123.131.060	1129	HOSRA	DB	'SY0:PROLOGUE.SYS',0 PROLOGUE FILE DESCRIPTOR

```

1132 ** RRH - RELOCATE CODE.
1133 * RRH IS CALLED TO RELOCATE THE HDOS CODE INTO HIGH MEMORY.
1135 *
1136 * ENTRY NONE
1137 * EXIT (DE) = DISPLACEMENT FACTOR
1138 * USES ALL
1139
1140
047.211 1141 RRH EQU *
1142
1143 * DETERMINE HIGH MEMORY.
1144
047.211 041.227.047 1145 LXI H,RRH2 START AT RRH2.
047.214 056 000 1146 MVI L,0 START AT 256 BOUNDARY
047.216 044 1147 RRH1 INR H TRY NEXT BLOCK.
047.217 176 1148 MOV A,M
047.220 064 1149 INR M
047.221 276 1150 CMP M
047.222 167 1151 MOV M,A RESTORE.
047.223 302 216 047 1152 JNE RRH1 WAS RAM
047.226 053 1153 DCX H (HL) = HIGHMEM.
1154
1155 * (HL) = HIGHMEM ADDRESS.
1156
000.001 1157 IF DEBUG
1158 RRH2 MVI A,140A ** DEBUG **
1159 CMP H
1160 JNC RRH2.5 DONT HAVE OVER 16 K
1161 LXI H,140000A RESTRICT FOR NOW ** DEBUG **
1162 ELSE
047.227 1163 RRH2 EQU *
1164 ENDIF
047.227 042.316.040 1165 RRH2.5 SHLD S,HIMEM SET HARDWARE HIGH MEM.
047.232 043 1166 INX H (HL) = LWA+1
047.233 174 1167 MOV A,H
047.234 326 040 1168 SUI 400
047.236 037 1169 RAR.
047.237 037 1170 RAR.
047.240 .346.077 1171 ANI 770 (A) = # OF K.
047.242 137 1172 MOV E,A
047.243 026 000 1173 MVI D,0
047.245 315 253 061 1174 CALL $TYPET
047.250 .000.123.131 1175 DB 0,(SYSTEM.HAS/. / +2000
047.264 076 002 1176 MVI A,2
047.266 .315.027.057 1177 CALL TDP TYPE NUMBER OF K.
047.271 315 253 061 1178 CALL $TYPET
047.274 113 040 117 1179 DB 'K OF RAM',2000
047.305 072 317 040 1180 LDA S,HIMEM+1 (A) = SIZE
047.310 326 120 1181 SUI 400+600
047.312 322 372 047 1182 JNC RRH3 ENOUGH ROOM
1183
1184 * NOT ENOUGH ROOM
1185
047.315 315 253 061 1186 CALL $TYPET
047.320 007 077 060 1187 DB BELL,'?01 HDOS REQUIRES AT LEAST 12K!',0,BELL+2000

```

047.362 257 1188 XRA A  
047.363 323 351 1189 OUT SC.ACETUR.IER  
047.365 323 373 1190 OUT SC.UART+USR CLEAR CONSOLE  
047.367 303 000 030 1191 JMP ROMBOOT  
1192  
1193 \* HAVE ENOUGH ROOM  
1194  
047.372 .052.314.040 1195 RRH3 LHLD S,HIMEM  
1196 \* LXI D,FWASYS-LWASYS-4 /79.11.GC/  
047.375 .021.020.363 1197 LXI D,FWASYS-LWASYS+1 /79.11.GC/  
050.000 031 1198 DAD D (HL) = NEW FWASYS  
050.001 .021.106.062 1199 LXI D,FWAREL  
050.004 175 1200 MOV A,L  
050.005 223 1201 SUB E  
050.006 117 1202 MOV C,A  
050.007 174 1203 MOV A,H  
050.010 232 1204 SBB D  
050.011 107 1205 MOV B,A (BC) = DISPLACEMENT  
050.012 305 1206 PUSH B SAVE  
050.013 .001.361.014. 1207 LXI B,LWASYS-FWASYS. (BC) = SYSTEM RESIDENCE LENGTH  
050.016 315 252 030 1208 CALL \$MOVE MOVE INTO PLACE  
1209  
1210 \* RELOCATE REFERENCES  
1211  
050.021 321 1212 POP D (DE) = RELOCATION FACTOR  
050.022 .052.004.047. 1213 LHLD SB,ORG+PIC.PTR  
050.025 001 000 047 1214 LXI B,SB.ORG  
050.030 .011 1215 DAD B (HL) = REL TABLE ADDRESS  
1216  
1217 \* RELOCATE CELLS IN BOOT CODE ITSELF  
1218  
050.031 .325. 1219 RRH4 PUSH D SAVE RELOCATION FACTOR  
050.032 136 1220 MOV E,M  
050.033 .043. 1221 INX H  
050.034 126 1222 MOV D,M  
050.035 .043. 1223 INX H (DE) = REL ADDRESS OF WORD TO RELOCATE  
050.036 172 1224 MOV A,D  
050.037 .263. 1225 ORA E  
050.040 312 055 050 1226 JZ RRH6 ALL DONE  
1227  
1228 \* SEE IF ADDRESS IS BEYOND FWAREL  
1229  
050.043 001 106 062 1230 LXI B,FWAREL (BC) = BREAK BETWEEN ABS PRESET AND REL HDDOS  
050.046 173 1231 MOV A,E  
050.047 221 1232 SUB C  
050.050 172 1233 MOV A,D  
050.051 230 1234 SBB B  
050.052 332 065 050 1235 JC RRH5 NOT BEYOND  
1236  
1237 \* LET REL ROUTINE RELOCATE REST OF CODE  
1238  
050.055 001 376 377 1239 RRH6 LXI B,-2  
050.060 011 1240 DAD B BACKUP (HL)  
050.061 .301. 1241 POP B (BC) = REL FACTOR  
050.062 303 175 033 1242 JMP REL RELOCATE AND EXIT  
1243

1244 \* (DE) = INDEX OF WORD TO RELOCATE  
1245 \* (HL) = RELOCATION TABLE ADDRESS  
1246 \* (BC) = CODE DISPLACEMENT FACTOR  
1247 \* ((SP)) = CODE RELOCATION FACTOR

1248

050.065 343 1249 RRH5 XTHL (HL) = CODE REL. FACTOR  
050.066 032 1250 LDAX D  
050.067 205 1251 ADD L RELOCATE WORD OF CODE  
050.070 022 1252 STAX D  
050.071 023 1253 INX D  
050.072 032 1254 LDAX D  
050.073 214 1255 ADC H  
050.074 022 1256 STAX D RELOCATE  
050.075 353 1257 XCHG (DE) = RELOCATION FACTOR  
050.076 341 1258 POP H (HL) = RELOCATION TABLE ENTRY ADDRESS  
050.077 303.031.050 1259 JMP RRH4 DO IT AGAIN

SRR - SET UP ROM REPLACEMENTS

14:00:55 16-MAY-80

1262 \*\* SRR - SET UP ROM REPLACEMENTS

1263 \*

1264 \* SET UP RAM REPLACEMENTS FOR THE ROM CODE.

1265 \*

1266 \*

1267

050.102

1268 SRR EQU \*

1269

050.102 041 043 072 1270

LXI H,DSKERR

050.105 042 233 040 1271

SHLD D.ERRT+1 SETUP.ERROR.TRAP.REFERENCE

1272

050.110 041 054 072 1273

LXI H,RAMCDE

050.113 042 161 040 1274

SHLD D.CDE+1 SETUP REPLACEMENT FOR 'R.CDE'

1275

050.116 041 120 072 1276

LXI H,RAMWRI

050.121 042 156 040 1277

SHLD D.WRITE+1 SETUP REPLACEMENT FOR 'R.WRITE'

1278

050.124 041 162 072 1279

LXI H,RAMSDP

050.127 042 206 040 1280

SHLD D.SDP+1 SETUP REPLACEMENT FOR 'R.SDP'

1281

050.132 311 1282

RET

```

1285 ** SLR - SETUP LOW MEMORY REFERENCES.
1286 *
1287
1288
050.133 041 106 062 1289 SLR EQU *
050.133 041 106 062 1290 LXI H,SYSCAL
050.136 042 062 040 1291 SHLD .UIVECT+18+1 SETUP SYSCALL LINKAGE
050.141 076 201 1292 MVI A,UD,CLK+UO,HLT DISABLE HALT PROCESSING &
050.143 062 010 040 1293 STA .MFLAG REQUEST CLOCK INTERRUPTS
1294
1295 * SETUP EXIT VECTOR AT 40100A
1296
050.146 315 052 060 1297 CALL $MOVEL
050.151 010 000 324 1298 DW SLRAL,SLRA,40100A
1299
1300 * SETUP MOTOR-ON DELAY TIME
1301
050.157 076 074 1302 MVI A,30*2
050.161 062 110 040 1303 STA D,XITA SET MOTOR ON DELAY
050.164 257 1304 XRA A
050.165 062 111 040 1305 STA D,XITA+1 SET NO HEAD SETTLE TIMEOUT
1306
1307 * SETUP LOW-MEMORY STUFF
1308
050.170 076 003 1309 MVI A,4-1 (A) = DEFAULT CLUSTER-1
050.172 062 033 041 1310 STA S,DCS SET DEFAULT CLUSTER SIZE
1311
050.175 041 106 062 1312 LXI H,FWASYS
050.200 042 320 040 1313 SHLD S,SYSM SET SYSTEM FWA
050.203 042 354 040 1314 SHLD S,RFWA SET RESIDENT CODE FWA
050.206 041 200 042 1315 LXI H,USERFWA
050.211 042 322 040 1316 SHLD S,USRML SET LWA USER MEMORY
1317
050.214 257 1318 XRA A
050.215 062 326 040 1319 STA S,CSLMID CLEAR CONSOLE MODE
050.220 062 330 040 1320 STA S,CUSOR CLEAR CURSOR ADDRESS
050.223 257 1321 XRA A
050.224 062 332 040 1322 STA S,CONFL CLEAR CONSOLE FLAGS
050.227 062 126 040 1323 STA D,ERTS CLEAR ERROR TRACK NUMBER
1324
050.232 041 244 074 1325 LXI H,HIGHDAT
050.235 042 346 040 1326 SHLD S,DLINK SET DATA LINK
050.240 041 347 072 1327 LXI H,OVLTAB
050.243 042 350 040 1328 SHLD S,OFWA
050.246 041 176 073 1329 LXI H,CHANTAB
050.251 042 352 040 1330 SHLD S,CFWA
050.254 041 367 072 1331 LXI H,DEVLST
050.257 042 354 040 1332 SHLD S,DFWA
1333
050.262 257 1334 XRA A
050.263 062 371 040 1335 STA S,OVLFL CLEAR OVL RESIDENCE
050.266 076 060 1336 MVI A,CTP,MLI+CTP,MLD
050.270 062 327 040 1337 STA S,CONTY INITIALIZE CONSOLE TYPE
1338
050.273 041 067 075 1339 LXI H,SECSCR
050.276 042 120 041 1340 SHLD S,SCR SET UP OF SYSTEM SCRATCH POINTER

```

1341  
1342 \* SETUP JUMP VECTORS  
1343  
050.301 076 303 1344 MVI A,MI,JMP  
050.303 062 040 041 1345 STA AIO,VEC  
050.306 315 052 060 1346 CALL \$MOVEL  
050.311 022 000 334 1347 DW SLRBL,SLRB,S.JUMPS SETUP JUMP VECTORS  
050.317 257 1348 XRA A  
050.320 062 061 041 1349 STA AIO,UNI AM BOOTING UNIT 0  
050.323 311 1350 RET  
1351  
050.324 1352 SLRA DS 0 CODE FOR 40100A  
050.324 257 1353 XRA A  
050.325 062 244 074 1354 STA SYSMODE  
050.330 076 001 1355 MVI A,1 FLAG RESET  
050.332 377 000 1356 DB SYSCALL,..EXIT  
000.010 1357 SLRAL EQU \*-SLRA  
377.377 1358 ERRPL SLRAL-9 ONLY ROOM FOR B.BYTES  
1359  
050.334 1360 SLRB DS 0 JUMP VECTOR CONTENTS  
000.000 1361 ERRNZ \*-SLRB+S.JUMPS-S.SDD  
050.334 303 263 071 1362 JMP SDD S.SDD  
000.000 1363 ERRNZ \*-SLRB+S.JUMPS-S.FASER  
050.337 303 210 063 1364 JMP FATSERR S.FASER  
000.000 1365 ERRNZ \*-SLRB+S.JUMPS-S.DIREA  
050.342 303 131 067 1366 JMP DIREAD S.DIREA  
000.000 1367 ERRNZ \*-SLRB+S.JUMPS-S.FCI  
050.345 303 260 070 1368 JMP FCI S.FCI  
000.000 1369 ERRNZ \*-SLRB+S.JUMPS-S.SCI  
050.350 303 237 071 1370 JMP SCI S.SCI  
000.000 1371 ERRNZ \*-SLRB+S.JUMPS-S.GUP  
050.353 303 301 071 1372 JMP GUP S.GUP  
000.022 1373 SLRBL EQU \*-SLRB

1376 \*\* SDV - SETUP SYSTEM DEFAULT VALUES.  
1377 \*  
1378 \* SDV SETS UP THE SYSTEM DEFAULT VALUES CONTAINED IN \*HOSTTAB\*,  
1379 \* AS DESCRIBED IN \*FLTDEF.COM\*  
1380 \*  
1381 \* THESE VALUES CAN BE SET IN THE HDDOS.SYS BINARY BY THE \*SET\*  
1382 \* UTILITY, AND ARE PROPAGATED INTO THE PROPER SPOTS AT  
1383 \* BOOT TIME.  
1384 \*  
1385 \* ENTRY NONE  
1386 \* EXIT NONE  
1387 \* USES ALL  
1388  
1389

050.356 072 011 047 1390 SDV LDA HOSTAB+FLT.CTY CONSOLE TYPE FLAGS  
050.361 062 327 040 1391 STA S.CONTY  
050.364 072 012 047 1392 LDA HOSTAB+FLT.CWI (A) = CONSOLE WIDTH  
050.367 062 331 040 1393 STA S.CONWI  
050.372 072 013 047 1394 LDA HOSTAB+FLT.CFC (A) = # OF FILL CHARACTERS NEEDED  
050.375 062 262 074 1395 STA CSLDLY SET PAD DELAY  
051.000 072 014 047 1396 LDA HOSTAB+FLT.CRF  
051.003 052 263 074 1397 LHLD CSLDCA (HL) = ADDRESS FOR CHARACTER NEEDING PAD  
051.006 167 1398 MOV M,A SET CHARACTER  
051.007 072 016 047 1399 LDA HOSTAB+FLT.TDT (A) = TRACK DELAY TIME  
051.012 062 115 040 1400 STA D.MAIA SET TRACK DELAY TIME  
051.015 072 017 047 1401 LDA HOSTAB+FLT.CDR  
051.020 062 343 040 1402 STA S.CDB SET CONSOLE DEFINITION BYTE  
051.023 052 020 047 1403 LHLD HOSTAB+FLT.CRD  
051.026 042 344 040 1404 SHLD S.BAUD SET CONSOLE BAUD RATE  
051.031 072 022 047 1405 LDA HOSTAB+FLT.BOP  
051.034 062 034 041 1406 STA S.BOOTF SET UP BOOT FLAGS  
051.037 072 023 047 1407 LDA HOSTAB+FLT.SAL  
051.042 062 245 074 1408 STA SALONE SET UP STAND-ALONE FLAG  
051.045 311 1409 RET

1412 \*\* SCD - SETUP CONSOLE DRIVER.  
1413 \*  
1414 \* SCD SETS UP INTERRUPT VECTORS FOR CONSOLE INPUT, AND  
1415 \* SETS UP THE USART.  
1416  
1417  
051.046 1418 SCD EQU \*  
000.001 1419 IF DEBUG  
1420 LDA 40077A  
1421 ANA A  
1422 RNZ AM IN HBUG  
1423 ENDIF  
051.046 041 233 064 1424 LXI H,SCINI  
051.051 042 046 .040 1425 SHLD ,UIVEC+7 .SETUP VECTOR.  
051.054 315 063 051 1426 CALL SCU  
051.057 315 220 051 1427 CALL ECI  
051.062 311 1428 RET  
051.063 1429 XTEXT SCU

1431X \*\* SCU - SETUP CONSOLE USART.  
1432X \*  
1433X \* SCU CONFIGURES THE CONSOLE USART.  
1434X \*  
1435X \* IF 8250  
1436X \* THEN PORT = 372-39  
1437X \* ELSE PORT = 340-70  
1438X \*  
1439X \*  
1440X \* ENTRY NONE  
1441X \* EXIT NONE  
1442X \* USES A,F,(BC),(HL)  
1443X  
1444X  
051.063 072 343 040 1445X SCU LDA S.CDB  
051.066 374 .001 1446X CPI CDB,H84  
051.070 312 133 051 1447X JZ SCU1 IF 8250  
1448X  
1449X \* PRESET 8251  
1450X  
051.073 076 201 1451X MVI A,2010  
051.075 323 373 1452X OUT SC,UART+USR .GET\_USART\_IN\_KNOWN\_STATE  
051.077 323 373 1453X OUT SC,UART+USR  
051.101 323 373 1454X OUT SC,UART+USR  
051.103 323 373 1455X OUT SC,UART+USR  
051.105 076 100 1456X MVI A,UCI.IR .RESET  
051.107 323 373 1457X OUT SC,UART+USR  
051.111 072 327 .040 1458X LDA S.CONTY  
051.114 346 010 1459X ANI CTP.2SB  
000.000 1460X ERRNZ CTP.2SB\*16+UM1,1B-UM1,2B  
051.116 007 1461X RLC  
051.117 007 1462X RLC  
051.120 007 1463X RLC  
051.121 007 1464X RLC

```

051.122 366 116 1465X ORI UMI.1B+UMI.L8+UMI.16X
051.124 323 373 1466X OUT SC.UART+USR
051.126 076 025 1467X MVI A,UCI.ER+UCI.RE+UCI.TE
051.130 323 373 1468X OUT SC.UART+USR
051.132 311 1469X RET
1470X
1471X * IS 8250
1472X
051.133 333 355 1473X SCU1 IN SC.ACE+UR.LSR /80.01.GC/
051.135 346 190 1474X ANI UC.TSE CHECK FOR SHIFT EMPTY /80.01.GC/
051.137 312 133 051 1475X JZ SCU1 /80.01.GC/
1476X
051.142 257 1477X XRA A /79.01.GC/
051.143 323 351 1478X OUT SC.ACE+UR.IER TURN OFF ANY INTERRUPTS /79.01.GC/
051.145 076 020 1479X MVI A,UC.L00 /79.01.GC/
051.147 323 354 1480X OUT SC.ACE+UR.MCR /79.01.GC/
051.151 052 344 040 1481X LHLD S.BAUD
051.154 076 200 1482X MVI A,UC.DLA
051.156 323 353 1483X OUT SC.ACE+UR.LCR ACCESS DIVISOR LATCHES
051.160 175 1484X MOV A,L
051.161 323 350 1485X OUT SC.ACE+UR.DLM SET LEAST SIGNIFICANT
051.163 174 1486X MOV A,H
051.164 346 177 1487X ANI 1770 TRIM STOP BITS
051.166 323 351 1488X OUT SC.ACE+UR.DLM SET MOST SIGNIFICANT
051.170 072 327 040 1489X LDA S.CONTY
051.173 346 010 1490X ANI CTF,2SB
051.175 017 1491X RRC
000.000 1492X ERRNZ CTF,2SB/2-UC,2SB
000.000 1493X ERRNZ UC.2SB-4 (A) = UC.2SB IF 2 STOP BITS
051.176 366 003 1494X ORI UC.8BW 8 BIT WORDS
051.200 323 353 1495X OUT SC.ACE+UR.LCR
051.202 076 156 1496X MVI A,AC.DLY /79.01.GC/
051.204 315 053 000 1497X CALL .DLY /79.01.GC/
051.207 333 350 1498X IN SC.ACE+UR.RBR GOBBLE ANY TRASH /79.01.GC/
051.211 333 354 1499X IN SC.ACE+UR.MCR /79.01.GC/
051.213 346 357 1500X ANI 3770-UC.L00 /79.01.GC/
051.215 323 354 1501X OUT SC.ACE+UR.MCR /79.01.GC/
051.217 311 1502X RET
051.220 1503 XTEXT ECI

```

1505X \*\* ECI - ENABLE CONSOLE INTERRUPTS

1506X \*

1507X \* ENTRY NONE

1508X \* EXIT NONE

1509X \* USES (PSW)

1510X \*

1511X

051.220 072 343 040 1512X ECI LDA S.CDB

051.223 376 001 1513X CPI CDB.H84

051.225 312 235 051 1514X JZ ECI1 IF 8250

1515X

1516X \* HAVE 8251

1517X

HDOS SYSTEM BOOT CODE

SCD - SETUP CONSOLE DRIVER

HEATH HBASM V1.4 01/20/78

PAGE 36

ECI

14:01:15 16-MAY-80

051.230 076 027	1518X	MVI A,UCI.RE+UCI.TE+UCI.ER+UCI.IE
051.232 323 373	1519X	OUT SC.UART+USR
051.234 311	1520X	RET
	1521X	
	1522X *	HAVE 8250
	1523X	
051.235 076 001	1524X ECI1	MVI A,UC.EDA
051.237 323 351	1525X	OUT SC.ACE+UR,IER
051.241 311	1526X	RET

1529 \*\* GVM - GIVE VERSION MESSAGE.  
1530 \*  
1531 \* ENTRY NONE  
1532 \* EXIT NONE  
1533 \* USES ALL  
1534  
1535  
051.242 315 220 051 1536 GVM CALL ECI  
051.245 315 136 031 1537 CALL \$TYPTX  
051.250 012 012 110 1538 DB NL,NL,'HDOS Version '  
051.267 061 056 066 1539 DB VERS'16+'0',',VER\$00001111B+'0'  
051.272 012 040 111 1540 DB NL,' Issue # 50.05.00',ENL  
051.315 311 1541 RET

1545 \*\* MSD - MOUNT SYSTEM DISK.

1546 \*

1547 \* MSD MOUNTS THE SYSTEM DISK.

1548 \*

1549 \* 1) ABORT DRIVER

1550 \* 2) READ LABEL RECORD

1551 \* 3) SET VOLUME NUMBER FOR DRIVER

1552 \*

1553 \* EXIT LABEL = LABEL SECTOR

1554

1555

051.316 076 007 1556 MSD MVI A,DC.ABT

051.320 315 130 040 1557 CALL SYDD ABORT DRIVER

051.323 001 000 001 1558 LXI B,256

051.326 021 211 047 1559 LXI D,LABEL

051.331 041 011 000 1560 LXI H,DDF.LAB

051.334 076 002 1561 MVI A,DC.RER READ REGARDLESS

051.336 315 130 040 1562 CALL SYDD

051.341 334 205 053 1563 CC BOOTERR BAD ERROR

1564

1565 \* CALL DEVICE MOUNT ROUTINE

1566

051.344 072 211 047 1567 LDA LABEL+LAB.SER

051.347 157 1568 MOV L,A

051.350 046 000 1569 MVI H,O (HL) = SERIAL NUMBER

051.352 076 010 1570 MVI A,DC.MOU

051.354 315 130 040 1571 CALL SYDD MOUNT UNIT

051.357 334 205 053 1572 CC BOOTERR BAD ERROR

1573

1574 \* SETUP ENTRY IN REVLIST

1575

051.362 315 364 070 1576 CALL GSP HL = POINTER TO SYSTEM UNIT DATA

051.365 353 1577 XCHG

051.366 052 216 047 1578 LHLD LABEL+LAB.GRT

051.371 353 1579 XCHG DE = GRT SECTOR ADDRESS

051.372 315 350 071 1580 CALL \$INDS SAVE GRT SECTOR

051.375 003 000 1581 DW UNT.GTS

051.377 353 1582 XCHG

052.000 052 214 047 1583 LHLD LABEL+LAB.DIS

052.003 353 1584 XCHG DE = FIRST SECTOR OF DIRECTORY

052.004 315 350 071 1585 CALL \$INDS SAVE FIRST SECTOR OF DIRECTORY

052.007 005 000 1586 DW UNT.DIS

052.011 345 1587 PUSH H

1588

052.012 041 052 052 1589 LXI H,MSDA

052.015 315 065 052 1590 CALL IMM ISSUE MOUNT MESSAGE

1591

1592 \* SEE IF DEVICE IS READ-ONLY

1593

052.020 341 1594 POP H

000.000 1595 ERRNZ UNT.FLG

1596 \* LXI D,UNT.FLG

1597 \* DAD D (HL) = UNIT FLAG

052.021 176 1598 MOV A,M

052.022 346 373 1599 ANI 377Q-DT.CW ASSUME READ ONLY

052.024 167 1600 MOV M,A

052.025 345 1601 PUSH H SAVE ADDRESS  
1602  
1603 \* ATTEMPT ZERO-LENGTH WRITE TO SEE IF WRITE PROTECTED  
1604  
052.026 001 000 000 1605 LXI B,0  
052.031 140 1606 MOV H,B  
052.032 150 1607 MOV L,B TRY SECTOR 0  
052.033 076 001 1608 MVI A,DC,WRI  
052.035 315 130 040 1609 CALL SYDD TRY WRITE  
052.040 341 1610 POP H  
052.041 332 050 052 1611 JC MSD1 IS WRITE-PROTECTED  
1612 \* LDA LABEL+LAB.SER SEE IF UNIT #0  
1613 \* ANA A  
1614 \* JZ MSD1 IS VOLUME 0.0. LEAVE WRITE PROTECTED  
052.044 176 1615 MOV A,M  
052.045 366 004 1616 ORI DT,CW  
052.047 167 1617 MOV M,A SET WRITEABLE  
052.050 247 1618 MSD1 ANA A CLEAR CARRY  
052.051 311 1619 RET  
052.052 115 157 165 1620  
052.052 115 157 165 1621 MSDA DB 'Mounted On', '+200Q

1623 \*\* IMM - ISSUE MOUNT MESSAGE.  
1624 \*  
1625 \* IMM TYPES THE MOUNTING MESSAGE:  
1626 \*  
1627 \* VOLUME NNN MOUNTED ON DEV:  
1628 \* LABEL: XXXX ... XXX  
1629 \*  
1630 \* ENTRY LABEL SECTOR READ  
1631 \* (HL)= ADDRESS OF MESSAGE VERB STRING (.PRINT FORMAT)  
1632 \* EXIT NONE  
1633 \* USES ALL  
1634  
1635

052.065 072 211 047 1636 IMM LDA LABEL+LAB.SER  
052.070 345 1637 PUSH H SAVE VERB  
052.071 117 1638 MOV C,A  
052.072 006 000 1639 MVI B,0  
052.074 041 146 052 1640 LXI H,IMMB  
052.077 076 003 1641 MVI A,3  
052.101 315 157 031 1642 CALL \$UDD UNPACK VOLUME NUMBER  
052.104 041 134 052 1643 LXI H,IMMA  
052.107 377 003 1644 DB SYSCALL,.PRINT PRINT MESSAGE  
052.111 341 1645 POP H  
052.112 377 003 1646 DB SYSCALL,.PRINT PRINT VERB  
052.114 041 153 052 1647 LXI H,IMMC  
052.117 377 003 1648 DB SYSCALL,.PRINT PRINT THE REST OF IT  
052.121 041 232 047 1649 LXI H,LABEL+LAB.LAB  
052.124 315 011 060 1650 CALL \$DTB DELETE TRAILING BLANKS  
052.127 075 1651 DCR A REMOVE 00 COUNT  
052.130 304 372 061 1652 CNZ \$TYPCC TYPE IF NON-NUL  
052.133 303 205 061 1653 JMP \$CRLF CRLF AND EXIT

HDD'S SYSTEM BOOT CODE  
MSD - MOUNT SYSTEM DISK..... IMM..... HEATH HBASM V1.4 01/20/78 PAGE 40  
14:01:19 16-MAY-80

1654  
052.136 012 126 157 1655 IMMA DB NL,'Volume,  
052.146 130 130 130 1656 IMMB DB 'XXX,' '+200Q  
052.153 123 131 060 1657 IMMC DB 'SY0!',NL,'Label:',', '+200Q

1661 \*\* PGT - PREPARE.GRT.  
1662 \*  
1663 \* PGT PREPARES THE GROUP RESERVATION TABLE BY READING BOTH THE  
1664 \* GRT AND THE RGT INTO MEMORY.  
1665 \*  
1666 \* THE GROUPS UNRESERVED VIA THE RGT ARE FLAGGED FREE (LINK TO GROUP 1)  
1667 \*  
1668 \* EACH DIRECTORY ENTRY IS THEN CHECKED, AND ITS GROUP IS  
1669 \* FOLLOWED THROUGH THE GRT. THE CHAIN IS DUPLICATED INTO THE GRT BEING  
1670 \* BUILT.  
1671 \*  
1672 \* WHEN THIS PROCESS IS COMPLETE, ANY UNUSED GROUPS ARE CHAINED TO  
1673 \* THE FREE LIST.  
1674 \*  
1675 \* ENTRY NONE  
1676 \* EXIT (HL) = SECTOR ADDRESS OF LAST DIRECTORY BLOCK CONTAINING FILES  
1677 \* USES ALL  
1678  
1679  
052.167 315 241 031 1680 PGT CALL \$WER WRITE ENABLE PROTECTED RAM  
052.172 001 000 001 1681 LXI B,256  
052.175 021 000 024 1682 LXI D,S,GRT0  
052.200 041 012 000 1683 LXI H,00F,RGT  
000.000 1684 ERRNZ DC.REA  
052.203 257 1685 XRA A (A) = DC.REA  
052.204 315 130 040 1686 CALL SYDD  
052.207 334 205 053 1687 CC BOOTERR  
1688  
052.212 001 000 001 1689 LXI B,256  
052.215 315 364 070 1690 CALL GSP  
052.220 345 1691 PUSH H SAVE SYSTEM UNIT POINTER  
052.221 315 234 030 1692 CALL \$INDL  
052.224 003 000 1693 DW UNT.GTS  
052.226 353 1694 XCHG HL = GRT SECTOR  
052.227 021 000 025 1695 LXI D,PGTA  
000.000 1696 ERRNZ DC.REA  
052.232 257 1697 XRA A (A) = DC.REA  
052.233 315 130 040 1698 CALL SYDD READ THE GRT SECTOR  
052.236 334 205 053 1699 CC BOOTERR  
1700  
052.241 257 1701 XRA A  
052.242 062 000 025 1702 STA PGTA CLEAR OLD FREE CHAIN  
052.245 341 1703 POP H RESTORE SYSTEM UNIT POINTER  
052.246 315 234 030 1704 CALL \$INDL  
052.251 005 000 1705 DW UNT.DIS  
052.253 353 1706 XCHG HL = FIRST DIRECTORY SECTOR  
052.254 042 065 077 1707 SHLD SECSCR+DIS.LNK SETUP FIRST DIRECTORY SECTOR LINK  
1708  
1709 \* READ DIRECTORY SECTOR  
1710  
052.257 052 065 077 1711 PGT3 LHLD SECSCR+DIS.LNK  
052.262 174 1712 MOV A,H  
052.263 265 1713 ORA L  
052.264 312 375 052 1714 JZ PGT7 ALL DONE  
052.267 001 000 002 1715 LXI B,512  
052.272 021 067 075 1716 LXI D,SECSCR

PGT - PREPARE.GRT

PGT

14:01:22 16-MAY-80

```

000.000 1717 ERRNZ DC,REA
052.275 257 1718 XRA A (A) = DC,REA
052.276 315 130 040 1719 CALL SYDD
052.301 334 205 053 1720 CC BOOTERR
1721
1722 * SCAN DIRECTORY FOR ENTRYS. TRANSFER THE CHAIN TO THE NEW GRT.
1723
052.304 041 067 075 1724 LXI H,SECSQR
1725
1726 * CHECK NEXT ENTRY
1727
052.307 176 1728 PGT4 MOV A,M (A) = 1ST CHARACTER OF NAME
000.000 1729 ERRNZ DF,EMF-377Q
052.310 074 1730 INR A
052.311 312 357 052 1731 JZ PGT6 SPACE IS EMPTY
000.000 1732 ERRNZ DF,CLR-376Q
052.314 074 1733 INR A
052.315 312 375 052 1734 JZ PGT7 ALL DONE
052.320 372 047 053 1735 JM PGTERR BAD VALUE HOLDING DIRECTORY
052.323 345 1736 PUSH H SAVE ADDRESS OF DIRECTORY ENTRY
052.324 021 020 000 1737 LXI D,DIR,FGN
052.327 031 1738 DAD D
052.330 156 1739 MOV L,M (L) = FIRST GROUP NUMBER
1740
1741 * COPY CHAIN TO GRT
1742
052.331 046 025 1743 PGT5 MVI H,PGTA/256
052.333 176 1744 MOV A,M
052.334 046 024 1745 MVI H,S,GRT0/256
052.336 065 1746 DCR M SEE IF FREE
052.337 167 1747 MOV M,A
052.340 304 047 053 1748 CNZ PGTERR WAS NOT FREE...DOUBLE LINKAGE
052.343 247 1749 ANA A
052.344 157 1750 MOV L,A
052.345 302 331 052 1751 JNZ PGT5 MORE TO GO
052.350 052 063 077 1752 LHLD SECSCR+DIS,SEC
052.353 042 044 053 1753 SHLD PGTB SAVE SECTOR ADDRESS OF BLOCK
052.356 341 1754 POP H (HL) = DIRECTORY SECTOR POINTER
1755
052.357 072 062 077 1756 PGT6 LD A SECSCR+DIS,ENL
052.362 315 101 030 1757 CALL $1A1A POINT TO NEXT ENTRY
052.365 176 1758 MOV A,M
052.366 247 1759 ANA A SEE IF ENTRY
052.367 302 307 052 1760 JNZ PGT4 MORE ENTRYS TO GO
052.372 303 257 052 1761 JMP PGT3 GET NEW SECTOR
1762
1763 * ALL DONE, LINK UNUSED GUYS
1764
052.375 016 000 1765 PGT7 MVI C,0 (C) = NEXT FREE GROUP
052.377 041 377 024 1766 LXI H,S,GRT0+255 GO FROM BACK TO FRONT
1767
053.002 076 001 1768 PGT8 MVI A,1
053.004 276 1769 CMP M
053.005 302 012 053 1770 JNE PGT9 NOT FREE
053.010 161 1771 MOV M,C LINK TO NEXT FREE
053.011 115 1772 MOV C,L SAVE THIS ONES INDEX

```

```

053.012 055 1773 PGT9 DCR L
053.013 302 002 053 1774 JNZ PGT8 NOT ALL PROCESSED
053.016 161 1775 MOV M,C SET FREE CHAIN
1776
1777 * UPDATE *GRT*
1778
053.017 001 000 001 1779 LXI B,256
053.022 315 364 070 1780 CALL GSP HL = SYSTEM UNIT POINTER
053.025 315 234 030 1781 CALL $INITL
053.030 003 000 1782 DW UNT.GTS
053.032 353 1783 XCHG HL = SECTOR FOR GRT
053.033 021 000 024 1784 LXI D,S.GRT0
053.036 076 001 1785 MVI A,DC.WRI
053.040 315 130 040 1786 CALL SYDD
1787
053.043 041 000 000 1788 LXI H,0 (HL) = SECTOR NUMBER OF LAST BLOCK WITH FILES
053.044 1789 PGTB EQU *-2
053.046 311 1790 RET

```

1792 \*\* PGTERR - LINKAGE ERROR IN DISK FILE STRUCTURE.

```

1793
053.047 315 136 031 1794 PGTERR CALL $TYPTX
053.052 012 007 077 1795 DB NL,BELL,'701 Disk Structure is Corrupt.',NL
053.113 103 157 156 1796 DB 'Contact HEATH Technical Correspondence for Assistance.',ENL
053.202 303 252 053 1797 JMP BOOTABT

```

1799 \*\* BOOTERR - ERROR DURING BOOT.

```

1800 *
1801 * I/O ERRORS COME HERE
1802
053.205 315 136 031 1803 BOOTERR CALL $TYPTX
053.210 012 007 077 1804 DB NL,BELL,'701 Disk I/O Error During Boot.',ENL
1805 * JMP BOOTABT

```

1807 \*\* BOOTABT - ABORT BOOT.

```

1808 *
1809
1810
053.252 315 136 031 1811 BOOTABT CALL $TYPTX
053.255 040 040 102 1812 DB '/... Boot Aborted. Will Restart...:/:/+2000
053.315 303 000 030 1813 JMP 30000A

```

1817 \*\* CDS - CLEAR DIRECTORY SPACES.

1818 \*  
1819 \* CDS IS CALLED TO FLAG THE UNUSED ENTRYS AT THE  
1820 \* END OF THE DIRECTORY AS CLEAR.1821 \*  
1822 \* WHEN A FILE IS DELETED, ITS ENTRY IS FLAGED EMPTY. CDS LOCATES  
1823 \* EMPTY SPOTS WHICH ARE AFTER THE LAST FILE IN THE DIRECTORY,  
1824 \* AND FLAGS THEM CLEAR.1825 \*  
1826 \* ENTRY... (HL) = SECTOR NUMBER OF LAST DIRECTORY BLOCK WITH FILES  
1827 \* EXIT NONE  
1828 \* USES ALL

1829

1830  
053.320 353 1831 CDS XCHG  
053.321 315.364.070 1832 CALL GSF  
053.324 315 327 071 1833 CALL \$INDLB A = UNIT CAPABILITY FLAG BYTE  
053.327 090.000 1834 DW UNT.FLG053.331 353 1835 XCHG  
053.332 346.004 1836 ANI IT.CW  
053.334 310 1837 RZ SYSTEM DISK IS WRITE DISABLED053.335 001.000.002 1838 LXI B,512  
053.340 021 067 075 1839 LXI D,SECSCR  
053.343 325 1840 PUSH D SAVE #SECSCR053.344 315 275 031 1841 CALL S.READ READ DIRECTORY BLOCK  
053.347 321 1842 POP D  
053.350 142 1843 MOV H,D  
053.351 153 1844 MOV L,E (DE) = (HL) = #SECSCR1845  
1846 \*. FIND LAST FILE NAME IN THIS BLOCK

1847

053.352 176 1848 CDS1 MOV A,M  
053.353 247 1849 ANA A  
053.354 312.375.053 1850 JZ CIS3 END OF BLOCK  
053.357 372 364 053 1851 JM CIS2 EMPTY OR CLEAR  
053.362 124 1852 MOV D,H  
053.363 135 1853 MOV E,L (DE) = ADDRESS OF THAT FILE NAME  
053.364 .072.062.077. 1854 CIS2 LDA SECSCR+DIS.ENL  
053.367 315 101 030 1855 CALL \$IADA.  
053.372 303.352.053 1856 JMP CIS1 TRY NEXT ONE1857  
1858 \*. ALL EMPTY SPOTS FOLLOWING THAT LAST NAME TO BE FLAGGED CLEAR

1859

053.375 353 1860 CIS3 XCHG (HL) = ADDRESS OF LAST FILE ENTRY

1861

053.376 176 1862 CIS4 MOV A,M (A) = ENTRY FIRST BYTE  
053.377 247 1863 ANA A  
054.000 312.022.054 1864 JZ CIS5 END OF BLOCK  
054.003 107 1865 MOV B,A SAVE ENTRY FLAG  
054.004 362.011.054 1866 JP CIS4.5 IS NOT EMPTY OR CLEAR  
054.007 066 376 1867 MVI M,DF,CLR IS CLEAR NOW  
054.011 .072.062.077. 1868 CIS4.5 LDA SECSCR+DIS.ENL  
054.014 315 101 030 1869 CALL \$IADA.  
054.017 303.376.053 1870 JMP CIS41871  
1872 \*. BLOCK IS CORRECTED, WRITE BACK TO DISK

1873  
054.022 305 1874 CDS5 PUSH B SAVE (B) FLAG  
054.023 001 000 002 1875 LXI B,512  
054.026 021 067 075 1876 LXI D,SECSCR  
054.031 052 063 077 1877 LHLD SECSCR+DIS.SEC  
054.034 315 330 031 1878 CALL S.WRITE  
054.037 301 1879 POP B  
1880  
1881 \* IF THE LAST ENTRY IN THIS BLOCK IS NOT CLEAR, MUST CONTINUE  
1882 \* CORRECTIONS TO NEXT BLOCK  
1883  
054.040 076 376 1884 MVI A,DF,CLR  
054.042 270 1885 CMP B  
054.043 310 1886 RF ALL CLEAR  
054.044 052 065 077 1887 LHLD SECSCR+DIS.LNK  
054.047 174 1888 MOV A,H  
054.050 265 1889 ORA L NO MORE TO CORRECT  
054.051 310 1890 RZ  
054.052 001 000 002 1891 LXI B,512  
054.055 021 067 075 1892 LXI D,SECSCR  
054.060 315 275 031 1893 CALL S.READ READ NEXT BLOCK  
054.063 041 067 075 1894 LXI H,SECSCR  
054.066 303 376 053 1895 JMP CDS4 TRY THIS ONE

LSO - LOCATE.SYSTEM.OVERLAY.

14:01:31 16-MAY-80

1898 \*\* LSO - LOCATE SYSTEM OVERLAY.  
 1899 \*  
 1900 \* LSO LOCATES THE SYSTEM OVERLAYS:  
 1901 \* \*HDSOVL0.SYS\*  
 1902 \* \*HDSOVL1.SYS\*  
 1903 \*  
 1904 \* AND SETS UP POINTERS AND OTHER TABLE DATA TO BOTH.  
 1905 \*  
 1906 \*  
 1907 \* IT IS READ, AND THE INFO USED TO SETUP THE CELLS.  
 1908 \*  
 1909 \* S.OMAX SYSTEM OVERLAY MAX  
 1910 \* S.SSN SWAP SECTOR NUMBER  
 1911 \* S.OSN OVERLAY SECTOR NUMBER  
 1912 \* S.OVLS OVERLAY SIZE  
 1913 \*  
 1914 \* ENTRY NONE  
 1915 \* EXIT NONE  
 1916 \* USES ALL  
 1917 \*  
 1918  
 054.071..021.045.055 1919 LSO .LXI D,LSOA  
 054.074 315 204 054 1920 CALL LSO. (HL) = SECTOR NUMBER  
 054.077..042.002.041 1921 SHLD S,SSN SET SWAP NUMBER  
 054.102 021 014 000 1922 LXI D,SB,OVMX/256  
 054.105..031 1923 RAD D (HL) = SECTOR FOR CODE  
 054.106 042 347 072 1924 SHLD OVLO\*OVL.ENS+OVLTAB+OVL,COD  
 054.111..315.021.055 1925 CALL LSO.. (HL) = LENGTH  
 054.114 042 324 040 1926 SHLD S.OMAX SET OVERLAY MAXIMUM SIZE  
 054.117..042.351.072 1927 SHLD OVLO\*OVL.ENS+OVLTAB+OVL,SIZ  
 1928  
 1929 \* SET UP \*HDSOVL2.SYS\*  
 1930  
 054.122..021.062.055 1931 .LXI D,LSOB  
 054.125 315 204 054 1932 CALL LSO. (HL) = SECTOR NUMBER FOR CODE  
 054.130..042.357.072 1933 SHLD OVL1\*OVL.ENS+OVLTAB+OVL,COD  
 054.133 315 021 055 1934 CALL LSO.. (HL) = LENGTH OF OVERLAY  
 054.136..042.361.072 1935 SHLD OVL1\*OVL.ENS+OVLTAB+OVL,SIZ  
 054.141 353 1936 XCHG  
 054.142..052.324.040 1937 LHLD S.OMAX  
 054.145 315 352 057 1938 CALL HLCPIE  
 054.150 320 1939 RNC S.OMAX >= SIZE OF THIS OVERLAY  
 054.151 315 136 031 1940 CALL \$TYPTX  
 054.154..012.077.060 1941 DB NL,'?91.Overlay too big',ENL  
 054.201 303 205 053 1942 JMP BOOTERR  
 1943  
 054.204 325 1944 LSO, PUSH D SAVE FILE NAME POINTER  
 054.205..001.015.000 1945 LXI B:DIRIDL (BC) = COUNT  
 054.210 041 062 041 1946 LXI H:AIO.DIR+DIR.NAM  
 054.213..315.252.030 1947 CALL \$MOVE MOVE IN NAME PATTERN  
 054.216 001 015 000 1948 LXI B:DIRIDL (BC) = MATCH LENGTH  
 054.221 052 214 047 1949 LHLD LABEL+LAB.RIS (HL) = DIRECTORY SECTOR FWA  
 054.224 315 331 056 1950 CALL LDE.. LOCATE DIRECTORY ENTRY  
 054.227..322.277.054 1951 JNC LSO1 GOTIT  
 1952  
 1953 \* MISSING OVERLAY FILE

1954  
054.232 315 136 031 1955 CALL \$TYPTX  
054.235 012 077 060 1956 DB NL,'?01 Missing File','+2000  
054.257 321 1957 POP D RESTORE FILE NAME POINTER  
054.260 001 015 000 1958 LXI B,DIRIDL SET UP COUNT  
054.263 041 062 041 1959 LXI H,AIO,DIR+DIR.NAM SET UP DESTINATION FOR FILE NAME  
054.266 315 252 030 1960 CALL \$MOVE MOVE IN NAME PATTERN  
054.271 315 222 061 1961 CALL \$TFN TYPE FILE NAME  
054.274 303 205 053 1962 JMP BOOTERR ABORT BOOT

1963  
1964 \* FOUND OVERLAY

1965  
054.277 321 1966 LS01 POP D DISCARD FILE NAME POINTER SINCE IT IS FOUND  
054.300 021 016 000 1967 LXI D,DIR.FLG  
054.303 031 1968 DAD D  
054.304 176 1969 MOV A,M (A) = FLAG BYTE  
054.305 346 020 1970 ANI DIF.CNT  
054.307 312 326 054 1971 JZ LS02 NOT CONTIGUOUS  
000.000 1972 ERRNZ DIR.FGN-DIR.FLG-2  
054.312 043 1973 INX H  
054.313 043 1974 INX H (HLO = #DIR.FGN  
054.314 136 1975 MOV E,M  
054.315 026 000 1976 MVI D,0 (DE) = FILE FIRST GROUP NUMBER  
054.317 072 220 047 1977 LDA LABEL+LAB.SPG  
054.322 315 007 031 1978 CALL \$MU86 (HL) = SECTOR NUMBER  
054.325 311 1979 RET

1980  
1981 \* OVERLAY IS NOT CONTIGUOUS.

1982  
054.326 315 136 031 1983 LS02 CALL \$TYPTX  
054.331 012 077 060 1984 DB NL,'?01 System Not SYSGENed Properly, or Files Damaged.',ENL  
055.016 303 205 053 1985 JMP BOOTERR

1986  
055.021 001 000 001 1987 LS0.. LXI B,256  
055.024 021 000 025 1988 LXI D,BUFF  
055.027 315 241 031 1989 CALL \$WER WRITE ENABLE RAM AREA  
055.032 315 275 031 1990 CALL S.READ READ FROM DISK  
055.035 052 002 025 1991 LHLD BUFF+PIC.LEN  
055.040 001 010 000 1992 LXI B,8  
055.043 011 1993 DAD B  
055.044 311 1994 RET

1995  
055.045 110 104 117 1996 LSOA DB 'HDOSOUL0','SYS',0,0 OVERLAY FILE NAME  
000.000 1997 ERRNZ \*-LSOA-BIRIDL LSOA IS ENTIRE SPECIFICATION  
055.062 110 104 117 1998 LS0B DB 'HDOSOVL1','SYS',0,0  
000.000 1999 ERRNZ \*-LSOB-BIRIDL

14:01:38 16-MAY-80

## SDT - SETUP DEVICE TABLE

```

2002 ** SDT - SETUP DEVICE TABLE.
2003 *
2004 * SDT SCANS THE SYSTEM DISK DIRECTORY LOOKING FOR FILES IN
2005 * THE FORM:
2006 *
2007 * XX DVD
2008 *
2009 *. THESE ENTRYS ARE BUILT INTO THE DEVICE TABLE
2010
2011
055.077 2012 SDT EQU *
055.077 052.214.047 2013 LHLD LABEL+LAB.DIS
055.102 042 376 027 2014 SHLD SDTA+DIS.LNK SET SECTOR NUMBER TO READ
2015
2016 * READ NEXT SECTOR
2017
055.105 052 376 027 2018 STD1 LHLD SDTA+DIS.LNK
055.110 174 2019 MOV A,H
055.111 265 2020 ORA L
055.112 310 2021 RZ NO MORE DIRECTORY, AM DONE
055.113 021 000 026 2022 LXI D,SDTA
055.116 001.000.002 2023 LXI B,512
055.121 315 241 031 2024 CALL $WER WRITE ENABLE RAM
055.124 315.275.031 2025 CALL S.READ READ DIRECTORY
2026
2027 *. RUN DOWN THROUGH ENTRYS LOOKING FOR XX,DVD
2028
055.127 041.000.026 2029 LXI H,SDTA
055.132 176 2030 SDT2 MOV A,M
055.133 247 2031 ANA A
055.134 312 105 055 2032 JZ STD1 END OF SECTOR
000.000 2033 ERRNZ DF,EMP-3770
055.137 074 2034 INR A
055.140 312.177.055 2035 JZ SDTA ENTRY IS EMPTY
000.000 2036 ERRNZ DF,CLR-3760
055.143 074 2037 INR A
055.144 310 2038 RZ NO MORE IN DIRECTORY
055.145 345 2039 PUSH H
055.146 043 2040 INX H
055.147 176 2041 MOV A,M
055.150 247 2042 ANA A
055.151 312.176.055 2043 JZ SDT3 IS ONE-CHARACTER NAME
055.154 043 2044 INX H
055.155 021.210.055 2045 LXI D,SDTB
055.160 001 013 000 2046 LXI B,SDTBL
055.163 315.060.030 2047 CALL $COMP COMPARE
055.166 302 176 055 2048 JNE SDT3 NOT MATCH
2049
2050 * GOT ONE
2051
055.171 341 2052 POP H
055.172 345 2053 PUSH H (HL) = ENTRY FWA
055.173 315 223 055 2054 CALL EDL ENTER DRIVER IN LIST
2055
2056 * TRY ANOTHER ENTRY
2057

```

055.176 341 2058 SDT3 POP H (HL) = ENTRY FWA  
055.177 072 373 027 2059 SDT4 LDA SDTATDIS.ENL  
055.202 315 101 030 2060 CALL \$DADA. ADVANCE  
055.205 303 132 055 2061 JMP SDT2 TRY.NEXT  
2062  
055.210 000 000 000 2063 SDTB DB 0,0,0,0,0,0,0,DVD',0,0 REQUIRED EXTENSION  
000.013 2064 SDTBL EQU \*-SDTB LENGTH OF PATTERN

2066 \*\* EDL - ENTER DEVICE IN DEVICE LIST.  
2067 \*  
2068 \* EDL ENTERS DEVICE DRIVER INFORMATION INTO THE  
2069 \* DEVLIST.  
2070 \*  
2071 \* THE FILE IS READ TO SETUP THE DEVICE TABLE ENTRY.  
2072 \*  
2073 \* ENTRY (HL) = FWA DIRECTROY ENTRY FOR DRIVER  
2074 \* EXIT DRIVER IN DEVLIST IF ALL OK  
2075 \* DRIVER IGNORED IF PROBLEMS  
2076 \* USES ALL  
2077  
2078  
055.223 136 2079 EDL MOV E,M  
055.224 043 2080 INX H  
055.225 126 2081 MOV D,M (DE) = NAME  
055.226 353 2082 XCHG  
055.227 042 264 056 2083 SHLD EDLNAM SET NAME FIELD IN DEVLIST ENTRY  
055.232 042 255 056 2084 SHLD EDLC SET NAME FOR MESSAGE  
2085  
2086 \* SETUP SECTOR ADDRESS FOR DRIVER  
2087  
055.235 041 017 000 2088 LXI H,DIR.FGN-1  
055.240 031 2089 DAD D (HL) = #DIR.FGN  
055.241 176 2090 MOV A,M (A) = FIRST GROUP  
055.242 062 302 056 2091 STA EDLDVG SET DRIVER FIRST GROUP  
2092  
2093 \* READ FIRST SECTOR OF DRIVER FILE  
2094  
055.245 137 2095 MOV E,A  
055.246 026 000 2096 MVI D,0 (DE) = GROUP  
055.250 072 376 072 2097 LDA DEVLIST+DEV.SPG (A) = SECTORS PER GROUP  
055.253 315 007 031 2098 CALL \$MUB6 (HL) = SECTOR ADDRESS OF 1ST GROUP  
055.256 021 000 025 2099 LXI D,BUFF  
055.261 001 000 001 2100 LXI B,256  
055.264 315 241 031 2101 CALL \$WER WRITE ENABLE RAM AREA  
055.267 315 275 031 2102 CALL S.READ READ IT  
2103  
2104 \* SEE IF PIC FILE  
2105  
055.272 052 000 025 2106 LHLD BUFF  
055.275 054 2107 INR L  
055.276 302 201 056 2108 JNZ EDL5 NOT BINARY  
055.301 076 001 2109 MVI A,FT.PIC  
055.303 274 2110 CMP H

EDL.....14:01:47...16-MAY-80.....

055.304 302 201 056 2111 JNE EDL5 NOT PIC  
2112  
2113 \* SET DEVICE CAPABILITY BYTE  
2114  
055.307 072 006 025 2115 LDA BUFF+DVD.DVD (A) = DRIVER FLAG  
055.312 376 307 2116 CPI DIVIFLY SEE IF DRIVER  
055.314 302 201 056 2117 JNE EDL5 NOT DRIVER  
055.317 072 007 025 2118 LDA BUFF+DRV.CAP.  
055.322 062 272 056 2119 STA EDLCAP SET DEVICE CAPABILITY FLAGS  
055.325 072 010 025 2120 LDA BUFF+DVD.MUM.  
055.330 062 274 056 2121 STA EDLMUM SET UP MOUNTED UNITS MASK  
055.333 072 011 025 2122 LDA BUFF+DVD.MNU  
055.336 062 275 056 2123 STA EDLMNU SET MAXIMUM NUMBER OF UNITS  
2124  
2125 \* ALLOCATE UNIT DESCRIPTOR TABLES  
2126  
055.341 072 011 025 2127 LDA BUFF+DVD.MNU A = MAX. NUMBER OF UNITS  
055.344 021 007 000 2128 LXI D, UNT.SIZ  
055.347 315 007 031 2129 CALL \$MU86 HL = MEMORY TO ALLOCATE  
055.352 315 224 030 2130 CALL \$CHL  
055.353 353 2131 XCHG  
055.356 052 356 040 2132 LHLD S,RFWA  
055.361 031 2133 DAD D HL = NEW FWA  
2134  
055.362 042 356 040 2135 SHLD S,RFWA  
055.365 042 320 040 2136 SHLD S,SYSM  
055.370 042 276 056 2137 SHLD EDLPTR  
2138  
2139 \* INITIALIZE THE UNIT DESCRIPTOR TABLE  
2140  
055.373 072 007 025 2141 LDA BUFF+DVD.CAP  
055.376 107 2142 MOV B,A B..= DEVICE CAPABILITY FLAGS  
055.377 021 012 025 2143 LXI D,BUFF+DVI.UFL  
056.002 072 011 025 2144 LDA BUFF+DVD,MNU  
2145  
056.005 075 2146 EDLO ICR A  
056.006 372 034 056 2147 JM EDLO.5 FINISHED WITH THE UNITS  
2148  
056.011 365 2149 PUSH PSW  
056.012 032 2150 LDAX D A..= FLAG VALUE FOR THIS UNIT  
056.013 240 2151 ANA B MAP OUT ILLEGAL BITS  
056.014 315 004 072 2152 CALL \$INDSR  
056.017 000 000 2153 DW UNT.FLG  
056.021 325 2154 PUSH D  
056.022 021 007 000 2155 LXI D, UNT.SIZ  
056.025 031 2156 DAD D HL = NEXT UNIT DESCRIPTOR  
056.026 321 2157 POP D  
056.027 023 2158 INX D MOVE TO NEXT UNIT  
056.030 361 2159 POP PSW  
056.031 303 005 056 2160 JMP EDLO  
2161  
056.034 2162 EDLO.5 EQU \*  
2163  
2164 \* SET LENGTH  
2165  
056.034 052 004 025 2166 LHLD BUFF+PIC.PTR (HL) = CODE LENGTH

056.037 001 000 376 2167 LXI B,-DVD.ENT  
056.042 .011 2168 DAD B (HL) = LEN OF DRIVER CODE  
056.043 322 201 056 2169 JNC EDLS TOO SMALL  
056.046 042 300 056 2170 SHLD EDLDVL SET DRIVER LENGTH  
2171  
2172 \* HAVE BUILT ENTRY FOR DEVLIST. INSERT  
2173  
056.051 052 354 040 2174 LHLD S,DFWA  
056.054 006 006 2175 MVI B,DEVCNT-1 (B) = MAX DRIVER COUNT  
000.005 2176 ERRMI DEVCNT-2 REQUIRE 2  
056.056 021 017 000 2177 LXI D,DEVELEN  
2178  
056.061 031 2179 EDL1 DAD D (HL) = ADDRESS OF NEXT ENTRY  
056.062 .174 2180 MOV A,M  
056.063 267 2181 ORA A  
000.000 2182 ERRNZ DV,EL DEVICE END OF LIST FLAG  
056.064 312 165 056 2183 JZ EDL3 GOT ONE  
056.067 005 2184 DCR B  
056.070 302 061 056 2185 JNZ EDL1 TRY NEXT  
2186  
2187 \* NO ROOM FOR IT.  
2188  
056.073 315 136 031 2189 CALL \$TYPTX  
056.076 012 007 077 2190 DB NL,BELL,'?01 Too Many Device Drivers.,' '+2000  
056.135 041 251 056 2191 EDL2 LXI H,EDLB TYPE NAME  
056.140 .076 012 2192 MVI A,10  
056.142 315 372 061 2193 CALL \$TYPCC TYPE NAME  
056.145 .315 136 031 2194 CALL \$TYPTX  
056.150 040 055 040 2195 DB ' - Ignored.',ENL  
056.164 .311 2196 RET  
2197  
2198 \* GOT SPOT..PUT IT IN  
2199  
056.165 .021 264 056 2200 EDL3 LXI B,EDLDEV  
056.170 001 017 000 2201 LXI B,DEVELEN  
056.173 .315 252 030 2202 CALL \$MOVE COPY INTO TABLE  
056.176 066 000 2203 MVI M,0 CLEAR NEXT ENTRY  
056.200 .311 2204 RET RETURN  
2205  
2206 \* ERROR IN DRIVER FORMAT  
2207  
056.201 .315 136 031 2208 EDL5 CALL \$TYPTX  
056.204 012 007 077 2209 DB NL,BELL,'?01 Format Error in Driver File', '+2000  
056.246 .303 135 056 2210 JMP EDL2  
2211  
056.251 .123 131 060 2212 EDLB DB 'SY0:' DEVICE NAME  
056.255 130 130 2213 EDLC DB 'XX' DRIVER NAME  
056.257 .056 104 126 2214 DB ',DVD',0  
2215  
056.264 .000 2216 EDLDEV EQU \*  
000.000 2217 ERRNZ \*-EDLDEV-DEV.NAM  
056.264 .040 040 2218 EDLNAM DB ' ' DEVICE NAME  
000.000 2219 ERRNZ \*-EDLDEV-DEV.RES  
056.266 .000 2220 DB 0 NOT RESIDENT  
000.000 2221 ERRNZ \*-EDLDEV-DEV.JMP  
056.267 .303 2222 DB 3030 JUMP OPCODE

000.000	2223	ERRNZ	*-EDLDEV-DEV.IDA
056.270.263.071	2224	DW	SDD DRIVER ADDRESS (STAND-IN DEVICE DRIVER)
000.000	2225	ERRNZ	*-EDLDEV-DEV.FLG
056.272.000	2226	EDLCAP	DB 0 FLAGS
000.000	2227	ERRNZ	*-EDLDEV-DEV.SPG
056.273.000	2228	DB	0 SECTORS PER GROUP
000.000	2229	ERRNZ	*-EDLDEV-DEV.MUM
056.274.000	2230	EDLMUM	DB 0 MOUNTED UNIT MASK
000.000	2231	ERRNZ	*-EDLDEV-DEV.MNU
056.275.001	2232	EDLMNU	DB 1 MAXIMUM NUMBER OF UNITS
000.000	2233	ERRNZ	*-EDLDEV-DEV.UNT
056.276.000.000	2234	EDLPTR	DW 0 UNIT POINTER
000.000	2235	ERRNZ	*-EDLDEV-DEV.IVL
056.300.000.000	2236	EDLDVL	DW 0 DRIVER LENGTH
000.000	2237	ERRNZ	*-EDLDEV-DEV.IVG
056.302.000	2238	EDLDVG	DB 0 DRIVER SECTOR FIRST GROUP NUMBER
000.000	2239	ERRNZ	*-EDLDEV-DEVLEN

2243 \*\* LDE - LOCATE DIRECTORY ENTRY.  
2244 \*  
2245 \* LDE LOCATES A DIRECTORY ENTRY CORRESPONDING TO THE AIO.DIR ENTRY.  
2246 \*  
2247 \* ENTRY (BC) = NUMBER OF CHARACTERS TO MATCH ON  
2248 \* EXIT 'C' CLEAR IF FOUND  
2249 \* AIO.DES SETUP  
2250 \* (HL) = ADDRESS OF DIRECTORY ENTRY IN SECSCR  
2251 \* 'C' SET IF NOT FOUND  
2252 \* (A) = CODE  
2253 \* USES ALL  
2254  
2255  
056.303 001.015.000 2256 LDE, LXI B,DIRIDL ENTRY FOR FULL NAME COMPARE.  
056.306 052.053.041 2257 LDE LHLD AIO.DTA  
056.311 021.012.000 2258 LXI D,DEV.UNT  
056.314 031 2259 DAD D HL = DEVICE UNIT TABLE  
056.315 072.061.041 2260 LDA AIO.UNI  
056.320 315.301.071 2261 CALL GUP HL = UNIT TABLE POINTER  
056.323 315.234.030 2262 CALL \$INDL  
056.326 005.000 2263 DW UNT.DIS  
056.330 353 2264 XCHG HL = SECTOR OF FIRST DIRECTORY BLOCK.  
2265  
2266 \*\* ENTRY FOR (HL) = SECTOR NUMBER TO START WITH  
2267  
056.331 305 2268 LDE, PUSH B SAVE COUNT.  
056.332 001.000.002 2269 LXI B,512  
056.335 021.067.075 2270 LXI D,SECSCR  
056.340 042.055.041 2271 SHLD AIO.DES ASSUME WILL FIND IN THIS BLOCK  
056.343 315.241.031 2272 CALL \$WER WRITE-ENABLE PROTECTED RAM.  
056.346 315.256.031 2273 CALL IREAD READ FRM DEVICE  
056.351 301 2274 POP B RESTORE (BC)  
2275  
2276 \* SCAN SECTOR FOR INFO  
2277  
056.352 041.067.075 2278 LXI H,DIS,ENT+SECSCR  
2279  
2280 \* COMPARE  
2281  
056.355 021.062.041 2282 LDE3 LXI D,AIO.DIR+DIR.NAM  
056.360 176 2283 MOV A,M  
056.361 247 2284 ANA A  
056.362 372.375.056 2285 JM LDE3.5 NO ENTRY  
056.365 305 2286 PUSH B SAVE COPY OF (BC)  
056.366 345 2287 PUSH H SAVE ADDRESS  
056.367 315.060.030 2288 CALL \$COMP COMPARE  
056.372 341 2289 POP H  
056.373 301 2290 POP B (BC) = COMPARE.COUNT  
056.374 310 2291 RE GOT MATCH  
056.375 021.027.000 2292 LDE3.5 LXI D,DIRELEN MISSED, SCAN TO NEXT ENTRY.  
057.000 031 2293 DAD D  
057.001 176 2294 MOV A,M  
057.002 247 2295 ANA A  
057.003 302.355.056 2296 JNZ LDE3 MORE IN SECTOR  
2297  
2298 \* DIDNT FIND IT IN THIS SECTOR, TRY NEXT.

2299  
057.006 .052.065.077. 2300 LHLD DIS.LNK+SECSCR  
057.011 042 055 041 2301 SHLD AIO.DES SET POSSIBLE SECTOR INDEX  
057.014 174. 2302 MOV A,H  
057.015 265. 2303 ORA L  
057.016 .302.331.056. 2304 JNZ LDE,. HAVE MORE SECTORS  
057.021 076 014 2305 MVI A,EC.FNF FILE NOT FOUND  
057.023 .067. 2306 STC  
057.024 311 2307 RET

2309 \*\* TDD.=.TYPE.DECIMAL.DIGITS.  
2310 \*  
2311 \*. TDD.TYPES.A.16.BIT.VALUE AS 1 TO 5 DECIMAL DIGITS.  
2312 \*  
2313 \*. ENTRY (D,E).=.VALUE  
2314 \* (A) = DIGIT COUNT  
2315 \*. EXIT VALUE.TYRED.  
2316 \* USES A,B,C,F  
2317  
2318  
057.025 .076.095. 2319 TDD. MVI A,5  
057.027 345 2320 TDD PUSH H  
057.030 .365. 2321 TDD1 PUSH PSW  
057.031 041 074 057 2322 LXI H,TDDA-2  
057.034 .097. 2323 RLC (A).=.DIGIT.NUMBER\*2  
057.035 315 072 030 2324 CALL \$1ADA  
057.040 .174. 2325 MOV A,M  
057.041 043 2326 INX H  
057.042 .146. 2327 MOV H,M  
057.043 157 2328 MOV L,A (HL) = MULTIPLE OF 10  
057.044 .353. 2329 XCHG (DE).=.DEVISOR,(HL).=.VALUE  
057.045 076 377 2330 MVI A,3770  
057.047 .031. 2331 TDD2 DAD D  
057.050 074 2332 INR A  
057.051 .332.047.057. 2333 JC TDD2 IF MORE TO GO  
057.054 306 060 2334 ADI '0'  
057.056 .315.310.061. 2335 CALL \$1TYPEC TYPE.DIGIT  
057.061 175 2336 MOV A,L  
057.062 .223. 2337 SUB E  
057.063 137 2338 MOV E,A REMOVE EXTRA SUBTRACTION  
057.064 .174. 2339 MOV A,H  
057.065 232 2340 SBB D  
057.066 .127. 2341 MOV D,A  
057.067 361 2342 POP PSW  
057.070 .075. 2343 ICR A  
057.071 .302.030.057. 2344 JNZ TDD1 IF MORE DIGITS  
057.074 .341. 2345 PDP H  
057.075 311 2346 RET EXIT  
2347  
057.076 2348 TDDA EQU \*  
057.076 .377.377. 2349 DW -1  
057.100 .366.377. 2350 DW -10  
057.102 .234.377. 2351 DW -100

057.104 030 374 2352 DW -1000  
057.106 360 330 2353 DW -10000

2355 \*\* SSD - SET SYSTEM DATE.  
2356 \*  
2357 \* SSD PROMPTS THE USER AS  
2358 \*  
2359 \* DATE (DD-MMM-YY)?  
2360 \*  
2361 \* THE 'DD-MMM-YY' FIELD IS REPLACED BY THE CURRENT  
2362 \* SYSTEM DATE, IF A VALID ONE IS IN MEMORY.  
2363 \*  
2364 \* IN THIS CASE, HITTING (CR) IN REPLY CAUSES THE CURRENT DATE  
2365 \* TO REMAIN.  
2366 \*

2367 \* ENTRY NONE  
2368 \* EXIT NONE  
2369 \* USES ALL

2370

2371

057.110 315 136 031 2372 SSD CALL \$TYPTX  
057.113 104 141 164 2373 DB 'Date ','+'+2000

2374  
2375 \* TRY TO DECODE THE CURRENT DATE

2376

057.121 021 211 047 2377 LXI D,SSDB  
057.124 052 310 040 2378 LHLD S,DATE  
057.127 353 2379 XCHG  
057.130 315 363 060 2380 CALL \$DAD DECODE AUGUSTAN DATE  
057.133 332 171 057 2381 JC SSD1 ILLEGAL  
057.136 .001 .011 .000 2382 LXI B,9  
057.141 021 277 040 2383 LXI D,S,DATE  
057.144 .041 .211 .047 2384 LXI H,SSDB  
057.147 315 060 030 2385 CALL \$COMP COMPARE TO EXPANDED VERSION  
057.152 .302 .171 .057 2386 JNE SSD1 NO GOOD  
2387

2388 \* CURRENT DATE IS IN VALID FORMAT.. OFFER AS DEFAULT

2389

057.155 .076 .011 2390 MVI A,9  
057.157 .041 .277 .040 2391 LXI H,S,DATE  
057.162 .315 .372 .061 2392 CALL \$TYPCC TYPE DATE  
057.165 257 2393 XRA A ALLOW DEFAULT  
057.166 .303 .207 .057 2394 JMP SSD2 FINISH PROMPT

2395

2396 \* CURRENT DATE IS NO GOOD.. MUST HAVE ONE SUPPLIED

2397

057.171 .315 .136 .031 2398 SSD1 CALL \$TYPTX  
057.174 104 104 055 2399 DB 'DD-MMM-Y','Y'+2000  
057.205 .076 .001 2400 MVI A,1 NO DEFAULT  
2401  
057.207 .062 .351 .057 2402 SSD2 STA SSDA FLAG DEFAULT ALLOWED OR NOT  
057.212 315 136 031 2403 CALL \$TYPTX  
057.215 .051 .077 2404 DB 'D?','+'+2000

..... 2405  
..... 2406 \* .GET.REPLY.  
..... 2407  
057.220..041.211.047. 2408 SSD3 LXI H,SSDB  
057.223 315 106 060 2409 CALL \$RTL READ TEXT LINE (UPPER CASE)  
057.226..342.220.057. 2410 JC SSD3 CTL-D STRUCK  
057.231 176 2411 MOV A,M  
057.232..247. 2412 ANA A  
057.233 302 246 057 2413 JNZ SSD4 GIVEN REPLY  
..... 2414  
..... 2415 \* HE DEFAULTED. SEE IF DEFAULT ALLOWED  
..... 2416  
057.236 072 351 057 2417 LDA SSDA  
057.241..247. 2418 ANA A  
057.242 310 2419 RZ DEFAULT OK  
057.243..303.254.057. 2420 JMP SSD5 MAKE IT MORE CLEAR WHAT WE WANT  
..... 2421  
..... 2422 \* CRACK DATE  
..... 2423  
057.246..315.153.060. 2424 SSD4 CALL \$CAD CONVERT AUGUSTAN DATE  
057.251 322 336 057 2425 JNC SSD6 DATE GOOD  
..... 2426  
..... 2427 \* HIS REPLY BAD. TRY AGAIN  
..... 2428  
057.254 315 136 031 2429 SSD5 CALL \$TYPTX  
057.257..007.040.105. 2430 DB BELL. ENTER DATE AS DD-MMM-YY (I.E., 02-JUL-77), ENL  
057.333 303 220 057 2431 JMP SSD3 TRY AGAIN  
..... 2432  
..... 2433 \* DATE IS GOOD. SETUP TWO DATE FIELDS FOR SYSTEM  
..... 2434  
057.336 353 2435 SSD6 XCHG  
057.337..042.310.040. 2436 SHLD S,DATE SET DATE CODE  
057.342 353 2437 XCHG  
057.343..041.277.040. 2438 LXI H,S,DATE  
057.346 303 363 060 2439 JMP \$DAD DECODE DATE INTO ASCII AND RETURN  
..... 2440  
057.351 000 2441 SSDA DB 0 =0 IFF DEFAULT DATE ALLOWED  
057.352. 2442 XTEXT HLCPDE  
..... 2443 HLCPDE - (HL) COMPARED TO (DE)  
..... 2444X \* THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).  
..... 2445X \*  
..... 2446X \*  
..... 2447X \* ENTRY: (HL)&(DE) SET UP  
..... 2448X \*  
..... 2449X \*  
..... 2450X \* EXIT: (PSW) =  
..... 2451X \* 'Z' SET IF (HL) = (DE)  
..... 2452X \* 'C' SET IF (HL) < (DE)  
..... 2453X \* 'C' CLEAR IF (HL) >= (DE)  
..... 2454X \*  
..... 2455X \*  
..... 2456X \* USES: (PSW)  
..... 2457X \*  
..... 2458X  
057.352..174. 2459X HLCPDE MOV A,H  
057.353 272 2460X CMP D 'C' SET => (A) < (D)  
057.354..300. 2461X RNZ

057.355 175 2462X MOV A,L  
057.356 273 2463X CMP E 'C' SET => (L) < (E)  
057.357 311 2464X RET  
057.360 2465 XTEXT CDEHL

2467X \*\* \$CDEHL - COMPARE (DE) TO (HL)  
2468X \*  
2469X \* \$CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.  
2470X \*  
2471X \* ENTRY NONE  
2472X \* EXIT 'Z' SET IF (DE) = (HL)  
2473X \* USES A,F  
2474X  
2475X  
030.216 2476X \$CDEHL EQU 30216A IN H17 ROM  
057.360 2477 XTEXT MCU

2479X \*\* MCU - MAP LOWER CASE TO UPPER CASE.  
2480X \*  
2481X \* MCU MAPS A LOWER CASE ALPHABETIC TO UPPER  
CASE.  
2482X \*  
2483X \*  
2484X \* ENTRY (A) = CHARACTER  
2485X \* EXIT (A) = CHARACTER RESULT  
2486X \* USES A,F  
2487X  
2488X  
057.360 376 141 2489X \$MCU CPI 'a'  
057.362 330 2490X RC NOT LOWER CASE  
057.363 376 173 2491X CPI 'z'+1  
057.365 320 2492X RNC NOT LOWER CASE  
057.366 326 040 2493X SUI 'a'-'A'  
057.370 311 2494X RET  
057.371 2495 XTEXT MLU

2497X \*\* MLU - MAP LOWER CASE LINE TO UPPER CASE.  
2498X \*  
2499X \* MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.  
2500X \*  
2501X \* ENTRY (HL) = LINE FWA  
2502X \* EXIT NONE  
2503X \* USES NONE  
2504X  
2505X  
057.371 365 2506X \$MLU PUSH PSW SAVE (PSW)  
057.372 345 2507X PUSH H SAVE FWA  
057.373 053 2508X DCX H ANTICIPATE INX H

057.374 043	2509X \$MLU1	INX	H	
057.375 176	2510X	MOV	A,M	(A) = CHARACTER
057.376 315 360 057	2511X	CALL	\$MCU	MAP CHAR TO UPPER
060.001 167	2512X	MOV	M,A	
060.002 247	2513X	ANA	A	
060.003 302 374 057	2514X	JNZ	\$MLU1	MORE TO GO
060.006 341	2515X	POP	H	RESTORE (HL)
060.007 361	2516X	POP	PSW	RESTORE (PSW)
060.010 311	2517X	RET		
060.011	2518	XTEXT	DTB	

2520X \*\* \$DTB - DELETE TRAILING BLANKS.  
2521X \*  
2522X \* \$DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.  
2523X \*  
2524X \* ENTRY (HL) = LINE FWA  
2525X \* EXIT (A) = LENGTH OF RESULT (INCLUDING 00 TERMINATOR BYTE)  
2526X \* USES A,F  
2527X  
2528X

060.011 325	2529X \$DTB	PUSH	D	SAVE (DE)
060.012 124	2530X	MOV	D,H	
060.013 135	2531X	MOV	E,L	(DE) = FWA
060.014 033	2532X	DCX	D	(DE) = FWA-1
060.015 176	2533X \$DTB1	MOV	A,M	
060.016 043	2534X	INX	H	
060.017 247	2535X	ANA	A	FIND END OF LINE
060.020 302 015 060	2536X	JNZ	\$DTB1	
060.023 053	2537X	DCX	H	(HL) = ADDRESS OF TERMINATING ZERO BYTE
	2538X			

2539X \* GOT END OF LINE. DELETE TRAILING BLANKS.

2540X  
060.024 053 2541X \$DTB2 DCX H BACKUP ONE CHARACTER  
060.025 315 216 030 2542X CALL \$CDEHL  
060.030 312 041 060 2543X JE \$DTB3 GONE PAST FRONT OF LINE. MUST BE ALL BLANKS.  
060.033 176 2544X MOV A,M  
060.034 376 040 2545X CPI ''  
060.036 312 024 060 2546X JE \$DTB2 GOT BLANK

2547X  
2548X \* HAVE TRIMED LINE. COMPUTE LENGTH

2549X				
060.041 043	2550X \$DTB3	INX	H	
060.042 066 000	2551X	MVI	M,0	TERMINATE LINE
060.044 175	2552X	MOV	A,L	
060.045 223	2553X	SUB	E	(A) = LENGTH +1 (FOR 00 BYTE)
060.046 353	2554X	XCHG		
060.047 043	2555X	INX	H	(HL) = LINE FWA
060.050 321	2556X	POP	D	RESTORE (DE)
060.051 311	2557X	RET		
060.052	2558	XTEXT	MOVEI	

2560X \*\* \$MOVEL - MOVE DATA  
2561X \*  
2562X \* \$MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.  
2563X \* IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM  
2564X \* FIRST TO LAST.  
2565X \*  
2566X \* IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM  
2567X \* LAST TO FIRST.  
2568X \*

2569X \* THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.

2570X \*

2571X \* CALL \$MOVEL

2572X \* DW COUNT

2573X \* DW FROM

2574X \* DW TO

2575X \*

2576X \* ENTRY ((SP)) = RET

2577X \* (RET+0) = COUNT (WORD VALUE)

2578X \* (RET+2) = FROM

2579X \* (RET+4) = TO

2580X \* EXIT TO (RET+6)

2581X \* (IE) = ADDRESS OF NEXT FROM BYTE

2582X \* (HL) = ADDRESS OF NEXT \*TO\* BYTE

2583X \* 'C' CLEAR

2584X \* USES ALL

2585X \*

2586X \*

060.052 .341 2587X \$MOVEL POP H (HL) = RET

060.053 116 2588X MOV C,M

060.054 .043 2589X INX H

060.055 106 2590X MOV B,M (BC) = COUNT

060.056 .043 2591X INX H

060.057 136 2592X MOV E,M

060.060 .043 2593X INX H

060.061 126 2594X MOV D,M (DE) = FROM

060.062 .043 2595X INX H

060.063 325 2596X PUSH D ((SP)) = FROM

060.064 .134 2597X MOV E,M

060.065 .043 2598X INX H

060.066 .126 2599X MOV D,M (DE) = TO

060.067 .043 2600X INX H

060.070 .343 2601X XTHL ((SP)) = RET, (HL) = FROM

060.071 353 2602X XCHG (DE) = FROM , (HL) = TO

060.072 .303.252.030 2603X JMP \$MOVE MOVE IT

060.075 2604 XTEXT RCHAR

2606X \*\* \$RCHAR - READ SINGLE CHARACTER FROM CONSOLE.

2607X \*

2608X \* ENTRY NONE

2609X \* EXIT (A) = CHARACTER

2610X \* USES A,F

2611X \*

2612X \*

BOOT COMMON DECKS

\$RCHAR 14:02:38 16-MAY-80

```

060.075 377 001 2613X $RCHAR DB      SYSCALL,.SCIN
060.077 332 075 060 2614X   JC      $RCHAR      NOT READY
060.102 311 2615X   RET
2616X
060.103 377 002 2617X $WCHAR DB      SYSCALL,.SCOUT
060.105 311 2618X   RET
060.106 2619    XTEXT MU10

```

2621X \*\* \$MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.

```

2622X *
2623X * (HL) = (DE)*10
2624X *
2625X * ENTRY (DE) = MULTIPLIER
2626X * EXIT 'C' CLEAR IF OK
2627X * (HL) = PRODUCT
2628X * 'C' SET IF ERROR
2629X * USES D,E,H,L,F
2630X
2631X
030.324 2632X $MU10 EQU 30324A IN H17 ROM
060.106 2633 XTEXT RTL

```

2635X \*\* \$RTL - READ TEXT LINE.

```

2636X *
2637X * $RTL READS A LINE FROM THE TERMINAL.
2638X *
2639X * CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
2640X * CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
2641X * $RTL RETURNS.
2642X *
2643X * ENTRY (HL) = BUFFER FWA
2644X * EXIT 'C' CLEAR IF OK
2645X * DATA IN BUFFER
2646X * (A)=,TEXT LENGTH
2647X * 'C' SET IF CTL-D STRUCK
2648X * USES A,F
2649X
2650X
060.106 315 115 060 2651X $RTL CALL $RTL $RTL IN UPPER CASE
060.111 330 2652X RC CTL-D
060.112 303 371 057 2653X JMP $MLU MAP LINE TO UPPER CASE
2654X
060.115 2655X $RTL EQU *
060.115 345 2656X PUSH H SAVE FWA
060.116 315 075 060 2657X $RTL1 CALL $RCHAR
060.121 376 004 2658X CPI CTL-D
060.123 312 150 060 2659X JE $RTL2 CTL-D STRUCK
060.126 167 2660X MOV M,A
060.127 043 2661X INX H
060.130 376 012 2662X CPI NL

```

\$RTL 14:02:50 16-MAY-80

060.132 302 116 060 2663X JNE \$RTL1  
060.135 053 2664X DCX H  
060.136 066 000 2665X MVI M,0  
060.140 043 2666X INX H  
2667X  
2668X \* ALL DONE. COMPUTE LENGTH  
2669X  
060.141 353 2670X XCHG (DE) = LWA+1  
060.142 343 2671X XTHL (HL) = FWA  
060.143 173 2672X MOV A,E  
060.144 225 2673X SUB L (A) = LENGTH  
060.145 247 2674X ANA A CLEAR CARRY  
060.146 321 2675X POP D RESTORE (DE)  
060.147 311 2676X RET  
2677X  
2678X \* CTL-D STRUCK  
2679X  
060.150 341 2680X \$RTL2 POP H (HL) = FWA  
060.151 067 2681X STC  
060.152 311 2682X RET  
060.153 2683 XTEXT CAD

2685X \*\* \$CAD - CODE AUGUSTAN DATE.  
2686X \*  
2687X \* \$CAD IS CALLED TO CODE AN AUGUSTAN DATE INTO THE FORM:  
2688X \*  
2689X \*  
2690X \*  
2691X \* I. O. I. 6 BITS.. I. 4 BITS.. I. 5 BITS.. I.  
2692X \*  
2693X \* YEAR-70 MON DAY  
2694X \* 1-63 1-12 1-31  
2695X \*  
2696X \* FROM THE FORM:  
2697X \*  
2698X \* DD-MMM-YY  
2699X \*  
2700X \* ENTRY (HL) = ADDRESS OF STRING  
2701X \* EXIT 'C' CLEAR IF OK  
2702X \* (DE) = 15 BIT VALUE  
2703X \* (HL) ADVANCED PAST '-'=YY  
2704X \* 'C' SET IF ERROR  
2705X \* USES ALL  
2706X  
2707X  
060.153 315 150 061 2708X \$CAD CALL \$000 DECODE DECIMAL DIGITS  
060.156 330 2709X RC ERROR  
060.157 172 2710X MOV A,D  
060.160 247 2711X ANA A  
060.161 067 2712X STC ASSUME TOO LARGE  
060.162 300 2713X RNZ TOO LARGE  
060.163 173 2714X MOV A,E  
060.164 247 2715X ANA A

060.165 067 2716X STC  
060.166 310 2717X RZ TOO SMALL FOR DD  
060.167 376 040 2718X CPI 32  
060.171 077 2719X CMC  
060.172 330 2720X RC TOO LARGE  
060.173 353 2721X XCHG (HL) = DAY  
060.174 076 040 2722X MVI A,100000B  
060.176 205 2723X ADD L  
060.177 157 2724X MOV L,A COUNT 1ST MONTH  
060.200 353 2725X XCHG (DE) = DD\*16+1, (HL) = ADDRESS  
2726X  
2727X \* DECODE MONTH  
2728X  
060.201 325 2729X PUSH D SAVE DD\*16+1  
060.202 176 2730X MOV A,M  
060.203 043 2731X INX H  
060.204 376 055 2732X CPI '-'  
060.206 302 250 060 2733X JNE CAD2 FORMAT ERROR  
060.211 021 316 060 2734X LXI D,CADA (DE) = MONTH TABLE ADDRESS  
060.214 001 003 000 2735X CAD1 LXI B,3  
060.217 345 2736X PUSH H SAVE TEXT ADDRESS, CADA ADDRESS  
060.220 325 2737X PUSH D  
060.221 315 060 030 2738X CALL \$COMP COMPARE  
060.224 321 2739X POP D (DE) = \*CADA ADDRESS  
060.225 312 253 060 2740X JE CAD3 GOT MONTH  
060.230 341 2741X POP H (HL) = BUFFER ADDRESS OF MMM-YY  
060.231 023 2742X INX D  
060.232 023 2743X INX D  
060.233 023 2744X INX D TRY NEXT MONTH  
060.234 343 2745X XTHL  
060.235 076 040 2746X MVI A,100000B  
060.237 315 101 030 2747X CALL \$DADA COUNT MONTH  
060.242 343 2748X XTHL  
060.243 032 2749X LDAX D (A) = ENTRY IN CADA  
060.244 247 2750X ANA A  
060.245 302 214 060 2751X JNZ CAD1 MORE MONTHS TO GO  
2752X  
2753X \* ERROR  
2754X  
060.250 341 2755X CAD2 POP H CLEAR STACK  
060.251 067 2756X STC  
060.252 311 2757X RET FLAG ERROR  
2758X  
2759X \* CRACK -YY  
2760X  
060.253 301 2761X CAD3 POP B DISCARD ADDRESS IF MMM-YY  
060.254 176 2762X MOV A,M  
060.255 376 055 2763X CPI '-'  
060.257 302 250 060 2764X JNE CAD2 NOT -  
060.262 043 2765X INX H  
060.263 315 150 061 2766X CALL \$000 DECODE DECIMAL DIGITS  
060.266 332 250 060 2767X JC CAD2 IF ERROR  
060.271 172 2768X MOV A,D  
060.272 247 2769X ANA A  
060.273 302 250 060 2770X JNZ CAD2 ERROR  
060.276 173 2771X MOV A,E (A) = YEAR

060.277 326 106 2772X SUI 70 SUBTRACT DISPLACEMENT  
060.301 332 250 060 2773X JC CAD2  
060.304 376 077 2774X CPI 63  
060.306 322 250 060 2775X JNC CAD2  
060.311 321 2776X POP D TOO LARGE  
(DE) = MONTH AND DAY  
060.312 207 2777X ADD A  
(A) = YEAR\*2  
060.313 202 2778X ADD D  
060.314 127 2779X MOV D,A MERGE WITH REST OF IT  
060.315 311 2780X RET  
2781X  
060.316 2782X CAIA DS 0 TABLE OF MONTHS  
060.316 112 101 116 2783X DB 'JANFEBMARAPRMMAYJUNJULAUGSEPOCTNOVDEC',0  
060.363 2784 XTEXT DAD

2786X \*\* \$DAD - DECODE AUGUSTAN DATE.  
2787X \*  
2788X \* \$DAD DECODES A 15 BIT DATE CODE OF THE FORMAT:  
2789X \*  
2790X \*  
2791X \* I 0 I 6 BITS I 4 BITS I 5 BITS I  
2792X \*  
2793X \* YEAR-70 MON DAY  
2794X \* 1-63 1-12 1-31  
2795X \*  
2796X \* TO THE FORM:  
2797X \*  
2798X \* DD-MMM-YY  
2799X \*  
2800X \* ENTRY (DE) = 15 BIT VALUE  
2801X \* (HL) = ADDRESS FOR DECODE  
2802X \* EXIT 'C' CLEAR IF OK  
2803X \* (DE) = (DE)+9  
2804X \* 'C' SET IF ERROR  
2805X \* USES ALL  
2806X  
2807X  
060.363 102 2808X \$DAD MOV B,D  
060.364 113 2809X MOV C,E  
060.365 021 040.000 2810X LXI D,32  
060.370 345 2811X PUSH H SAVE ADDRESS  
060.371 315.196.030 2812X CALL \$DU66 (DE) = DAY, (HL) = YEAR & MONTH  
060.374 343 2813X XTHL (HL) = ADDRESS  
060.375 102 2814X MOV B,D  
060.376 113 2815X MOV C,E  
060.377 173 2816X MOV A,E  
061.000 247 2817X ANA A  
061.001 312.101.061 2818X JZ DAD1 BAD VALUE  
061.004 076 002 2819X MVI A,2  
061.006 315.157.031 2820X CALL \$UDI UNPACK DAY  
061.011 066 055 2821X MVI M,'-'  
061.013 043 2822X INX H  
061.014 301 2823X POP B (BC) = YEAR & MONTH  
061.015 021 020.000 2824X LXI D,16

061.020 345 2825X PUSH H SAVE ADDRESS  
061.021 315.106.030 2826X CALL \$DU66  
061.024 343 2827X XTHL (HL) = ADDRESS, ((SP)) = YEAR  
061.025 173 2828X MOV A,E  
061.026 207 2829X ADD A  
061.027 203 2830X ADD E (A) = 3\*MONTH  
061.030 312 101 061 2831X JZ DAD1 BAD VALUE  
061.033 376.047 2832X CPI 13\*3  
061.035 322 101 061 2833X JNC DAD1 TOO LARGE  
061.040 353 2834X XCHG (DE) = ADDRESS  
061.041 041 101 061 2835X LXI H,DADB-3  
061.044 315.101.030 2836X CALL \$DADA, (HL) = ADDRESS OF MONTH  
061.047 001 003 000 2837X LXI B,3  
061.052 353 2838X XCHG (HL) = BUFFER ADDR., (DE) = ADDR IN 'DADB'  
061.053 315 252 030 2839X CALL \$MOVE MOVE MONTH IN  
061.056 066.055 2840X MVI M,'-'  
061.060 043 2841X INX H  
061.061 301 2842X POP B (BC) = YEAR  
061.062 171 2843X MOV A,C  
061.063 306.106 2844X ADI 70  
061.065 376 144 2845X CPI 100  
061.067 077 2846X CMC  
061.070 330 2847X RC TOO LARGE  
061.071 117 2848X MOV C,A (BC) = YEAR  
061.072 076 002 2849X MVI A,2  
061.074 315.157.031 2850X CALL \$UDD UNPACK YEAR  
061.077 247 2851X ANA A  
061.100 311 2852X RET  
2853X  
2854X \* ILLEGAL FORMAT,..(NOT ALL ILLEGALS EXIT HERE!)  
2855X  
061.101 341 2856X DAD1 POP H RESTORE STACK  
061.102 067 2857X STC FLAG ERROR  
061.103 311 2858X RET  
2859X  
061.104 112.141.156 2860X DADR DB 'JanFebMarAprMayJunJulAusSepOctNovDec'  
061.150 2861 XTEXT DU66

2863X \*\* \$DU66 - UNSIGNED 16 / 16 DIVIDE.  
2864X \*  
2865X \* (HL) = (BC)/(DE)  
2866X \*  
2867X \* ENTRY (BC), (DE) PRESET  
2868X \* EXIT (HL) = RESULT  
2869X \* (DE) = REMAINDER  
2870X \* USES ALL  
2871X  
2872X  
030.106 2873X \$DU66 EQU 30106A IN.H17.ROM  
061.150 2874 XTEXT DBB

\$000

14:03:17 16-MAY-80

2876X \*\* \$000 - DECODE DECIMAL DIGITS.  
2877X \*  
2878X \* \$000 DECODES A STRING OF DECIMAL DIGITS INTO A DECIMAL INTEGER.  
2879X \*  
2880X \* THE CHARACTERS ARE TAKEN OUT OF MEMORY. CONVERSION STOPS WITH THE  
2881X \* FIRST NON-DIGIT CHARACTER FOUND.  
2882X \*  
2883X \* ENTRY (HL) = ADDRESS OF CHARACTERS  
2884X \* EXIT 'C' CLEAR IF OK  
2885X \* (DE) = NUMBER  
2886X \* (HL) = INDEX OF FIRST NON-DIGIT ENCOUNTERED  
2887X \* 'C' SET IF ERROR  
2888X \* USES A,F,D,E,H,L

2889X

2890X

061.150	021 000 000	2891X	\$000	LXI	D,0	(DE) = ACCUM
		2892X				
061.153	176	2893X	\$0001	MOV	A,M	
061.154	326 060	2894X		SUI	'0'	
061.156	077	2895X		CMC		
061.157	320	2896X		RNC		TOO SMALL
061.160	376 012	2897X		CPI	10	
061.162	320	2898X		RNC		TOO LARGE
061.163	043	2899X		INX	H	ADVANCE ADDRESS
061.164	345	2900X		PUSH	H	SAVE (HL)
061.165	315 324 030	2901X		CALL	\$MU10	(HL) = ACCUM*10
061.170	353	2902X		XCHG		(DE) = ACCUM
061.171	341	2903X		POP	H	(HL) = ADDRESS OF STRING
061.172	330	2904X		RC		OVERFLOW
061.173	203	2905X		ADD	E	
061.174	137	2906X		MOV	E,A	
061.175	076 000	2907X		MVI	A,0	
061.177	212	2908X		ADC	D	
061.200	127	2909X		MOV	D,A	
061.201	322 153 061	2910X		JNC	\$0001	NOT OVERFLOW
061.204	311	2911X		RET		
061.205		2912		XTEXT	UDD	

2914X \*\* \$UDD - UNPACK DECIMAL DIGITS.  
2915X \*  
2916X \* UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF  
2917X \* DECIMAL DIGITS. THE RESULT IS ZERO FILLED.

2918X \*  
2919X \* ENTRY (B,C) = ADDRESS VALUE  
2920X \* (A) = DIGIT COUNT  
2921X \* (H,L) = MEMORY ADDRESS  
2922X \* EXIT (HL) = (HL) + (A)  
2923X \* USES ALL

2924X

2925X

031.157	2926X	\$UDD	EQU	31157A	IN H17 ROM
061.205	2927		XTEXT	DADA	

2929X \*\* \$DATA - PERFORM (H,L) = (H,L) + (0,A)

2930X \* ENTRY (H,L) = BEFORE VALUE

2932X \* (A) = BEFORE VALUE

2933X \* EXIT (H,L) = (H,L) + (0,A)

2934X \* 'C' SET IF OVERFLOW

2935X \* USES F,H,L

2936X

2937X

030.072 2938X \$DATA EQU 30072A IN H17.ROM  
061.205 2939 XTEXT CRLF

2941X \*\* \$CRLF - TYPE CARRIAGE RETURN/ LINE FEED

2942X \*

2943X \* \$CRLF IS USED TO GENERATE PADDED CRLF'S.

2944X \*

2945X \* ENTRY NONE

2946X \* EXIT (A) = 0

2947X \* USES A,F

2948X

2949X

061.205 076 012 2950X \$CRLF MVI A,NL  
061.207 377.002 2951X DB SYSCALL:,SCOUT  
061.211 257 2952X XRA A  
061.212 311 2953X RET  
061.213 2954 XTEXT TYPT2 TYPTX

2956X \*\* \$TYPTX - TYPE TEXT.

2957X \*

2958X \* \$TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.

2959X \*

2960X \* IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.

2961X \* A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.

2962X \*

2963X \* ENTRY (RET) = TEXT

2964X \* EXIT TO (RET+LENGTH)

2965X \* USES A,F

2966X

2967X

031.136 2968X \$TYPTX EQU 31136A IN H17.ROM  
031.144 2969X  
061.213 2970X \$TYPTX EQU 31144A IN H17.ROM  
2971 XTEXT TYPCH

2973X \*\* \$TYPCH - TYPE SINGLE CHARACTER.

2974X \*  
2975X \* ENTRY (RET) = CHARACTER

2976X \* EXIT TO (RET)+1

2977X \* (A) = CHARACTER TYPED

2978X

2979X

061.213 343 2980X \$TYPCH XTHL (HL) = RETURN ADDRESS

061.214 176 2981X MOV A,M (A) = CHARACTER

061.215 043 2982X INX H

061.216 343 2983X XTHL RESTORE ADVANCED EXIT ADDRESS

2984X

2985X \*\* \$TYPC. - TYPE SINGLE CHARACTER.

2986X \*

2987X \* ENTRY (A) = CHARACTER

2988X \* EXIT TO (RET)

2989X

061.217 377 002 2990X \$TYPC. DB SYSCALL,.SCOUT

061.221 311 2991X RET

061.222 2992 XTEXT TFN

2994X \*\* \$TFN - TYPE FILE NAME.

2995X \*

2996X \* \$TFN TYPES THE FILE WHOSE NAME APPEARS IN AIO.XXX

2997X \*

2998X \* ENTRY NONE

2999X \* EXIT NONE

3000X \* USES A,F,B,H,L

3001X

3002X

061.222 .041 062 .041 3003X \$TFN LXI H,AIO,DIR+DIR,NAM

061.225 006 010 3004X MVI B,8

061.227 315 240 061 3005X CALL \$TFN1 TYPE NAME

061.232 315 213 061 3006X CALL \$TYPCH

061.235 .054 3007X DB ,:,

061.236 006 003 3008X MVI B,3

3009X

061.240 176 3010X \$TFN1 MOV A,M

061.241 247 3011X ANA A

061.242 304 217 061 3012X CNZ \$TYPC.

061.245 .043 3013X INX H

061.246 005 3014X DCR B

061.247 302 240 061 3015X JNZ \$TFN1

061.252 311 3016X RET

061.253 3017 XTEXT TYPET

BOOT.COMMON.DECKS

\$TYPET

14:03:43 16-MAY-80

3019X \*\* \$TYPET - TYPE TEXT.  
 3020X \*  
 3021X \* \$TYPET IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE  
 3022X \* AT TASK TIME RATHER THAN AT INTERRUPT TIME.  
 3023X \*  
 3024X \* IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,  
 3025X \* A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE OF THE MESSAGE.  
 3026X \*  
 3027X \* This routine modified to accomodate H8-4 Ports by G.Chandler, 1-SEP-78.  
 3028X \* This routine assumes that the ports have been previously initialized,  
 3029X \* and that S.CDB has been previously initialized.  
 3030X \*  
 3031X \* ENTRY (RET) = TEXT  
 3032X \* EXIT TO (RET+LENGTH)  
 3033X \* USES A,F  
 3034X  
 3035X  
 061.253 343 3036X \$TYPET XTHL (HL) = TEXT ADDRESS  
 061.254 315 261 061 3037X CALL \$TYPET TYPE IT  
 061.257 343 3038X XTHL  
 061.260 311 3039X RET  
 3040X  
 061.261 176 3041X \$TYPET. MOV A,M  
 061.262 346 177 3042X ANI 1770  
 061.264 304 310 061 3043X CNZ \$TYPEC. IF NOT CRLF  
 061.267 247 3044X ANA A  
 061.270 314 301 061 3045X CZ \$TYPET1 IS CRLF ~  
 061.273 276 3046X CMP M  
 061.274 043 3047X INX H  
 061.275 300 3048X RNE WAS 200 BIT SET  
 061.276 303 261 061 3049X JMP \$TYPET.  
 3050X  
 3051X \* TYPE CRLF  
 3052X  
 061.301 315 253 061 3053X \$TYPET1 CALL \$TYPET  
 061.304 015 212 3054X DB CR,LF+2000  
 061.306 257 3055X XRA A RESTORE (A)  
 061.307 311 3056X RET

3058X \*\* \$TYPEC. - TYPE SINGLE CHARACTER.  
 3059X \*  
 3060X \* IF CR, PADD WITH 4 ZERO BYTES  
 3061X \*  
 3062X \* ENTRY (A) = CHARACTER  
 3063X \* EXIT (A) = CHARACTER  
 3064X \* USES A,F  
 3065X  
 3066X  
 061.310 365 3067X \$TYPEC. PUSH PSW SAVE CHAR  
 061.311 072 343 040 3068X LDA S.CDB  
 061.314 376 001 3069X CPI CDB,H84  
 061.316 312 336 061 3070X JZ TYPEC2 IF H8-4 PORT  
 3071X

\$TYPEC.

14:03:48 16-MAY-80

3072X \* HAVE 8251 PORT FOR CONSOLE

3073X

061.321 333 373	3074X	TYPEC1	IN	SC.UART+USR
061.323 346 001	3075X	ANI	USR.TXR	
061.325 312 321 061	3076X	JZ	TYPEC1	NOT READY
061.330 361	3077X	POP	PSW	
061.331 323 372	3078X	OUT	SC.UART+UDR	
061.333 303 350 061	3079X	JMP	TYPEC3	
	3080X			

3081X \* HAVE 8250 PORT FOR CONSOLE

3082X

061.336 333 355	3083X	TYPEC2	IN	SC.ACE+UR.LSR
061.340 346 040	3084X	ANI	UC.THE	
061.342 312.336.061	3085X	JZ	TYPEC2	NOT READY
061.345 361	3086X	POP	PSW	
061.346 323 350	3087X	OUT	SC.ACE+UR.THR	
	3088X			

061.350 376 015 3089X TYPEC3 CPI CR

061.352 300 3090X RNE NOT CR

3091X

3092X \* IS CR. PADD 4 TIMES

3093X

061.353 076 004	3094X	MVI	A,4	
061.355 365	3095X	TYPEC4	PUSH	PSW
061.356 257	3096X	XRA	A	
061.357 315.310.061	3097X	CALL	\$TYPEC,	
061.362 361	3098X	POP	PSW	
061.363 .075	3099X	ICR	A	
061.364 302 355 061	3100X	JNZ	TYPEC4	
061.367 .076.015	3101X	MVI	A,CR	
061.371 311	3102X	RET		
061.372	3103	XTEXT	..MUB6	

3105X \*\* \$MUB6 - MULTIPLY BX16 UNSIGNED.

3106X \*

3107X \* \$MUB6 MULTIPLIES A 16 BIT VALUE BY A 8

3108X \* BIT.VALUE.

3109X \*

3110X \* ENTRY ..(A)..= MULTIPLIER

3111X \* ..(DE).. = MULTIPLICAND

3112X \* EXIT ..(HL)..= RESULT

3113X \* 'Z' SET IF NOT OVERFLOW

3114X \* USES ..A,x,F,x,H,x,L

3115X

3116X

031.007 3117X \$MUB6 EQU 31007A IN H17 ROM

061.372 3118 XTEXT..TYPEC.

3120X \*\* \$TYPCC - TYPE A CHARACTER STRING BY COUNT.  
3121X \*  
3122X \* \$TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES  
3123X \* THE CHARACTER ADDRESS AND COUNT.  
3124X \*  
3125X \* ENTRY (HL) = ADDRESS  
3126X \* (A) = COUNT  
3127X \* EXIT (HL) = LAST CHARACTER ADDRESS+1  
3128X \* USES A,F,H,L  
3129X  
3130X

061.372 3131X \$TYPCC EQU \*  
061.372 247 3132X ANA A  
061.373 310 3133X RZ NOTHING\_TO\_TYPE  
061.374 365 3134X PUSH PSW SAVE COUNT  
061.375 176 3135X MOV A,X,M (A)=CHARACTER  
061.376 043 3136X INX H  
061.377 377.002 3137X DB SYSCALL,,SCOUT  
062.001 361 3138X POP PSW  
062.002 .075 3139X DCR A  
062.003 303 372 061 3140X JMP \$TYPCC

3142. \*\* PATCH.=PATCH AREA

3143  
062.006 3144 PATCH DS 0  
062.006 014 012 101 3145 DB FF,NL,'ANOTHER FINE HEATH SOFTWARE PRODUCT'  
3146  
000.033 3147 ERRMI 64-\*+PATCH /79.06.sc/  
062.053 3148 DS 64-\*+PATCH /79.06.sc/

3150 \*\*\*\*=  
3151 \*\*\*\*=  
3152 \*\*

3153 \*\* BE VERY CAREFUL ABOUT THE PLACEMENT OF THESE BUFFERS, AND NOTE \*\*

3154 \*\* THAT THE \*LABEL\* BUFFER OVERLAYS CODE. (MAKE SURE THAT THE \*\*

3155 \*\* CODE WHICH IS OVERLAID IS NO LONGER NEEDED AT OVERLAY TIME.) \*\*

G. Chandler \*\*

3157 \*\* 79.11.sc. \*\*

3158 \*\*

3159 \*\*\*\*=  
3160 \*\*\*\*=  
3161

025.000 3162 PGTA EQU S.GRT1 256 BYTE BUFFER /79.11.GC/

3163  
025.000 3164 BUFF EQU S.GRT1 256 BYTE BUFFER /79.11.GC/

3165  
026.000 3166 SDTA EQU S.GRT2 512 BYTE BUFFER /79.11.GC/

047.211 3168 LABEL EQU RRH 256 BYTE BUFFER /79.11.GC/

3169

047.211 3170 SSDB EQU RRH DATE BUFFER /79.12.GC/  
062.106 3171 OVBUFE EQU \* END OF OVERLAID BUFFERS  
3172  
3173 \*\* WE MUST MAKE SURE THAT THERE IS ENOUGH MEMORY IN 8K SO THAT  
3174 \* THE RESIDENT CODE WILL BE MOVED COMPLETELY ABOVE 'OVBUFE'  
3175  
000.265 3176 ERRMI 100000A-LENSYS-OVBUFE-20 NOT ENOUGH ROOM FOR EVERYBODY

3180 \*\*\* SYSCALL DISPATCH.  
3181 \*  
3182 \* THE SYSCALL DISPATCH HANDLER IS ENTERED VIA A SYSCALL INSTRUCTION.  
3183 \*  
3184 \* IF THE PROCESSOR IS IN RESIDENT CODE, IT IS CALLED.  
3185 \*  
3186 \* ALL CALLS WHICH INVOKE THE OVERLAY CODE HAVE THEIR STACK POINTER  
3187 \* VALUE SAVED. THIS IS A KLUDGE FOR STACK PRESERVATION VIA 'LINK'.  
3188 \*  
3189 \* IF THE REQUIRED OVERLAY IS RESIDENT, IT IS CALLED.  
3190 \*  
3191 \* IF THE OVERLAY IS NOT RESIDENT, LOAD IT, RELOCATE IT, AND CALL IT.  
3192 \*  
3193 \* ENTRY (SP) = RET  
3194 \* (RET) = SYSCALL INDEX  
3195 \* EXIT 'C' SET IF ILLEGAL CODE  
3196 \* (A) = EC.ILC  
3197 \* TO PROCESSOR IF A GOOD LOAD  
3198 \* (SP) = PSW  
3199 \* (SP+2) = RETURN ADDRESS (ADVANCED PAST CODE).  
3200 \* USES A,F  
3201  
3202  
062.106 3203 FWAREL EQU \* ABS ADDRESS TO START RELOCATION  
3204 CODE +R REMAINING CELLS ARE RELOCATED  
062.106 3205 FWASYS EQU \* SYSTEM.FWA  
3206  
062.106 3207 SYSCAL EQU \*  
062.106 062 006 041 3208 STA S.CACC SAVE (A)  
062.111 343 3209 XTHL  
062.112 176 3210 MOV A,M (A) = CODE  
062.113 062.007.041 3211 STA S.CODE SET SYSTEM CODE  
062.116 043 3212 INX H ADVANCE RETURN ADDRESS  
062.117 343 3213 XTHL  
000.001 3214 IF DEBUG  
3215 CPI .READ  
3216 JC SYSCO IS CONSOLE FUNCTION  
3217 PUSH PSW \*.\*.DEBUG.\*.  
3218 CALL JGL CLEAR ACTIVE CHANNEL  
3219 POP PSW \*.\*.DEBUG.\*.  
3220 ENDIF  
062.120 345 3221 PUSH H.  
062.121 041 371 040 3222 LXI H,S.OVLFLL STORE S.OVLFLL ON STACK WITHOUT  
062.124 146 3223 MOV H,M DAMAGING REGISTERS.  
062.125 343 3224 XTHL  
062.126 315.222.062 3225 SYSCO CALL SYSCALO CAUSE ALL SYSCALLS TO RETURN HERE.  
3226  
3227 \* ALL SYSCALLS RETURN HERE.  
3228 \*  
3229 \* LOAD ANY POSTPONED DEVICE DRIVERS, AND SEEZ IF A CTL-C OR CTL-Z  
3230 \* WAS STRUCK.  
3231  
062.131 365 3232 PUSH PSW  
062.132 072.244.074 3233 LDA SYSMODE  
062.135 075 3234 DCR A  
062.134 302.150.062 3235 JNZ SYSC1 DONT RESTORE USER IF NOT FIRST LEVEL CALL

```

062.141 072 371 040 3236 LDA S.OVLFL
000.000 3237 ERRNZ OVL.UCS-2000
062.144 247 3238 ANA A
062.145 374 257 033 3239 CM RUC RESTORE USER CODE, IF SWAPPED
3240
062.150 361 3241 SYSC1 POP PSW
062.151 343 3242 XTHL (H) = OLD S.OVLFL
062.152 365 3243 PUSH PSW
062.153 174 3244 MOV A,H
062.154 346 002 3245 ANI OVL.RES
062.156 312 174 062 3246 J2 SYSC2 WAS NOT PERMANENTLY RESIDENT BEFORE
062.161 174 3247 MOV A,H
062.162 346 014 3248 ANI OVL.NUM
062.164 017 3249 RRC
062.165 017 3250 RRC
000.000 3251 ERRNZ OVL.NUM-00001100B
062.166 315 360 062 3252 CALL LDON WAS PERM. RESIDENT BEFORE
062.171 334 210 063 3253 CC FATSER WAS OVERLAY TOO BIG
3254
062.174 072 361 040 3255 SYSC2 LDA S.DLDIDA+1
062.177 247 3256 ANA A
062.200 304 001 071 3257 CNZ LDD LOAD DEVICE DRIVER IF PENDING
062.203 041 244 074 3258 LXI H,SYSMODE
062.206 363 3259 DI LOCK OUT CONSOLE INTERRUPTS UNTIL *CPA*
062.207 065 3260 DCR M DECREMENT NESTED SYSCALL COUNT
062.210 314 222 031 3261 CZ $WDR IF RETURNING TO USER, WRITE DISABLE ROM
062.213 315 232 070 3262 CALL CPA CHECK PENDING ABORT
062.216 361 3263 POP PSW
062.217 341 3264 POP H RESTORE USER (HL)
062.220 373 3265 EI
062.221 311 3266 RET EXIT
3267
3268
000.001 3269 IF DEBUG
3270 JGL PUSH B
3271 PUSH H
3272 LXI H,AIO.IDA
3273 MVI B,AIO.CHA-AIO.DDA
3274 CALL $ZERO
3275 POP H
3276 POP B
3277 RET ** DEBUG ***
3278 ENDIF
3280
062.222 345 3280 SYSCL0 EQU *
062.222 345 3281 PUSH H SAVE (HL)
062.223 041 244 074 3282 LXI H,SYSMODE
062.226 064 3283 INR M COUNT NESTED SYSCALL
062.227 315 241 031 3284 CALL $WER WRITE ENABLE RAM AREA
062.232 376 040 3285 CPI .LINK
062.234 322 271 062 3286 JNC SYSCL2 IS IN OVERLAY
062.237 376 012 3287 CPI .SYSRES
062.241 332 251 062 3288 JC SYSCL1 IS RESIDENT
062.244 076 003 3289 MVI A,EC.ILC

```

```

062.246 067      3290    STC
062.247 341      3291    POP H
062.250 311      3292    RET
                                RESTORE (HL)
                                ERROR
062.251 041 334 062  3293
062.254 207      3294 * DISPATCH RESIDENT CALLS
062.255 315 101 030  3295
062.260 176      3296 SYSCAL1 LXI H,SYSCALA
062.264 207      3297 ADD A (A) = CODE#2
062.255 315 101 030  3298 CALL $DADA. (HL) = TABLE ADDRESS
062.260 176      3299 MOV A,M
062.261 043      3300 INX H
062.262 146      3301 MOV H,M
062.263 157      3302 MOV L,A (HL) = CODE ADDRESS
062.264 343      3303 XTHL PUT ON STACK
062.265 072 006 041  3304 LDA S,CACC (A) = (ACC) UPON CALL
062.270 311      3305 RET ENTER PROCESSOR CODE
                                3306
                                3307 * DISPATCH OVERLAID CALLS
                                3308
062.271 041 010 000  3309 SYSCAL2 LXI H,8
062.274 071      3310 DAD SP
062.275 042 035 041  3311 SHLD S,OVSTK SAVE STACK VALUE
062.300 365      3312 PUSH FSW SAVE CODE
                                3313
062.301 376 200     3314 CPI .MOUNT
062.303 322 313 062  3315 JNC SYSCAL3 SECOND OVERLAY REQUIRED
062.306 076 000     3316 MVI A,OVL0 HDOSOVL.SYS
062.310 303 315 062  3317 JMP SYSCAL4
                                3318
062.313 076 001     3319 SYSCAL3 MVI A,OVL1 HDOSOVL2.SYS
                                3320
062.315 315 360 062  3321 SYSCAL4 CALL LDON LOAD INDEXED OVERLAY
062.320 334 210 063  3322 CC FATSERR OVERLAY TOO BIG
                                3323
                                3324 * OVERLAY IS NOW LOADED
                                3325
062.323 315 152 071  3326 CALL OTI
062.326 094 090     3327 DW OVL.ENT (HL) = ADDRESS OF ENTRY POINT
062.330 315 211 030  3328 CALL $HLIHL (HL) = ENTRY POINT
062.333 351      3329 PCHL ENTER CODE
                                3330
                                3331 ** TABLE OF SYSCALL ROUTINES.
                                3332 *
                                3333 * DW ADDR ENTRY ADDRESS
                                3334
                                3335
062.334      3336 SYSCAL4 DS 0
                                3337
062.334 264 063     3338 DW EXIT RETURN TO MONITOR
                                3339
062.336 123 064     3340 DW SCIN READ FROM SYSTEM CONSOLE
                                3341
062.340 360 065     3342 DW SCOUT WRITE TO SYSTEM CONSOLE
                                3343
062.342 362 066     3344 DW PRINT WRITE LINE TO SYSTEM CONSOLE

```

062.344 276 066	3345		DW	READ	READ DATA
	3346			3347	
062.346 327 066	3348		DW	WRITE	WRITE DATA
	3349			3350	
062.350 376 066	3350		DW	CONSL	SET/READ CONSOLE OPTIONS
062.352 024 067	3351		DW	CLRCO	CLEAR CONSOLE TYPE AHEAD
	3352			3353	
062.354 052 067	3353		DW	LOADO	LOAD SPECIFIED OVERLAY
	3354			3355	
062.356 125 067	3355		DW	VERSN	

3357 \*\* LDON - LOAD OVERLAY BY NUMBER  
3358 \*  
3359 \* LOAD THE SPECIFIED OVERLAY ACCORDING TO THE NUMBER SPECIFIED.  
3360 \* THE NUMBER CORRESPONDS TO THE INDEX IN TABLE SYSCALB.  
3361 \* SET THE ENTRY POINT AND FLAG BYTE IN THE OVERLAY TABLE.  
3362 \*  
3363 \* IF THE OVERLAY IS ALREADY PRESENT, IT IS NOT LOADED.  
3364 \*  
3365 \* IF A SMALLER OVERLAY IS ALREADY LOADED, IT IS TAKEN AS  
3366 \* A FATAL SYSTEM ERROR.

3367 \*  
3368 \* \*\*\*\*\*  
3369 \* \*  
3370 \* \* OVERLAID CALLS TO OTHER OVERLAYS WILL PROBABLY NOT WORK \*  
3371 \* \*  
3372 \* \*\*\*\*\*  
3373 \*  
3374 \*  
3375 \* ENTRY: ..(A)... = INDEX OF OVERLAY TO BE LOADED.  
3376 \*  
3377 \* EXIT: ..(PSW)... = 'C' CLEAR IF NO ERROR  
3378 \* = 'C' SET IF ERROR  
3379 \* ..(A)... = ERROR CODE  
3380 \*  
3381 \* USES: ..(FLAGS)  
3382 \*  
3383 \*

062.360 305 3384 LDON PUSH B  
062.361 325 3385 PUSH B  
062.362 345 3386 PUSH H  
3387

062.363 376 002 3388 CPI OVLMAX  
062.365 322 174 063 3389 JNC LDONS INDEX IS TOO BIG  
3390

3391 \* CHECK TO SEE IF OVERLAY IS PERMANENTLY RESIDENT  
3392

062.370 365 3393 PUSH PSW SAVE OVERLAY INDEX  
062.371 315 152 071 3394 CALL OTI  
062.374 006 000 3395 DW OVL.FLB (HL) = ADDRESS OF FLAG BYTE  
062.376 176 3396 MOV A,M  
062.377 346 002 3397 ANI OVL.RES

063.001 302 036 063 3398 JNZ LDON0 OVERLAY IS PERMANENTLY RESIDENT  
063.004 361 3399 POP PSW RESTORE OVERLAY INDEX  
3400  
3401 \* CHECK TO SEE IF OVERLAY IS PRESENTLY IN MEMORY  
3402  
063.005 365 3403 PUSH PSW SAVE OVERLAY INDEX  
063.006 207 3404 ADD A  
063.007 207 3405 ADD A A = A\*4  
000.000 3406 ERRNZ OVL:NUM-000001100B  
063.010 107 3407 MOV B,A (B) = OVERLAY SOUGHT  
063.011 072 371 040 3408 LIA S.OVFL  
063.014 037 3409 RAR  
063.015 322 044 063 3410 JNC LDON2 NO OVERLAY LOADED  
000.000 3411 ERRNZ OVL.IN-1  
3412  
3413 \* CHECK TO SEE IF CURRENT OVERLAY IS THE ONE SOUGHT  
3414  
063.020 027 3415 RAL  
063.021 346 014 3416 ANI OVL.NUM  
063.023 270 3417 CMP B  
063.024 312 036 063 3418 JZ LDON0 CURRENT == SOUGHT  
063.027 052 376.040 3419 LHLD S.OVLS  
063.032 353 3420 XCHG (DE) = OLD OVERLAY SIZE  
063.033 303 047 063 3421 JMP LDON3  
3422  
063.036 361 3423 LDON0 POP PSW  
063.037 247 3424 ANA A CLEAR CARRY  
063.040 341 3425 LDON1 POP H  
063.041 321 3426 POP D  
063.042 301 3427 POP B  
063.043 311 3428 RET  
3429  
3430 \* LOAD THE NEW OVERLAY  
3431  
063.044 021 377 377 3432 LDON2 LXI D,377377A LARGE (DE) IF NO PRESENT OVERLAYS  
063.047 072 032.041 3433 LDON3 LIA S.MOUNT  
063.052 247 3434 ANA A  
063.053 .074.051 3435 MVI A,EC:NOS NO OPERATING SYSTEM  
063.055 312 202 063 3436 JZ LDON6 NO O.S.  
063.060 170 3437 MOV A,B (A) = OVERLAY INDEX \* 4  
063.061 017 3438 RRC  
063.062 017 3439 RRC (A) = OVERLAY INDEX  
063.063 315 152 071 3440 CALL OTI  
063.066 .000.000 3441 DW OVL.COD (HL) = ADDRESS OF CODE ENTRY  
063.070 345 3442 PUSH H  
063.071 315 211 030 3443 CALL \$HLIHL  
063.074 042 004 041 3444 SHLD S.OSN SET NEW OVERLAY SECTOR NUMBER  
063.077 341 3445 POP H  
063.100 043 3446 INX H  
063.101 043 3447 INX H  
000.000 3448 ERRNZ OVL.SIZ-OVL.COD-2  
063.102 315 211 030 3449 CALL \$HLIHL  
063.105 072 371 040 3450 LIA S.OVFL  
063.110 346 200 3451 ANI OVL.UCS  
063.112 312 133 063 3452 JZ LDON4 NO USER CODE SWAPPED  
063.115 315 352 057 3453 CALL HLCPDE

SYSCALL\_DISPATCH

LDON

14:04:17 16-MAY-80

063.120	312 133 063	3454	JZ	LDON4	NEW SIZE = PRESENT SIZE
063.123	332 133 063	3455	JC	LDON4	NEW SIZE < PRESENT SIZE
063.126	076 053	3456	MVI	A,EC.OTL	NEW SIZE > PRESENT SIZE
063.130	303 202 063	3457	JMP	LDON6	
		3458			
		3459	*	SET. ENTRY. POINT. AND FLAG. OVERLAY. (IN MEMORY)	
		3460			
063.133	042 376 040	3461	LION4	S.OVLS	SET NEW OVERLAY SIZE
063.136	315 012 033	3462	CALL	LIO	
063.141	072 371 040	3463	LIA	S.OVLF	
063.144	346 363	3464	ANI	3770-OVL.NUM	
063.146	260	3465	ORA	B	
063.147	062 371 040	3466	STA	S.OVLF	SET OVERLAY NUMBER IN FLAG BYTE
063.152	361	3467	POP	PSW	RESTORE OVERLAY INDEX
063.153	365	3468	PUSH	PSW	SAVE OVERLAY INDEX
063.154	315 152 071	3469	CALL	OTI	OVERLAY TABLE INDEXING
063.157	004 000	3470	IW	OVL.ENT	(HL) = ADDRESS OF THIS OVERLAY'S OVL.ENT BYTE
063.161	353	3471	XCHG		
063.162	052 000 041	3472	LHLD	S.OVLE	
063.165	353	3473	XCHG		(DE) = OVERLAY.ENTRY.ADDRESS
063.166	163	3474	MOV	M,E	
063.167	043	3475	INX	H	
063.170	162	3476	MOV	M,D	SET OVERLAY ENTRY ADDRESS IN OVERLAY TABLE
063.171	303 036 063	3477	JMP	LIONO	RETURN
		3478			
063.174	076 052	3479	LDON5	MVI	A,EC,IOI
063.176	067	3480	STC		ILLEGAL OVERLAY INDEX
063.177	303 040 063	3481	JMP	LDON1	FLAG ERROR
		3482			
063.202	063	3483	LDON6	INX	SP
063.203	063	3484	INX	SP	REMOVE OLD (PSW) FROM STACK
063.204	067	3485	STC		FLAG ERROR
063.205	303 040 063	3486	JMP	LDON1	

3488	*	FATAL SYSTEM ERROR
3489		

063.210	315 136 031	3490	FATSER CALL	\$TYPTX	
063.213	012 007 077	3491	DB	NL,BELL,'?02 FATAL SYSTEM ERROR ?',BELL,ENL	
063.247	257	3492	FATSER1 XRA	A	
063.250	062 010 040	3493	STA	.MFLAG	
063.253	323 373	3494	OUT	SC.UART+USR	CLEAR CONSOLE UART
063.255	323 351	3495	OUT	SC.ACE+UR.IER	
063.257	373	3496	EI		
063.260	166	3497	HLT		
063.261	303 247 063	3498	JMP	FATSER1	

3501 \*\*\* EXIT - EXIT USER PROGRAM.  
3502 \*  
3503 \* EXIT IS CALLED TO RETURN CONTROL TO THE SYSTEM COMMAND  
3504 \* PROGRAM.  
3505 \*  
3506 \* MVI A,FLAG =0 FOR NORMAL, =1 FOR ABORT  
3507 \* DB SYSCALL,.EXIT  
3508 \*  
3509 \* FOR A NORMAL EXIT, THE CONTROL CHARACTER VECTORS ARE CLEARED,  
3510 \* AND SYSCMD IS ENTERED.  
3511 \*  
3512 \* FOR AN ABORT EXIT, THE DISK DRIVER IS RESET.  
3513 \*  
3514 \* /79.06.sc/ IF (. NO SYSTEM DISK AND S.ALONE IS SET)  
3515 \* OR  
3516 \* ( SYSTEM DISK IS STILL MOUNTED )  
3517 \*  
3518 \* NORMAL LINK TO \*SYSCMD.SYS\*  
3519 \*  
3520 \* ELSE  
3521 \*  
3522 \* EXIT TO REBOOT CODE  
3523 \*  
3524 \*  
3525 \*  
063.264 061 200 042 3526 EXIT EQU \*  
063.264 061 200 042 3527 LXI SP,STACK RESET STACK /79.12.GC/  
063.267 345 3528 PUSH PSW SAVE CODE FOR LINKED PROGRAM  
063.270 247 3529 ANA A SET CONDITION CODES  
063.271 076 201 3530 MVI A,U0,CLK+U0,HLT  
063.273 062 010 040 3531 STA .MFLAG REFRESH MFLAG  
063.276 312 312 063 3532 JZ EXIT1 NOT TO ABORT  
063.301 257 3533 XRA A  
063.302 062 061 041 3534 STA A10,UNI SET SYSTEM DISK  
063.305 076 007 3535 MVI A,DC.ABT  
063.307 315 130 040 3536 CALL SYID ABORT SYSTEM DISK  
3537  
063.312 377 056 3538 EXIT1 DB SYSCALL,.CLEARA CLEAR ALL BUT THE LINK CHANNEL  
063.314 072 032 041 3539 LDA S.MOUNT  
063.317 247 3540 ANA A  
063.320 302 332 063 3541 JNZ EXIT2 SYSTEM IS MOUNTED  
063.323 072 245 074 3542 LDA SALONE  
063.326 247 3543 ANA A  
063.327 312 353 063 3544 JZ EXIT3 STAND-ALONE SWITCH IS NOT SET  
3545  
3546 \* LOAD EXIT OVERLAY  
3547  
063.332 361 3548 EXIT2 POP PSW RESTORE LINK CODE  
063.333 041 063 064 3549 LXI H,EXITA  
063.336 061 200 042 3550 LXI SP,STACK RESET STACK  
063.341 377 040 3551 DB SYSCALL,.LINK LINK TO EXIT PROCESSOR  
3552  
3553 \* COULD NOT LINK  
3554  
063.343 385 3555 PUSH PSW SAVE CODE  
063.344 072 032 041 3556 LDA S.MOUNT

HDOS : RESIDENT HDOS CODE  
EXIT = PROCESS.EXIT.SYSCALL

HEATH HBASM V1.4 01/20/78 PAGE 79  
14:04:22 14-MAY-80

063.347 247 3557 ANA A  
063.350 302.014.064 3558 JNZ EXIT4 CONSIDERED FATAL BECAUSE SYSTEM DISK  
3559  
063.353 041.122.064 3560 EXIT3 LXI H,EXITF  
063.356 377.003 3561 DB SYSCALL,.PRINT MAKE SURE WE ARE ON A NEW LINE  
063.360 076.377 3562 MVI A,-1  
063.362 377.055 3563 DB SYSCALL,.CLEAR CLEAR THE LINK CHANNEL  
063.364 041.115.064 3564 LXI H,EXITE  
063.367 315.031.064 3565 CALL EXITS DISMOUNT SY2:  
063.372 041.110.064 3566 LXI H,EXITB  
063.375 315.031.064 3567 CALL EXITS DISMOUNT SY1:  
064.000 041.103.064 3568 LXI H,EXTC  
064.003 315.031.064 3569 CALL EXITS DISMOUNT SY0:  
064.006 315.031.070 3570 CALL RRP  
064.011 303.000.030 3571 JMP ROMBOOT  
3572  
3573 \* ERROR - COULD NOT LINK TO \*SY0:SYSCMD.SYS\*  
3574  
064.014 041.044.064 3575 EXIT4 LXI H,EXITB  
064.017 377.003 3576 DB SYSCALL,.PRINT PRINT MESSAGE  
(A) = CODE  
064.021 361 3577 POP PSW  
064.022 046.000 3578 MVI H,0  
064.024 377.057 3579 DB SYSCALL,.ERROR \*TYPE ERROR  
064.026 315.210.063 3580 CALL FATSERR HALT  
3581  
064.031 377.201 3582 EXIT5 DB SYSCALL,.IMDUN  
064.033 320 3583 RNC NO ERROR  
064.034 376.042 3584 CPI EC,NVM  
064.036 310 3585 RZ NO VOLUME MOUNTED NOT CONSIDERED FATAL  
064.037 377.057 3586 DB SYSCALL,.ERROR  
064.041 315.210.063 3587 CALL FATSERR HALT  
064.044 .012.007.072 3588 EXITB DR NL,BELL,'P02.Cent.Run.'  
064.063 123.131.060 3589 EXITA DB 'SY0:SYSCMD.SYS',0,ENL  
064.103 123.131.060 3590 EXITC DB 'SY01',0  
064.110 123.131.061 3591 EXITD DB 'SY1',0  
064.115 123.131.062 3592 EXITE DB 'SY2',0  
064.122 212 3593 EXITF DB ENL

3597 \*\*\* SCIN - SYSTEM CONSOLE INPUT.  
3598 \*  
3599 \* SCIN TAKES A SINGLE CHARACTER FROM THE CONSOLE INPUT  
3600 \* BUFFER, IF ANY ARE AVAILABLE.  
3601 \*  
3602 \* L1 DB SYSCALL,.SCIN  
3603 \* JC L1 CHARACTER NOT READY  
3604 \*  
3605 \* ENTRY NONE  
3606 \* EXIT 'C' SET IF NO CHARACTER  
3607 \* '/C' CLEAR IF CHARACTER  
3608 \* (A) =CHARACTER  
3609 \* USES A,F  
3610  
3611  
064.123 3612 SCIN EQU \*  
064.123 072 326 040 3613 LDA S.CSLMD  
000.000 3614 ERRNZ CSL.CHR-1  
064.126 037 3615 RAR  
064.127 345 3616 PUSH H SAVE,(HL)  
064.130 315 136 064 3617 CALL SCIN1 GET CHARACTER  
064.133 373 3618 EI  
064.134 341 3619 POP H  
064.135 311 3620 RET  
3621  
3622  
3623 \*\* GET CHARACTER FROM BUFFER.  
3624  
064.136 363 3625 SCIN1 DI  
064.137 332 150 064 3626 JC SCIN2 NOT LINE MODE  
3627  
3628 \* LINE INPUT FORM  
3629  
064.142 072 246 074 3630 LDA CSLLCNT  
064.145 326 001 3631 SUI 1 'C' SET IF NO LINES  
064.147 330 3632 RC NO LINE YET  
3633  
3634 \* TAKE CHARACTER  
3635  
064.150 052 254 074 3636 SCIN2 LHLD SCIOUT  
064.153 072 252 074 3637 LDA SCIIN  
064.156 275 3638 CMP L SEE IF EMPTY  
064.157 067 3639 STC  
064.160 310 3640 RE EMPTY  
064.161 176 3641 MOV A,M (A) = CHARACTER  
064.162 365 3642 PUSH PSW  
064.163 315 277 067 3643 CALL ABP ADVANCE BUFFER POINTER  
064.166 042 254 074 3644 SHLD SCIOUT UPDATE POINTER  
064.171 361 3645 POP PSW (A) = CHARACTER READ  
3646  
3647 \* MAP LOWER CASE TO UPPER, IF 'CTP.MLI' SET  
3648  
064.172 376 141 3649 CPI 'a' LOWER CASE 'A'  
064.174 332 213 064 3650 JC SCIN2.5 NOT LOWER CASE  
064.177 376 173 3651 CPI 'z'+1 LOWER CASE 'Z'  
064.201 322 213 064 3652 JNC SCIN2.5 NOT LOWER CASE

SCIN.....

SCIN.....

14:04:30 16-MAY-80

064.204 147 3653 MOV H,A (H) = CHARACTER  
064.205 .072 327.040 3654 LDA S,CONTY  
000.000 3655 ERRNZ 'a'-'A'-CTP.MLI  
064.210 346.040 3656 ANI CTP.MLI (A) = 0400 IF TO MAP  
064.212 254 3657 XRA H (A) = MAPPED CHARACTER  
064.213 376.012 3658 SCIN2.5 CPI NL  
064.215 312 223 064 3659 JE SCIN3 IS NEW LINE  
064.220 376.004 3660 CPI CTL0 SEE IF CTL0  
064.222 300 3661 RNE IS NOT NEW LINE  
3662  
3663 \* END OF A LOGICAL LINE  
3664  
064.223 041 246 074 3665 SCIN3 LXI H,CSLLCNT  
064.226 .065 3666 DCR M COUNT LINE  
064.227 360 3667 RP NOT UNDERFLOW  
064.230 066.000 3668 MVI M,O  
064.232 311 3669 RET

3672 \*\* SCINI - SYSTEM CONSOLE INPUT INTERRUPT.  
3673 \*  
3674  
3675  
064.233 365 3676 SCINI EQU \*  
064.234 345 3677 PUSH PSW  
064.235 315 247 064 3678 PUSH H  
064.240 341 3679 CALL SCINIO PROCESS CHARACTER.  
064.241 315 232 070 3680 POP H  
064.244 361 3681 CALL CPA CHECK FOR PENDING ABORT  
064.245 373 3682 POP PSW  
064.246 311 3683 EI  
064.246 311 3684 RET EXIT  
3685  
3686 \* PROCESS CHARACTER INTERRUPT  
3687  
064.247 072 343 040 3688 SCINIO LDA S.CDB  
064.252 376 001 3689 CPI CDB.H84  
064.254 312 264 064 3690 JZ SCINIO1 IF 8250  
3691  
3692 \* HAVE 8251  
3693  
064.257 333 372 3694 IN SC.UART+UDR  
064.261 303 266 064 3695 JMP SCINIO2  
3696  
3697 \* HAVE 8250  
3698  
064.264 333 350 3699 SCINIO1 IN SC.ACE+UR.RBR  
3700  
064.266 346 177 3701 SCINIO2 ANI 1770 TRIM PARITY  
064.270 310 3702 RZ NULL CHARACTER  
064.271 376 012 3703 CPI LF  
064.273 310 3704 RE IGNORE LINE-FEEDS  
3705  
3706 \* SEE IF SPECIAL CONTROL CHARACTER:  
3707 \*  
3708 \* CTL-A,B,C, CTL-Z, CTL-D, CTL-P, CTL-Q, CTL-S  
3709  
064.274 376 032 3710 CPI CTLZ  
064.276 312 310 064 3711 JE SCINIO2 CTL-Z  
064.301 376 004 3712 CPI 04  
064.303 322 317 064 3713 JNC SCINIO3 NOT CTL-A, CTL-B, OR CTL-C  
064.306 356 002 3714 XRI 2 CANCEL EFFECT OF NEXT INSTRUCTION  
3715  
3716 \* HAVE CTL-A,B,C OR CTL-Z  
3717  
064.310 356 002 3718 SCINIO2 XRI 2 REMOVE '2' BIT IN "Z (32Q -> 30Q)  
064.312 346 013 3719 ANI CC.FLG+CZ.FLG MASK OFF FLAG  
064.314 303 250 065 3720 JMP PSC PROCESS SPECIAL CHARACTER AND EXIT  
3721  
3722 \* SEE IF CTL-D THROUGH CTL-S  
3723  
064.317 062 247 074 3724 SCINIO3 STA SCIPRE SET PREVIOUS CHARACTER  
064.322 376 017 3725 CPI CTL0  
064.324 332 362 064 3726 JC SCINIO4 NONE OF THESE  
064.327 376 024 3727 CPI 'T'-'@'

```

064.331 322 362 064 3728 JNC SCINI4 NONE OF THESE
064.334 376 022 3729 CPI 'R'-'@'
064.336 312 362 064 3730 JE SCINI4 DONT TAKE CTL-R
3731
3732 * IS CTL-O THROUGH CTL-S
3733
064.341 207 3734 ADD A (A) = 2*CODE
064.342 041 200 065 3735 LXI H,SCINI4-'0'-'0'+ '@'+ '@' (HL) = TABLE.FWA..BIAS
064.345 315 101 030 3736 CALL $DADA. (HL) = TABLE ADDRESS
064.350 072 332 040 3737 LDA S,CONFL
064.353 246 3738 ANA M CLEAR BITS
064.354 043 3739 INX H
064.355 256 3740 XRA M SET BITS
064.356 062 332 040 3741 STA S,CONFL
064.361 311 3742 RET DONE
3743
3744 * IS NOT AN 'ANYTIME' CONTROL CHARACTER. SEE IF LINE MODE
3745
064.362 147 3746 SCINI4 MOV H,A (H) = CHARACTER
064.363 072 326 040 3747 LDA S,CSLMD
000.000 3748 ERRNZ CSL.CHR-1
064.366 037 3749 RAR
064.367 332 130 065 3750 JC SCINI8 IS CHARACTER MODE
3751
3752 * IS LINE MODE. SEE IF RUBOUT OR CTL-U
3753
064.372 072 327 040 3754 LDA S,CONTY
064.375 346 002 3755 ANI CTF,BKM SEE IF MAPPING BKSP TO RUBOUT
064.377 312 010 065 3756 JZ SCIN4.3 NOT MAPPING
065.002 076 010 3757 MVI A,BKSP
065.004 274 3758 CMP H
065.005 312 015 065 3759 JE SCIN4.5 IS BKSP/RUBOUT
065.010 174 3760 SCIN4.3 MOV A,H
065.011 074 3761 INR A
065.012 362 055 065 3762 JP SCINI6 NOT RUBOUT
3763
3764 * IS RUBOUT. TYPE FLAGS AND REMOVE CHARACTER
3765
065.015 315 201 071 3766 SCIN4.5 CALL RRC REMOVE REGULAR CHARACTER
065.020 310 3767 RE NONE TO REMOVE
065.021 147 3768 MOV H,A (H) = REMOVED CHARACTER
065.022 072 327 040 3769 LDA S,CONTY
000.000 3770 ERRNZ CTF,BKS-200Q
065.025 247 3771 ANA A
065.026 372 217 065 3772 JM SCINI11 CAN BACKSPACE: ECHO <BKSP BLANK BKSP>
065.031 072 250 074 3773 LDA CSLRBF
065.034 356 057 3774 XRI ///
065.036 312 051 065 3775 JZ SCINIS ALREADY SET
065.041 062 250 074 3776 STA CSLRBF
065.044 345 3777 PUSH H
065.045 315 002 066 3778 CALL SCOUT1 TYPE ///
065.050 341 3779 POP H
065.051 174 3780 SCINIS MOV A,H
065.052 303 002 066 3781 JMP SCOUT1 ECHO CHARACTER
3782
065.055 072 250 074 3783 SCINI6 LDA CSLRBF

```

065.060 354 057 3784 XRI // SEE IF RUBOUT PENDING  
065.062 302 077 065 3785 JNZ SCINI65 NOT PENDING  
065.065 062 250 074 3786 STA CSLRBF CLEAR FLAG  
065.070 076 057 3787 MVI A,'/'  
065.072 345 3788 PUSH H  
065.073 315 002 066 3789 CALL SCOUT1  
065.076 341 3790 POP H  
065.077 174 3791 SCINI65 MOV A,H (A) = INPUT CHARACTER  
065.100 376 025 3792 CPI 'U'-@  
065.102 302 130 065 3793 JNE SCINI8 NOT CTL-U  
3794  
3795 \* IS CTL-U  
3796  
065.105 315 201 071 3797 SCINI7 CALL RRC REMOVE REGULAR CHARACTER  
065.110 302 105 065 3798 JNZ SCINI7 MORE GO TO  
065.113 076 136 3799 MVI A,'C'  
065.115 315 021 066 3800 CALL SCOUT2 TYPE ^  
065.120 076 125 3801 MVI A,'U'  
065.122 315 021 066 3802 CALL SCOUT2 TYPE 'U'  
065.125 303 261 066 3803 JMP CRLF NEW LINE AND EXIT  
3804  
3805 \* HAVE REGULAR CHARACTER, STORE IF ROOM  
3806  
065.130 345 3807 SCINI8 PUSH H SAVE CHAR  
065.131 052 252 074 3808 LHLD SCIIN  
065.134 345 3809 PUSH H SAVE IN POINTER  
065.135 315 277 067 3810 CALL ABP ADVANCE BUFFER POINTER  
065.140 072 254 074 3811 LDA SCIDOUT  
065.143 275 3812 CMP L  
065.144 302 156 065 3813 JNE SCINI9 HAVE ROOM  
3814  
3815 \* TOO FULL, BEEP CHARACTER  
3816  
065.147 341 3817 POP H  
065.150 361 3818 POP PSW  
065.151 076.007 3819 MVI A,BELL  
065.153 303 002 066 3820 JMP SCOUT1 BEEP  
3821  
3822 \* HAVE ROOM, WILL STORE CHARACTER  
3823  
065.156 042 252 074 3824 SCINI9 SHLD SCIIN  
065.161 341 3825 POP H (HL) = POINTER  
065.162 361 3826 POP PSW (A) = CHAR  
065.163 167 3827 MOV M,A STORE  
065.164 376 004 3828 CPI CTLD  
065.166 312 201 065 3829 JE SCINI95 IS CTL-D  
065.171 376 015 3830 CPI CR  
065.173 302 205 065 3831 JNE SCINI10 NOT CR  
065.176 076 012 3832 MVI A,NL  
065.200 167 3833 MOV M,A STORE NL  
3834  
3835 \* HAVE SEEN END OF LOGICAL LINE  
3836  
065.201 041 246 074 3837 SCINI95 LXI H,CSLLCNT  
065.204 064 3838 INR M COUNT LINE  
065.205 147 3839 SCINI10 MOV H,A (H) = CHAR

3840  
3841 \* SEE IF TO ECHO  
3842  
065.206 072.326.040 3843 LDA S,CSLMD  
000.000 3844 ERRNZ CSL,ECH-200R  
065.211 027 3845 RAL  
065.212 174 3846 MOV A,H (A) = CHA  
065.213 322.002.066 3847 JNC SCOUT1 AM TO ECHO  
065.216 311 3848 RET SUPPRESS ECHO  
3849  
3850 \* HAVE BACKSPACE FOR TERMINAL WITH BACKSPACE CAPABILITY:  
3851 \* ISSUE BKSP, BLANK, BKSP  
3852  
065.217 076.010 3853 SCINI11 MVI A,BKSP  
065.221 315 021 066 3854 CALL SCOUT2  
065.224 076.040 3855 MVI A,' '  
065.226 315 021 066 3856 CALL SCOUT2 PRINT BKSP, <BLANK>, BKSP  
065.231 076.010 3857 MVI A,BKSP  
065.233 303 021 066 3858 JMP SCOUT2 PRINT AND EXIT  
3859  
3860  
3861 \*\* PROCESSING FOR CTL-P, CTL-O, CTL-Q, CTL-S  
3862 \*  
3863 \* FIRST BYTE = CLEAR MASK FOR S.CONFL  
3864 \* 2ND BYTE = XOR MASK FOR S.CONFL  
3865  
065.236 377 001 3866 SCINIA DB -1,CO,FLG CTL-O  
065.240 376.000 3867 DB 377Q-CO,FLG,0 CTL-P  
065.242 177 000 3868 DB 377Q-CS,FLG,0 CTL-Q  
065.244 377.000 3869 DB -1,0 CTL-R  
065.246 177 200 3870 DB 377Q-CS,FLG,CS,FLG CTL-S  
3871

3873 \*\* PSC - PROCESS SPECIAL CHARACTER.  
3874 \*  
3875 \* PSC IS CALLED WHEN A SPECIAL INTERRUPT CHARACTER IS DETECTED  
3876 \* (CTL-A, CTL-B, CTL-C, CTL-Z). PSC DECIDES IF SPECIAL  
3877 \* SERVICE WILL BE NEEDED (IF REQUESTED BY USER FOR CTL-A, -B, AND -C,  
3878 \* OR UPON 2ND CTL-Z).  
3879 \*  
3880 \* IF SERVICE IS NEEDED, THE SERVICE ADDRESS IS STORED IN  
3881 \* S.CAADR.  
3882 \*  
3883 \* ENTRY (A) = CHARACTER DETECTED  
3884 \* EXIT S.CAADR > 256 IF PROCESSING NEEDED  
3885 \* USES A,F,H,L  
3886  
3887  
065.250 305 3888 PSC PUSH B  
065.251 315 256 065 3889 CALL PSC1 PROCESS  
065.254 301 3890 POP B  
065.255 311 3891 RET  
3892

065.254 107 3893 PSC1 MOV B,A (B) = CHARACTER  
065.257 346 010 3894 ANI CZ,FLG  
065.261 302 307 065 3895 JNZ PSC2 IS CTL-Z  
3896  
3897 \* IS CTL-A, -B, OR -C  
3898  
065.264 170 3899 MOV A,B  
065.265 207 3900 ADD A,(A)=.2\*CODE  
065.266 310 3901 RZ NONE  
065.267 .041.333.040. 3902 LXI H,S.CCTAB-2.  
065.272 315 101 030 3903 CALL \$DADA. (HL) = ADDRESS OF ADDRESS  
065.275. 315.211.030. 3904 CALL \$HLIHL  
065.300 174 3905 MOV A,H  
065.301. 247. 3906 ANA A  
065.302 310 3907 RZ NONE TO SET  
065.303. 042.333.040. 3908 SHLD S,CAAIR SET CONSOLE ABORT ADDRESS  
065.306 311 3909 RET  
3910  
3911 \* IS CTL-Z  
3912  
065.307 072 032 041 3913 PSC2 LDA S,MOUNT  
065.312. 247. 3914 ANA A  
065.313 310 3915 RZ SYSTEM IS NOT MOUNTED, IGNORE  
065.314. 373. 3916 EI  
065.315 076 136 3917 MVI A,'/'  
065.317. 315.002.064. 3918 CALL SCOUT1  
065.322 076 132 3919 MVI A,'Z'  
065.324. 315.002.066. 3920 CALL SCOUT1  
065.327 072 247 074 3921 LDA SCIPRE  
065.332. 376.032. 3922 CPI CTLZ  
065.334 312 351 065 3923 JE PSC3 2ND CTL-Z  
065.337. 076.032. 3924 MVI A,CTLZ  
065.341 062 247 074 3925 STA SCIPRE SET CTL-Z AS PREVIOUS CHARACTER  
3926  
3927 \* TYPE '?' WARNING  
3928  
065.344 076 077 3929 MVI A,'?'  
065.346. 303.002.066. 3930 JMP SCOUT1 OUTPUT AND RETURN  
3931  
3932 \*. 2ND CTL-Z HIT  
3933  
065.351. 041.264.063. 3934 PSC3 LXI H,EXIT  
065.354 042 333 040 3935 SHLD S,CAAIR CONSOLE ABORT ADDRESS  
065.357. 311. 3936 RET

3939 \*\*\* SCOUT - SYSTEM CONSOLE OUTPUT.  
3940 \*  
3941 \* SCOUT OUTPUTS A SINGLE CHARACTER TO THE CONSOLE. CURSOR POSITIONING  
3942 \* IS KEPT TRACK OF. A 'NL' CHARACTER INDICATES A NEW LINE, 'CR' AND  
3943 \* 'LF' CHARACTERS SHOULD NOT BE USED.  
3944 \*  
3945 \* NOTE THAT THERE ARE SOME GAMES PLAYED WITH THE PARITY BIT,  
3946 \* SEE \*CRLF\* FOR DISCUSSION.  
3947 \*  
3948 \* MVI A,CHAR  
3949 \* DB SYSCALL,.SCOUT  
3950 \*  
3951 \* ENTRY (A) = CHARACTER  
3952 \* EXIT (A) = CHARACTER  
3953 \* USES NONE  
3954  
3955  
.065.360 3956 SCOUT EQU \*  
065.360 365 3957 PUSH PSW SAVE CHAR  
.065.361 345 3958 PUSH H, SAVE (HL)  
065.362 147 3959 MOV H,A (A) = CHARACTER  
.065.363 .072.332.040 3960 SCOUT0 LDA S.CONFL  
000.000 3961 ERRNZ CS.FLG-200Q  
.065.366 247 3962 ANA A  
065.367 372 363 065 3963 JM SCOUT0 AM IN WAIT MODE  
000.000 3964 ERRNZ CO.FLG-1  
065.372 037 3965 RAR  
.065.373 174 3966 MOV A,H (A) = CHARACTER  
065.374 324 002 066 3967 CNC SCOUT1 PERFORM I/O IF NOT CTL-O  
.065.377 341 3968 PQF H  
066.000 361 3969 POP PSW  
.066.001 311 3970 RET  
  
3972 \*\* SCOUT1 - OUTPUT CHARACTER.  
3973 \*  
3974 \* SCOUT1 IGNORES CTL-A AND CTL-S, AND IS USED BY HDOS CODE.  
3975 \* WHICH MUST NOT BE HELD UP.  
3976 \*  
3977 \* SCOUT1 MAY BE CALLED WITH INTERRUPTS DISABLED.  
3978 \*  
3979 \* ENTRY (A) = CHARACTER  
3980 \* EXIT NONE  
3981 \* USES A,F,H,L  
3982  
3983  
.066.002 346.177 3984 SCOUT1 ANI 1770 TRIM  
3985  
3986 \* IF LOWER CASE MAPPING TURNED ON, DO IT  
3987  
.066.004 376.140 3988 CPI 140Q  
066.006 332 021 066 3989 JC SCOUT2 NOT LOWER CASE  
.066.011 147 3990 MOV H,A  
066.012 072 327 040 3991 LDA S.CONFL  
.066.015 207 3992 ADD A

066.016 346 040 3993 ANI CTP.MLOX2 (A) = 040Q IF MAPPING  
066.020 254 3994 XRA H CLEAR BIT IF MAPPING

3995  
3996  
3997 \*\* SOME ROUTINES CALL HERE (SCOUT ITSELF, RECURSIVELY)  
3998 \* TO OUTPUT CHARACTERS WITHOUT THE CPU OVERHEAD OF SCOUT OR  
3999 \* SCOUT1.

4000

066.021 376 011 4001 SCOUT2 CPI TAB

066.023 302 056 066 4002 JNE SCOUT4 NOT TAB

4003  
4004 \* HAVE TAB, EXPAND TO COLUMN

4005

066.026 072 327 040 4006 LDA S.CONTY  
000.000 4007 ERRNZ CTP.TAB-1

066.031 037 4008 RAR

066.032 076 011 4009 MVI A,TAB

066.034 332 056 066 4010 JC SCOUT4 TERMINAL WILL TAKE TABS

066.037 076 040 4011 SCOUT3 MVI A,' '

066.041 315 056 066 4012 CALL SCOUT4 TYPE BLANK

066.044 072 330 040 4013 LDA S.CUSOR

066.047 075 4014 BCR A

066.050 346 007 4015 ANI 7

066.052 302 037 066 4016 JNZ SCOUT3 NOT TO FIELD

066.055 311 4017 RET DONE

4018

4019 \* TYPE CHARACTER. (A) =CHARACTER

4020

066.056 376 012 4021 SCOUT4 CPI NL

066.060 312 261 066 4022 JE CRLF IS A NEW LINE

066.063 041 330 040 4023 LXI H,S.CUSOR (M) = CONSOLE CURSOR POINTER

066.066 376.015 4024 CPI CR

066.070 302 075 066 4025 JNE SCOUT5 NOT CR

066.073 066.001 4026 MVI M,1 CLEAR POINTER

066.075 376 010 4027 SCOUT5 CPI BKSP

066.077 302 110 066 4028 JNE SCOUT6 IS NOT BKSP

066.102 065 4029 DCR M

066.103 302.110.066 4030 JNZ SCOUT6 NOT UNDERFLOW

066.106 066 001 4031 MVI M,1 RESET

066.110 376.011 4032 SCOUT6 CPI TAB

066.112 302 126 066 4033 JNE SCOUT7 NOT TAB

066.115 176 4034 MOV A,M

066.116 306 007 4035 ADI 7Q

066.120 346.370 4036 ANI 3700

066.122 074 4037 INR A ADJUST COLUMN COUNT TO NEXT TAB

066.123 167 4038 MOV M,A

066.124 076 011 4039 MVI A,TAB

066.126 346.177 4040 SCOUT7 ANI 1770 TRIM TO 7 BITS

066.130 376 040 4041 CPI '/' SEE IF PRINTING

066.132 332.147.066 4042 JC SCOUT8 NOT PRINTING

4043

4044 \* CHECK FOR CHARACTER WRAP?

4045

066.135 365 4046 PUSH PSW SAVE CURRENT CHARACTER

066.136 043 4047 INX H

000.000 4048 ERRNZ S.CONWI-S.CUSOR-1

066.137 176 4049 MOV A,M (A) = CONSOLE WIDTH  
066.140 053 4050 DCX H  
000.000 4051 ERRNZ S.CONWI-S.CUSOR-1  
066.141 276 4052 CMP M  
066.142 334 261 066 4053 CC CRLF WIDTH < S.CUSOR AND ABOUT TO OUTPUT A PRINTING  
066.145 361 4054 POP PSW CHARACTER  
4055  
066.146 064 4056 INR M COUNT CHARACTER  
4057  
4058 \* OUTPUT CHARACTER.  
4059  
066.147 376 014 4060 SCOUT8 CPI FF  
066.151 312 222 066 4061 JE SCOUT10 IS FORM FEED  
066.154 365 4062 PUSH PSW SAVE CHAR  
066.155 072 343 040 4063 SCOUT9 LDA S.CDB  
066.160 376 001 4064 CPI CDB.HB4  
066.162 312 202 066 4065 JZ SCOUT92 IF 8250  
4066  
4067 \* HAVE 8251  
4068  
066.165 333 373 4069 SCOUT91 IN SC.UART+USR  
066.167 346 001 4070 ANI USR.TXR  
066.171 312 165 066 4071 JZ SCOUT91 NOT READY  
066.174 361 4072 POP PSW  
066.175 323 372 4073 OUT SC.UART+UDR  
066.177 303 214 066 4074 JMP SCOUT95  
4075  
4076 \* HAVE 8250  
4077  
066.202 333 355 4078 SCOUT92 IN SC.ACE+UR.LSR  
066.204 346 040 4079 ANI UC.THE  
066.206 312 202 066 4080 JZ SCOUT92  
066.211 361 4081 POP PSW  
066.212 323 350 4082 OUT SC.ACE+UR.THR  
4083  
066.214 376 377 4084 SCOUT95 CPI 377Q SEE IF TO PAD  
066.215 4085 SCOUTA EQU \*-1  
066.216 314 240 066 4086 CE SC0LY MUST DELAY FOR PADS  
066.221 311 4087 RET /79.04.GC/  
4088  
4089 \* IS FORM FEED  
4090  
066.222 076 012 4091 SCOUT10 MVI A,10  
066.224 365 4092 SCOUT11 PUSH PSW SAVE LINE FEED COUNT  
066.225 076 212 4093 MVI A,LF+200Q  
066.227 315 021 066 4094 CALL SCOUT2 OUTPUT LINE FEED  
066.232 361 4095 POP PSW  
066.233 075 4096 DCR A  
066.234 302 224 066 4097 JNZ SCOUT11 MORE TO GO  
066.237 311 4098 RET

4100 \*\* SCDLY - ISSUE DELAY (VIA 00 CHARACTERS)

4101 \*  
4102 \* ENTRY NONE

4103 \* EXIT NONE

4104 \* USES A,F

4105

4106

066.240 072 262 074 4107 SCDLY LIA CSLILY

066.243 247 4108 SCDLY1 ANA A

066.244 310 4109 RZ NO MORE PADS.

066.245 365 4110 PUSH PSW

066.246 345 4111 PUSH H SAVE REGISTERS

066.247 257 4112 XRA A

066.250 315 021 066 4113 CALL SCOUT2 WRITE PAD

066.253 341 4114 POP H

066.254 361 4115 POP PSW

066.255 075 4116 DCR A

066.256 303 243 066 4117 JMP SCDLY1 DELAY UNTIL DONE.

4119 \*\* CRLF - START NEW LINE.

4120 \*

4121 \* NOTE THAT CRLF DOESN'T WANT THE 'LF' TO BE TAKEN AS A  
'NL', AND THUS TRIGGER A RECURSIVE LOOP. WE GET AROUND THAT.

4122 \* BY SETTING THE PARITY BIT FOR IT, SO THAT IT FAILS THE

4123 \* CPI NL

4124 \* TEST. THE PARITY BIT IS STRIPPED (AGAIN, FOR MOST) BEFORE

4125 \* THE CHAR IS PASSED TO THE USART.

4126 \* ENTRY NONE

4127 \* EXIT NONE

4128 \* USES A,F

4129 \*

4130 \*

4131

066.261 345 4132 CRLF PUSH H SAVE (HL)

066.262 076 015 4133 MVI A,CR

066.264 315 002 066 4134 CALL SCOUT1

066.267 076 212 4135 MVI A,LF+2000

066.271 315 021 066 4136 CALL SCOUT2 OUTPUT IT

066.274 341 4137 POP H

066.275 311 4138 RET

4141 \*\*\* READ - PROCESS READ SYSCALL.  
4142 \*  
4143 \* READ PROCESSES READ SYSCALLS. IF A SERIAL DEVICE, PASS TO  
4144 \* DRIVER. IF A STORAGE DEVICE, HANDLE STORAGE MAPPING.  
4145 \*  
4146 \* MVI A,CHAN  
4147 \* LXI B,COUNT MUST BE MULTIPLE OF 256  
4148 \* LXI D,ADDR  
4149 \* DB SYSCALL,.READ READ DATA FROM CHANNEL  
4150 \*  
4151 \* ENTRY (A) = I/O CHANNEL NUMBER  
4152 \* (BC) = DATA COUNT  
4153 \* (DE) = ADDRESS FOR DATA  
4154 \* EXIT (BC) = COUNT LEFT  
4155 \* (DE) = NEXT UNUSED ADDRESS  
4156 \* 'C' CLEAR IF ALL OK  
4157 \* 'C' SET IF ERROR  
4158 \* (A) = ERROR CODE  
4159 \* USES ALL  
4160  
4161  
066.276..315.260.070 4162 READ CALL FCI FETCH CHANNEL INFO  
066.301 330 4163 RC ERROR  
066.302 .247 4164 ANA A  
066.303 312 344 031 4165 JZ ERR.FNO FILE NOT OPEN  
000.000. 4166 ERRNZ FT.OR-2  
066.306 037 4167 RAR  
066.307 .037 4168 RAR  
066.310 322 350 031 4169 JNC ERR.ILR ILLEGAL REQUEST  
000.000. 4170 ERRNZ FT.BD-1  
066.313 027 4171 RAL  
066.314 .076.000. 4172 MVI A:DC,REA (A) = DEVICE CODE  
066.316 322 040 041 4173 JNC , AIO.VEC IF NOT DIRECTORY DEVICE, CALL DRIVER  
066.321..315.131.067. 4174 CALL DIREAD DIRECTORY READ  
066.324 303 237 071 4175 JMP SCI STORE CHANNEL INFORMATION AND EXIT

4178 \*\*\* WRITE - PROCESS WRITE SYSCALL.  
4179 \*  
4180 \* MVI A,CHAN  
4181 \* LXI B,COUNT MUST BE MULTIPLE OF 256  
4182 \* LXI D,ADDR  
4183 \* DB SYSCALL,. WRITE WRITE DATA TO CHANNEL  
4184 \*  
4185 \* ENTRY (A) = CHANNEL #  
4186 \* (BC) = DATA COUNT  
4187 \* (DE) = ADDRESS  
4188 \* EXIT (BC) = COUNT LEFT  
4189 \* (DE) = NEXT ADDRESS  
4190 \* 'C' CLEAR IF OK  
4191 \* 'C' SET IF ERROR  
4192 \* (A) = ERROR CODE  
4193 \* USES ALL  
4194  
4195  
066.327 315 260 070 4196 WRITE CALL FCI FETCH CHANNEL INFORMATION  
066.332 330 4197 RC ERROR  
066.333 247 4198 ANA A  
066.334 312 344 031 4199 JZ ,ERR,FNO FILE NOT OPEN  
066.337 147 4200 MOV H,A SAVE COPY IN H  
066.340 346 004 4201 ANI FT,OW SEE IF OPEN FOR WRITE  
066.342 312 350 031 4202 JZ ERR,ILR ILLEGAL REQUEST  
066.345 174 4203 MOV A,H  
000.000 4204 ERRNZ FT,DD-1  
066.346 037 4205 RAR  
066.347 076 001 4206 MVI A,DC,WRI REQUEST WRITE  
066.351 322 040 041 4207 JNC AIO,VEC NOT DIRECTORY DEVICE  
066.354 315 211 067 4208 CALL DIWRITE DIRECTOREID WRITE  
066.357 303 237 071 4209 JMP SCI STORE CHANNEL INFO

4212 \*\*\* PRINT - PRINT CONSOLE LINE.  
4213 \*  
4214 \* PRINT CAUSES A CODED LINE TO BE PRINTED AT THE CONSOLE.  
4215 \*  
4216 \* LXI H,LINEADDR  
4217 \* DB SYSCALL,PRINT  
4218 \*  
4219 \*. THE LAST CHARACTER IN THE LINE SHOULD HAVE THE 2000 BIT SET.  
4220 \*  
4221 \*. ENTRY... (HL) = LINE ADDRESS  
4222 \* EXIT (HL) = LWA OF MESSAGE +1  
4223 \*. USES A,F,H,L  
4224  
4225  
066.362 176 4226 PRINT MOV A,M (A) = CODE  
066.363 346.177 4227 ANI 1770 CLEAR FLAG BIT  
066.365 315 360 065 4228 CALL SCOUT TYPE IT  
066.370.274 4229 CMP M  
066.371 043 4230 INX H  
066.372.312.362.066. 4231 JE PRINT NOT 2000 SET  
066.375 311 4232 RET

```

4236 *** CONSL - SET AND CLEAR CONSOLE FLAGS.
4237 *
4238 * CONSL IS CALLED TO SET, CLEAR, OR READ BITS IN THE VARIOUS
4239 * CONSOLE FLAGS.
4240 *
4241 * THE CALLER PASSES AN INDEX INTO THE PROPER FLAG, A
4242 * MASK TO INDICATE THE EFFECTED BITS, AND A SET OF NEW VALUES
4243 * FOR THOSE BITS.
4244 *
4245 * INDEX =
4246 *
4247 * 0 S.CSLMD
4248 * 1 S.CONTY
4249 * 2 S.CUSR
4250 * 3 S.CONWI
4251 * 4 S.CONFL
4252 *
4253 * ENTRY (A) = INDEX
4254 * (B) = NEW VALUES
4255 * (C) = MASK ('1'..BIT FOR EVERY BIT TO CHANGE)
4256 * EXIT 'C' CLEAR IF NO ERROR
4257 * (A) = NEW VALUE
4258 * 'C' SET IF ERROR
4259 * (A) = ERROR CODE
4260 * USES ALL
4261
4262
066.376 4263 CONSL EQU *
066.376 376 005 4264 CPI 5
067.000 322 350 031 4265 JNC ERR,ILR ILLEGAL REQUEST
067.003 041 326 040 4266 LXI H,S.CSLMD
067.006 315 101 030 4267 CALL $DADA. (HL) = ADDRESS FOR BYTE
067.011 171 4268 MOV A,C
067.012 240 4269 ANA B CLEAR..(B)..TO..THE..BITS..TO..BE..ALTERED
067.013 107 4270 MOV B,A
067.014 171 4271 MOV A,C
067.015 057 4272 CMA (A) = -MASK
067.016 363 4273 DI INTERLOCK CONSOLE
067.017 246 4274 ANA M CLEAR EFFECTED BITS
067.020 260 4275 ORA B SET NEW VALUES
067.021 167 4276 MOV M,A REPLACE
067.022 373 4277 EI
067.023 311 4278 RET

```

4282 \*\*\* CLRCO - CLEAR CONSOLE BUFFERS.

4283 \* CLRCO CLEARS THE CONSOLE TYPE-AHEAD BUFFER.

4285 \* CTL-Q AND CTL-S FLAGS ARE ALSO CLEARED,

4286 \*

4287 \* ENTRY NONE

4288 \* EXIT NONE

4289 \* USES ALLL

4290

4291

067.024 363 4292 CLRCO DI

067.025 .041.245.074. 4293 LXI H:CSLIBUF

067.030 .042.252.074 4294 SHLD SCIN

067.033 .042.254.074. 4295 SHLD SCIOUT CLEAR POINTER

067.036 257 4296 XRA A

067.037 .062.246.074. 4297 STA CSLLCNT CLEAR LINE COUNT

067.042 .062.250.074 4298 STA CSLRBF CLEAR RUBOUT BUFFER

067.045 .062.332.040. 4299 STA S,CONFL CLEAR CTL-Q AND CTL-S

067.050 373 4300 EI

067.051 .311 4301 RET

4304 \*\*\* LOADO - LOAD SPECIFIED OVERLAY  
4305 \*  
4306 \* LOADO LOADS THE OVERLAY SPECIFIED THROUGH THE INDEX.  
4307 \*  
4308 \* OVERLAY INDEX  
4309 \*  
4310 \* HDOSOVL 0  
4311 \* HDOSOVL2 1  
4312 \*  
4313 \*  
4314 \* \*\*\*\*\*  
4315 \* \*  
4316 \* \* NOTE: THIS CALL SHOULD NOT BE MADE FROM ANOTHER OVERLAY \*  
4317 \* \* UNLESS IT IS THE OVERLAY TO BE LOADED. \*  
4318 \* \*  
4319 \* \*\*\*\*\*  
4320 \*  
4321 \*  
4322 \* ENTRY: (A) = OVERLAY INDEX  
4323 \*  
4324 \* EXIT: (PSW) = 'C' CLEAR IF NO ERRORS  
4325 \* ('C' SET... IF... ERRORS)  
4326 \* (A) = ERROR CODE  
4327 \*  
4328 \* USES: ALL  
4329 \*  
4330 \*  
067.052 4331 LOADO ERU \*  
067.052 365 4332 PUSH PSW SAVE THE OVERLAY INDEX  
067.053 315.360.062 4333 CALL LDON LOAD THE SPECIFIED OVERLAY  
067.056 332 122 067 4334 JC LOADO2 ERROR  
067.061 .072.371.040 4335 LDA S.OVLF<sub>L</sub>  
067.064 346 200 4336 ANI OVL.UCS  
067.066 .302.120.067 4337 JNZ LOADO1 USER CODE IS SWAPPED  
067.071 361 4338 POP PSW RESTORE OVERLAY INDEX  
067.072 .315.152.071 4339 CALL OTI OVERLAY TABLE INDEX  
067.075 006 000 4340 DW OVL.FLB (HL) = ADDRESS OF FLAG BYTE  
067.077 176 4341 MOV A,M  
067.100 346 002 4342 ANI OVL.RES IT IS ALREADY RESIDENT.  
067.102 300 4343 RNZ  
067.103 176 4344 MOV A,M  
067.104 .366.002 4345 ORI OVL.RES  
067.106 167 4346 MOV M,A FLAG OVERLAY AS PERM. RESIDENT  
067.107 .053 4347 DCX H  
067.110 053 4348 DCX H (HL) = OVERLAY ENTRY POINT  
000.000 4349 ERRNZ OVL.FLB-OVL,ENT-2  
067.111 315 211 030 4350 CALL \$HLIHL (HL) = ENTRY ADDRESS  
067.114 .042.320.040 4351 SHLD S.SYRM SET OVERLAY ENTRY POINT AS HDOS LOWER BOUND  
067.117 311 4352 RET  
4353  
067.120 076 021 4354 LOADO1 MVI A,EC.NEM NOT ENOUGH MEMORY  
067.122 067 4355 LOADO2 STC FLAG ERROR  
067.123 341 4356 POP H REMOVE SAVED OVERLAY INDEX  
067.124 311 4357 RET

HDOS - RESIDENT HDOS CODE ..... HEATH H8ASM V1.4 01/20/78 PAGE 97  
VERSN .... RETURN.HDOS.VERSION.NUMBER..... VERSN.....14:05:19...16-MAY-80.....

4361 \*\* VERSN - RETURN HDOS VERSION NUMBER'

4362 \*

4363 \* VERSN RETURNS THE HDOS VERSION NUMBER AS A ONE BYTE BCD NUMBER.  
4364 \* A DECIMAL IS ASSUMED BETWEEN THE HIGH AND LOW ORDER NIBLES.

4365 \*

4366 \*

4367 \* ENTRY NONE

4368 \*

4369 \* EXIT (PSW) = (A) = VERSION NUMBER

4370 \*

4371 \* USES (PSW)

4372 \*

4373

067.125 076.026 4374 VERSN MVI A,VERS

067.127 247 4375 ANA A

CLEAR CARRY

067.130 311 4376 RET

4380 \*\* DIREAD - DIRECTORIED READ.  
4381 \*  
4382 \* DIREAD READS THE SPECIFIED NUMBER OF SECTORS FROM A  
4383 \* DIRECTORIED DEVICE. THE DATA IS READ FROM THE CURRENT  
4384 \* FILE POSITION.

4385 \*  
4386 \* ENTRY (B) = SECTOR COUNT

4387 \* (C) = 0

4388 \* (DE) = ADDRESS FOR DATA

4389 \* AIO,XXX,SETUP

4390 \* EXIT (BC) = COUNT LEFT

4391 \* (DE) = NEXT FREE ADDRESS

4392 \* 'C' CLEAR IF OK

4393 \* 'C' SET IF ERROR

4394 \* (A) = CODE

4395 \* USES ALL

4396

4397

067.131 4398 DIREAD EQU \*

067.131 170 4399 MOV A,B

067.132 247 4400 ANA A

067.133 310 4401 RZ NOTHING TO READ

067.134 325 4402 PUSH D SAVE (DE)

067.135 315 002 032 4403 CALL DCA DETERMINE CONTINUOUS AREA

067.140 321 4404 POP D

067.141 072 113 041 4405 LDA AIO.EOF

067.144 037 4406 RAR

067.145 330 4407 RC EXIT IF EOF

067.146 305 4408 PUSH B

067.147 315 145 033 4409 CALL PDI PREPARE DEVICE I/O

000.000 4410 ERRNZ DC.REA

067.152 315 162 067 4411 CALL DIREAD1 PERFORM I/O

067.155 301 4412 POP B

067.156 322 131 067 4413 JNC DIREAD IF NOT ERROR

067.161 311 4414 RET

4416 \*\* DIREAD1 - PERFORM I/O

4417 \*

4418 \* DIREAD1 CALLS THE I/O DRIVER, AFTER COMPUTING THE COMPLETION ADDRESS

4419 \* (WHICH THE DRIVER WILL NOT RETURN)

4420 \*

4421 \* ENTRY (A) = OPERATION CODE

4422 \* (BC) = COUNT

4423 \* (DE) = ADDRESS

4424 \* (HL) = SECTOR NUMBER

4425 \* EXIT (PSW) AS FROM DRIVER

4426 \* (BC) AS FROM DRIVER

4427 \* (DE) = (BC ON ENTRY) + (DE ON ENTRY)

4428 \* (HL) AS FROM DRIVER

4429 \* USES ALL

4430

4431

067.162 353 4432 DIREAD1 XCHG (HL) = I/O ADDRESS

067.163 345 4433 PUSH H SAVE

067.164 011	4434	DAD B	(HL) = NEW ADDRESS
067.165 343	4435	XTHL	(HL) = I/O ADDRESS ((SP)) = NEW ADDRESS
067.166 353	4436	XCHG	RESTORE AS UPON ENTRY
067.167 315 040 041	4437	CALL AIO.VEC	CALL DRIVER
067.172 321	4438	POP D	(DE) = NEW ADDRESS
067.173 311	4439	RET	

4441 \*\* DIWRITE - DIRECTORY DEVICE WRITE.  
4442 \*  
4443 \* DIWRITE WRITES THE SPECIFIED NUMBER OF SECTORS TO A DIRECTORIED  
DEVICE.  
4444 \*  
4445 \*  
4446 \* ENTRY (B) = COUNT  
4447 \* (C) = 0  
4448 \* (DE) = TEXT ADDRESS  
4449 \* AIO.XXX SETUP  
4450 \* EXIT (BC) = COUNT LEFT  
4451 \* (DE) = ADDRESS  
4452 \* 'C' CLEAR, IF OK  
4453 \* 'C' SET IF ERROR  
4454 \* (A) = ERROR CODE  
4455 \* USES ALL  
4456  
4457

067.174 305	4458	DWRIT1 PUSH B	SAVE COUNT
067.175 315 145 033	4459	CALL PDI	PREPARE FOR DEVICE I/O
067.200 076 001	4460	MVI A,DC.WRI	
067.202 315 162 067	4461	CALL DIREAD1	PERFORM I/O
067.205 301	4462	POP B	(BC) = COUNT LEFT
067.206 076 023	4463	MVI A,EC.WF	WRITE FAIL (IF CARRY SET)
067.210 330	4464	RC	RETURN IF ERROR
	4465		

067.211	4466	DIWRITE EQU *	
067.211 170	4467	MOV A,B	
067.212 247	4468	ANA A	
067.213 310	4469	RZ	NO MORE
067.214 325	4470	PUSH D	
067.215 315 002 032	4471	CALL DCA	DETERMINE CONTIGUOUS AREA
067.220 321	4472	POP D	
067.221 072 113 041	4473	LDA AIO.EOF	
067.224 037	4474	RAR	
067.225 322 174 067	4475	JNC DWRIT1	IF NOT EOF
	4476		

	4477 *	MUST APPEND SECTORS TO END OF THE FILE.	
	4478 *	ALLOCATE THE SPACE.	
	4479		
067.230 170	4480	DWRIT2 MOV A,B	
067.231 247	4481	ANA A	
067.232 310	4482	RZ	NO MORE
067.233 052 116 041	4483	LHLD AIO.CHA	
067.236 076 037	4484	MVI A,IOC.DIR+DIR.FLG-IOC.IDA	
067.240 315 101 030	4485	CALL \$DADA.	(HL) = \$DIR.FLG IN CHANNEL
067.243 176	4486	MOV A,M	

067.244 346 357 4487 ANI 377Q-DIF.CNT IS NOT CONTIGUOUS ANY MORE (IF IT EVER WAS)  
067.246 167 4488 MOV M,A  
067.247 325 4489 PUSH D  
067.250 315 311 067 4490 CALL ACA ALLOCATE CONTINUOUS AREA  
067.253 321 4491 POP D  
067.254 072 112 041 4492 LDA A10,EOM  
067.257 037 4493 RAR  
067.260 330 4494 RC EXIT IF EOM  
4495  
4496 \* NOT OUT OF SPACE, WRITE IT.  
4497  
067.261 305 4498 PUSH B  
067.262 315 145 033 4499 CALL PDI PREPARE DEVICE I/O  
067.265 076 001 4500 MVI A,DC,WRI  
067.267 315 162 067 4501 CALL DIREAD1 PERFORM I/O  
067.272 301 4502 POP B  
067.273 322 230 067 4503 JNC DIWRIT2 GO AGAIN  
067.276 311 4504 RET RETURN WITH ERROR CODE

4508 \*\* ABP - ADVANCE BUFFER POINTERS.

4509 \*  
4510 \* ABP ADVANCES THE BUFFER POINTER TO THE NEXT BYTE. IF THE  
4511 \* POINTER OVERFLOWS, IT IS WRAPPED.

4512 \*  
4513 \* ENTRY (HL) = POINTER  
4514 \* EXIT (HL) POINTED TO NEXT  
4515 \* USES A,F,H,L

4516

4517 067.277 043 4518 ABP INX H INCREMENT

067.300 072 260 074 4519 LDA SCILWA

067.303 275 4520 CMP L

067.304 300 4521 RNE NOT OVER END

067.305 052 256 074 4522 LHLD SCIFWA

067.310 311 4523 RET

4525 \*\* ACA - ALLOCATE CONTINUOUS AREA.

4526 \*  
4527 \* ACA IS CALLED TO APPEND SECTORS TO THE END OF A FILE.  
4528 \* IT ALLOCATES AS MANY CONTINUOUS SECTORS AS IT CAN UNTIL  
4529 \* ENOUGH ARE ALLOCATED, OR A BREAK IN THE CONTINUITY IS REQUIRED.

4530 \*  
4531 \* FIRST, THE REMAINING SECTORS IN THE GROUP ARE USED.

4532 \* 2ND, ACA ATTEMPTS TO OBTAIN THE IMMEDIATELY FOLLOWING GROUP.

4533 \* 3RD, ACA TRY'S TO LOCATE A VIRGIN CLUSTER

4534 \* 4TH, ACA TAKES ANY FREE GROUPS.

4535 \*

4536 \* ENTRY (B) = SECTOR COUNT

4537 \* AIO.XXX SETUP

4538 \* EXIT (B) = SECTORS NOT ALLOCATED

4539 \* AIO.CNT = AMOUNT ALLOCATED

4540 \* AIO.EOM = EC.EOM\*2+1, IF END OF MEDIA

4541 \* AIO.LGN, AIO.LSI UPDATED FOR ADDITIONS

4542 \* AIO.CGN, AIO.CSI = AIO.LGN, AIO.LSI

4543 \* AIO.TFP = SETUP WITH GROUP AND INDEX OF START OF AREA

4544 \* USES ALL

4545

4546

067.311 016 000 4547 ACA MVI C,0 (C) = COUNT ALLOCATED

067.313 052 051 041 4548 ACA0 LHLD AIO.LGN (L) = AIO.LGN, (H) = AIO.LSI

000.000 4549 ERRNZ AIO.LSI-AIO.LGN-1

067.316 042 114 041 4550 SHLD AIO.TFP SAVE WRITE ADDRESS

4551

067.321 041 052 041 4552 ACA1 LXI H,AIO.LSI (M) = LAST SECTOR INDEX

067.324 072 046 041 4553 LDA AIO.SPG (A) = SECTORS PER GROUP

067.327 226 4554 SUB M (A) = SECTORS LEFT IN GROUP

067.330 312 354 067 4555 JZ ACA3 NONE LEFT

067.333 270 4556 CMP B

067.334 332 340 067 4557 JC ACA2 NOT TOO MANY IN GROUP FOR NEED

067.337 170 4558 MOV A,B DONT TAKE MORE THAN WE NEED

067.340 127 4559 ACA2 MOV D,A (D) = AMOUNT IN GROUP

067.341 206 4560 ADD M

067.342 167 4561 MOV M,A ADVANCE AIO.LSI  
067.343 172 4562 MOV A,D (A) = AMOUNT ALLOCATED FROM GROUP  
067.344 201 4563 ADD C  
067.345 117 4564 MOV C,A ADVANCE TOTAL ALLOCATED COUNT  
067.346 170 4565 MOV A,B  
067.347 222 4566 SUB D DECREMENT NEEDED COUNT  
067.350 107 4567 MOV B,A  
067.351 312 016 070 4568 JZ ACA? GOT ALL WE NEED  
4569  
4570 \* FINISHING THE GROUP WASN'T ENOUGH, TRY TO GET THE FOLLOWING  
4571 \* GROUP.  
4572  
067.354 056 051 4573 ACA3 MVI L,#AIO.LGN (HL) = #AIO.LGN  
000.041 4574 SET AIO.LGN/256  
000.000 4575 ERRNZ AIO.LSI/256-, MUST BE IN SAME PAGE  
067.356 126 4576 MOV D,M (D) = AIO.LGN  
067.357 024 4577 INR D (D) = FOLLOWING GROUP \*  
067.360 315 133 032 4578 CALL FFB FIND FREE BLOCK  
067.363 332 004 070 4579 JC ACA8 END OF MEDIA  
067.364 302 016 070 4580 JNZ ACA? COULDNT GET ONE CONTIGUOUS  
4581  
4582 \* GOT A BLOCK, CHAIN IT TO THE FILE  
4583  
067.371 315 354 031 4584 CALL CFF CHAIN FREE BLOCK TO FILE  
067.374 171 4585 MOV A,C  
067.375 247 4586 ANA A  
067.376 312 313 067 4587 JZ ACA0 AM STILL LOOKING FOR THE START  
070.001 303 321 067 4588 JMP ACA1 GO SOME MORE  
4589  
4590  
4591 \*\* END OF MEDIA EXIT, FLAG EOM IF COULDNT ALLOCATE ANY  
4592  
070.004 171 4593 ACA8 MOV A,C  
070.005 247 4594 ANA A  
070.006 302 016 070 4595 JNZ ACA9 GIVE HIM WHAT WE DID GET, ANYWAY...  
070.011 076 005 4596 MVI A,EC.EOM\*2+1  
070.013 062 112 041 4597 STA AIO.EOM SET EOM  
4598  
4599 \*\* NORMAL EXIT, (C) = AMOUNT ALLOCATED  
4600  
070.016 171 4601 ACA9 MOV A,C  
070.017 062 111 041 4602 STA AIO.CNT SET COUNT  
070.022 052 051 041 4603 LHLD AIO.LGN  
070.025 042 047 041 4604 SHLD AIO.CGN UPDATE CURRENT=LAST  
000.000 4605 ERRNZ AIO.LSI-AIO.LGN-1  
000.000 4606 ERRNZ AIO.CSI-AIO.CGN-1  
070.030 311 4607 RET  
070.031 4608 XTEXT BRF

4610X \*\* BRP - BAUD RATE PROMPT  
4611X \*  
4612X \* Prompt console for baud rate determining spaces at interrupt time  
4613X \* if current console is .8250.. Should be used before jumping to  
4614X \* ROMBOOT.  
4615X \*  
4616X \* ENTRY S.CDB = CONSOLE DEFINITION BYTE, describes current console.  
4617X \* EXIT NONE  
4618X \* USES NONE  
4619X \*

4620X  
070.031.365.....4621X BRP PUSH PSW  
070.032 377 007 4622X DB SYSCALL,.CLR00 CLEAR ANY TYPE-AHEAD  
070.034.315.136.031 4623X CALL \$TYPTX  
070.037 012 111 156 4624X DB NL,'Install a Bootable Disk in SY0:. Hit Return to Reboot'  
070.127.240.....4625X DB '/+2000  
070.130 377 001 4626X BRPO DB SYSCALL,SCIN WAIT FOR A NEWLINE  
070.132.376.012.....4627X CPI NL  
070.134 302 130 070 4628X JNZ BRPO  
070.137.072.343.040.4629X LDA S.CDB  
070.142 376 001 4630X CPI CDB.HB4  
070.144.392.216.070.4631X JNZ BRP1 IF .NOT. .8250  
4632X  
070.147.315.136.031 4633X CALL \$TYPTX  
070.152 012 124 171 4634X DB NL,'Type spaces to determine BAUD RATE',ENL  
4635X  
070.216 076 156 4636X BRP1 MVI A,AC.DLY  
070.220 315.053.000 4637X CALL ,DLY WAIT FOR CHARACTER TO BE OUTPUT  
070.223 257.....4638X XRA A  
070.224.323.351.....4639X OUT SC,ACE+UR.IER...CLEAR CONSOLE  
070.226 323 373.....4640X OUT SC.UART+USR  
070.230.361.....4641X POP PSW  
070.231 311.....4642X RET

4644.\*\* CPA - CHECK FOR PENDING ABORT,  
4645 \*  
4646 \* CPA IS CALLED WHEN A CONSOLE ABORT MAY BE PROCESSED,  
4647 \* IF THE SYSTEM IS READY, AND AND ABORT  
4648 \* IS PENDING, PROCESS IT.  
4649 \*  
4650 \* CPA SHOULD BE CALLED WITH INTERRUPTS DISABLED, SO THAT  
4651 \* ANOTHER INTERRUPT CHARACTER CANNOT OCCUR DURING CPA PROCESSING.  
4652 \* THIS GUARANTEES THAT A USER PROGRAM WILL BE (.INTERRUPTED.) WITH  
4653 \* THE PROGRAM COUNTER IN THE USER CODE, NEVER IN HDOS CODE.  
4654 \*  
4655 \* UPON ENTRY TO THE USER INTERRUPT ROUTINE,  
4656 \* ((SP)+0) = RETURN ADDRESS (IF USER WISHES TO RESUME NORMAL PROCESSING)  
4657 \* ((SP)+2) = USER PSW  
4658 \* ((SP)+4) = USER INTERRUPTED ADDRESS  
4660 \*  
4661 \* THE USER REGISTER VALUES FOR B,C,D,E,H, AND L ARE STILL  
4662 \* IN THE REGISTERS.

4663 \* ENTRY ((SP)+0) = RETURN ADDRESS.  
4664 \* ((SP)+2) = USER PSW  
4665 \* ((SP)+4) = USER INTERRUPTED ADDRESS.  
4666 \* EXIT TO \*RET\* IF NONE, OR DISABLED  
4667 \* TO PROCESSOR IF READY AND OK.  
4668 \* USES A,F  
4669 \*  
4670  
4671  
070.232 072.244.074 4672 CFA LDA SYSMODE  
070.235 247 4673 ANA A  
070.236 390 4674 RNZ IN SYSCALL MODE  
4675  
4676 \* WILL ALLOW PROCESSING  
4677  
070.237 072.334.040 4678 LDA S.CAADR+1 (A) = HIGH BYTE ABORT ADDRESS  
070.242 247 4679 ANA A  
070.243 310 4680 RZ NO ABORT PENDING  
4681  
4682 \* HAVE ABORT PROCESS IT  
4683  
070.244 345 4684 PUSH H  
070.245 052 333 040 4685 LHLD S.CAADR (HL) = ADDRESS FOR JUMP  
070.250 252 4686 XRA A  
070.251 062 334 040 4687 STA S.CAADR+1 CLEAR  
070.254 074 4688 INR A SET (A) <> 0  
070.255 343 4689 XTHL RESTORE (HL), SET PROCESSOR  
070.256 373 4690 EI  
070.257 311 4691 RET ENTER ROUTINE

4693 \*\* FCI -- FETCH CHANNEL INFORMATION.  
4694 \*  
4695 \* FCI COPIES THE CHANNEL INFORMATION FROM THE  
4696 \* CHANNEL TABLE INTO THE ACTIVE I/O TABLE.  
4697 \*  
4698 \* AIO.VEC = DRIVER ADDRESS  
4699 \* AIO.XXX.SETUP IF DIRECTORY DEVICE  
4700 \* AIO.CTA = ADDRESS OF CHANNEL AREA  
4701 \*  
4702 \* ENTRY (A) = CHANNEL NUMBER  
4703 \* EXIT (A) = CHANNEL STATUS BYTE  
4704 \* (HL) = ADDRESS OF FILE STATUS BYTE  
4705 \* 'C' SET IF ERROR  
4706 \* (A) = ERROR CODE  
4707 \* USES A,F,H,L  
4708  
4709  
070.260 052 352 040 4710 FCI LHLD S.CFWA (HL) = CHANNEL TABLE FWA  
000.000 4711 ERRNZ IOCCTR-1 CHANNEL 3770 IS FIRST IN LIST  
070.263 074 4712 INR A (A) = INDEX OF CHANNEL IN CHANTAB  
070.264 365 4713 PUSH PSW SAVE INDEX  
070.265 361 4714 FCI1 POP PSW (A) = INDEX  
070.266 247 4715 ANA A

FCI 14:05:45 14-MAY-80

070.267	312 311 070	4716	JZ	FCI2	GOT IT
070.272	075	4717	DCR	A	DECREMENT COUNT
070.273	365	4718	PUSH	PSW	SAVE INDEX
070.274	176	4719	MOV	A,M	
070.275	043	4720	INX	H	
070.276	146	4721	MOV	H,M	
070.277	157	4722	MOV	L,A	FOLLOW LINK
070.300	264	4723	DRA	H	
070.301	302 265 070	4724	JNZ	FCI1	MORE TO FOLLOW
		4725			
		4726	*		CHANNEL DOES NOT EXIST. FLAG ERROR
		4727			
070.304	361	4728	POP	PSW	
070.305	076. 016	4729	MVI	A,EC,ICN	ILLEGAL CHANNEL NUMBER
070.307	067	4730	STC		
070.310	311	4731	RET		
		4732			
		4733	*		GOT CHANNEL
		4734			
070.311	305	4735	FCI2	PUSH	B
070.312	325	4736	PUSH	D	SAVE REGISTERS
070.313	315 327. 071	4737	CALL	\$INDLB	A = UNIT CODE
070.316	022 000	4738	DW	IOC.UNI	/80.02.GC/
070.320	062 061 041	4739	STA	AIO.UNI	/80.02.GC/
070.323	043	4740	INX	H	INSURE UNIT SET UP FOR SEQ
070.324	043	4741	INX	H	/80.02.GC/
000.000		4742	ERRNZ	IOC.DDA-2	MOVE.PAST.LINK
070.325	042 116. 041	4743	SHLD	AIO.CHA	POINT TO DDA
070.330	043	4744	INX	H	SET BLOCK ADDRESS
070.331	043	4745	INX	H	
000.000		4746	ERRNZ	IOC.FLG-IOC.DDA-2	(HL) = #IOC.FLG
070.332	345	4747	PUSH	H	SAVE ADDRESS
070.333	176	4748	MOV	A,M	(A) = TYPE
070.334	346. 001	4749	ANI	FT.DD	SEE IF DIRECTORY TYPE
070.336	053	4750	DCX	H	
070.337	053	4751	DCX	H	
000.000		4752	ERRNZ	IOC.DDA-IOC.FLG+2	(HL) = #IOC.DDA
070.340	353	4753	XCHG		
070.341	041 041 041	4754	LXI	H,AIO.DDA	
070.344	001 003 000	4755	LXI	B,IOC.SQL	
070.347	312 354 070	4756	JZ	FCI3	IS SEQUENTIAL
070.352	016 021	4757	MVI	C,IOC.DIL	IS DIRECTORY
070.354	315 252 030	4758	CALL	\$MOVE	MOVE DATA
070.357	341	4759	POP	H	
070.360	176	4760	MOV	A,M	(A) = FLAG
070.361	321	4761	POP	D	
070.362	301	4762	POP	B	
070.363	311	4763	RET		

4765 \*\* GSP GET SYSTEM POINTER  
4766 \*  
4767 \* GET THE SYSTEM POINTER  
4768 \*  
4769 \*  
4770 \* ENTRY: NONE  
4771 \*  
4772 \* EXIT: HL = SYSTEM DEVICE UNIT POINTER  
4773 \*  
4774 \* USES: PSW,HL  
4775 \*  
4776  
070.364 052 354 040 4777 GSP LHLD S.DFWA  
070.367 325 4778 PUSH D  
070.370 021 012 000 4779 LXI D,DEV.UNIT HL = POINTER TO UNIT TABLE POINTER  
070.373 031 4780 DAD D  
070.374 321 4781 POP D  
070.375 257 4782 XRA A  
070.376 303 301 071 4783 JMP GUP

4785 \*\* LDD - LOAD DEVICE DRIVER.  
4786 \*  
4787 \* LDD IS CALLED TO PERFORM THE SUSPENDED LOAD OF A DEVICE DRIVER.  
4788 \*  
4789 \* IF SOME OVL CODE WISHES TO LOAD A DEVICE DRIVER, IT MUST  
4790 \* SUSPEND THE REQUEST, SINCE THE DEVICE DRIVER WILL OVERLAY THE  
4791 \* OVL CODE. AFTER THE OVL CODE EXITS, THE RESIDENT CODE WILL CALL  
4792 \* LDD TO PERFORM THE ACTUAL LOAD, OVER THE OVL.  
4793 \*  
4794 \* ENTRY DD.IOC = POINTER TO IOC.DDA  
4795 \* DD.LDA = LOAD ADDRESS  
4796 \* DD.LEN = LOAD LENGTH  
4797 \* DD.GRP = SECTOR INDEX ON SYSTEM DEVICE  
4798 \* DD.DTA = DEV.RES ADDRESS  
4799 \* DD.OPE = OPEN CODE (DC,OPR,DC,OPW,DC,OPU)  
4800 \* EXIT OVL CODE DESTROYED  
4801 \* USES NONE

4802  
4803  
071.001 315 054 031 4804 LDD CALL \$SAVALL SAVE REGS

4805  
4806 \* CLEAR OVL RESIDENT FLAG  
4807  
071.004 041 371 040 4808 LXI H.S.OVLFL  
071.007 176 4809 MOV A,M  
071.010 346 376 4810 ANI 3770-OVL.IN  
071.012 167 4811 MOV M,A CLEAR IN FLAG  
4812  
4813 \* LOAD OVERLAY  
4814  
071.013 .052.362.040. 4815 LHLD S.DBLLEN (HL) = LENGTH  
071.016 104 4816 MOV B:H  
071.017 115 4817 MOV C:L (BC) = LENGTH

071.020 052 360 040 4818 LHLD S.DDLDA (HL) = LOAD ADDRESS  
071.023 345 4819 PUSH H SAVE FOR LATER  
071.024 353 4820 XCHG  
071.025 041 066 077 4821 LXI H:SECSCR+511 FORCE NEW DISK READ RIGHT AWAY  
4822  
4823 \* LOAD BINARY  
4824  
071.030 315 101 071 4825 LDD2 CALL LDD8 FIND NEXT BYTE  
071.033 176 4826 MOV A:M (A) = NEXT BYTE  
071.034 022 4827 STAX D COPY  
071.035 023 4828 INX D  
071.036 013 4829 DCX B  
071.037 170 4830 MOV A:B  
071.040 261 4831 ORA C  
071.041 302 030 071 4832 JNZ LDD2 MORE TO GO  
4833  
4834 \* CODE ALL LOADED. RELOCATE IT  
4835  
071.044 301 4836 POP B (BC) = REL FACTOR  
071.045 .005 4837 ICR B  
071.046 005 4838 ICR B  
000.000 4839 ERRNZ DVD,ENT-2000A... ASSUME DRIVER ENTRY = 2000A  
071.047 315 101 071 4840 LDD3 CALL LDD8  
071.052 136 4841 MOV E:M  
071.053 315 101 071 4842 CALL LDD8  
071.056 126 4843 MOV D:M (DE) = REL ADDRESS OF WORD TO RELOCATE  
071.057 172 4844 MOV A:D  
071.060 263 4845 ORA E  
071.061 312 323 032 4846 JZ LDD4 ALL DONE  
071.064 353 4847 XCHG (HL) = REL ADDRESS OF WORD TO RELOCATE  
071.065 011 4848 ADD B (HL) = ABS ADDRESS OF WORD TO RELOCATE  
071.066 176 4849 MOV A:M  
071.067 201 4850 ADD C  
071.070 167 4851 MOV M:A  
071.071 043 4852 INX H  
071.072 176 4853 MOV A:M  
071.073 210 4854 ADC B  
071.074 167 4855 MOV M:A  
071.075 353 4856 XCHG RESTORE (HL)  
071.076 303.047.071 4857 JMP LDD3  
4858  
4859 \* SETUP ENTRY ADDRESSES IN TABLES  
4860  
032.323 4861 LDD4 EQU 32323A USE WHATS IN ROM  
4862  
032.361 4863 FCNL EQU 32361A USE FCNL IN ROM  
4864

4866 \*\* LDD8 - READ A BYTE FROM THE FILE.  
4867 \*  
4868 \* NOTE THAT S.IDDGRP CONTAINS THE GROUP NUMBER FOR THE CURRENT SECTOR.  
4869 \* SINCE LDD READS 2 SECTORS AT A TIME, AND 2 SECTORS  
4870 \* MAKES ONE GROUP, LDD8 FOLLOWS THE GROUP CHAIN TO THE NEXT  
4871 \* GROUP, UPON INITIAL ENTRY OF LDD, S.IDDGRP = THE FIRST  
4872 \* GROUP OF THE DRIVER, WHICH CONTAINS OPTION  
4873 \* INFO AND IS NOT LOADED.  
4874 \*  
4875 \* ENTRY (HL) = SECSCR POINTER OF CURRENT BYTE  
4876 \* S.IDDSEC = SECTOR NUMBER OF CURRENT SECTOR  
4877 \* EXIT (HL) = ADDRESS OF NEXT BYTE  
4878 \* USES A,F,H,L  
4879  
000.000 4880 ERRNZ HOS,SPG-2 REQUIRE 2 SECTORS PER GROUP  
000.000 4881 ERRNZ DVD,ENT-2000A FIRST 2 SECTORS ARE IGNORED  
4882  
071.101 054 4883 LDD8 INR L POINT TO NEXT BYTE  
071.102 300 4884 RNZ GOT IT  
071.103 044 4885 INR H MAYBE IN NEXT GROUP  
071.104 345 4886 PUSH H  
071.105 041 067 076 4887 LXI H,SECSCR+256  
071.110 174 4888 MOV A,H  
071.111 341 4889 POP H  
071.112 274 4890 \* MVI A,SECSCR+256/256  
071.113 310 4891 CMP H  
4892 RE OK, IN SECOND SECTOR NOW  
4893  
071.114 305 4894 \* MUST READ ANOTHER  
4895  
071.115 325 4896 PUSH B  
071.116 021 067 075 4897 PUSH D  
071.116 021 067 075 4898 LXI D,SECSCR (DE) = ADDRESS  
071.121 001 000 002 4899 LXI B,512 (BC) = COUNT  
071.124 072 364 040 4900 LDA S.IDDGRP (A) = GROUP OF DEVICE DRIVER  
071.127 041 000 024 4901 LXI H,S,GRTO  
071.132 157 4902 MOV L,A (HL) = GRT POINTER FOR NEXT ONE  
071.133 176 4903 MOV A,M (A) = GROUP FOR NEXT ONE  
071.134 062 364 040 4904 STA S.IDDGRP  
000.000 4905 ERRNZ HOS,SPG-2 CODE ASSUMES 2 SECTORS PER GROUP  
071.137 157 4906 MOV L,A (L) = GROUP  
071.140 143 4907 MOV H,E (H) = 0  
071.141 051 4908 DAD H (HL) = SECTOR ADDRESS  
071.142 325 4909 PUSH D SAVE \$SECSCR  
071.143 315 275 031 4910 CALL S,READ READ IT  
071.146 341 4911 POP H (HL) = \$SECSCR  
071.147 321 4912 POP D RESTORE (DE) AND (BC)  
071.150 301 4913 POP B  
071.151 311 4914 RET

4916 \*\* OTI - OVERLAY TABLE INDEX  
4917 \*  
4918 \* OTI COMPUTES THE OVERLAY TABLE INDEX ADDRESS BASED ON THE OVERLAY  
4919 \* INDEX, (AS DEFINED IN LOADO,) AND THE OFFSET INTO THE TABLE ENTRY.  
4920 \*  
4921 \* USE: CALL OTI  
4922 \* DW offset  
4923 \*  
4924 \*  
4925 \* ENTRY: (A) = OVERLAY INDEX  
4926 \*  
4927 \* EXIT: (HL) = ADDRESS OF THE SPECIFIED TABLE ENTRY  
4928 \*  
4929 \* USES: (PSW), (HL)  
4930 \*  
4931  
071.152 207 4932 OTI ADD A (A) = 2\*(A)  
071.153 207 4933 ADD A (A) = 4\*(A)  
071.154 207 4934 ADD A (A) = 8\*(A)  
000.000 4935 ERRNZ OVL,ENS-8  
071.155 041 347 072 4936 LXI H,OVLTAB TABLE FIRST WORD ADDRESS  
071.160 315 101.030 4937 CALL \$DADA, (HL) = TABLE ENTRY ADDRESS  
071.163 353 4938 XCHG  
071.164 343 4939 XTHL SAVE (DE)  
071.165 325 4940 PUSH D SAVE TABLE ENTRY ADDRESS  
071.166 136 4941 MOV E,M  
071.167 043 4942 INX H  
071.170 126 4943 MOV D,M (DE) = TABLE ENTRY OFFSET  
071.171 043 4944 INX H (HL) = RETURN ADDRESS  
071.172 343 4945 XTHL (HL) = TABLE ENTRY ADDRESS  
071.173 031 4946 DAD D (HL) = TABLE ENTRY OFFSET ADDRESS  
071.174 321 4947 POP D (DE) = RETURN ADDRESS  
071.175 353 4948 XCHG (HL) = OLD (DE)  
071.176 343 4949 XTHL (HL) = OLD (DE)  
071.177 353 4950 XCHG (HL) = TAB. ENTRY OFFSET ADDR, (DE) = OLD (DE)  
071.200 311 4951 RET

4953 \*\* RRC - REMOVE REGULAR CHARACTER.  
4954 \*  
4955 \* RRC REMOVES THE LAST CHARACTER IN THE INPUT CIRCULAR BUFFER,  
4956 \* IF IT IS NOT A NEW-LINE CHARACTER (00).  
4957 \*  
4958 \* ENTRY NONE  
4959 \* EXIT 'Z' SET IF NO CHARACTERS, OR LAST ONE IS '00'  
4960 \* 'Z' CLEAR IF GOT CHARACTER  
4961 \* (A) = CHARACTER  
4962 \* USES A,F,H,L  
4963 \*  
4964 \*  
071.201 052 252 074 4965 RRC LHLD SCIIN  
071.204 072 254 074 4966 LDA SCIOUT  
071.207 275 4967 CMP L  
071.210 310 4968 RE NONE

RRC 14:06:04 16-MAY-80

071.211 072 256 074 4969	LDA	SCIFWA
071.214 275 4970	CMP	L
071.215 302 223 071 4971	JNE	RRC1
071.220 052 260 074 4972	LHLD	SCILWA
071.223 053 4973 RRC1	DCX	H
071.224 176 4974	MOV	A,M
071.225 376 012 4975	CPI	NL
071.227 310 4976	RE	
071.230 376 004 4977	CPI	CTLD
071.232 310 4978	RE	
071.233 042 252 074 4979	SHLD	SCIIN
071.236 311 4980	RET	

4982 \*\* SCI - STORE CHANNEL INFORMATION.

4983 \*  
4984 \* SCI SAVES THE ACTIVE CHANNEL INFORMATION BACK  
4985 \* INTO THE CHANNEL BLOCK.4986 \*  
4987 \* ENTRY NONE  
4988 \* EXIT NONE  
4989 \* USES NONE  
4990

4991

071.237 315 054 031 4992	SCI	CALL	\$SAVALL
071.242 052 116 041 4993	LHLD	AIO.CHA	
000.000 4994	ERRNZ	IOC.FLG-IOC.DDA-2	
071.245 043 4995	INX	H	
071.246 043 4996	INX	H	(HL) = IOB.FLG ADDRESS
071.247 .001 .010 .000 4997	LXI	B,IOC.DRL	(BC) = LEN
071.252 021 043 041 4998	LXI	D,AIO.FLG	
071.255 .315 .252 .030 .4999	CALL	\$MOVE	MOVE DATA
071.260 303 047 031 5000	JMP	\$RSTALL	RESTORE ALL REGS

5002 \*\* SDD - STAND-IN DEVICE DRIVER.

5003 \*  
5004 \* SDD IS SETUP AS THE DEVICE DRIVER ADDRESS FOR DRIVERS WHICH  
5005 \* ARE NOT IN MEMORY. IF THE REQUEST IS AN OPEN, POSTPONE IT  
5006 \* UNTIL 'LDR' LOADS THE OVERLAY. OTHERWISE, IS A FATAL.  
5007 \* SYSTEM ERROR.5008 \*  
5009 \* ENTRY (A) = CODE  
5010 \* EXIT NONE  
5011 \* USES A,F

5012

5013

071.263 376 003 5014	SDD	CPI	DC.OPR
071.265 334 210 063 5015	CC	FATSER	
071.270 376 006 5016	CPI	DC.OPU+1	
071.272 324 210 063 5017	CNC	FATSER	
071.275 062 370 040 5018	STA	S.DDOFC	SET CODE

HRS - RESIDENT HRS CODE .....  
RESIDENT SUBROUTINES.....

HEATH H8ASH V1.4 01/20/78 PAGE 111

SDD

14:06:08 16-MAY-80

071.300 311 5019 RET

071.301 5022 XTEXT TBRA

5024X \*\* \$TBRA - BRANCH RELATIVE THOUGH TABLE.  
5025X \*  
5026X \* \$TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE  
5027X \* JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE  
5028X \* ADDRESS OF THE BYTE, YIELDING THE PROCESSOR ADDRESS.  
5029X \*  
5030X \* CALL \$TBRA  
5031X \* DB LAB1-\* INDEX = 0 FOR LAB1  
5032X \* DB LAB2-\* INDEX = 1 FOR LAB2  
5033X \* DB LABN-\* INDEX = N-1 FOR LABN  
5034X \*  
5035X \* ENTRY (A) = INDEX  
5036X \* (RET) = TABLE FWA  
5037X \* EXIT TO COMPUTED ADDRESS  
5038X \* USES F,H,L  
5039X  
5040X

031.076 5041X \$TBRA EQU 31076A IN H17 ROM  
071.301 5042 XTEXT GUP

5044X \*\* GUP - GET UNIT POINTER  
5045X \*  
5046X \* GET THE UNIT SPECIFIC DATA POINTER FOR THE SPECIFIED UNIT  
5047X \*  
5048X \*  
5049X \* ENTRY: A = UNIT NUMBER  
5050X \* HL = ADDRESS OF UNIT TABLE  
5051X \*  
5052X \* EXIT: HL = ADDRESS OF TABLE ENTRY FOR SPECIFIED UNIT  
5053X \*  
5054X \* USES: PSW,HL  
5055X \*

5056X  
071.301 325 5057X GUP PUSH D  
071.302 365 5058X PUSH PSW  
071.303 315.211.030 5059X CALL \$HLIHL HL = POINTER TO UNIT TABLE  
071.306 361 5060X POP PSW SAVE A  
071.307 345 5061X PUSH H  
071.310 021 007 000 5062X LXI D,UNT.SIZ  
071.313 315 007 031 5063X CALL \$MUB6 HL = HL \* UNT.SIZ  
071.316 321 5064X POP D  
071.317 031 5065X PDP D  
071.320 321 5066X POP D  
071.321 311 5067X RET  
071.322 5068 XTEXT HLIHL

\$HLIHL.....14:06:22...14-MAY-80.

5070X \*\* \$HLIHL - LOAD HL INDIRECT THROUGH HL.

5071X \*  
5072X \* (HL) = ((HL))  
5073X \*  
5074X \* ENTRY NONE  
5075X \* EXIT NONE  
5076X \* USES A,H,L  
5077X

030.211 5078X \$HLIHL EQU 30211A IN H17 ROM  
071.322 5079 XTEXT ILDEHL

5081X \*\* ILDEHL - INDEXED LOAD OF DE FROM HL

5082X \*  
5083X \* 'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS  
5084X \* INCREMENTED BY TWO.  
5085X \*

5086X \* ENTRY: ..HL..... = ADDRESS OF FULL WORD VALUE

5087X \*  
5088X \* EXIT: ..DE..... = (HL)  
5089X \* HL = HL + 2  
5090X \*

5091X \* USES: DE

5092X \*

5093X \*

071.322 136 5094X ILDEHL MOV E,M  
071.323 043 5095X INX H  
071.324 126 5096X MOV D,M  
071.325 043 5097X INX H  
071.326 311 5098X RET  
071.327 5099 XTEXT INDL

5101X \*\* \$INDL - INDEXED LOAD.

5102X \*

5103X \* \$INDL LOADS DE WITH THE TWO BYTES AT ..(HL)+DISPLACEMENT.

5104X \*  
5105X \* THIS ACTS AS AN INDEXED FULL WORD LOAD.

5106X \*

5107X \* (DE) = ..( (HL) + DISPLACEMENT ..)

5108X \*

5109X \* ENTRY ..((RET)) = DISPLACEMENT (FULL WORD)

5110X \* (HL) = TABLE ADDRESS

5111X \* EXIT TO (RET+2)

5112X \* USES A,F,D,E

5113X \*

5114X \*

030.234 5115X \$INDL EQU 30234A IN H17 ROM  
071.327 5116 XTEXT INDXX

5118X \*\* \$INDLB - INDEXED LOAD BYTE  
5119X \*  
5120X \* BYTE INDEXED LOAD PRIMITIVE  
5121X \*  
5122X \* ENTRY: HL = BASE ADDRESS  
5123X \* (RET) = FULL WORD RELOCATION  
5124X \*  
5125X \* EXIT: A = ( HL + (RET) )  
5126X \*  
5127X \* USES: A  
5128X \*  
5129X  
071.327 353 5130X \$INDLB XCHG DE = BASE  
071.330 343 5131X XTHL SAVE .DE.  
071.331 325 5132X PUSH D SAVE BASE  
071.332 305 5133X PUSH B SAVE .BC.  
5134X  
071.333 116 5135X MOV C,M  
071.334 043 5136X INX H  
071.335 106 5137X MOV B,M BC = OFFSET  
071.336 043 5138X INX H HL = .RET.  
5139X  
071.337 353 5140X XCHG HL = BASE  
071.340 011 5141X DAD B HL = BASE + OFFSET  
071.341 176 5142X MOV A,M A = ('BASE + OFFSET')  
071.342 353 5143X XCHG HL = .RET.  
5144X  
071.343 301 5145X POP B RESTORE .BC.  
071.344 321 5146X POP D RESTORE BASE  
071.345 343 5147X XTHL HL = .DE. ; (SP) = .RET.  
071.346 353 5148X XCHG DE = .DE. ; HL = BASE  
071.347 311 5149X RET

5151X \*\* \$INDS - INDEXED STORE  
5152X \*  
5153X \* INDEXED STORE PRIMITIVE.  
5154X \*  
5155X \* ENTRY: HL = BASE ADDRESS  
5156X \* DE = VALUE TO STORE  
5157X \*  
5158X \* EXIT: ( HL + (RET) ) = DE  
5159X \*  
5160X \* USES: NONE  
5161X \*  
5162X  
071.350 315 032 072 5163X \$INDS CALL XCHGBC  
071.353 343 5164X XTHL SAVE .BC.  
071.354 325 5165X PUSH D  
071.355 315 322 071 5166X CALL ILDEHL DE = OFFSET  
071.360 315 032 072 5167X CALL XCHGBC BC = .RET.  
071.363 353 5168X XCHG DE = BASE ; HL = OFFSET  
071.364 031 5169X DAD D HL = BASE + OFFSET  
071.365 353 5170X XCHG

071.366 343 5171X XTHL SAVE BASE  
071.367 353 5172X XCHG DE = VALUE  
071.370 315 025 072 5173X CALL ISDEHL  
071.373 341 5174X POP H HL = BASE  
071.374 315 032 072 5175X CALL XCHGBC  
071.377 343 5176X XTHL RESTORE .BC.  
072.000 315 032 072 5177X CALL XCHGBC  
072.003 311 5178X RET.

5180X \*\* \$INDSB - INDEXED BYTE STORE  
5181X \*  
5182X \* INDEXED BYTE STORE.  
5183X \*  
5184X \* ENTRY: A = VALUE TO STORE  
5185X \* HL = BASE ADDRESS  
5186X \* (RET) = OFFSET  
5187X \*  
5188X \* EXIT: NONE  
5189X \*  
5190X \* USES: PSW  
5191X \*  
5192X  
072.004 353 5193X \$INDSB XCHG DE = BASE  
072.005 343 5194X XTHL SAVE .DE.  
072.006 325 5195X PUSH D SAVE .DE.  
072.007 305 5196X PUSH B SAVE .BC.  
5197X  
072.010 116 5198X MOV C,M  
072.011 043 5199X INX H  
072.012 106 5200X MOV B,M BC = OFFSET  
072.013 043 5201X INX H HL = .RET.  
5202X  
072.014 353 5203X XCHG HL = BASE  
072.015 011 5204X DAD B HL = BASE + OFFSET  
072.016 167 5205X MOV M,A (.BASE + OFFSET) = A  
072.017 353 5206X XCHG  
5207X  
072.020 301 5208X POP B RESTORE .BC.  
072.021 321 5209X POP D RESTORE .DE.  
072.022 343 5210X XTHL HL = .DE. ; (SP) = .RET.  
072.023 353 5211X XCHG DE = .DE. ; HL = BASE.  
072.024 311 5212X RET  
072.025 5213 XTEXT ISDEHL

5215X \*\* ISIDEHL - INDEXED STORE OF DE AT HL  
5216X \*  
5217X \* STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'  
5218X \* BY 2.  
5219X \*  
5220X \* ENTRY: DE = VALUE  
5221X \* HL = ADDRESS OF VALUE  
5222X \*  
5223X \* EXIT: (HL) = DE  
5224X \* HL = HL + 2  
5225X \*  
5226X \* USES: HL  
5227X \*  
5228X  
072.025 163 5229X ISIDEHL MOV M,E  
072.026 043 5230X INX H  
072.027 162 5231X MOV M,D  
072.030 043 5232X INX H  
072.031 311 5233X RET  
072.032 5234 XTEXT MOVE

5236X \*\* \$MOVE - MOVE DATA  
5237X \*  
5238X \* \$MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.  
5239X \* IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM  
FIRST TO LAST.  
5240X \*  
5241X \*  
5242X \* IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM  
LAST TO FIRST.  
5243X \*  
5244X \*  
5245X \* THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.  
5246X \*  
5247X \* ENTRY (BC) = COUNT  
5248X \* (DE) = FROM  
5249X \* (HL) = TO  
5250X \* EXIT MOVED  
5251X \* (DE) = ADDRESS OF NEXT FROM BYTE  
5252X \* (HL) = ADDRESS OF NEXT \*TO\* BYTE  
5253X \* 'C' CLEAR  
5254X \* USES ALL  
5255X  
5256X  
030.252 5257X \$MOVE EQU 30252A IN H17 ROM  
072.032 5258 XTEXT DADA2

HIDS - RESIDENT HIDS CODE  
COMMON DECKS..... \$DADA..... HEATH H8ASM V1.4 01/26/78 PAGE 117  
14:06:53 16-MAY-80

5260X \*\* \$DADA. - ADD (0,A) TO (H,L)

5261X \*

5262X \* ENTRY NONE

5263X \* EXIT (HL) = (HL) + (0A)

5264X \* USES A,F,H,L

5265X

5266X

030.101 5267X \$DADA. EQU 30101A IN H17 ROM  
072.032 5268 XTEXT SAVALL

5270X \*\* \$RSTALL - RESTORE ALL REGISTERS.

5271X \*

5272X \* \$RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND

5273X \*. RETURNS TO THE PREVIOUS CALLER.

5274X \*

5275X \* ENTRY (SP) = PSW

5276X \* (SP+2) = BC

5277X \* (SP+4) = DE

5278X \* (SP+6) = HL

5279X \* (SP+8) = RET

5280X \* EXIT TO \*RET\*, REGISTERS RESTORED

5281X \* USES ALL

5282X

5283X

031.047 5284X \$RSTALL EQU 31047A IN H17 ROM

5286X \*\* \$SAVALL - SAVE ALL REGISTERS ON STACK.

5287X \*

5288X \* \$SAVALL SAVES ALL THE REGISTERS ON THE STACK.

5289X \*

5290X \* ENTRY NONE

5291X \* EXIT (SP) = PSW

5292X \* (SP+2) = BC

5293X \* (SP+4) = DE

5294X \* (SP+6) = HL

5295X \* USES H,L

5296X

5297X

031.054 5298X \$SAVALL EQU 31054A IN H17 ROM  
072.032 5299 XTEXT COMP

5301X \*\* \$COMP - COMPARE TWO CHARACTER STRINGS.

5302X \*

5303X \* \$COMP COMPARES TWO BYTE STRINGS.

5304X \*

5305X \* ENTRY (C) = COMPARE COUNT

5306X \* (DE) = FWA OF STRING #1

5307X \* (HL) = FWA OF STRING #2

5308X \* EXIT 'Z' CLEAR, IS MIS-MATCH

\$COMP 14:07:03 16-MAY-80

5309X \* (C) = LENGTH REMAINING  
5310X \* (DE) = ADDRESS OF MISMATCH IN STRING#1  
5311X \* (HL) = ADDRESS OF MISMATCH IN STRING #2  
5312X \* 'C' SET, HAVE MATCH  
5313X \* (C) = 0  
5314X \* (DE) = (DE) + (OC)  
5315X \* (HL) = (HL) + (OD)  
5316X \* USES A,F,C,D,E,H,L  
5317X

030.060 5318X 03060A IN H17 ROM  
072.032 5319X \$COMP EQU XTEXT XCHGBC

5322X \*\* XCHGBC - XCHG BC  
5323X \*  
5324X \* EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.  
5325X \*  
5326X \* ENTRY: BC = ORIGINAL BC  
5327X \* HL = ORIGINAL HL  
5328X \*  
5329X \* EXIT: BC = ORIGINAL HL  
5330X \* HL = ORIGINAL BC  
5331X \*  
5332X \* USES: BC,HL  
5333X \*

072.032 365 5334X  
072.033 170 5335X XCHGBC PUSH PSW  
072.034 104 5336X MOV A,B  
072.035 147 5337X MOV B,H  
072.036 171 5338X MOV H,A  
072.037 115 5339X MOV A,C  
072.040 157 5340X MOV C,L  
072.041 361 5341X MOV L,A  
072.042 311 5342X POP PSW  
072.043 5343X RET  
5344 XTEXT ZERO

5346X \*\* \$ZERO - ZERO MEMORY  
5347X \*  
5348X \* \$ZERO ZEROS A BLOCK OF MEMORY.  
5349X \*  
5350X \* ENTRY (HL) = ADDRESS  
5351X \* (B) = COUNT  
5352X \* EXIT (A) = 0  
5353X \* USES A,B,F,HL  
5354X  
5355X

031.212 5356X \$ZERO 31212A IN H17 ROM  
072.043 5357 XTEXT WER

HDOS - RESIDENT HDOS CODE  
COMMON DECKS

\$WER

HEATH HBASM V1.4 01/26/78  
14:07:13 16-MAY-80

PAGE 119

5359X \*\* \$WER - WRITE ENABLE RAM.

5360X \*  
5361X \* \$WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S  
RAM AREA.

5362X \*  
5363X \*  
5364X \* ENTRY NONE  
5365X \* EXIT NONE  
5366X \* USES NONE

5367X

5368X

031.241 5369X \$WER EQU 31241A IN H17 ROM

5371X \*\* \$WDR - WRITE DISABLE RAM.

5372X \*  
5373X \* \$WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S  
RAM AREA.

5374X \*  
5375X \*  
5376X \* ENTRY NONE  
5377X \* EXIT NONE  
5378X \* USES NONE

5379X

5380X

031.222 5381X \$WDR EQU 31222A IN H17 ROM  
072.043 5382 XTEXT CHL

5384X \*\* \$CHL - COMPLEMENT (HL).

5385X \*  
5386X \* (HL) = - (HL) TWO'S COMPLEMENT

5387X \*  
5388X \* ENTRY NONE  
5389X \* EXIT NONE  
5390X \* USES A,F,H,L

5391X

5392X

030.224 5393X \$CHL EQU 30224A IN H17 ROM

5396 \*\* THE FOLLOWING ROUTINES ARE REPLACEMENTS FOR THE H17 ROM CODE.

5398 \*\* DSKERR - DISK ERROR.

5399 \*

5400

072.043 064 5401 DSKERR INR M COUNT ERROR  
000.001 5402 IF DEBUG  
5403 MVI A,'!' \*\* DEBUG \*\*  
5404 OUT 3720 FLAG TO CONSOLE  
5405 ENDIF  
072.044 .052 262.040 5406 LHLD D,SECNT  
072.047 043 5407 INX H  
072.050 042 262.040 5408 SHLD D,SECNT  
072.053 311 5409 RET

5411 \*\* CDE - COUNT DISK ERRORS.

5412 \*

5413 \*

CDE IS CALLED WHEN A DISK SOFT ERROR OCCURS. IF THERE HAVE  
OCCURED 10 SOFT ERRORS FOR THIS OPERATION, THEN A HARD ERROR

IS FLAGGED.

5416 \*

5417 \* ENTRY NONE

5418 \* EXIT 'C' SET IF HARD ERROR

5419 \* INTERRUPTS DISABLED

5420 \* USES A,F,H,L

5421

000.012 5422 ERPTCNT EQU 10 ERROR REPEAT COUNT (DEFINED IN ROMDD)

5423

072.054 373 5424 RAMCDE EI  
072.055 .315 213.040 5425 CALL R,STZ SEEK TRACK ZERO  
072.060 315 166 040 5426 CALL D,SDT SEEK DESIRED TRACK  
072.063 .247 5427 ANA A CLEAR CARRY  
072.064 .052 262 040 5428 LHLD D,SECNT  
072.067 .043 5429 INX H  
072.070 .042 262 040 5430 SHLD D,SECNT INCREMENT COUNT  
072.073 .041 264.040 5431 LXI H,P,DECNT (HL) = #OPERATION ERROR COUNT  
072.076 065 5432 DCR M  
072.077 .360 5433 RP NOT TOO MANY  
072.100 053 5434 DCX H  
072.101 ,053 5435 DCX H (HL) = #D,SECNT  
5436  
000.000 5437 ERRNZ D,SECNT-D,DECNT+2  
072.102 076 366 5438 MVI A,-ERPTCNT  
072.104 .206 5439 ADD M REMOVE SOFT COUNT  
072.105 167 5440 MOV M,A  
000.000 5441 ERRNZ D,SECNT-D,HECNT-1  
072.106 053 5442 DCX H (HL) = #D,HECNT  
072.107 .064 5443 INR M COUNT HARD ERROR  
072.110 .052 240 040 5444 LHLD D,TT  
072.113 .042 126 040 5445 SHLD D,ERTS RECORD ERROR TRACK AND SECTOR

072.116 067 5446 STC  
072.117 311 5447 RET EXIT WITH 'C' SET

5449 \*\* R.WRITE - PERFORM DISK WRITE.  
5450 \*  
5451 \* PARTIALLY REPLACES ROM CODE IN H17ROM (ROMDD OR H17ROM LISTING)  
5452 \*  
5453 \* SEE LISTING FOR DETAILS.  
5454  
5455

034.370 5456 WRITE1 EQU 34370A  
035.132 5457 WRITE8 EQU 35132A  
5458  
072.120 5459 RAMWRI EQU \*  
072.120 5460 R.WRITE EQU \*  
072.120 345 5461 PUSH H SAVE BLOCK NUMBER  
072.121 315 205 040 5462 CALL D.SDP SET DEVICE PARAMETERS  
072.124 052 275 040 5463 LHLD D.OPW  
072.127 043 5464 INX H  
072.130 042 275 040 5465 SHLD D.OPW COUNT OPERATION  
072.133 333 177 5466 IN DP.DC SEE IF DISK WRITE PROTECTED  
072.135 346 004 5467 ANI DF.WP  
072.137 067 5468 STC  
072.140 076 025 5469 MVI A,EC.WP  
072.142 302 132 035 5470 JNZ WRITE8 DISK IS WRITE PROTECTED

5471  
5472 \* READY TO WRITE SECTOR  
5473 \*  
5474 \* (BC) = COUNT  
5475 \* (DE) = ADDRESS  
5476 \* ((SP)) = SECTOR NUMBER  
5477

072.145 041 377 000 5478 LXI H,3770  
072.150 011 5479 DAD B  
072.151 104 5480 MOV B,H (B) = # OF SECTORS TO WRITE  
072.152 170 5481 MOV A,B  
072.153 247 5482 ANA A  
072.154 312 132 035 5483 JZ WRITE8 NONE TO WRITE  
072.157 303 370 034 5484 JMP WRITE1 RE-JOIN ROM

5486 \*\* R.SDP - SETUP DEVICE PARAMETERS  
5487 \*  
5488 \* SDP SETS UP ARGUMENTS FOR THE SPECIFIC UNIT.  
5489 \*  
5490 \* D.BUCTL = MOTOR ON  
5491 \* D.TRKPT = ADDRESS OF DEVICE TRACK NUMBER  
5492 \*  
5493 \*  
5494 \* Modified: /79.10.GC/ by G. Chandler  
5495 \* Enable the access of "SY2!"

5496 \*  
5497 \* ENTRY: AIO.UNI = UNIT NUMBER  
5498 \*  
5499 \* EXIT: (HL) = (D,TRKPT)  
5500 \*  
5501 \* USES: (PSW), (HL)  
5502 \*  
5503  
036.073 5504 R.SDP. EQU 036073A THE GOOD ROM CODE ENTRY POINT  
5505  
072.162 5506 RAMSDP EQU \*  
072.162 5507 R.SDP EQU \*  
5508  
072.162 076 012 5509 MVI A,ERPTCNT  
072.164 062 264 040 5510 STA D,OECNT SET MAX ERROR COUNT FOR THE OPERATION  
072.167 072 061 041 5511 LDA AIO,UNI  
072.172 365 5512 PUSH PSW  
072.173 376 002 5513 CPI 1+1  
5514  
072.175 332.073.036 5515 JC R.SDP. UNIT 0 OR 1  
000.000 5516 ERRNZ DF.DS0-2  
000.000 5517 ERRNZ DF.DS1-4  
5518  
072.200 076 003 5519 MVI A,3 UNIT 2  
000.000 5520 ERRNZ DF.DS2-8  
072.202 303.073.036 5521 JMP R.SDP.

5525 \*\*\* TTDVD - RESIDENT TT DEVICE DRIVER.\*  
5526  
5527  
072.205 315 076 031 5528 TTDVD EQU \*  
072.205 315 076 031 5529 CALL \$TBRA  
072.210 .037 5530 DB TTREAD-\* READ  
072.211 112 5531 DB TTWRITE-\* WRITE  
072.212 .007 5532 DB TTART-\* READR  
072.213 012 5533 DB TTOPEN-\* OPENR  
072.214 .011 5534 DB TTOPEN-\* OPENW  
072.215 004 5535 DB TTART-\* OPENU  
072.216 .024 5536 DB TTNOP-\* CLOSE  
072.217 023 5537 DB TTNOP-\* ABORT  
072.220 .001 5538 DB TTART-\* MOUNT  
5539  
072.221 .074 .027 5540 TTART MVJ A,EC,DDA DEVICE DRIVER ABORT  
072.223 067 5541 STC  
072.224 .311 5542 RET  
5543  
072.225 .072 .332 .040 5544 TTOPEN LDA S,CONFL  
072.230 346 376 5545 ANI 377Q-CO,FLG CLEAR CTL-0  
072.232 .062 .332 .040 5546 STA S,CONFL  
5547 \* LDA S,CSLMII /79.02.04.GC/  
5548 \* ANI CSL,ECH /79.02.04.GC/  
5549 \* ORI CSL,WRP SET WRAP MODE /79.02.04.GC/  
5550 \* MVJ A,CSL,WRP /79.02.GC/  
5551 \* STA S,CSLMD SET WRAP MODE /79.04.GC/  
072.235 .257 5552 XRA A  
072.236 062 322 072 5553 STA EOFFLG CLEAR EOF ON INPUT FLAG  
072.241 .311 5554 RET  
5555  
072.242 .247 5556 TTNOP ANA A  
072.243 .311 5557 RET DO NOTHING

5559 \*\* TTREAD - READ  
5560 \*  
5561  
072.244 .022 5562 TTR2 STAX D STORE CHAR  
072.245 023 5563 INX D  
072.246 .013 5564 DCX B  
5565  
072.247 .001 5566 TTREAD EQU \*  
072.247 072 322 072 5567 LDA EOFFLG  
072.252 .037 5568 RAR  
072.253 330 5569 RC IS EOF  
5570  
072.254 170 5571 MOV A,B  
072.255 .261 5572 ORA C  
072.256 310 5573 RZ ALL DONE  
5574  
5575 \* TAKE A CHAR  
5576  
072.257 072 334 040 5577 TTR1 LDA S,CAADRT1  
072.262 .247 5578 ANA A

072.263 302 300 072 5579 JNZ TTREOF  
072.266 377 001 5580 DB SYSCALL,,SCIN  
072.270 332 257 072 5581 JC TTR1 WAIT TILL GOT IT  
072.273 376 004 5582 CPI 04  
072.275 302 244 072 5583 JNE TTR2 NOT CTL-D  
5584  
5585 \* HAVE EOF CHARACTER. FILL THIS SECTOR WITH 0's  
5586  
072.300 076 003 5587 TTREOF MVI A,EC.EOF\*2+1  
072.302 .062.322.072 5588 STA EOFFLG FLAG.EOF  
072.305 257 5589 TTR4 XRA A  
072.306 .022 5590 STAX D STORE 0  
072.307 023 5591 INX D  
072.310 .013 5592 DCX B  
072.311 171 5593 MOV A,C  
072.312 261 5594 ORA C  
072.313 302 305 072 5595 JNZ TTR4  
072.316 076.001 5596 MVI A,EC.EOF  
072.320 067 5597 STC  
072.321 .311 5598 RET  
5599  
5600  
072.322 000 5601 EOFFLG DB 0 EOF FLAG

072.323 .5603 TTWRITE EQU \*  
072.323 072 334 040 5604 LDA S.CAADDR+1  
072.326 .247. 5605 ANA A ALL DONE  
072.327 300 5606 RNZ  
072.330 .170. 5607 MOV A,B  
072.331 261 5608 ORA C  
072.332 .310. 5609 RZ ALL DONE  
072.333 032 5610 LDAX D  
072.334 .247. 5611 ANA A  
072.335 312 342 072 5612 JZ TTW2 NULL CHARACTER  
072.340 .377.002. 5613 DB SYSCALL,,SCOUT  
072.342 023 5614 TTW2 INX D  
072.343 .013. 5615 RCX B  
072.344 303 323 072 5616 JMP TTWRITE

5619 \*\* RELOCATABLE RAM CELLS.  
5620 \*.  
5621 \* THESE CELLS RESIDE AT THE TOP OF THE MONITOR.  
5622  
5623

5625 \*\* TABLE OF OVERLAY DATA  
5626 \*.  
5627 \* THIS TABLE IS GENERATED AT BOOT-UP TIME  
5628 \*.  
5629  
5630  
000.002 5631 OVLCNT EQU 2  
5632  
072.347 5633 OVLTAB EQU \*  
5634  
072.347 5635 DS OVL.ENS OVERLAY \*HDOSOVL.SYS\*  
5636  
072.357 5637 DS OVL.ENS OVERLAY \*HDOSOVL2.SYS\*  
5638  
000.002 5639 OVLMAX EQU \*-OVLTAB/OVL.ENS  
5640  
000.000 5641 ERRMI OVLMAX-OVLCNT

5643 \*\* DEVICE LIST  
5644  
000.007 5645 DEVCNT EQU 7 INITIALLY 7 DEVICES  
5646  
072.367 5647 DEVLIST DS 0 DEVICE TABLE  
000.000 5648 ERRNZ .\*-DEVLIST-DEV.NAM.  
072.367 123 131 5649 DB 'SY' DEVICE NAME  
000.000 5650 ERRNZ .\*-DEVLIST-DEV.RES.  
072.371 003 5651 DB DR.IM+DR.FR PERMANENTLY RESIDENT  
000.000 5652 ERRNZ .\*-DEVLIST-DEV.JMP.  
072.372 303 5653 DB 3030 JUMP OPCODE  
000.000 5654 ERRNZ .\*-DEVLIST-DEV.DDA.  
072.373 130 040 5655 DW SYDD DRIVER ADDRESS  
000.000 5656 ERRNZ .\*-DEVLIST-DEV.FLG.  
072.375 007 5657 DB DT.DD+DT.CR+DT.CW  
000.000 5658 ERRNZ .\*-DEVLIST-DEV.SPG.  
072.376 002 5659 DB 2 SECTORS PER GROUP  
000.000 5660 ERRNZ .\*-DEVLIST-DEV.MUM.  
072.377 001 5661 DB 1 MOUNTED UNIT MASK  
000.000 5662 ERRNZ .\*-DEVLIST-DEV.MNU.  
073.000 003 5663 DB 3 MAXIMUM NUMBER OF UNITS  
000.000 5664 ERRNZ .\*-DEVLIST-DEV.UNT.  
073.001 142 073 5665 DW SYSUNT SY: UNIT TABLE  
000.000 5666 ERRNZ .\*-DEVLIST-DEV.DVL.  
073.003 000 000 5667 DW 0 DRIVER LENGTH  
000.000 5668 ERRNZ .\*-DEVLIST-DEV.DVG.  
073.005 000 5669 DB 0 DRIVER GROUP NUMBER  
000.000 5670 ERRNZ .\*-DEVLIST-DEVLEN.

073.006	5672	TTDEV	DS	0	TT DEVICE TABLE ENTRY
000.000	5673	ERRNZ	*-TTDEV-DEV.NAM		
073.006 124 124	5674	DB	'TT'	DEVICE NAME	
000.000	5675	ERRNZ	*-TTDEV-DEV.RES		
073.010 003	5676	DB	DR.1M+DR.PR	PERMANENTLY RESIDENT	
000.000	5677	ERRNZ	*-TTDEV-DEV.JMP		
073.011 303	5678	DB	303R	JUMP OPCODE	
000.000	5679	ERRNZ	*-TTDEV-DEV.DDA		
073.012 205 072	5680	DW	TTDEV	DRIVER ADDRESS	
000.000	5681	ERRNZ	*-TTDEV-DEV.FLG		
073.014 006	5682	DB	DT.CR+DT.CW		
000.000	5683	ERRNZ	*-TTDEV-DEV.SPG		
073.015 000	5684	DB	0	SECTORS PER GROUP	
000.000	5685	ERRNZ	*-TTDEV-DEV.MUM		
073.016 001	5686	DB	1	MOUNTED MASK	
000.000	5687	ERRNZ	*-TTDEV-DEV.MNU		
073.017 001	5688	DB	1	MAXIMUM NUMBER OF UNITS	
000.000	5689	ERRNZ	*-TTDEV-DEV.UNT		
073.020 167 073	5690	DW	TTOUNT	UNIT TABLE	
000.000	5691	ERRNZ	*-TTDEV-DEV.DVL		
073.022 000 000	5692	DW	0	DRIVER LENGTH	
000.000	5693	ERRNZ	*-TTDEV-DEV.DVG		
073.024 000	5694	DB	0	DRIVER GROUP NUMBER	
000.000	5695	ERRNZ	*-TTDEV-DEVELEN		

073.025 000	5697	DB	0	NO MORE DEVICES	
073.026	5698	DS	DEVCNT-2*DEVELEN	ROOM FOR 5 MORE DEVICES	
073.141 000	5699	DB	0	BYTE USED IF LAST DEVLST ENTRY USED	

073.142	5701	SYSUNT	DS	0	
000.000	5702	ERRNZ	UNT.FLG-0		
073.142 007	5703	DB	DT.DD+DT.CR+DT.CW	SY0:	
000.000	5704	ERRNZ	UNT.GRT-1		
073.143 000 024	5705	DW	S.GRT0		
073.145	5706	DS	UNT.SIZ-3		
000.000	5707	ERRNZ	UNT.FLG-0		
073.151 007	5708	DB	DT.DD+DT.CR+DT.CW	SY1:	
000.000	5709	ERRNZ	UNT.GRT-1		
073.152 000 025	5710	DW	S.GRT1		
073.154	5711	DS	UNT.SIZ-3		
000.000	5712	ERRNZ	UNT.FLG-0		
073.160 007	5713	DB	DT.DD+DT.CR+DT.CW	SY2:	
000.000	5714	ERRNZ	UNT.GRT-1		
073.161 000 026	5715	DW	S.GRT2		
073.163	5716	DS	UNT.SIZ-3		

073.167 5721 TTOUNT DS 0  
000.000 5722  
073.167. 006 5723 ERRNZ UNT.FLG-0  
073.170 5724 DB DT.CR+DT.CW TT0:  
5725 DS UNT.SIZ-1

5727 \*\* INITIAL CHANNEL TABLE.  
5728  
000.006 5729 CHANCNT EQU .6. CHANNELS  
5730  
073.176 5731 CHANTAB EQU .\*  
5732  
5733 \* NOTE THAT THE FIRST CHANNEL IS CHANNEL 377Q, WHICH IS THE  
OVERLAY CHANNEL. THE .CLEARA FUNCTION ASSUMES THIS, AS  
5734 \* DOES. FCT.  
5735 \*

073.176. 250. 073 5737 DW \*+IOCELEN LINK - CHANNEL 377.  
073.200 000 000 000 5738 DB 0,0,0  
000.000 5739 ERRNZ IOCCID-1 USER CHANNEL #9. FOLLOWS.  
073.203 5740 DS IOCELEN-5  
073.250. 322. 073 5741 DW \*+IOCELEN LINK - CHANNEL 0.  
073.252 000 000 000 5742 DB 0,0,0  
073.255 5743 DS IOCELEN-5  
073.322 374 073 5744 DW \*+IOCELEN LINK - CHANNEL 1  
073.324. 000. 000. 000 5745 DR 0,0,0  
073.327 5746 DS IOCELEN-5  
073.374. 046. 074 5747 DW \*+IOCELEN LINK - CHANNEL 2.  
073.376 000 000 000 5748 DB 0,0,0  
074.001 5749 DS IOCELEN-5  
074.046 120 074 5750 DW \*+IOCELEN LINK - CHANNEL 3  
074.050. 000. 000. 000 5751 DR 0,0,0  
074.053 5752 DS IOCELEN-5  
074.120. 172. 074 5753 DW \*+IOCELEN LINK - CHANNEL 4.  
074.122 000 000 000 5754 DR 0,0,0  
074.125 5755 DS IOCELEN-5  
074.172 000 000 5756 DW 0 NULL LINK - CHANNEL 5  
074.174. 000. 000. 000 5757 DR 0,0,0  
074.177 5758 DS IOCELEN-5  
5759  
5760 \* OVL LOAD ADDRESS

074.244 5761  
5762 HIGHDAT EQU \*  
5763  
5764  
5765 \*\* SYSTEM MODE, NON-ZERO WHEN PROCESSING SYSCALL.  
5766  
000.000 5767 ERRNZ \*-HIGHDAT-M.SYSM  
074.244 000 5768 SYSMODE DB 0  
5769  
000.000 5770 ERRNZ \*-HIGHDAT-M.SALO  
074.245. 000. 5771 SALONE DB 0 STAND ALONE FLAG, != 0 => CAN GO STAND ALONE  
5772  
000.000 5773 ERRNZ \*-HIGHDAT-M.CSLC

074.246 000 5774 CSLLCNT DB 0 LINES ENTERED IN BUFFER  
000.000 5775 ERRNZ \*-HIGHDAT-M.CPRE  
074.247 000 5776 SCIPRE DB 0 PREVIOUSLY INPUT CHARACTER  
000.000 5777 ERRNZ \*-HIGHDAT-M.CRUB  
074.250 000 5778 CSLRBF DB 0 RUBOUT FLAG  
000.003 5779  
000.010 5780 CC.FLG EQU 00000011B CTL CHARACTERS FLAG  
000.000 5781 CZ.FLG EQU 00001000B CTL-Z.FLAG  
000.000 5782 ERRNZ \*-HIGHDAT-M.CINT  
074.251 000 5783 SCINTEL.DB 0 SYSTEM CONSOLE INTERRUPT FLAGS  
000.000 5784  
074.252 265 074 5785 ERRNZ \*-HIGHDAT-M.CIN  
000.000 5786 SCIIN DW CSLIBUF IN POINTER  
074.254 265 074 5787 ERRNZ \*-HIGHDAT-M.COUT  
000.000 5788 SCIOUT DW CSLIBUF OUT POINTER  
074.256 265 074 5789 ERRNZ \*-HIGHDAT-M.CFWA  
000.000 5790 SCIFWA DW CSLIBUF  
074.260 032 075 5791 ERRNZ \*-HIGHDAT-M.CLWA  
074.260 032 075 5792 SCILWA DW CSLIBFE END POINTER  
000.000 5793  
074.262 004 5794 ERRNZ \*-HIGHDAT-M.CDLY  
000.000 5795 CSLDLY DB 4 PAD CHARACTER COUNT  
074.263 215 066 5796 ERRNZ \*-HIGHDAT-M.CICA  
074.263 215 066 5797 CSLICA DW SCOUTA ADDRESS OF DELAY CHARACTER  
074.263 215 066 5798  
074.265 5799  
075.032 5800 CSLIBUF DS 101  
075.032 5801 CSLIRFE DS 0 END OF BUFFER  
075.032 5802  
075.032 5803 \* PATCH AREA  
075.032 5804  
075.041 266 261 337 5805 DB 377Q-'B',377Q-'Y',377Q-'L',377Q-'G',377Q-'A',377Q-'C',377Q-'I',377Q-'N',377Q-'R',377Q-'E',377Q-'M',377Q-'E'  
075.050 262 275 255 5806 DB 377Q-'I',377Q-'N',377Q-'L',377Q-'R',377Q-'E',377Q-'M',377Q-'E'  
075.057 337 260 271 5807 DB 377Q-'M',377Q-'B',377Q-'R',377Q-'A',377Q-'N',377Q-'C',377Q-'E'  
075.066 014 5808 DB 377Q-'L',377Q-'O',377Q-'F',377Q-'J',377Q-'G',377Q-'L'  
075.066 014 5809 DB FF  
075.067 5810  
075.067 5811 SECSCR EQU \* SYSTEM SCRATCH AREA  
075.067 5812 DS 512  
077.067 5813  
077.067 5814 LWASYS EQU \* END OF MONITOR  
014.361 5815 LENSYS EQU LWASYS-FWASYS  
077.067 5816  
077.067 5817 \* PATCH AREA FOR RELOCATION TABLE  
077.067 5818  
077.076 266 271 337 5819 DB 377Q-'I',377Q-'F',377Q-'U',377Q-'L',377Q-'C',377Q-'N',377Q-'R',377Q-'D',377Q-'T',377Q-'H',377Q-'S'  
077.105 337 252 337 5820 DB 377Q-'L',377Q-'U',377Q-'R',377Q-'C',377Q-'N',377Q-'G'  
077.114 253 337 276 5821 DB 377Q-'L',377Q-'U',377Q-'R',377Q-'C',377Q-'N',377Q-'G'  
077.123 265 275 336 5822 DB 377Q-'T',377Q-'A',377Q-'G',377Q-'D',377Q-'I'  
077.126 014 5823 DB 377Q-'J',377Q-'B',377Q-'I'  
077.126 014 5824 DB FF  
077.127 5825  
077.127 5826  
077.127 5827 DS 4  
077.127 5828  
077.127 5829

HDDS - RESIDENT HDDS CODE  
DATA AREAS

CHAN

HEATH HBASIC V1.4 01/20/78  
14:07:54 16-MAY-80

PAGE 129

5830  
5831  
5832 LON G  
5833 LON C  
077.133 103 050 111 5834 END  
050 117 050  
125 050 134  
050 176 050  
233 050 241  
050 247 050  
255 050 274  
050 326 050  
335 050 340  
050 343 050  
346 050 351  
050 354 050  
376 050 004  
051 043 051  
047 051 363  
051 373 051  
005 052 216  
052 255 052  
260 052 273  
052 305 052  
351 052 360  
052 .023 .053  
322 053 325  
053 341 053  
365 053 012  
054 027 054  
032 054 045  
054 056 054  
064 054 107  
054 120 054  
131 054 137  
054 251 055  
015 056 270  
056 321 056  
336 056 353  
056 007 057  
127 062 133  
062 137 062  
157 062 167  
062 172 062  
201 062 204  
062 214 062  
224 062 235  
062 242 062  
252 062 304  
062 311 062  
316 062 321  
062 324 062  
334 062 336  
062 340 062  
342 062 344  
062 346 062

350 062 352  
062 354 062  
356 062 366  
062 372 062  
002 063 016  
063 025 063  
034 063 056  
063 064 063  
113 063 121  
063 124 063  
131 063 155  
063 172 063  
200 063 206  
063 262 063  
277 063 321  
063 324 063  
330 063 334  
063 351 063  
354 063 365  
063 370 063  
373 063 376  
063 001 064  
004 064 007  
064 015 064  
027 064 042  
064 131 064  
140 064 143  
064 151 064  
154 064 164  
064 167 064  
175 064 202  
064 216 064  
224 064 236  
064 242 064  
255 064 262  
064 277 064  
304 064 315  
064 320 064  
325 064 332  
064 337 064  
343 064 370  
064 000 065  
006 065 013  
065 016 065  
027 065 032  
065 037 065  
042 065 046  
065 053 065  
056 065 063  
065 066 065  
074 065 103  
065 106 065  
111 065 116  
065 123 065  
126 065 132  
065 136 065

HDOS - RESIDENT HDOS CODE  
DATA AREAS

CHAN

HEATH H8ASH V1.4 01/20/78  
14:07:55 16-MAY-80

PAGE 131

141 065 145  
065 154 065  
157 065 167  
065 174 065  
202 065 214  
065 222 065  
227 065 234  
065 252 065  
262 065 320  
065 325 065  
330 065 335  
065 342 065  
347 065 352  
065 370 065  
375 065 007  
066 024 066  
035 066 042  
066 053 066  
061 066 071  
066 100 066  
104 066 113  
066 133 066  
143 066 152  
066 163 066  
172 066 200  
066 207 066  
217 066 230  
066 235 066  
241 066 251  
066 257 066  
265 066 272  
066 277 066  
322 066 325  
066 330 066  
355 066 360  
066 366 066  
373 066 026  
067 031 067  
034 067 040  
067 043 067  
054 067 057  
067 067 067  
073 067 153  
067 157 067  
203 067 226  
067 251 067  
270 067 274  
067 301 067  
306 067 331  
067 335 067  
352 067 364  
067 367 067  
377 067 002  
070 007 070  
135 070 145  
070 233 070

HDOS - RESIDENT HDOS CODE  
DATA AREAS

CHAN

HEATH H8ASM V1.4 01/20/78  
14:07:55 16-MAY-80

PAGE 132

270 070 302  
070 314 070  
350 070 377  
070 026 071  
031 071 042  
071 050 071  
054 071 077  
071 106 071  
117 071 156  
071 202 071  
205 071 212  
071 216 071  
221 071 234  
071 266 071  
273 071 351  
071 356 071  
361 071 371  
071 375 071  
001 072 237  
072 250 072  
264 072 271  
072 276 072  
303 072 314  
072 336 072  
343 072 001  
073 012 073  
020 073 176  
073 250 073  
322 073 374  
073 046 074  
120 074 252  
074 254 074  
256 074 260  
074 263 074  
000 000

ASSEMBLY COMPLETE.

5834 STATEMENTS

0 ERRORS DETECTED.

7810 BYTES FREE

**HDOS - RESIDENT HDOS CODE  
CROSS REFERENCE TABLE**

XREF VI.1  
PAGE 133

.CLEAR	000055	392L	3563
.CLEARA	000056	393L	3538
.CLOSE	000046	385L	
.CLRCDD	000007	369L	4622
.CONSL	000006	368L	
.CRC	002347	274E	
.CRCSUM	040027	294E	
.CTC	002172	268E	
.CTLCC	000041	380L	
.CTLFLG	040011	290E	
.DECODE	000053	390L	
.DELET	000050	387L	
.DISMT	000061	396L	
.DLEIDS	040021	292E	
.ILY	000053	263E	1497 4637
.IMMMS	000203	407L	
.IMOUN	000201	405L	3582
.DOD	003122	277E	
.DODA	003356	279E	
.DSPMOD	040007	288E	
.DSPPROT	040006	287E	
.DUMP	001374	265E	
.ERROR	000057	394L	3579 3586
.EXIT	000000	362L	1127 1356
.HORN	002140	267E	
.IBENT	000000	262E	
.IOWRK	040002	285E	
.LINK	000040	379L	1124 3285 3551
.LOAD	001267	264E	
.LOADD	000062	397L	
.LOADO	000010	370L	
.MFLAG	040010	289E	1293 3493 3531
.MONMS	000202	406L	
.MOUNT	000200	404L	3314
.NAME	000054	391L	
.OPENC	000045	384L	
.OPENR	000042	381L	
.OPENU	000044	383L	
.OPENW	000043	382L	
.PCHL	002264	270E	
.POSIT	000047	386L	
.PRINT	000003	365L	1644 1646 1648 3561 3576
.RCK	003260	278E	
.READ	000004	366L	
.REGI	040005	286E	
.REGPTR	040035	297E	
.RENAM	000051	388L	
.RESET	000204	408L	
.RNB	002331	273E	
.RNP	002325	272E	
.SCIN	000001	363L	2613 4626 5580
.SCOUT	000002	364L	2617 2951 2990 3137 5613
.SETTP	000052	389L	
.SRS	002265	271E	
.START	040000	284E	
.SYSRES	000012	372L	3287
.TICCNT	040033	296E	
.TPERR	002205	269E	

:TPERRX	040031	295E						
:UIVEC	040037	298E	1291	1425				
:VERS	000011	371L						
:WNB	003024	276E						
:WNP	003017	275E						
:WRITE	000005	367L						
ABP	067277'	3643	3810	4518L				
AC.DLY	000156	41E	1496	4636				
ACA	067311'	4490	4547L					
ACA0	067313'	4548L	4587					
ACA1	067321'	4552L	4588					
ACA2	067340'	4557	4559L					
ACA3	067354'	4555	4573L					
ACA8	070004'	4579	4593L					
ACA9	070016'	4568	4580	4595	4601L			
AIO.CGN	041047	1012L	4604	4606				
AIO.CHA	041116	1027L	1102	4483	4743	4993		
AIO.CNT	041111	1023L	4602					
AIO.CSI	041050	1013L	4606					
AIO.IDA	041041	1008E	4754					
AIO.DES	041055	1017L	2271	2301				
AIO.DEV	041057	1018L						
AIO.DIR	041062	1021L	1946	1959	2282	3003		
AIO.DTA	041053	1016L	2257					
AIO.EOF	041113	1025L	4405	4473				
AIO.EOM	041112	1024L	4492	4597				
AIO.FLG	041043	1009L	4998					
AIO.GRT	041044	1010L						
AIO.LGN	041051	1014L	4548	4549	4573	4574	4603	4605
AIO.LSI	041052	1015L	4549	4552	4575	4605		
AIO.SPG	041046	1011L	4553					
AIO.TFP	041114	1026L	4550					
AIO.UNI	041061	1019L	1349	2260	3534	4739	5511	
AIO.VEC	041040	1007L	1345	4173	4207	4437		
BELL	000007	193E	1187	1187	1795	1804	2190	2209
BKSP	000010	195E	3757	3853	3857	4027		
BOOT.F	000001	987E	1120					
BOOTTABT	053252	1797	1811L					
BOOTERR	053205	1563	1572	1687	1699	1720	1803L	1942
BRP	070031'	3570	4621L					
BRP0	070130'	4626L	4628					
BRP1	070216'	4631	4636L					
BUFF	025090	1988	1991	2099	2106	2115	2118	2120
		2166	3164E					
C,DSYN	000375	184E						
C,STX	000002	197E						
C,SYN	000026	196E						
CAD1	060214	2735L	2751					
CAD2	060250	2733	2755L	2764	2767	2770	2773	2775
CAD3	060253	2740	2761L					
CARA	060314	2734	2782L					
CB.CLI	000100	232E	247					
CB.MTL	000040	231E						
CB.SPK	000200	233E						
CB.SSI	000020	230E						
CC.FLG	000003	3719	5780E					
CDB.HB4	000001	930E	1446	1513	3069	3689	4064	4630
CDB.HB5	000000	929E	1067					

CDS	053320	1113	1831L
CDS1	053352	1848L	1856
CDS2	053364	1851	1854L
CDS3	053375	1850	1860L
CDS4	053376	1862L	1870 1895
CDS4.5	054011	1866	1868L
CDS5	054022	1864	1874L
CFF	031354	315E	4584
CHANCNT	000006	5729E	
CHANTAB	073176	1329	5731E
CLRC0	067024	3351	4292L
CO.FLG	000001	906E	3866 3867 3964 5545
CONSL	066376	3350	4263E
CPA	070232	3262	3681 4672L
CR	000015	189E	1061 3054 3089 3101 3830 4024 4133
CRLF	066261	3803	4022 4053 4132L
CS.FLG	000200	907E	3868 3870 3870 3961
CSL.CHR	000001	884E	3614 3748
CSL.ECH	000200	882E	3844
CSL.WRF	000002	883E	
CSLDICA	074263	1397	5797L
CSLDILY	074262	1395	4107 5795L
CSLIBFE	075032	5792	5801L
CSLIBUF	074265	4293	5786 5788 5790 5800L
CSLLCNT	074246	3630	3665 3837 4297 5774L
CSLRBF	074250	3773	3776 3783 3786 4298 5778L
CTLA	000001	204E	
CTLB	000002	205E	
CTLC	000003	206E	
CTLD	000004	207E	2658 3660 3828 4977
CTLO	000017	208E	3725
CTLP	000020	209E	
CTLQ	000021	210E	
CTLS	000023	211E	
CTLZ	000032	212E	3710 3922 3924
CTP.2SB	000010	892E	1055 1093 1459 1460 1490 1492
CTP.BKM	000002	893E	1055 3755
CTP.BKS	000200	889E	3770
CTP.MLI	000040	890E	1055 1336 3655 3656
CTP.MLO	000020	891E	1055 1336 3993
CTP.TAB	000001	894E	4007
CZ.FLG	000010	3719	3894 5781E
D.ABORT	040141	791L	
D.CDE	040160	796L	1274
D.CON	040110	741L	741
D.DLY	040235	811L	
D.DLYHS	040244	829L	
D.DLYMO	040243	828L	
D.DRVTB	040251	834L	
D.DIS	040163	797L	
D.DVCTL	040242	826L	
D.E.CHK	040267	845L	
D.E.HCK	040270	846L	
D.E.HSY	040266	844L	
D.E.MDS	040265	843L	
D.E.TRK	040272	848L	
D.E.VOL	040271	847L	
D.ERR	040265	842L	

HDOS™ RESIDENT HDOS™ CODE  
CROSS REFERENCE TABLE

XREF V1.1  
PAGE 137

D.ERRL	040273	849L					
D.ERRT	040232	810L	1271				
D.ERTS	040126	777L	1323	5445			
D.HECNT	040261	836L	5441				
D.LPS	040177	801L					
D.LPSA	040116	768L					
D.MAI	040171	799L					
D.MAIA	040115	767L	1400				
D.MAO	040174	800L					
D.MOUNT	040133	789L					
D.OECNT	040264	838L	5431	5437	5510		
D.OPR	040273	853L					
D.OPW	040275	854L	5463	5465			
D.RAM	040240	744L	821	856			
D.RAML	000037	856E					
D.RDB	040202	802L					
D.READ	040147	793L					
D.READR	040152	794L					
D.SDP	040205	803L	1280	5462			
D.SDPA	040117	769L					
D.SDPP	040120	770L					
D.SDT	040166	798L	5426				
D.SECNT	040262	837L	5406	5408	5428	5430	5437
D.STS	040210	804L					
D.STSA	040121	771L					
D.STSB	040122	772L					
D.STZ	040213	805L	5425				
D.SYDD	040130	788L					
D.TRKPT	040245	831L					
D.TS	040241	824L					
D.TT	040240	823L	5444				
D.UDLY	040216	806L					
D.VEC	040130	743L	786				
D.VOLPT	040247	832L					
D.WHDA	040123	773L					
D.WNB	040227	809L					
D.WNHA	040124	774L					
D.WRITE	040112	764L					
D.WRITE	040113	765L					
D.WRITE	040114	766L					
D.WRITE	040155	795L	1277				
D.WSC	040221	807L					
D.WSCA	040125	775L					
D.WSP	040224	808L					
D.XIT	040144	792L					
D.XITA	040110	763L	1303	1305			
D.XOK	040136	790L					
DAD1	061101	2818	2831	2833	2856L		
DADB	061104	2835	2860L				
DC.ABT	000007	558L	1556	3535			
DC.CLO	000006	557L					
DC.LOD	000011	560L					
DC.MAX	000012	561L					
DC.MOU	000010	559L	1570				
DC.OPR	000003	554L	5014				
DC.OPU	000005	556L	5016				
DC.OPW	000004	555L					
DC.REA	000000	551L	1684	1696	1717	4172	4410

DC.RER 000002	553L	1561						
DC.WRI 000001	552L	1608	1785	4206	4460	4500		
DCA 032002	317E	4403	4471					
DDF.BOL 000011	621E							
DDF.BOO 000000	620L							
DDF.LAB 000011	622L	1560						
DDF.RGT 000012	623L	1683						
DDF.USR 000014	624L							
DEBUG 000001	2E	1079	1157	1419	3214	3269	5402	
DEV.DDA 000004	443L	2223	5654	5679				
DEV.DVG 000016	455L	2237	5668	5693				
DEV.DVL 000014	454L	2235	5666	5691				
DEV.FLG 000006	444L	2225	5656	5681				
DEV.JMP 000003	442L	2221	5652	5677				
DEV.MNU 000011	451L	2231	5662	5687				
DEV.MUM 000010	450L	2229	5660	5685				
DEV.NAM 000000	434L	2217	5648	5673				
DEV.RES 000002	438L	2219	5650	5675				
DEV.SPC 000007	449L	2097	2227	5658	5683			
DEV.UNT 000012	452L	2233	2258	4779	5664	5689		
DEVCNT 000007	2175	2176	5645E	5698				
DEVELLEN 000017	457E	2177	2201	2239	5670	5695	5698	
DEVLST 072367'	1331	2097	5647L	5648	5650	5652	5654	5656
	5666	5668	5670					
DF.CLR 000376	477E	1732	1867	1884	2036			
DF.DI 000040	160E							
DF.DSO 000002	156E	5516						
DF.DS1 000004	157E	5517						
DF.DS2 000010	158E	5520						
DF.EMP 000377	476E	1729	2033					
DF.HD 000001	150E							
DF.MO 000020	159E							
DF.SD 000010	153E							
DF.ST 000100	161E							
DF.TO 000002	151E							
DF.WG 000001	155E							
DF.WP 000004	152E	5467						
DF.WR 000200	162E							
DIF.CNT 000020	682E	1970	4487					
DIF.LOC 000100	680E							
DIF.SYS 000200	679E							
DIF.WF 000040	681E							
DIR.ALD 000025	492L							
DIR.CLU 000015	485L							
DIR.CRD 000023	491L							
DIR.EXT 000010	480L							
DIR.FGN 000020	488L	1737	1972	2088				
DIR.FLG 000016	486L	1967	1972	4484				
DIR.LGN 000021	489L							
DIR.LSI 000022	490L							
DIR.NAM 000000	479L	1946	1959	2282	3003			
DIR.PRD 000013	481L							
DIR.VER 000014	482L							
DIREAD 067131'	1366	4174	4398E	4413				
DIREAD1 067162'	4411	4432L	4461	4501				
DIRELEN 000027	494E	502	539	1021	2292			
DIRIDL 000015	483E	1945	1948	1958	1997	1999	2256	
DIS.ENL 001373	506L	1756	1854	1868	2059			



**HDOS - RESIDENT HDOS CODE  
CROSS REFERENCE TABLE**

XREF VI.1  
PAGE 140

**HDOS - RESIDENT HDOS CODE  
CROSS REFERENCE TABLE**

XREF V1.1  
PAGE 141

**HDOS - RESIDENT HDOS CODE  
CROSS REFERENCE TABLE**

XREF VI.i  
PAGE 142

IOC.LGN	000012	529L
IOC.LNK	000000	515L
IOC.LSI	000013	530L
IOC.SPG	000007	526L
IOC.SQL	000003	523E 4755
IOC.UNI	000022	536L 4738
IOCCTD	000001	543E 4711 5739
IOCELEN	000052	541E 5737 5740 5741 5743 5744 5746 5747 5749 5750 5752 5753 5755 5758
IP.PAI	000360	223E
ISDEHL	072025	5173 5229L
LAB.DAT	000000	636E
LAB.DIS	000003	632L 1583 1949 2013
LAB.GRT	000005	633L 1578
LAB.IND	000001	631L
LAB.LAB	000021	643L 644 1649
LAB.LBL	000074	644E
LAB.NOD	000002	638E
LAB.SER	000000	630L 1567 1636
LAB.SPG	000007	634L 1977
LAB.SYS	000001	637E
LAB.VER	000011	641L
LAB.VLT	000010	640L
LABEL	047211	1559 1567 1578 1583 1636 1649 1949 1977 2013 3168E
LDD	071001	3257 4804L
LDD2	071030	4825L 4832
LDD3	071047	4840L 4857
LDD4	032323	4846 4861E
LDD8	071101	4825 4840 4842 4883L
LDE	056306	2257L
LDE..	056303	2256L
LDE..	056331	1950 2268L 2304
LDE3	056355	2282L 2296
LDE3.5	056375	2285 2292L
LDO	033012	325E 3462
LDON	062360	3252 3321 3384L 4333
LDON0	063036	3398 3418 3423L 3477
LDON1	063040	3425L 3481 3486
LDON2	063044	3410 3432L
LDON3	063047	3421 3433L
LDON4	063133	3452 3454 3455 3461L
LDON5	063174	3389 3479L
LDON6	063202	3436 3457 3483L
LENSYS	014361	3176 5815E
LF	000012	190E 3054 3703 4093 4135
LOAD0	067052	3353 4331E
LOAD01	067120	4337 4354L
LOAD02	067122	4334 4355L
LS0	054071	1114 1919L
LS0.	054204	1920 1932 1944L
LS0..	055021	1925 1934 1987L
LS01	054277	1951 1966L
LS02	054326	1971 1983L
LS0A	055045	1919 1996L 1997
LS0B	055062	1931 1998L 1999
LBWASYS	077087	1197 1207 5814E 5815
M.CICA	000017	709L 5796
M.CDLY	000016	708L 5794

M.CFWA	000012	706L	5789
M.CIN	000006	704L	5785
M.CINT	000005	703L	5782
M.CLWA	000014	707L	5791
M.CQUT	000010	705L	5787
M.CPRE	000003	701L	5775
M.CRUB	000004	702L	5777
M.CSLC	000002	700L	5773
M.FDX	000303	257E	
M.PAMB	000021	256E	
M.SALO	000001	699L	5770
M.SYSM	000000	698L	5767
MI.CFI	000374	29E	
MI.JMP	000303	30E	1344
MI.RET	000311	31E	
MSD	051316	1110	1556L
MSD1	052050	1611	1618L
MSDA	052052	1589	1621L
NL	000012	201E	202 1538 1538 1540 1655 1657 1795 1795 1804 1941 1956
		1984	2190 2209 2662 2950 3145 3491 3588 3658 3832 4021 4624 4627
		4634	4975
NUL2	000000	192E	
NULL	000200	191E	
OP.CTL	000360	224E	
OP.DIG	000360	225E	
OP.SEG	000361	226E	
DTI	071152	3324	3394 3440 3449 4339 4932L
DVBUFE	062106	3171E	3176
DVL,COD	000000	415L	1924 1933 3441 3448
DVL,ENS	000010	420E	1924 1927 1933 1935 4935 5635 5637 5639
DVL,ENT	000004	417L	3327 3470 4349
DVL,FLB	000006	418L	3395 4340 4349
DVL,IN	000001	954E	3411 4810
DVL,NUM	000014	956E	3248 3251 3406 3416 3464
DVL,RES	000002	955E	3245 3397 4342 4345
DVL,SIZ	000002	416L	1927 1935 3448
DVL,UCS	000200	957E	3237 3451 4336
DVLO	000000	426L	1924 1927 3316
DVL1	000001	427L	1933 1935 3319
DVLCNT	000002	5631E	5641
DVLMAX	000002	3388	5639E 5641
DVLTAB	072347	1327	1924 1927 1933 1935 4936 5633E 5639
PATCH	062006	3144L	3147 3148
PCHL	032361	4863E	
PDI	033145	327E	4409 4459 4499
PGT	052167	1112	1680L
PGT3	052257	1711L	1761
PGT4	052307	1728L	1760
PGT5	052331	1743L	1751
PGT6	052357	1731	1756L
PGT7	052375	1714	1734 1765L
PGT8	053002	1768L	1774
PGT9	053012	1770	1773L
PGTA	025000	1695	1702 1743 3162E
PGTB	053044	1753	1789E
PGTERR	053047	1735	1748 1794L
PIC,COD	000006	656L	663
PIC, ID	000000	651L	



HDOS™ RESIDENT HDOS CODE  
CROSS REFERENCE TABLE

XREF VI.1

PAGE 145

S.GRT2	026000	734E	3166	5718
S.GUP	041027	981L	1371	
S.HIMEM	040316	871L	1165	1180 1195
S.INT	040343	746L	924	
S.JUMPS	041010	975L	1347	1361 1363 1365 1367 1369 1371
S.MOUNT	041032	983L	1118	3433 3539 3556 3913
S.OFWA	040350	937L	1328	
S.OMAX	040324	877L	1926	1937
S.OSN	041004	966L	3444	
S.OVLE	041000	963L	3472	
S.OVLFL	040371	959L	1335	3222 3236 3408 3450 3463 3466 4335 4808
S.OVLS	040376	962L	3419	3461
S.OVSTK	041035	991L	3311	
S.READ	031275	307E	1841	1893 1990 2025 2102 4910
S.RFWA	040356	940L	1314	2132 2135
S.SCI	041024	980L	1369	
S.SCR	041120	1029L	1340	
S.SDD	041010	976L	1361	
S.SOVR	041146	748L	750	
S.SSN	041002	965L	1921	
S.SYSM	040320	873L	1313	2136 4351
S.TIME	040312	870L	1101	1102
S.UCSF	040372	960L		
S.UCSL	040374	961L		
S.USRM	040322	875L	1316	
S.VAL	040277	745L	866	
S.WRITE	031330	309E	1878	
SALONE	074245	1408	3542	5771L
SB.ORG	047000	1034E	1040	1213 1214
SB.OVMX	014000	1035E	1922	
SC.ACE	000350	40E	1189	1473 1478 1480 1483 1485 1488 1495 1498 1499 1501
		1525	3083	3087 3495 3699 4078 4082 4639
SC.UART	000372	109E	1190	1452 1453 1454 1455 1457 1466 1468 1519 3074 3078
		3494	3694	4069 4073 4640
SCD	051046	1109	1418E	
SCDLY	066240	4086	4107L	
SCDLY1	066243	4108L	4117	
SCI	071237	1370	4175	4209 4992L
SCIFWA	074256	4522	4967	5790L
SCIIN	074252	3637	3808	3824 4294 4965 4979 5786L
SCILWA	074260	4519	4972	5792L
SCIN	064123	3340	3612E	
SCIN1	064136	3617	3625L	
SCIN2	064150	3626	3636L	
SCIN2.5	064213	3650	3652	3658L
SCIN3	064223	3659	3665L	
SCIN4.3	065010	3756	3760L	
SCIN4.5	065015	3759	3766L	
SCINI	064233	1424	3676E	
SCINIO	064247	3679	3688L	
SCINIO1	064264	3690	3699L	
SCINIO2	064266	3695	3701L	
SCINII0	065205	3831	3839L	
SCINII1	065217	3772	3853L	
SCINII2	064310	3711	3718L	
SCINII3	064317	3713	3724L	
SCINII4	064362	3726	3728	3730 3746L
SCINIS	065051	3775	3780L	

SCINI6	065055'	3762	3783L
SCINI65	065077'	3785	3791L
SCINI7	065105'	3797L	3798
SCINI8	065130'	3750	3793 3807L
SCINI9	065156'	3813	3824L
SCINI95	065201'	3829	3837L
SCINIA	065236'	3735	3866L
SCINTFL	074251'	5783L	
SCIOUT	074254'	3636	3644 3811 4295 4966 5788L
SCIPRE	074247'	3724	3921 3925 5776L
SCOUT	065360'	3342	3956E 4228
SCOUTO	065363'	3960L	3963
SCOUT1	066002'	3778	3781 3789 3820 3847 3918 3920 3930 3967 3984L 4134
SCOUT10	066222'	4061	4091L
SCOUT11	066224'	4092L	4097
SCOUT2	066021'	3800	3802 3854 3856 3858 3989 4001L 4094 4113 4136
SCOUT3	066037'	4011L	4016
SCOUT4	066056'	4002	4010 4012 4021L
SCOUT5	066075'	4025	4027L
SCOUT6	066110'	4028	4030 4032L
SCOUT7	066126'	4033	4040L
SCOUT8	066147'	4042	4060L
SCOUT9	066155'	4063L	
SCOUT91	066165'	4069L	4071
SCOUT92	066202'	4065	4078L 4080
SCOUT95	066214'	4074	4084L
SCOUTA	066215'	4085E	5797
SCU	051063	1426	1445L
SCU1	051133	1447	1473L 1475
SDD	071263'	1362	2224 5014L
SDT	055077	1115	2012E
SDT2	055132	2030L	2061
SDT3	055176	2043	2048 2058L
SDT4	055177	2035	2059L
SDTA	026000	2014	2018 2022 2029 2059 3166E
SDTB	055210	2045	2063L 2064
SDTBL	000013	2046	2064E
SDV	050356	1104	1108 1390L
SECSCR	075067'	1339	1707 1711 1716 1724 1752 1756 1839 1854 1868 1876 1877
		1887	1892 1894 2270 2278 2300 4821 4887 4898 5811E
SLR	050133	1107	1289E
SLRA	050324	1298	1352L 1357
SLRAL	000010	1298	1357E 1358
SLRB	050334	1347	1360L 1361 1363 1365 1367 1369 1371 1373
SLRBL	000022	1347	1373E
SRR	050102	1106	1268E
SSD	057110	1116	2372L
SSD1	057171	2381	2386 2398L
SSD2	057207	2394	2402L
SSD3	057220	2408L	2410 2431
SSD4	057246	2413	2424L
SSD5	057254	2420	2429L
SSD6	057336	2425	2435L
SSDA	057351	2402	2417 2441L
SSDB	047211	2377	2384 2408 3170E
STACK	042200	752E	1078 3527 3550
STACKL	001032	750E	
STD1	055105	2018L	2032

**HDOS - RESIDENT HDOS CODE  
CROSS REFERENCE TABLE**

XREF VIII

PAGE 147

HDOS - RESIDENT HDOS CODE  
CROSS REFERENCE TABLE

XREF V1.1  
PAGE 148

UC.IIP	000001	58E						
UC.LOO	000020	78E	1479	1500				
UC.MSI	000010	55E						
UC.OR	000002	82E						
UC.OU1	000004	76E						
UC.OU2	000010	77E						
UC.PE	000004	83E						
UC.PEN	000010	67E						
UC.RI	000100	96E						
UC.RLS	000200	97E						
UC.RSI	000004	54E						
UC.RTS	000002	75E						
UC.SB	000100	70E						
UC.SKF	000040	69E						
UC.TER	000004	92E						
UC.THE	000040	86E	3084	4079				
UC.TRE	000002	53E						
UC.TSE	000100	87E	1474					
UCI.ER	000020	131E	1467	1518				
UCI.IE	000002	133E	1518					
UCI.IR	000100	129E	1456					
UCI.RE	000004	132E	1467	1518				
UCI.RD	000040	130E						
UCI.TE	000001	134E	1467	1518				
UDR	000000	106E	3078	3694	4073			
UF.FCT	000100	177E						
UF.RDA	000001	174E						
UF.ROR	000002	175E						
UF.RPE	000004	176E						
UF.TBM	000200	178E						
UMI.16X	000002	124E	1465					
UMI.1B	000100	114E	1460	1465				
UMI.1X	000001	123E						
UMI.2B	000300	116E	1460					
UMI.64X	000003	125E						
UMI.HB	000200	115E						
UMI.L5	000000	119E						
UMI.L6	000004	120E						
UMI.L7	000010	121E						
UMI.L8	000014	122E	1465					
UMI.FA	000020	118E						
UMI.FE	000040	117E						
UNT.RIS	000005	466L	1586	1705	2263			
UNT.FLG	000000	463L	1595	1834	2153	5703	5709	5715
UNT.GRT	000001	464L	5705	5711	5717			
UNT.GTS	000003	465L	1581	1693	1782			
UNT.SIZ	000007	468E	2128	2155	5062	5707	5713	5719
UO.CLK	000001	249E	1292	3530				
UO.DDU	000002	248E						
UO.HLT	000200	246E	1292	3530				
UO.NFR	000100	247E						
UP.IP	000174	168E						
UP.FC	000175	169E						
UP.SC	000176	171E						
UP.SR	000176	172E						
UP.ST	000175	170E						
UR.DLL	000000	47E	1485					
UR.DLM	000001	49E	1488					

## HDOS - RÉSIDENT HDOS CODE

XREF V1.1

PAGE 149

UR.IER	000001	51E	1189	1478	1525	3495	4639
UR.IIR	000002	57E					
UR.LCR	000003	61E	1483	1495			
UR.LSR	000005	80E	1473	3083	4078		
UR.MCR	000004	73E	1480	1499	1501		
UR.MSR	000006	89E					
UR.RBR	000000	43E	1498	3699			
UR.THR	000000	45E	3087	4082			
USERFWA	042200	753E	1315				
USR	000001	107E	1190	1452	1453	1454	1455
			4069	4640			
USR.FE	000040	138E					
USR.OE	000020	139E					
USR.FE	000010	140E					
USR.RXR	000002	142E					
USR.TXE	000004	141E					
USR.TXR	000001	143E	3075	4070			
VERS	000026	353E	1539	1539	4374		
VERSN	067125'	3355	4374L				
WRITE	066327'	3348	4196L				
WRITE1	034370	5456E	5484				
WRITE8	035132	5457E	5470	5483			
XCHGRC	072032'	5163	5167	5175	5177	5335L	

8284 BYTES FREE

