

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

K	K	SSSS	PPPP	RRRR	000	U	U	L			
K	K	S	P	P	R	R	0	0	U	U	L
K	K	S	P	P	R	R	0	0	U	U	L
KKK		SSS	PPPP	RRRR	0	0	U	U	L		
K	K		S	P	R	R	0	0	U	U	L
K	K		S	P	R	R	0	0	U	U	L
K	K	SSSS	P		R	R	000	UUUUU	LLLLL		

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

N	N				SSSS	999	1	000		H	H	000	L	EEEE	SSSS						
N	N		t		S	9	9	11	0	0		H	H	0	0	L	E	S			
NN	N	ooo	ttt	eee	##	S	9	9	1	0	00		H	H	0	0	L	E	S		
N	N	N	o	o	t	e	e	##	SSS	9999	1	0	0	0	-----	HHHHH	0	0	L	EEEE	SSS
N	NN	o	o	t	eeee		S		9	1	00	0		H	H	0	0	L	E		S
N	N	o	o	t	e	##	S		9	1	0	0		H	H	0	0	L	E		S
N	N	ooo	tt	eeee	##	SSSS	999	111	000		H	H	000	LLLLL	EEEE	SSSS					

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

*
* * * * * N O T I C E * * * * *
*
* ATTENTION!! NEW BIN NUMBERS AT HILL CENTER!
* -STARTS AUGUST 29- CHECK CAMPUS MAIL
* FOR DESCRIPTIVE MEMO OR PICK ONE UP AT HILL
* I/O COUNTER OR AID STATION-
* RESET BIN IN WYLBUR TO CONFORM!
*

FFFFFFFFFFFFFFF	00000000	RRRRRRRRRR	TTTTTTTTTTTTTT	HHH	HHH
FFFFFFFFFFFFFFF	00000000	RRRRRRRRRR	TTTTTTTTTTTTTT	HHH	HHH
FFFFFFFFFFFFFFF	00000000	RRRRRRRRRR	TTTTTTTTTTTTTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFFFFFFFFFFFF	000 000	RRRRRRRRRR	TTT	HHHHHHHHHHHHHH	HHH
FFFFFFFFFFFFF	000 000	RRRRRRRRRR	TTT	HHHHHHHHHHHHHH	HHH
FFFFFFFFFFFFF	000 000	RRRRRRRRRR	TTT	HHHHHHHHHHHHHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	000 000	RRR RRR	TTT	HHH	HHH
FFF	00000000	RRR RRR	TTT	HHH	HHH
FFF	00000000	RRR RRR	TTT	HHH	HHH
FFF	00000000	RRR RRR	TTT	HHH	HHH

LLL	SSSSSSSSSS	TTTTTTTTTTTTTT		111
LLL	SSSSSSSSSS	TTTTTTTTTTTTTT		111
LLL	SSSSSSSSSS	TTTTTTTTTTTTTT		111
LLL	SSS	TTT		111111
LLL	SSS	TTT		111111
LLL	SSS	TTT		111111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSSSSSSS	TTT		111
LLL	SSSSSSSS	TTT		111
LLL	SSSSSSSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT		111
LLL	SSS	TTT	111
LLL	SSS	TTT	111
LLL	SSS	TTT	111
LLL	SSS	TTT	111
LLLLLLLLLLLLLLLL	SSSSSSSSSS	TTT	11111111
LLLLLLLLLLLLLLLL	SSSSSSSSSS	TTT	11111111
LLLLLLLLLLLLLLLL	SSSSSSSSSS	TTT	11111111

START Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:44:56 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*
File PS:<KSPROUL>FORTH.LST.1, created: 11-Sep-82 20:29:48, printed: 11-Sep-82 20:45:11
Job parameters: Request created:11-Sep-82 20:31:14 Page limit:108 Forms:TN Account:OPS Note:S910-HOLES
File parameters: Copy: 1 of 1 Spacing:SINGLE File format:ASCII Print mode:ASCII /DELETE

Faint header text at the top of the page, possibly containing a title or reference number.

Second line of faint text, likely a subtitle or secondary reference.

Large block of very faint text, possibly a list or a paragraph of content.

Text block in the middle of the page, appearing as a separate section or entry.

Text block below the middle, continuing the list or content.

Text block at the bottom of the page, possibly a footer or concluding text.

.MAIN. CROSS 6(54) 11-SEP-82 20:29
FORTH.M65 Table of contents

1	HISTORY & DOCUMENTATION
31	System Equates
140	Format Disk
296	Disk Drivers - Screen Read Routine
358	Disk Drivers - Screen Write Routine
458	Disk Drivers - Sub-Routines
746	Disk Drivers - Command Tables
823	Error Routine
840	VARIABLES

```
1          .SBTTL HISTORY & DOCUMENTATION
2          .TITLE FORTH SUPPORT
3
4          ; Copyright by:
5          ; Keith A. Sproul
6          ; 1368 Noah Road
7          ; North Brunswick,
8          ; New Jersey 08902
9          ; (201) 246-3749
10
11         ; STARTED SPRING 1982
12         ; 15-JUL-82 GOT BACK WORKING ON FORTH AGAIN
13
14         ; DISK FORMAT:
15         ; 08-MAY-82 STARTED PROGRAM
16
17         ; TRACK SECTORS BYTES/SECTOR COMMENTS
18         ; -----
19         ; 0 26 256 CODOS COMPATIBLE
20         ; 1-77 16 512 8K/TRACK
21
22
23         ; TRACK 0 CODOS COMPATIBLE (00-19)
24         ; 1-76 (00-0F)
25         ; 16 SECTORS/TRACK
26         ; 512 BYTES/SECTOR
27         ; GPL SAME AS IF 26 SECTORS
28         ; EACH TRACK IS SKEWED by FIVE
29
```

```

31          .SBTTL System Equates
32
33          .LIST  CND
34
35          0001      KSV      =      1          ; KSPROUL VERSION
36          0000      KIM      =      0          ; KIM VERSION
37          0000      AIM      =      0          ; AIM VERSION
38          0000      KIMA     =      0          ; KIM USER AT $A000
39
40          01        .IFEQ   KSV-1          ; K-SPROUL VERSION
41          .IFT
42          C000      SYSRAM  =      $C000
43          8000      USRRAM  =      $8000
44          A000      VM.ORG  =      $A000
45          2000      ORG     =      $2000
46          00        .ENDC
47
48          01        .IFEQ   KIM-1          ; STANDARD KIM VERSION
49          .IFT
50          SYSRAM  =      $C000
51          USRRAM  =      $6000
52          VM.ORG  =      $A000
53          ORG     =      $2000
54          00        .ENDC
55
56
57          01        .IFEQ   KIMA-1        ; STANDARD KIM VERSION
58          .IFT
59          SYSRAM  =      $C000
60          USRRAM  =      $A000
61          VM.ORG  =      $3000
62          ORG     =      $2000
63          00        .ENDC
64
65          01        .IFEQ   AIM-1        ; STANDARD AIM VERSION
66          .IFT
67          SYSRAM  =      $8000
68          USRRAM  =      $4000
69          VM.ORG  =      $6000
70          ORG     =      $2000
71          00        .ENDC
72
73          8000      DMABUF  =      USRRAM+$0000 ; DMA BUFFER
74          8000      FORMDA  =      USRRAM+$0000 ; FORMAT DATA AREA
75

```

```

77
78          .LIST    ME
79          .MCALL   DSKEQU
80          0000
(1)          ;***** FLOPPY DISK CONTROLLER REGISTER ADDRESSES *****
(1)
(1)          CG03      CODOS   =      SYSRAM+$0603      ; CODOS WARM START ADDRESS
(1)          DFOO      DSKIPL  =      SYSRAM+$1FOO      ; DISK BOOT ENTRY POINT
(1)          DFE8      FDCIRQ  =      SYSRAM+$1FE8      ; K-1013 INTERRUPT REG.          (READ-ONLY)
(1)          DFE8      FDCHWC  =      SYSRAM+$1FE8      ; DMA MODE (BO) WRITE PROT (B1) (WRITE-ONLY)
(1)          DFEA      FDCDMA  =      SYSRAM+$1FEA      ; FDC DMA ADDRESS REGISTER
(1)          DFEE      FDCMSR  =      SYSRAM+$1FEE      ; FDC NEC-765 MAIN STATUS REGISTER
(1)          DFEF      FDCDR   =      SYSRAM+$1FEF      ; FDC NEC-765 DATA REGISTER
(1)
(1)          ;*****
81
82          FO06      CTYIN   =      $FO06
83          FO09      CTYSPA  =      $FO09
84          FO0C      CTYOUT  =      $FO0C
85          FO0F      GETBYT  =      $FO0F
86          FO12      PRTBYT  =      $FO12
87          FO15      CRLF    =      $FO15
88
89          B800      .      =      $B800
90
91
92          B800      4C      54      B8      FORTH:  JMP      FORMAT      ; FORMAT DISK
93          B803      4C      0F      B8              JMP      TREAD        ; READ SCREEN
94          B806      4C      27      B8              JMP      TWRITE       ; WRITE SCREEN
95          B809      4C      55      B9              JMP      RDSCRN       ; $B809
96          B80C      4C      CA      B9              JMP      WRSCRN       ; $B80C
97
98          B80F      20      15      FO      TREAD:  JSR      CRLF          ; TEST-READ
99          B812      A9      52              LDA      #'R
100         B814      20      0C      FO      JSR      CTYOUT
101         B817      20      09      FO      JSR      CTYSPA
102         B81A      20      3F      B8      JSR      GETNO
103         B81D      A9      80              LDA      #DMABUF^
104         B81F      8D      07      B0      STA      ADDR+1
105         B822      20      55      B9      JSR      RDSCRN
106         B825      00              BRK
107         B826      00              BRK
108
109         B827      20      15      FO      TWRITE: JSR      CRLF          ; TEST-WRITE
110         B82A      A9      57              LDA      #'W
111         B82C      20      0C      FO      JSR      CTYOUT
112         B82F      20      09      FO      JSR      CTYSPA
113         B832      20      3F      B8      JSR      GETNO
114         B835      A9      80              LDA      #DMABUF^
115         B837      8D      07      B0      STA      ADDR+1
116         B83A      20      CA      B9      JSR      WRSCRN
117         B83D      00              BRK
118         B83E      00              BRK
119
120

```



```
121 B83F A9 23          GETNO: LDA    #'#  
122 B841 20 OC FO      JSR    CTYOUT  
123 B844 20 OF FO      JSR    GETBYT  
124 B847 8D 05 BO      STA    SCRNO+1  
125 B84A 20 OF FO      JSR    GETBYT  
126 B84D 8D 04 BO      STA    SCRNO+0  
127 B850 60            RTS
```

```
128  
129          ;.SBTTL Disk Debug Routines
```

```
130  
131          ;          .MCALL FDCRST          ; FDC RESET  
132          ;          FDCRST  
133  
134          ;          .MCALL PRESLT         ; PRINT FDC RESULTS  
135          ;          PRESLT  
136  
137  
138
```

```

140          .SBTTL  Format Disk
141
142
143          0001          NSTAG  =          1          ; NUMBER OF SECTORS TO STAGGER PER TRACK
144
145          ; FORMAT PROGRAM:
146          ; FORMATS VMDSYS DISKS ONLY          ALSO USED FOR FORTH.
147
148  B851  00          FABORT: BRK
149  B852  00          BRK
150  B853  00          BRK
151
152  B854  20  15  FO  FORMAT: JSR  CRLF
153  B857  A9  3F  FO          LDA  #'?
154  B859  20  0C  FO          JSR  CTYOUT
155  B85C  20  06  FO          JSR  CTYIN
156  B85F  C9  59  FO          CMP  #'Y
157  B861  DO  EE          BNE  FABORT
158  B863  D8          CLD          ; INSURE BINARY ARITHMETIC
159  B864  A9  01          LDA  #1
160  B866  8D  00  BO          STA  DRIVE
161  B869  AD  00  BO          LDA  DRIVE          ; GET DRIVE NUMBER
162  B86C  20  AC  BA          JSR  RECAL          ; RECALIBRATE IT TO TRACK 0
163  B86F  90  03          BCC  RCL.OK
164  B871  4C  1C  B9          JMP  FMT.ER          ; OUT-OF-RANGE BRANCH
165  B874  A9  00          RCL.OK: LDA  #0          ; SET DMA READ MODE, UNPROTECT SYSRAM
166  B876  8D  E8  DF          STA  FDCHWC
167  B879  A9  00          LDA  #0          ; INITIALIZE TRACK NUMBER
168  B87B  8D  01  BO          STA  TRACK
169  B87E  8D  EE  BB          STA  SIDE          ; SET TO SIDE 0
170
171          ; CREATE FORMATTING DATA AREA FOR TRACK ZERO
172
173  B881  A2  00          LDX  #0          ; INITIALIZE SECTOR NUMBER
174  B883  A0  00          LDY  #0          ; INITIALIZE POINTER INTO FORMAT DATA AREA
175  B885  AD  01  BO  FORM1: LDA  TRACK          ; PUT TRACK #
176  B888  99  00  80          STA  FORMDA,Y
177  B88B  C8          INY
178  B88C  AD  EE  BB          LDA  SIDE          ; SIDE #
179  B88F  99  00  80          STA  FORMDA,Y
180  B892  C8          INY
181  B893  8A          TXA          ; SET SECTOR #
182  B894  99  00  80          STA  FORMDA,Y
183  B897  C8          INY
184  B898  AD  D1  BB          LDA  FMTOCM+3          ; NUMBER OF BYTES PER SECTOR
185  B89B  99  00  80          STA  FORMDA,Y          ; INTO FORMAT DATA AREA IN MEMORY
186  B89E  C8          INY
187  B89F  E8          INX          ; INCREMENT SECTOR #
188  B8A0  CO  68          CPY  #26*4          ; TEST IF 26 SECTORS GONE THROUGH
189  B8A2  DO  E1          BNE  FORM1          ; CONTINUE GENERATING DATA IF NOT
190  B8A4  20  13  B9          JSR  FMTDRS          ; SET UP DRIVE&HEAD
191  B8A7  8D  DO  BB          STA  FMTOCM+2
192

```

```

194
195 B8AA A2 24 LDX #FMTOCM-DSKCMD
196 B8AC 20 20 B9 JSR FMT.SR ; FORMAT TRACK ZERO
197 B8AF B0 6B BCS FMT.ER
198 B8B1 EE 01 B0 INC TRACK ; POINT TO TRACK 1
199 B8B4 A9 00 LDA #0
200 B8B6 8D 02 B0 STA SECTOR ; SECTOR #
201
202 B8B9 AD 00 B0 FMT.LP: LDA DRIVE ; GET DRIVE #
203 B8BC AE 01 B0 LDX TRACK ; GET TRACK #
204 B8BF 20 D2 BA JSR SEEK ; SEEK TO NEXT TRACK
205
206 B8C2 A0 00 FORM2: LDY #0 ; SET UP FORMAT DATA
207 B8C4 AE 02 B0 LDX SECTOR ; GET SECTOR #
208 B8C7 AD 01 B0 FORM3: LDA TRACK
209 B8CA 99 00 80 STA FORMDA,Y
210 B8CD C8 INY
211 B8CE AD EE BB LDA SIDE
212 B8D1 99 00 80 STA FORMDA,Y
213 B8D4 C8 INY
214 B8D5 AD 02 B0 LDA SECTOR
215 B8D8 29 0F AND #$0F ; MASK OFF ALL BUT NEEDED BITS
216 B8DA 99 00 80 STA FORMDA,Y ; SET SECTOR #
217 B8DD C8 INY
218 B8DE AD D8 BB LDA FMTCMD+3 ; GET BYTES/SECTOR CODE
219 B8E1 99 00 80 STA FORMDA,Y
220 B8E4 C8 INY
221 B8E5 EE 02 B0 INC SECTOR ; BUMP UP SECTOR #
222 B8E8 C0 40 CPY #16*4 ; CHECK IF AT END OF TRACK
223 B8EA D0 DB BNE FORM3
224 B8EC AD 02 B0 LDA SECTOR ; BUMP UP BY NSTAG (-5)
225 B8EF 18 CLC
226 B8F0 69 01 ADC #NSTAG ; NUMBER TO STAGER/TRACK
227 B8F2 8D 02 B0 STA SECTOR
228 B8F5 20 13 B9 JSR FMTDRS ; SET UP DRIVE&HEAD
229 B8F8 8D D7 BB STA FMTCMD+2
230
231 B8FB A2 2B LDX #FMTCMD-DSKCMD
232 B8FD 20 20 B9 JSR FMT.SR ; FORMAT TRACK
233 B900 B0 1A BCS FMT.ER
234 B902 EE 01 B0 INC TRACK
235 B905 AD 01 B0 LDA TRACK
236 B908 C9 4D CMP #77
237 B90A D0 AD BNE FMT.LP
238 B90C AD 00 B0 LDA DRIVE
239 B90F 20 AC BA JSR RECAL
240 B912 60 RTS
241
242
243 B913 AD EE BB FMTDRS: LDA SIDE ; SET UP SECOND BYTE OF FORMAT COMMAND
244 B916 0A ASL A ; ACCORDING TO DRIVE NUMBER AND SIDE
245 B917 0A ASL A
246 B918 OD 00 B0 ORA DRIVE
247 B91B 60 RTS
248

```

249
250
251
252
253
254
255

B91C
B91C A9 2B
B91E 38
B91F 60

FMT.ER: LDA #FMTERR
SEC
RTS

; SO STACK WON'T WRAP AROUND
; ERROR WHILE FORMATING

```

257
258
259           ;      SET UP THE FORMAT COMMAND
260
261 B920  8A           FMT.SR: TXA           ;      SAVE DISK CMD TABLE OFFSET
262 B921  48           PHA
263 B922  A9  80      LDA  #FORMDA^       ;      SET DMA ADDRESS FOR FORMATTING DATA
264 B924  20  76  BB  JSR  DMASET         ;      CALCULATE CORRESPONDING DMA ADDRESS
265                                           ;      AND PUT INTO DMA ADDRESS REGISTER
266 B927  B0  F3      BCS  FMT.ER
267 B929  68         PLA
268 B92A  AA         TAX
269 B92B  20  0A  BB  JSR  COMAND          ;      SEND THE COMMAND TO THE DISK CONTROLLER
270 B92E  AD  E8  DF  FMTSR1: LDA  FDCIRQ   ;      WAIT UNTIL INTERRUPT REQUEST FROM FDC
271 B931  30  FB      BMI  FMTSR1
272 B933  20  30  BB  JSR  RESULT         ;      READ STATUS INTO STATUS AREA
273 B936  AD  DE  BB  LDA  DSKSTS+0       ;      CHECK STATUS REGISTER 0
274 B939  29  D8      AND  #$D8          ;      MASK OUT NON-ERROR BITS
275 B93B  DO  DF      BNE  FMT.ER         ;      GO TO ERROR IF ANY OF REMAINDER SET
276 B93D  AD  DF  BB  LDA  DSKSTS+1       ;      CHECK STATUS REGISTER 1
277 B940  29  35      AND  #$35          ;      MASK OUT NON-ERROR BITS
278 B942  DO  D8      BNE  FMT.ER         ;      GO TO ERROR IF ANY OF REMAINDER SET
279 B944  AD  E0  BB  LDA  DSKSTS+2       ;      CHECK STATUS REGISTER 2
280 B947  29  33      AND  #$33          ;      MASK OUT NON-ERROR BITS
281 B949  DO  D1      BNE  FMT.ER         ;      GO TO ERROR IF ANY OF REMAINDER SET
282
283 B94B  A0  00      LDY  #0             ;      WAIT AWHILE BEFORE SEEKING TO PREVENT
284 B94D  EA           FMTSR2: NOP         ;      OVERLAP WITH ERASE CURRENT ON DISK DRIVE
285 B94E  88         DEY
286 B94F  DO  FC      BNE  FMTSR2
287 B951  A9  00      LDA  #0
288 B953  18         CLC
289 B954  60         RTS
290
291
292
293
294

```

```

296 .SBTTL Disk Drivers - Screen Read Routine
297
298 ; READ SCREEN
299 B955 20 6A BA RDSCRN: JSR DSKINI ; DSKINI CALLS SPECIFY
300 B958 AD 00 BO LDA DRIVE
301 B95B 20 8C BA JSR SDRVST ; SENSE DRIVE STATUS
302 B95E A9 00 LDA #0
303 B960 8D EF BB STA ERRCNT
304 B963 8D 08 BO STA DSKERR
305 B966 20 38 BA JSR SCRNTS ; CALC TRACK/SECTOR #
306 B969 BO 5B BCS RDSFER ; READ-SCREEN-FATAL ERROR
307 B96B AD 07 BO RSRTRY: LDA ADDR+1 ; READ-SECTOR RE-TRY
308 B96E 20 76 BB JSR DMASET ; SET THE DMA PAGE #
309 B971 BO 53 BCS RDSFER ; READ-SCREEN-FATAL ERROR
310 B973 AD 00 BO LDA DRIVE ; GET DRIVE #
311 B976 AE 01 BO LDX TRACK ; GET TRACK #
312 B979 29 07 AND #$07 ; JUST TO BE SAFE
313 B97B 48 PHA ; SAVE DRIVE #
314 B97C 8D BC BB STA RDS.CM+2 ; DRV
315 B97F 8E BD BB STX RDS.CM+3 ; C
316 B982 4A LSR A ; ISOLATE HEAD # & PUT INTO H FIELD
317 B983 4A LSR A
318 B984 8D BE BB STA RDS.CM+4 ; H
319 B987 68 PLA ; GET DRIVE #
320 B988 20 D2 BA JSR SEEK ; SEEK TO CORRECT TRACK
321 B98B BO 2B BCS RDSERR ; BRANCH IF ERROR
322 B98D A9 01 LDA #$01 ; SET DMA DIRECTION TO READ
323 B98F 8D E8 DF STA FDCHWC
324 B992 AE 02 BO LDX SECTOR
325 B995 8E BF BB STX RDS.CMD+5 ; SET as 1st SECTOR TO READ
326 B998 E8 INX ; BUMP UP FOR TWO SECTORS
327 B999 8E C1 BB STX RDS.CMD+7 ; & AS LAST SECTOR
328 B99C A2 10 LDX #RDS.CM-DSKCMD ; RELATIVE ADDRESS OF CMD
329 B99E 20 0A BB JSR COMMAND
330 B9A1 BO 15 BCS RDSERR ; BRANCH IF ERROR
331
332 B9A3 AD E8 DF RDIRQW: LDA FDCIRQ ; WAIT FOR IRQ FROM FDC
333 B9A6 30 FB BMI RDIRQW
334 B9A8 20 30 BB JSR RESULT ; GET FDC RESULTS
335 B9AB BO 0B BCS RDSERR
336 B9AD 20 54 BB JSR RERROR ; CHECK IF ERROR
337 ; RETURN WITH ERROR CONDITION IN CARRY
338 ; CARRY = 1 = ERROR
339 ; CARRY = 0 = OKAY
340 B9B0 BO 06 BCS RDSERR
341 B9B2 A9 00 LDA #0 ; RETURN OKAY
342 B9B4 8D 08 BO STA DSKERR
343 B9B7 60 RTS
344
345 B9B8 8D 08 BO RDSERR: STA DSKERR ; SAVE DISKERR
346 B9BB EE EF BB INC ERRCNT
347 B9BE AD EF BB LDA ERRCNT
348 B9C1 C9 05 CMP #5 ; ALLOW FOR 5 RE-TRYS
349 B9C3 90 A6 BLT RSRTRY
350 B9C5 60 RTS

```

FORTH SUPPORT
FORTH.M65

CROSS 6(54) 11-SEP-82 20:29 Page 4-1
Disk Drivers - Screen Read Routine

```
351  
352 B9C6 8D 08 B0 RDSFER: STA DSKERR ; READ-SCREEN-FATAL ERROR  
353 B9C9 60 RTS  
354  
355  
356
```

```

358 .SBTTL Disk Drivers - Screen Write Routine
359
360 ; WRITE SCREEN
361
362 B9CA 20 6A BA WRSCRN: JSR DSKINI
363 B9CD 20 97 BA JSR SPECFY
364 B9DO A9 00 LDA #0
365 B9D2 8D EF BB STA ERRCNT
366 B9D5 20 38 BA JSR SCRNTS
367 B9D8 B0 5A BCS WRSFER ; WRITE-SCREEN FATAL ERROR
368 B9DA AD 07 BO WSRTRY: LDA ADDR+1 ; WRITE-SECTOR RE-TRY
369 B9DD 20 76 BB JSR DMASET ; SET THE DMA PAGE #
370 B9E0 B0 52 BCS WRSFER ; WRITE-SCREEN FATAL ERROR
371 B9E2 AD 00 BO LDA DRIVE ; GET DRIVE #
372 B9E5 AE 01 BO LDX TRACK ; GET TRACK #
373 B9E8 29 07 AND #$07 ; JUST TO BE SAFE
374 B9EA 48 PHA ; SAVE
375 B9EB 8D C6 BB STA WRS.CM+2 ; DRV
376 B9EE 8E C7 BB STX WRS.CM+3 ; C
377 B9F1 4A LSR A ; ISOLATE HEAD # & PUT INTO H FIELD
378 B9F2 4A LSR A
379 B9F3 8D C8 BB STA WRS.CM+4 ; H
380 B9F6 68 PLA
381 B9F7 20 D2 BA JSR SEEK ; SEEK TO CORRECT TRACK
382 B9FA B0 2B BCS WRSERR
383 B9FC A9 00 LDA #$00 ; SET DMA DIRECTION TO WRITE
384 B9FE 8D E8 DF STA FDCCHWC
385 BAO1 AE 02 BO LDX SECTOR
386 BAO4 8E C9 BB STX WRS.CMD+5 ; SET as 1st SECTOR TO READ
387 BAO7 E8 INX ; BUMP UP FOR TWO SECTORS
388 BAO8 8E CB BB STX WRS.CMD+7 ; & AS LAST SECTOR
389 BAOB A2 1A LDX #WRS.CM-DSKCMD ; RELATIVE ADDRESS OF CMD
390 BAOD 20 0A BB JSR COMAND
391 BA10 B0 15 BCS WRSERR
392 BA12 AD E8 DF WRIRQW: LDA FDCIRQ ; WAIT FOR IRQ FROM FDC
393 BA15 30 FB BMI WRIRQW
394 BA17 20 30 BB JSR RESULT ; GET FDC RESULTS
395 BA1A B0 0B BCS WRSERR
396 BA1C 20 54 BB JSR RERROR ; CHECK IF ERROR
397 ; RETURN WITH ERROR CONDITION IN CARRY
398 ; CARRY = 1 = ERROR
399 ; CARRY = 0 = OKAY
400 BA1F B0 06 BCS WRSERR
401 BA21 A9 00 LDA #0
402 BA23 8D 08 BO STA DSKERR
403 BA26 60 RTS
404
405 BA27 8D 08 BO WRSERR: STA DSKERR
406 BA2A EE EF BB INC ERRCNT
407 BA2D AD EF BB LDA ERRCNT
408 BA30 C9 03 CMP #3 ; ONLY 3 WRITE RE-TRYS
409 BA32 90 A6 BLT WSRTRY
410
411 BA34 8D 08 BO WRSFER: STA DSKERR
412 BA37 60 RTS

```


FORTH SUPPORT
FORTH.M65

CROSS 6(54) 11-SEP-82 20:29 Page 5-1
Disk Drivers - Screen Write Routine

413
414

```
416  
417 BA38          SCRNTS:          ; CALCULATE SCREEN TRACK/SECTOR #S  
418          ; LDA #1  
419          ; STA DRIVE  
420          ; JSR CRLF  
421          ; LDA SCRNO+1  
422          ; JSR PRTBYT  
423          ; LDA SCRNO+0  
424          ; JSR PRTBYT  
425 BA38 AD 04 BO LDA SCRNO+0  
426 BA3B 29 07 AND #07          ; 8 SCREENS/TRACK  
427 BA3D 0A ASL A          ; *2 (2 SECTORS/SCREEN)  
428 BA3E 8D 02 BO STA SECTOR  
429 BA41 4E 05 BO LSR SCRNO+1          ; /2  
430 BA44 6E 04 BO ROR SCRNO+0  
431 BA47 4E 05 BO LSR SCRNO+1          ; /4  
432 BA4A 6E 04 BO ROR SCRNO+0  
433 BA4D 4E 05 BO LSR SCRNO+1          ; /8  
434 BA50 6E 04 BO ROR SCRNO+0  
435 BA53 AD 04 BO LDA SCRNO+0  
436 BA56 18 CLC  
437 BA57 69 01 ADC #1          ; TRACK 0 IS NOT USEABLE  
438 BA59 C9 4D CMP #77  
439 BA5B BO 05 BGE SCR.ER          ; SCREEN ERROR  
440 BA5D 8D 01 BO STA TRACK  
441          ; JSR CTYSPA  
442          ; LDA TRACK  
443          ; JSR PRTBYT  
444          ; JSR CTYSPA  
445          ; LDA SECTOR  
446          ; JSR PRTBYT  
447 BA60 18 CLC  
448 BA61 60 RTS  
449  
450 BA62 A9 4C BO SCR.ER: LDA #76  
451 BA64 8D 01 BO STA TRACK  
452 BA67 A9 31 LDA #SCRERR  
453 BA69 60 RTS  
454  
455  
456
```

```

458          .SBTTL  Disk Drivers - Sub-Routines
459
460
461          ;          .MCALL  DISKSR          ; DISK SUB-ROUTINES
462          ;          DISKSR
463          ; .MACRO  DISKSR
464
465          ; SUBROUTINES THAT DON'T CHANGE.
466          ; THIS INCLUDES THE FOLLOWING SUBROUTINES:
467
468          ;          DSKINI          INIT & RESET IF NEEDED THE FDC
469          ;          SDRVST          SENSE DRIVE STATUS
470          ;          RECAL          RECALIBRATE DRIVE IN A
471          ;          SEEK          SEEK DRIVE A TO TRACK X
472          ;          SPECFY          SET DISK PARAMETERS
473          ;          SINTST          SENSE INTERRUPT STATUS
474          ;          COMAND          COMMAND PHASE OF FDC
475          ;          RESULT          RESULT PHASE OF FDC
476          ;          RERROR          SENSE IF ANY ERRORS IN RESULT PHASE
477          ;          DMASET          SET DMA REG ACORDING TO PAGE # IN ACC
478
479          BA6A          DSKINI:          ; INIT the FDC & SYSTEM
480          BA6A      AD  EE  DF          LDA      FDCMSR          ; CHECK Main Status Registre
481          BA6D      10  FB          BPL      DSKINI          ; WAIT UNTIL Request for Master GOES TRUE
482          BA6F      29  10          AND      #$10          ; CHECK DIO
483          BA71      FO  O3          BEQ      .+5          ; IF NOT WANTING TO TALK
484          BA73      20  .30  BB          JSR      RESULT          ; ELSE LISTEN
485          BA76      A2  32          LDX      #INVCMD-DSKCMD ; 02-SEP-82
486          BA78      20  OA  BB          JSR      COMAND
487          BA7B      AD  E8  DF          DSKINW: LDA      FDCIRQ          ; WAIT FOR IRQ FROM FDC
488          BA7E      30  FB          BMI      DSKINW
489          BA80      20  30  BB          JSR      RESULT          ; GET FDC RESULTS
490
491          BA83      20  97  BA          JSR      SPECFY
492          BA86      A9  01          LDA      #$01
493          BA88      8D  E8  DF          STA      FDCHWC          ; TURN WRITE PROTECT OFF
494          BA8B      60          RTS          ; & DATA DIR = READ FROM DISK
495
496          ; SENSE DRIVE STATUS
497          BA8C      8D  B9  BB          SDRVST: STA      SDSTCM+2          ; PUT DRIVE # IN SENSE DRV STAT CMD 2nd BYTE
498          BA8F      A2  OD          LDX      #SDSTCM-DSKCMD ; SENSE DRIVE STATUS CMD OFFSET
499          BA91      20  OA  BB          JSR      COMAND
500          BA94      4C  30  BB          JMP      RESULT          ; GET RESULT
501          ; & RETURN
502
503
504

```

```
506
507           ; FLOPPY DISK RESET AND SPECIFY. THIS ROUTINE FIRST RESETS THE
508           ; FLOPPY DISK CONTROLLER AND THEN ESTABLISHES THE SEEK RATE, HEAD
509           ; LOAD TIME, HEAD UNLOAD TIME, AND DMA OPTION BY USING A SPECIFY
510           ; COMMAND. SPECIFY DATA MUST BE PREFORMATTED IN MEMORY AT SPECCM.
511
512
513 BA97 AD E8 DF SPECFY: LDA FDCIRQ ; TEST IF AN INTERRUPT IS PENDING
514 BA9A 30 03 BMI SPECF1 ; JUMP AHEAD IF NOT
515 BA9C 20 A4 BA JSR SINTST ; EXECUTE SENSE INTERRUPT STATUS COMMAND
516
517 BA9F A2 00 SPECF1: LDX #SPECCM-DSKCMD ; SET ADDRESS OF SPECIFY COMMAND
518 BAA1 4C 0A BB JMP COMAND ; EXECUTE THE SPECIFY COMMAND
519 ; & RETURN
520
521
522 BAA4 SINTST:
523 BAA4 A2 04 LDX #SNSICM-DSKCMD ; OFF SET FOR SENSE INTERUPT STATUS CMD
524 BAA6 20 0A BB JSR COMAND ; EXECUTE THE CMD
525 BAA9 4C 30 BB JMP RESULT ; GET THE RESULTS
526 ; & RETURN
527
528
```

```

530
531 ; FLOPPY DISK RECALIBRATE
532 ; THIS ROUTINE SEEKS THE DRIVE NUMBER LOADED INTO A TO TRACK ZERO
533 ; REGARDLESS OF WHERE IT CURRENTLY IS.
534 ; THE STATUS BYTES AT DSKSTS ARE MODIFIED AS A RESULT OF THE
535 ; RECALIBRATE OPERATION.
536 ;
537 ; RECALIBRATION IS PERFORMED BY ISSUEING A RECALIBRATE COMMAND,
538 ; WAITING FOR INTERRUPT REQUEST TO GO TRUE, AND PERFORMING A
539 ; SENSE INTERRUPT STATUS COMMAND.
540
541 BAAC 29 03 RECAL: AND #03 ; PUT DRIVE NUMBER INTO SECOND BYTE OF
542 BAAE 8D B2 BB STA RECLCM+2 ; RECALIBRATE COMMAND
543 BAB1 A2 06 LDX #RECLCM-DSKCMD ; EXECUTE THE RECALIBRATE COMMAND
544 BAB3 20 0A BB JSR COMAND
545 BAB6 AD E8 DF RECAL1: LDA FDCIRQ ; READ DISK CONTROLLER INTERRUPT REQUEST
546 ; (FDC IRQ JUMPER REMOVED or IRQ DISABLED)
547 BAB9 30 FB BMI RECAL1 ; WAIT UNTIL THE FDC REQUESTS AN INTERRUPT
548 BABB 20 A4 BA JSR SINTST ; EXECUTE SENSE INTERRUPT STATUS COMMAND
549
550 BABE AD DF BB LDA DSKSTS+1 ; LOOK AT PRESENT CYLINDER NUMBER
551 BAC1 DO 0B BNE RECALE ; ERROR IF NON-ZERO
552 BAC3 AD DE BB LDA DSKSTS+0 ; LOOK AT ST-0 STATUS REGISTER
553 BAC6 29 F8 AND #F8 ; DELETE DON'T CARE BITS
554 BAC8 C9 20 CMP #20 ; REMAINING BITS MUST BE $20
555 BACA DO 02 BNE RECALE ; GO TO ERROR RETURN
556 BACC 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
557 BACD 60 RTS
558
559 BACE A9 2A RECALE: LDA #RCLERR ; SET RECALIBRATE ERROR
560 BADO 38 SEC
561 BAD1 60 RTS
562

```

```

564
565 ; FLOPPY DISK SEEK
566 ; THIS ROUTINE SEEKS THE DRIVE (A) TO TRACK (X)
567 ; THE STATUS BYTES AT DSKSTS ARE MODIFIED AS A RESULT OF THE
568 ; SEEK OPERATION.
569 ; SEEK IS PERFORMED BY ISSUING A SEEK COMMAND,
570 ; WAITING FOR INTERRUPT REQUEST TO GO TRUE, AND PERFORMING A
571 ; SENSE INTERRUPT STATUS COMMAND.
572
573 ; A = DRIVE
574 ; X = TRACK
575
576 BAD2 29 03 SEEK: AND #$03 ; FORMAT DRIVE NUMBER INTO SECOND BYTE OF
577 BAD4 8D B5 BB STA SEEKCM+2 ; SEEK COMMAND AND SELECT HEAD 0 2nd BYTE
578 BAD7 8E B6 BB STX SEEKCM+3 ; STORE NEW CYLINDER NUMBER INTO 3rd BYTE
579 BADA 8A TXA ; SAVE NEW CYLINDER NUMBER ON THE STACK
580 BADB 48 PHA
581 BADC A2 09 LDX #SEEKCM-DSKCMD ; EXECUTE THE SEEK COMMAND
582 BADE 20 0A BB JSR COMAND
583
584 BAE1 AD E8 DF SEEK1: LDA FDCIRQ ; READ DISK CONTROLLER INTERRUPT REQUEST
585 BAE4 30 FB BMI SEEK1 ; WAIT UNTIL THE FDC REQUESTS AN INTERRUPT
586 BAE6 20 A4 BA JSR SINTST ; EXECUTE SENSE INTERRUPT STATUS COMMAND
587
588 BAE9 68 PLA ; COMPARE PRESENT CYLINDER NUMBER WITH
589 BAEA CD DF BB CMP DSKSTS+1 ; DESIRED CYLINDER NUMBER
590 BAED DO 09 BNE SEEKER ; ERROR IF NOT THE SAME
591 BAEF AD DE BB LDA DSKSTS+0 ; LOOK AT ST-0 STATUS REGISTER
592 BAF2 29 F8 AND #$F8 ; DELETE DON'T CARE BITS
593 BAF4 C9 20 CMP #$20 ; REMAINING BITS MUST BE $20
594 BAF6 FO OE BEQ SEEKOK ; GO TO ERROR RETURN
595
596 BAF8 AD DE BB SEEKER: LDA DSKSTS+0 ; GET ST-0
597 BAFB 29 03 AND #$03 ; CHECK IF CURRENT DRIVE AT FAULT
598 BAFD CD B5 BB CMP SEEKCM+2
599 BBOO DO 04 BNE SEEKOK ; IF NOT, THEN ASSUME OKAY
600 ; ELSE FALL THROUGH
601 BBO2 A9 29 LDA #SEKERR ; DISK DRIVE SEEK ERROR
602 BBO4 38 SEC
603 BBO5 60 RTS
604
605 BBO6 A9 00 SEEKOK: LDA #0 ; CLEAR CARRY FOR NORMAL RETURN
606 BBO8 18 CLC
607 BBO9 60 RTS
608
609

```

```

611          ; SEND COMMAND TO FLOPPY DISK CONTROLLER
612          ; ENTER WITH RELATIVE ADDRESS OF COMMAND BYTES IN X (RELATIVE TO
613          ; DSKCMD) AND NUMBER OF BYTES IN COMMAND IN Y.
614          ; ROUTINE SENDS THE BYTES TO THE FLOPPY DISK CONTROLLER AND
615          ; RETURNS WITH THE CARRY FLAG OFF.
616          ; IF AN ERROR IS DETECTED, THE CARRY FLAG IS ON.
617          ; IF THE CONTROLLER IS BUSY, THE ERROR RETURN IS TAKEN
618
619 BBOA          COMAND:
620 BBOA AD EE DF          LDA FDCMSR          ; LOOK AT MAIN STATUS REGISTER
621 BBOD 29 10          AND #$10          ; LOOK AT ALL OF THE BUSY BITS
622 BBOF DO 1B          BNE CMDPHE          ; ERROR IF EXECUTING PREVIOUS
623          ; COMMAND (EXCEPT SEEK)
624 BB11 BC AA BB          LDY DSKCMD,X          ; GET NUMBER OF BYTES IN CMD <KS>
625 BB14 E8          INX          ; POINT TO NEXT ENTRY IN CMD TABLE <KS>
626 BB15 AD EE DF          CMDPH1: LDA FDCMSR          ; LOOK AT MAIN STATUS REGISTER
627 BB18 10 FB          BPL CMDPH1          ; WAIT UNTIL REQUEST FOR MASTER GOES TRUE
628 BB1A 29 40          AND #$40          ; TEST DATA DIRECTION BIT
629 BB1C DO OE          BNE CMDPHE          ; ERROR IF FDC WANTS TO TALK
630 BB1E BD AA BB          LDA DSKCMD,X          ; GET A COMMAND BYTE
631 BB21 8D EF DF          STA FDCDR          ; STORE IT IN THE DISK CONTROLLER
632 BB24 E8          INX          ; POINT S TO NEXT COMMAND BYTE
633 BB25 88          DEY          ; DECREMENT COMMAND BYTE COUNT
634 BB26 DO ED          BNE CMDPH1          ; GO TRANSFER NEXT BYTE IF NOT DONE
635 BB28 18          CLC          ; CLEAR CARRY FOR NORMAL RETURN
636 BB29 A9 00          LDA #0
637 BB2B 60          RTS
638
639 BB2C A9 2E          CMDPHE: LDA #FCMDER          ; FDC COMMAND ERROR
640 BB2E 38          SEC
641 BB2F 60          RTS
642

```

```

644 ; RECEIVE STATUS FROM FLOPPY DISK CONTROLLER
645 ; STATUS BYTES ARE STORED SEQUENTIALLY IN MEMORY STARTING AT
646 ; DSKSTS, THE NUMBER READ IS DETERMINED BY THE FDC BUSY STATUS
647 ; ROUTINE READS THE STATUS BYTES IN THE RESULT PHASE AND RETURNS
648 ; WITH THE CARRY FLAG OFF.
649
650
651 BB30 A2 00 RESULT: LDX #0 ; INIT INDEX POINTER
652 BB32 AD EE DF RESULT1: LDA FDCMSR ; LOOK AT MAIN STATUS REGISTER
653 BB35 10 FB DF BPL RESULT1 ; WAIT UNTIL REQUEST FOR MASTER GOES TRUE
654 BB37 29 40 AND #$40 ; TEST DATA DIRECTION BIT (DIO)
655 BB39 FO 15 BEQ RSLPHE ; ERROR IF FDC WANTS TO LISTEN
656 BB3E AD EF DF LDA FDCDR ; GET A STATUS BYTE FROM THE DATA REGISTER
657 BB3E 9D DE BB STA DSKSTS,X ; PUT IF INTO MEMORY
658 BB41 E8 INX ; POINT X TO NEXT STATUS BYTE
659 BB42 EA NOP ; DAMN SLOW CONTROLLER CHIP! 12US RESPONSE
660 BB43 EA NOP ; TIME FROM READ TO VALID BUSY STATUS
661 BB44 EA NOP
662 BB45 A9 10 LDA #$10 ; LOOK AT BUSY BIT IN MAIN STATUS REGISTER
663 BB47 2D EE DF AND FDCMSR
664 BB4A DO E6 BNE RESULT1 ; GO FOR ANOTHER STATUS BYTE IF STILL BUSY
665 BB4C 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
666 BB4D A9 00 LDA #0
667 BB4F 60 RTS
668
669 BB50 A9 2F RSLPHE: LDA #FRSLER ; FDC RESULT ERROR
670 BB52 38 SEC
671 BB53 60 RTS
672

```



```

674
675 BB54 AD DE BB RERROR: LDA DSKSTS+0 ; CHECK STATUS REGISTER 0
676 BB57 29 D8 BB AND #$D8 ; MASK OUT NON-ERROR BITS
677 BB59 FO O9 BEQ RERR1 ; JUMP AHEAD IF NO OBVIOUS ERROR
678 BB5B C9 40 CMP #$40 ; TEST IF ABNORMAL TERMINATION ERROR
679 BB5D DO 15 BNE R..ERR ; TRUE ERROR IF NOT
680 BB5F AD DF BB LDA DSKSTS+1 ; IF ABNORMAL, TEST IF END OF CYLINDER
681 ; <KS> CHECK FOR $04
682 BB62 10 10 BPL R..ERR ; TRUE ERROR IF NOT, OK IF SO
683 BB64 AD DF BB RERR1: LDA DSKSTS+1 ; CHECK STATUS REGISTER 1
684 BB67 29 35 AND #$35 ; MASK OUT NON-ERROR BITS
685 BB69 DO O9 BNE R..ERR ; GO TO ERROR IF ANY OF REMAINDER SET
686 BB6B AD EO BB LDA DSKSTS+2 ; CHECK STATUS REGISTER 2
687 BB6E 29 33 AND #$33 ; MASK OUT NON-ERROR BITS
688 BB70 DO O2 BNE R..ERR ; GO TO ERROR IF ANY OF REMAINDER SET
689 BB72 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
690 BB73 60 RTS
691
692 BB74 38 R..ERR: SEC
693 BB75 60 RTS
694

```

```

696           ;      SUBROUTINE TO ACCEPT THE PAGE ADDRESS OF A MEMORY BUFFER AND
697           ;      SET THE DMA ADDRESS REGISTER WITH THE APPROPRIATE VALUE.  THE
698           ;      BUFFER ADDRESS MUST BE IN THE DISK CONTROLLER RAM.
699           ;      RETURN WITH CARRY CLEAR IF ADDRESS IS OK, SET IF ADDRESS IS
700           ;      INVALID.
701           ;      FOR THIS ROUTINE TO WORK, THE ORIGIN OF THE USER RAM MUST BE
702           ;      EQUATED TO USRRAM AND THE ORIGIN OF THE SYSTEM RAM MUST BE
703           ;      EQUATED TO SYSRAM.
704           ;      ENTER WITH BUFFER PAGE ADDRESS IN A, EXIT WITH DMA ADDRESS IN A
705           ;      AND THE DMA ADDRESS REGISTER.
706
707 BB76 C9 80      DMASET: CMP      #USRRAM/256      ; COMPARE WITH BEGINNING OF USER RAM
708 BB78 90 OE      BCC      DMAST1      ; JMP IF LESS THAN USER RAM
709 BB7A C9 AO      CMP      #USRRAM/256+32      ; COMPARE WITH END OF USER RAAM
710 BB7C B0 OA      BCS      DMAST1      ; JUMP IF NOT IN USER RAM
711 BB7E 38         SEC          ; IF IN USER RAM, COMPUTE RELATIVE PAGE
712 BB7F E9 80      SBC      #USRRAM/256      ; ADDRESS IN USER RAM
713
714 BB81 48         PHA          ; SAVE RESULT
715 BB82 A9 00      LDA      #USRRAM/32&FFF      ; TEST IF USRRAM IS ON AN ODD 4K BOUNDARY
716 BB84 30 12      BMI      DMAST2      ; JUMP ON ODD BOUNDARY
717 BB86 10 16      BPL      DMAST3      ; JUMP ON EVEN BOUNDARY
718 BB88 C9 CO      DMAST1: CMP      #SYSRAM/256      ; COMPARE WITH BEGINNING OF SYSTEM RAM
719 BB8A 90 1A      BCC      DMASTE      ; ERROR IF NOT IN DISK CONTROLLER RAM
720 BB8C C9 DF      CMP      #SYSRAM/256+31      ; COMPARE WITH END OF USABLE SYSTEM RAM
721 BB8E B0 16      BCS      DMASTE      ; ERROR IF NOT IN DISK CONTROLLER RAM
722 BB90 38         SEC          ; IF IN SYSTEM RAM, COMPUTE RELATIVE PAGE
723 BB91 E9 AO      SBC      #SYSRAM/256-32      ; ADDRESS IN SYSTEM RAM
724 BB93 48         PHA          ; SAVE RESULT
725 BB94 A9 00      LDA      #SYSRAM/32&FFF      ; TEST IF SYSRAM IS ON AN ODD 4K BOUNDARY
726 BB96 10 06      BPL      DMAST3      ; JUMP ON EVEN BOUNDARY
727 BB98 68         DMAST2: PLA          ; RETRIEVE RELATIVE ADDRESS
728 BB99 49 10      EOR      #$10          ; FLIP BIT 6 (AFTER SHIFT) FOR ODD BOUNDARY
729 BB9B 4C 9F BB   JMP      DMAST4
730
731 BB9E 68         DMAST3: PLA          ; RETRIEVE RELATIVE ADDRESS
732 BB9F 0A         DMAST4: ASL      A          ; MULTIPLY RELATIVE PAGE ADDRESS BY 4
733 BBA0 0A         ASL      A
734 BBA1 8D EA DF   STA      FDCDMA      ; AND PUT RESULT IN DMA ADDRESS REGISTER
735 BBA4 18         CLC
736 BBA5 60         RTS
737
738
739 BBA6 A9 30      DMASTE: LDA      #DMAERR      ; DMA PAGE ERROR
740 BBA8 38         SEC
741 BBA9 60         RTS
742
743
744

```

```
746          .SBTTL  Disk Drivers - Command Tables
747
748          ;      LIST OF DISK COMMANDS, STARTS IN PROGRAM AREA BUT MUST BE IN
749          ;      RAM SO THAT CERTAIN BYTES OF THE COMMANDS CAN BE CHANGED
750
751  BBAA      DSKCMD:          ; START OF PREFORMATTED DISK COMMANDS
752  BBAA 03   SPECCM: .BYTE 3      ; #      ; THREE BYTES IN SPECIFY COMMAND
753  BBAB 03   .BYTE $03      ; CMD   ; SPECIFY COMMAND
754  BBAC AF   .BYTE $AF      ; SRT   ; SEEK SPEED=6ms HEAD UNLOAD TIME=240ms
755  BBAD 24   .BYTE $24      ; HLT   ; HEAD LOAD TIME=40ms DMA MODE
756
757  BBAE 01   SNSICM: .BYTE 1     ; #      ; 1 BYTE IN SENSE INT. CMD
758  BBAF 08   .BYTE $08      ; CMD   ; SENSE INTERRUPT STATUS COMMAND
759
760  BBBO 02   RECLCM: .BYTE 2     ; #      ; 2 BYTES IN RECALIBRATE CMD
761  BBB1 07   .BYTE $07      ; CMD   ; RECALIBRATE COMMAND
762  BBB2 00   .BYTE 00       ; DRV   ; DRIVE NUMBER IN BITS 0-1
763
764  BBB3 03   SEEKCM: .BYTE 3     ; #      ; 3 BYTES IN SEEK CMD
765  BBB4 0F   .BYTE $0F      ; CMD   ; SEEK COMMAND
766  BBB5 00   .BYTE 00       ; DRV   ; DRIVE NUMBER IN BITS 0-1, SIDE NUMBER B2
767  BBB6 00   .BYTE 00       ; NCN   ; NEW CYLINDER NUMBER
768
769  BBB7 02   SDSTCM: .BYTE 2     ; #      ; 2 BYTES IN SENSE DRIVE STATUS CMD
770  BBB8 04   .BYTE $04      ; CMD   ; SENSE DISK STATUS COMMAND
771  BBB9 00   .BYTE 00       ; DRV   ; DRIVE NUMBER IN BITS 0-1, SIDE NUMBER B2
772
773
```

```

775
776                                     ; READ SCREEN CMD
777 BBBA 09      RDS.CM: .BYTE 9      ; #      ; 9 BYTES IN READ DATA CMD
778 BBBB 46      .BYTE $46      ; CMD   ; READ DATA, MFM, READ DEL DATA
779 BBBC 00      .BYTE 00       ; DRV   ; HEAD ZERO, DRIVE NUMBER IN BITS 0-1
780 BBBD 00      .BYTE 00       ; C     ; NEEDS PRESENT CYLINDER NUMBER
781 BBBE 00      .BYTE 00       ; H     ; NEEDS HEAD NUMBER
782 BBBF 00      .BYTE 00       ; R     ; READ FROM SECTOR X
783 BBBCO 02     .BYTE $02      ; N     ; 512 BYTES/SECTOR
784 BBC1 00      .BYTE 00       ; EOT   ; TO SECTOR X
785 BBC2 0E      .BYTE $0E      ; GPL   ; GAP LENGTH FOR 26 SECTORS, 256 BYTES/SECT
786 BBC3 FF      .BYTE $FF      ; DTL   ; DATA LENGTH = $FF SINCE N IS NON-ZERO
787
788                                     ; WRITE SCREEN CMD
789 BBC4 09      WRS.CM: .BYTE 9      ; #      ; 9 BYTES IN READ DATA CMD
790 BBC5 45      .BYTE $45      ; CMD   ; READ DATA, MFM, READ DEL DATA
791 BBC6 00      .BYTE 00       ; DRV   ; HEAD ZERO, DRIVE NUMBER IN BITS 0-1
792 BBC7 00      .BYTE 00       ; C     ; NEEDS PRESENT CYLINDER NUMBER
793 BBC8 00      .BYTE 00       ; H     ; NEEDS HEAD NUMBER
794 BBC9 00      .BYTE 00       ; R     ; READ FROM SECTOR X
795 BBBCA 02     .BYTE $02      ; N     ; 512 BYTES/SECTOR
796 BBCB 00      .BYTE 00       ; EOT   ; TO SECTOR X
797 BBCC 0E      .BYTE $0E      ; GPL   ; GAP LENGTH FOR 26 SECTORS, 256 BYTES/SECT
798 BBCCD FF     .BYTE $FF      ; DTL   ; DATA LENGTH = $FF SINCE N IS NON-ZERO
799
800                                     ; FORMAT CMD FOR TRACK 0
801 BBCE 06      FMTOCM: .BYTE 6      ; #      ; 6 BYTES IN FORMAT COMMAND
802 BBCF 4D      .BYTE $4D      ; CMD   ; FORMAT A DOUBLE DENSITY TRACK COMMAND
803 BBDO 00      .BYTE 00       ; DRV   ; HEAD ZERO, DRIVE NUMBER IN BITS 0 AND 1
804 BBD1 01      .BYTE $01      ; N     ; 256 BYTES/SECTOR
805 BBD2 1A      .BYTE $1A      ; SC    ; 26 SECTORS/TRACK
806 BBD3 36      .BYTE $36      ; GPL   ; GAP LENGTH FOR 26 SECTORS, 256 BYTES/SECT
807 BBD4 00      .BYTE $00      ; D     ; DATA FIELD FILLER BYTE
808
809                                     ; FORMAT CMD FOR REST OF DISK
810 BBD5 06      FMTCMD: .BYTE 6      ; #      ; 6 BYTES IN FORMAT COMMAND
811 BBD6 4D      .BYTE $4D      ; CMD   ; FORMAT A DOUBLE DENSITY TRACK COMMAND
812 BBD7 00      .BYTE 00       ; DRV   ; HEAD ZERO, DRIVE NUMBER IN BITS 0 AND 1
813 BBD8 02      .BYTE $02      ; N     ; 512 BYTES/SECTOR
814 BBD9 10      .BYTE $10      ; SC    ; 16 SECTORS/TRACK
815 BBDA 36      .BYTE $36      ; GPL   ; GAP LENGTH FOR 26 SECTORS, 256 BYTES/SECT
816 BBDB 20      .BYTE $20      ; D     ; DATA FIELD FILLER BYTE <SPACE FOR FORTH>
817
818 BBDC 01      INVCMD: .BYTE 1      ; #      ; 1 BYTE IN INVALID COMMAND
819 BBDD 00      .BYTE 0        ; CMD   ; 00 IS AN INVALID COMMAND
820
821

```

```
823          .SBTTL Error Routine
824
825
826
827          0029          SEKERR =          41          ; Seek Error
828          002A          RCLERR =          42          ; Recalibrate Error
829          002B          FMTERR =          43          ; Format Error
830          002C          RDERR  =          44          ; Read Error
831          002D          WRERR  =          45          ; Write Error
832          002E          FCMDER =          46          ; FDC Comand Error
833          002F          FRSLER =          47          ; FDC Result Error
834          0030          DMAERR =          48          ; DMA Page Error
835          0031          SCRERR =          49          ; SCREEN Error
836
837
838
```

```
840          .SBTTL  VARIABLES
841
842
843 BBDE 0010      DSKSTS: .BLKB  16
844 BBEE 0001      SIDE:   .BLKB  1          ; USED IN FORMATING ONLY
845 BBFE 0001      ERRCNT: .BLKB  1          ; ERROR COUNT
846
847
848          B000          . =          $B000
849 B000 B001      DRIVE:  . = .+1          ; DISK DRIVE
850 B001 B002      TRACK:  . = .+1          ; CURRENT TRACK #
851 B002 B003      SECTOR: . = .+1          ; CURRENT SECTOR #
852 B003 B004      RDWR:   . = .+1          ; READ/WRITE FLAG (NOT USED)
853 B004 B006      SCRNO:  . = .+2          ; SCREEN #
854 B006 B008      ADDR:   . = .+2          ; DMA ADDRESS
855 B008 B009      DSKERR: . = .+1          ; DISK ERROR
856
857          B800          .END  FORTH
```


DSKEQU 79# 80

.BLKR	843	844	845											
.BYTE	752	753	754	755	757	758	760	761	762	764	765	766	767	769
	770	771	777	778	779	780	781	782	783	784	785	786	789	790
	791	792	793	794	795	796	797	798	801	802	803	804	805	806
	807	810	811	812	813	814	815	816	818	819				
.END	857													
.ENDC	46	54	63	71										
.IFEQ	40	48	57	65										
.IFT	41	49	58	66										
.LIST	33	78												
.MCALL	79													
.PAGE	76	193	256	415	505	529	563	610	643	673	695	774		
.SBTTL	1	31	140	296	358	458	746	823	840					
.TITLE	2													

Errors detected: 0

*,FORTH=FORTH

Run-time: 0 1 0 Seconds

Core used: 9K

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

* * * L P T S P L R u n L o g * * *

20:44:56 LPDAT LPTSPL version 104(16650) Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit
20:44:56 LPDAT Job FORTH sequence #1425 on Printer 0 [LOCAL] at 11-Sep-82 20:44:56
20:45:03 LPMSG Starting File PS:<KSPROUL>FORTH.LST.1
20:46:47 LPMSG Finished File PS:<KSPROUL>FORTH.LST.1
20:46:47 LPEND Summary: 32 Pages of Output
20:46:47 LPEND 12 Disk Pages Read
20:46:47 LPEND 14.350 Seconds CPU Time Used

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job FORTH Req #910 for KSPROUL Date 11-Sep-82 20:46:47 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

