

.REM _

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

IDENTIFICATION

PRODUCT CODE: AC-E785D-MC
PRODUCT NAME: CXRKBD0 RK611/RK06,RK07 MOD
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

.REM -

1. ABSTRACT

RKB IS AN IOMODX THAT EXERCISES RK06/RK07 DRIVES ON A RK611 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. IN DUAL PORT MODE THESE ARE DONE BACK AND FORTH BETWEEN TWO PORTS OF THE DRIVE. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RK06/RK07 DISK DRIVES WITH TWO RK611 CONTROLLERS

SOFTWARE: MUST USE THE FOLLOWING OR LATER REVISIONS OF MONITOR
STANDARD MONITOR: QABL
11/70 MONITOR: QAGC
SHORT MONITOR: QAED

STORAGE:: RKB REQUIRES:
1. DECIMAL WORDS: 2073
2. OCTAL WORDS: 04031
3. OCTAL BYTES: 10062

3. PASS DEFINITION

SINGLE PORT:

ONE PASS OF THE RKB MODULE CONSISTS OF 765 CYCLES OF THE BASIC TEST SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

DUAL PORT:

ONE PASS OF THE RKB MODULE CONSISTS OF 300 CYCLES OF THE BASIC TEST SEQUENCE. A-PORT DOES A WRITE, WRITE-CHECK, READ, DATA COMPARE ON THE CURRENT SECTOR (CALLED BLK1 THROUGHOUT PROGRAM) AND THEN WRITES BLK 0 WITH BIT 4 SET IN THE FLAG WORD (THE FIRST WORD OF BLOCK ZERO), AND THE SECOND WORD WITH THE CURRENT BLK. A-PORT THEN SITS IN A LOOP, PERIODICALLY READING BLK 0 TO SEE IF B-PORT HAS MODIFIED THE FLAG. B-PORT SITS IN A LOOP WAITING FOR THE FLAG IN BLK ZERO TO BE WRITTEN WITH BIT4 SET. WHEN IT IS B-PORT PICKS THE CURRENT BLK FROM THIS DATA READ FROM BLK 0 AND GOES TO THAT SECTOR AND READS THE DATA THAT A-PORT JUST WROTE. THEN B-PORT WRITES THE DATA BACK AND WRITE CHECKS IT. NEXT IT RE-WRITES BLK 0 WITH BIT4 CLEARED AND WITH BIT2 SET, INDICATING TO A-PORT THAT HE IS DONE. A-PORT AFTER READING BLK 0 AND SEEING THE FLAGS REVERSED, RE-READS THE DATA WHICH IT HAD WRITTEN AND AGAIN DOES AN IN-CORE COMPARE. THIS VERIFIES THE ABILITY OF A-PORT TO RE-READ DATA WHICH IT HAD ORIGINALLY WROTE BUT THAT HAD BEEN READ AND RE-WRITTEN

97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152

BY B-PORT. B-PORT DOES NOT DO ANY IN CORE COMPARISONS.

FAULTY SINCE A-PORT WAS JUST ABLE TO SUCCESSFULLY USE THIS BLK.

16 LOCATIONS STARTING AT BADSPT ARE REVERVED FOR ENTRY OF 16 BADSPOTS BY USER ON ERROR TYPEOUTS WHICH DUMP THE RK REGRISTERS, THE LAST ITEM TYPED IN THE TABLE IS THE CURRENT BLK NUMBER. ENTER THIS INTO THE BAD BLK TABLE TO AVOID ERRORS FROM KNOWN MEDIA BAD SPOTS. THIS TABLE ONLY MAKES SENSE ON THE A-PORT SIDE SINCE B-PORT ALWAYS GETS ITS BLK ADDRESSES FROM A-PORT. IF YOU MODIFY THE WRITE BUFFER SIZE, YOU MUST ADD SECTORS TO THE BAD BLK TABLE TO AVOID THE ERRORS. FOR EXAMPLE, IF BLK 3474 IS BAD AND YOU DOUPLE THE WRITE TRANSFER SIZE TO 1000 OCTAL BYTES YOU MUST ADD BLK 3473 TO THE TABLE SO THE PROGRAM DOES NO START A TRANSFER THAT WILL EXTEND ON INTO THE KNOWN BAD BLK.

4. EXECUTION TIME

ONE PASS OF RKB RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177440, VECTOR: 210, RR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

7. MODULE OPERATION

TEST SEQUENCE:

- A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
- B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
- C. GET A DRIVE ADDRESS
- D. IF THE NUMBER OF CYCLES IS COMPLETE FOR THIS DRIVE GO TO C ELSE, GET A DISK ADDRESS AND A FRESH BLOCK OF DATA
- E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
- F. DO A WRITE-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
- G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
- H. DO A DATA-CHECK -- IF ERRORS, REPORT AND CONTINUE
- I. IF END OF PASS, REPORT AND GO TO C, ELSE GO TO D

8. OPERATION OPTIONS

153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

SR1 BIT 0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR
IS ASSUMED AND THE DRIVE IS DROPPED
SR1 BIT 0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND
THE TESTING CONTINUES
SR1 BIT 2 SET(1):
ON ENCOUNTERING A BAD SECTOR ALWAYS PRINT A MESSAGE
SR1 BIT 2 CLEAR(0):
ON ENCOUNTERING A BAD SECTOR ONLY PRINT A MESSAGE IF
SECTOR IS NOT IN THE BAD SECTOR FILE
SR1 BIT 4 SET(1):
WILL EXERCISE PORT B IN DUAL PORT MODE OF OPERATION
SR1 BIT 4 CLEAR(0):
DUAL PORT WILL EXERCISE PORT A. IN SINGLE PORT THIS
BIT MUST BE 0 IN ORDER TO RUN.
SR1 BIT 6 SET(0):
WRITE/READ DATA STARTING AT A RANDOM SECTOR
SR1 BIT 6 CLEAR(1):
WRITE/READ DATA STARTING AT EVERY THIRD SECTOR
SR1 BIT 7 SET(1):
DUAL PORT MODE SELECTED
SR1 BIT 7 CLEAR(0):
SINGLE PORT MODE SELECTED.

9. NON-STANDARD PRINTOUTS

- A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN
THE DEC/K11 DOCUMENT
- B. ERROR MESSAGES DUMP THE CONTENTS OF 13 RK611 REGISTERS
IN THE FOLLOWING ORDER:

RCS1	RKWC	RKBA	RKDA	RKCS2	RKDS	RKER	RKASO
RKDC	RKECPS	RKECPT	BLK1				

ERRORS GENERATED IN RKER

ILF	; ILLEGEAL FUNCTION
SKI	; SEEK INCOMPLETE
NXF	; NON-EXECTUABLE FUNCTION
DRP	; DRIVE PARITY ERROR
FMT	; FORMAT
DTV	; DRIVE TYPE ERROR

```

209 ECH ;ERROR CORRECTION HARD
210 BSE ;BAD SECTOR ERROR
211 HVP ;HEADER VRC ERROR
212 COE ;CYLINDER OVERFLOW ERROR
213 IDA ;INVALID DISK ADDRESS ERROR
214 WLE ;WRITE LOCK
215 DTE ;DRIVE TIMING ERROR
216 OPI ;OPERATION INCOMPLETE
217 UNS ;DRIVE UNSAFE
218 DCK ;DATA CHECK
219
220 ERRORS GENERATED IN RKDS
221
222 ACL ;DRIVE AC POWER LOW
223 DCL ;DRIVE DC POWER LOW
224 DOT ;DRIVE OFF TRACK
225 WRL ;WRITE LOCK
226
227 ERRORS GENERATED IN RKCS2
228
229 UFE ;UNIT FIELD ERROR
230 MDS ;MULTIPLE DRIVE SELECT
231 PCE ;PROGRAMMING ERROR
232 NEV ;NON EXISTANT MEMORY
233 NED ;NON EXISTANT DRIVE
234 UPE ;UNIBUS PARITY ERROR
235 WCE ;WRITE CHECK ERROR
236 DLT ;DATA LATE
237
238 ERRORS GENERATED IN RKCS1
239
240 CDT ;CONTROLLER DRIVE TYPE
241 CTO ;CONTROLLER TIMEOUT
242 CFM ;CONTROLLER FORMAT
243 SPA ;SERIAL PARITY ERROR
244 CEP ;CONTROLLER ERROR

```

CAUTION: THE PK06 SUBSYSTEM WILL GENERATE TWO (2) INTERRUPTS WHEN ISSUED AN EXPLICIT SEEK COMMAND. WHEN NO HEAD MOTION IS REQUIRED ONLY ONE INTERRUPT REACHES THE PROCESSOR. HOWEVER, ON SYSTEMS THAT HAVE MANY DEVICES BEING EXERCISED THE SECOND INTERRUPT MAY OCCUR WHILE THE MODULE IS SERVICING THE FIRST (PRIORITY ZERO) WHICH CAN LEAD TO CONFUSION.

```

254 .NLIST MC,TTM,CND,MD
255 .LIST ME
256
257
258
259
260
261
262 .SBTTL CALL MONITOR FOR SETUP
263
264 ; TITLE RKBD DEC/X11 SYSTEM EXERCISER MODULE
265 ; DDXCUM VERSION 6 23-MAY-78
266 .LIST BIN
267
268 *****
269 BEGIN:
270 MODNAM: .ASCII /RKBD / ;MODULE NAME
271 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
272 ADDR: 177440+0 ;1ST DEVICE ADDR.
273 VECTOR: 210+0 ;1ST DEVICE VECTOR.
274 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
275 BR2: .BYTE PRTV0+0 ;2ND BR LEVEL.
276 DVID1: 0+1 ;DEVICE INDICATOR 1.
277 SR1: OPEN ;SWITCH REGISTER 1
278 SR2: OPEN ;SWITCH REGISTER 2
279 SR3: OPEN ;SWITCH REGISTER 3
280 SR4: OPEN ;SWITCH REGISTER 4
281 *****
282 STAT: 150000 ;STATUS WORD.
283 INIT: START ;MODULE START ADDR.
284 SPOINT: MODSP ;MODULE STACK POINTER.
285 PASCNT: 0 ;PASS COUNTER.
286 ICONF: 765 ;# OF ITERATIONS PER PASS=765
287 SPCNT: 0 ;LOC TO COUNT ITERATIONS
288 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
289 SOFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
290 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
291 SYSPNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
292 RANNO: 0 ;# OF SYS ERRORS ACCUMULATED
293 CONFIG: ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
294 RES1: 0 ;RESERVED FOR MONITOR USE
295 SVR0: OPEN ;RESERVED FOR MONITOR USE
296 SVR1: OPEN ;RESERVED FOR MONITOR USE
297 SVR2: OPEN ;LOC TO SAVE R0.
298 SVR3: OPEN ;LOC TO SAVE R1.
299 SVR4: OPEN ;LOC TO SAVE R2.
300 SVR5: OPEN ;LOC TO SAVE R3.
301 SVR6: OPEN ;LOC TO SAVE R4.
302 CSRA: OPEN ;LOC TO SAVE R5.
303 SHADR: OPEN ;LOC TO SAVE R6.
304 ACSR: OPEN ;ADDR OF CURRENT CSR.
305 WASADP: WASADP ;ADDR OF GOOD DATA, OR
306 ERRTYP: OPEN ;CONTENTS OF CSR.
307 ASB: OPEN ;ADDR OF BAD DATA, OR
308 AWAS: OPEN ;STATUS REG CONTENTS.
309 RSTRT: RSTRT ;TYPE OF ERROR
;EXPECTED DATA.
;ACTUAL DATA.
;RESTART ADDRESS AFTER END OF PASS

```

310 000114 000000
311 000116 000000
312 000120 000000
313 000122 000124
314 000124 000366
315 000126 000000
316 000130 000000
317 000132 000400
318 000134 000000
319 000136 000000
320 000140 000000
321 000142 000000
322 000144 000000
323 000146 000000
324 000150 000000
325 000252
326
327

WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 124 ;MODULE IDENTIFICATION NUMBER=124
RBUFVA: RBUF ;READ BUFFER VIRTUAL ADDRESS
RBUFPA: OPEN ;READ BUFFER PHYSICAL ADDRESS
RBUPEA: OPEN ;READ BUFFER EA BITS
RBUFSZ: 256 ;SIZE OF THE READ BUFFER
WRUFPA: OPEN ;WRITE RUFFER PHYSICAL ADDRESS
WRUFEA: OPEN ;WRITE RUFFER EA BITS
WRUFPS: 1024 ;WRITE RUFFER SIZE REQUESTED
WRUFSS: 1024 ;WRITE RUFFER SIZE AVAILABLE
CDERCT: OPEN ;CDATA/DATCK ERROR COUNT
CDWDCT: OPEN ;CDATA/DATCK WORD COUNT
FREE: OPEN ;RESERVED FOR FUTURE USE
MODSP:
;*****

328
329
330 000252 000000
331 000254 000000
332 000256 000000
333 000260 000000
334 000262 000000
335 000264 000000
336 000266 000000
337 000270 000000
338 000272 000000
339 000274 000000
340 000276 000000
341 000300 000000
342 000302 000000
343 000304 000000
344 000306 000000
345 000310 000000
346 000312 000000
347 000314 000000
348 000316 000000
349 000318 000000
350 000322 000000
351 000324 000000
352 000326 000000
353 000330 000000
354 000332 001000
355 000334 000000
356 000336 000000
357 000340 000000
358 000342 000000
359 000344 000000
360 000346 000000
361 000350 000000
362 000352 000000
363 000354 000000
364 000356 000765
365 000360 000300
366 000362 000300
367 000364 000000
368 000366 000400
369 000368 000000
370 000370 000000
371 000372 000000
372 000374 000000
373 000376 000000
374 000400 000000
375 000402 000000
376 000404 000000
377 000406 000000
378 000410 000000
379 000412 000000
380 000414 000000
381 000416 000000
382 000420 000000
383 000422 000000

.SBTTL CONSTANTS AND STORAGE
BADSPT: C ;BAD SPOT TABLE, PUT IN THE OCTAL
;BLOCK NUMBER, AS SHOWN IN ERROR TYPEOUTS
;OF KNOWN BAD BLOCKS.
;THERE'S ROOM FOR 16.
;BLK 0 IS NEVER CONSIDERED BAD
;IF YOURS IS, YOU CANNOT RUN THIS PROGRAM
;GO FIND ANOTHER PACK AND TRY AGAIN

FLAG: C ;FLAG WHICH IS PASSED BACK BETWEEN PORTS
AVFLAG: C ;DRIVE AVAILABLE TIMER
TCYL: C ;HOLDING LOCATION FOR CYLINDER ADDR
TSEC: C ;" " SECTOR
TIMER: C ;TIMES OUT A PORT
CYLADR: C ;DESIRED CYLINDER
SECADR: C ;DESIRED TRACK
WDCNT: C ;REPETITION COUNT
CYLCNT: 512.
RUFADR: C
CNT: C ;END OF PASS COUNTER
FUNC: C ;DESIRED FUNCTION
XMEW: C ;EXTENDED MEMORY BITS
DSKADR: C ;CURRENT SELECTED DRIVE NUMBER
DRICE: C ;AVAILABLE DRIVES
DRIVE: C ;CURRENT SELECTED DRIVE
DRIVE: C ;DRIVE COUNTER
BLK1: C ;SECTOR/BLOCK COUNTER
ICNT: .WORD 765 ;ITERATION COUNT
DPICNT: .WORD 300 ;DUAL PORT ITERATION COUNT
WCNT1: C ;WORD COUNT FOR WRITES
WCNT2: C ;WORD COUNT FOR READS
RBUF: C BLKW 256. ;INPUT BUFFER
CLK1: C ;TIMEOUT CLOCK
CLK2: C ;SOFT ERROR FLAG
INTRFR: C ;INTERRUPT EXPECTED FLAG
RKCS1: C ;CONTROLLER STATUS REGISTER
RKWC: C ;WORD COUNT (2'S COMPLEMENT)
RKBA: C ;BUS ADDR (LOW 16 BITS)
RKDA: C ;TRACK AND SECTOR ADDR
RKCS2: C ;CONTR AND STATUS REG 2
RKDS: C ;DRIVE STATUS REG
RKER: C ;ERROR REG
RKASOF: C ;ATTENTION REG
RKADR: C ;CYLINDER ADDR
RKECPS: C ;ECC BAD BIT LOCATION
RKECPT: C ;ECC PATTREN

```

001424 000354
001426 177777
053057
036000
137400
004070
001430 000016
001464 000000
001466 000000
001470 007404
001472 001504
001474 007414
001476 000000
001500 177777
001502 000000
001504 000000
001506 000000

```

```

BLK: BLK1
177777
ERHARD=BIT0|BIT1|BIT2|BIT3|BITS|BIT9|BIT10|BIT12|BIT14
SHARD=BIT10|BIT11|BIT12|BIT13
S2HARD=BIT8|BIT9|BIT10|BIT11|BIT12|BIT13|BIT15
DSHARD=BIT3|BIT4|BIT5|BIT11
ERRBF: .PLKW 16
CNTR1: 0
CNTR2: 0
DRP: DRVE
      NUMB
      DRPED
DRPMSC: 0
      -1
HSRD: 0
NUMB: 0 WORD 0
RETRY: 0

```

```

001510 001542
001512 001544
001514 001546
001516 001550
001520 001552
001522 001554
001524 001556
001526 001560
001530 001562
001532 001564
001534 001566
001536 000354
001540 177777
001542 000000
001544 000000
001546 000000
001550 000000
001552 000000
001554 000000
001556 000000
001560 000000
001562 000000
001564 000000
001566 000000
001570 000007
001572 000001
001574 000011
001576 000005
001600 100100
001602 100000
001604 000013
001606 000003
001610 000000

```

```

TABLE: SVCS1
        SVWC
        SVBA
        SVDA
        SVCS2
        SVDS
        SVER
        SVASOF
        SVDC
        SVECP5
        SVECP7
        BLK1
        177777
;END OF TABLE INDICATOR
SVCS1: 0 ;SAVE OF CONTROLLER STATUS REGISTER
SVWC: 0 ;SAVE OF WORD COUNT (2'S COMPLEMENT)
SVBA: 0 ;SAVE OF BUS ADDR (LOW 16 BITS)
SVDA: 0 ;SAVE OF TRACK AND SECTOR ADDR
SVCS2: 0 ;SAVE OF CNTR AND STATUS REG 2
SVDS: 0 ;SAVE OF DRIVE STATUS REG
SVER: 0 ;SAVE OF ERROR REG
SVASOF: 0 ;SAVE OF ATTENTION REG
SVDC: 0 ;SAVE OF CYLINDER ADDR
SVECP5: 0 ;SAVE OF ECC BAD BIT LOCATION
SVECP7: 0 ;SAVE OF ECC PATTREN
;
; HOUSEKEEPING COMMANDS
; IF RK07 DRIVE IS SELECTED, THE BIT10 OF EACH COMMAND MUST BE SET
; IF RK06, MUST BE RESET
;
UNLOAD: .WORD 7 ;UNLOAD COMMAND
SELDRV: .WORD 1 ;SELECT A DRIVE
SPINDL: .WORD 11 ;STARTS THE SPINDLE
CLRDRV: .WORD 5 ;CLEAR THE DRIVE
DISINT: .WORD #BIT15|BIT6 ;CLEAR CCR ERROR WITH INTERRUPT DISABLE
CLRCTR: .WORD #BIT15 ;CLEAR CONTROLLER ERROR
RECALX: .WORD 13 ;RECALIPRATE
PACKAC: .WORD 3 ;PACK ACKNOWLEDGE
RKFLG: .WORD 0 ;SOFTWARE RK06, RK07 FLAG
;BIT10=1 IF RK07, RESET IF RK06

```

```
445 001612 012767 006000 176276 START: SPTTL MAIN LOOP
446 001612 012767 003000 176276 MOV #3072, WDFR
447 001612 012767 003000 176266 MOV #1536, WDT0
448 001634 012767 000010 176264 MOV #R, INTB
449 001634 012767 000001 176512 MOV #1, BLK1 ;START AT BLOCK(SECTOR) 1
450 001642 012767 000001 176442 MOV #1, FLAG ;SET 1ST TIME FLAG BIT
451 001654 004767 000572 JSR PC, SETUP
452 001654 004767 001040 JSR PC, CH*DRV ;CHECK ALL DRIVES
453 001660 012706 000252 RESTR: MOV #MODSP, SP ;RESTORE STACK
454 001660 104412 000000 GETPAS, BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
455 001672 016767 176234 176464 MOV RBUF$Z, WCNT2 ;SIZE OF READ BUFFER
456 001700 005467 176460 NEG WCNT2
457 001700 005467 176460 CLR CNT ;RESET END-OF-PASS COUNTER
458 001710 005367 176460 CLR INTPLG ;CLEAR EXPECTED INTERRUPT FLAG
459 001714 012767 177777 LOOP1: MOV #-1, DRVVE ;INITIALIZE DRIVE COUNT
460 001722 012767 177777 MOV #-1, DSKADR ;INITIALIZE DRIVE ADDRESS
461 001730 104414 000000 JSR PC, PICKR ;GO PICK A BLK1
462 001730 104414 000000 GBUFS, BEGIN ;GET WRITE BUFFER INFORMATION
463 001740 016767 176176 MOV WBUF$Z, WCNT1 ;SAVE WRITE BUFFER SIZE
464 001740 005467 176410 NEG WCNT1
465 001752 004767 LOOP2: JSP PC, NITDRV ;GO PICK A DRIVE
466 001752 004767 001260 RR LOOP4 ;RETURNS HERE IF ALL DRIVES DONE
467 001752 004767 000462 JSR PC, CYCLE ;GO DO A CYCLE ON THIS DRIVE
468 001764 000772 BP LOOP2 ;DO IT TO NEXT DRIVE
469 001766 042767 000601 176316 LOOP4: BIT #1, FLAG ;CLEAR FIRST TIME FLAG
470 001774 005767 176346 TST DVICE ;ANYBODY LEFT TO CHECK?
471 001780 001067 000006 BNE ZS ;RR IF YES
472 001780 001067 000006 JMP ENTST ;ELSE DROP MODULE
473 002008 164413 000000 2$: ENDITS, BEGIN ;SIGNAL END OF ITERATION.
474 002012 000740 RR LOOP1 ;MONITOR SHALL TEST END OF PASS
475 ;RR BACK IF NO
476
477
478
479
480
481 002014 ENTST: MSGNS, BEGIN, DRPMG ;ASCII MESSAGE CALL WITH COMMON HEADER
482 002014 104403 000000 001476 ENDS, BEGIN ;
483 002022 104410 000000 ;
484
```

```
485 002026 032767 000020 175762 CYCLE: BIT #20, SR1 ;R-PORT?
486 002034 001076 RNE CYCLER ;RR IF SO, ELSE DO A-PORT
487 002036 052767 000004 176246 BLS #4, FLAG ;SET THE A-PORT FLAG
488 002044 004767 004654 JSR PC, BLKADR ;CONVERT BLK TO DISK ADDR
489 002050 004767 001374 JSP PC, WRITE ;GO WRITE A BLOCK
490 002054 004767 001426 JSR PC, WRITCK ;GO DO WRITE CHECK
491 002060 004767 001436 JSR PC, READ ;GO READ A BLOCK
492 002064 104412 000000 CDATAS, BEGIN, RBUFP ;REQUEST FOR MONITOR TO CHECK DATA
493 002072 002074 +2 ; IF ERROR, CONTINUE
494 002074 002074 000200 175714 BIT #17, SR1 ;SINGLE PORT?
495 002102 001452 BEQ S ;YES-GET OUT
496
497 ;NOW UPDATE BLOCK 0 FOR R-PORT'S INFORMATION
498
499
500 002104 004767 004730 JSP PC, CLPRB ;CLEAR THE READ BUFFER
501 002110 016767 176176 MOV FLAG, RBUF ;PUT INFO IN FIRST WORD
502 002116 016767 176232 MOV BLK1, RBUF+2 ;PUT CURRENT ADDR IN NXT
503 002124 004767 001524 JSP PC, WRTO ;GO UPDATE BLOCK 0
504 002130 012767 050000 MOV #5000, TIMER ;INIT WAIT LOOP
505 002136 004767 004716 JSR PC, RELEASE ;GIVE THE DRIVE TO B
506 002142 005367 176154 DFC TIMER
507 002146 001004 BNE ZS
508 002150 004767 004434 JSR PC, DROP
509 002154 007765 PORTHC
510 002156 006267 PC
511 002160 012700 000002 RTS #2, R0
512 002164 000000 4$:
513 002164 104407 000000 BREAKS, BEGIN ;TEMPORARY RETURN TO MONITOR.
514 002170 104407 000000 BREAKS, BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
515 002174 005300 DEC R0
516 002176 001372 RLE 4$
517 002200 004767 JSR PC, RD00 ;GO READ BLOCK 0
518 002204 032767 000002 176154 BIT #2, RBUF ;HAS R UPDATED BLK 0 YET?
519 002212 001751 BEQ S ;RR BACK IF NOT
520 002214 004767 JSR PC, READ ;GO READ THE DATA B WROTE
521 002220 104412 000000 CDATAS, BEGIN, RBUFP ;REQUEST FOR MONITOR TO CHECK DATA
522 002222 002230 +2 ; IF ERROR, CONTINUE
523 002230 000207 RTS PC
```



```

24 002232 012767 000000 176062 CYCLER: MOV #60000,TIMER
25 002240 032767 001456 176040 1S: JSR PC,RD00 ;GO SEE IF A IS DONE YET
26 002248 032767 000001 176040 BIT #1,FLAG ;IS THIS THE FIRST TIME SINCE START?
27 002252 001410 000004 176104 BEQ #2,RBUF ;BR IF NO, SKIP THIS CHECK
28 002254 032767 000001 176074 BIT #4,RBUF ;HAS A WRITTEN THIS BLOCK?
29 002262 001410 000001 176074 BEQ #3,RBUF ;BR IF NO
30 002272 032767 000002 176064 2S: BIT #1,RBUF ;HAS A SET THE FIRST TIME FLAG?
31 002274 001404 000002 176064 PEQ #2,RBUF ;BR IF NOT, MUST WAIT
32 002302 001421 004550 3S: JSR PC,RELEASE ;HAS A WRITTEN THIS BLOCK SINCE B DID?
33 002304 004767 000002 176064 MOV #2,R0 ;BR IF SO, ELSE
34 002310 012700 000002 4S: ;GIVE PORT TO A
35 002314 104407 000000 ;TEMPORARY RETURN TO MONITOR...
36 002320 104407 000000 ;THEN CONTINUE AT NEXT INSTRUCTION.
37 002324 005300 DEC R0
38 002326 001372 BNE 4S
39 002330 005367 DEC TIMER
40 002334 001341 BNE 1S
41 002336 004767 JSR PC,DROP
42 002342 007767 PORTHG
43 002344 000207 PC
44 002350 004767 176016 6S: RTS RBUF+2,RLK1 ;GET THE CURRENT RLK1
45 002354 004767 004344 JSR PC,RLKADR ;GENERATE DISK ADDR FROM IT
46 002356 004767 001136 JSR PC,READ ;GO READ WHAT A WROTE
47 002364 004767 001170 JSR PC,WRITEB ;GO WRITE IT BACK OUT
48 002370 004767 001222 JSR PC,WRITEB ;GO WRITE CHECK IT
49 002374 004767 004440 JSR PC,CLPRR ;GO CLEAR BUFFER
50 002400 052767 000002 175704 BIS #2,FLAG ;SET BIT SAYING B'S DONE
51 002406 016767 175700 MOV FLAG,RBUF ;PUT INFO INTO RBUF
52 002414 004767 001234 JSR PC,WRTOO ;GO WRITE IT FOR A TO SEE
53 002418 004767 004434 JSR PC,RELEASE ;GIVE DRIVE TO A
54 002424 012700 000002 7S: MOV #2,R0
55 002430 104407 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
56 002434 104407 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
57 002440 005300 DEC R0
58 002442 001372 BNE 7S
59 002444 000207 RTS PC
  
```

```

64 .SBTTL INITIALIZATION
65
66 002446 016767 175342 175672 SETUP: MOV DVID1,DVICE ;GET DRIVE INDICATOR
67 002448 004767 000013 000041 CMPB #13,@#41 ;IF RK IS LOAD MEDIUM THEN
68 002452 001024 BEGIN ;BEGIN
69 002464 113700 000040 MOV #40,R0 ;GET LOAD-DEVICE NUMBER
70 002470 012701 000001 MOV #1,R1 ;INITIALIZE DEVICE MASK
71 002474 105700 10S: RSTR R0 ;WHILE NOT POINTING AT LOAD DEVICE DO
72 002476 001403 REG 20$ BEGIN
73 002500 006301 ASL R1 ;POINT AT NEXT DEVICE
74 002502 105300 DECR R0 ;COUNT SHIFTS
75 002504 007773 BR 10$ END
76 002506 130167 175634 20S: BITR R1,DVICE ;IF LOAD-DEVICE IS SELECTED THEN
77 002514 140167 BEQ 35$ BEGIN
78 002516 140167 BICR PC,DVICE ;DROP LOAD-DEVICE
79 002520 012767 175626 MOV #LOAD,DRPMSG ;SPECIFY LOAD-DEVICE DROP
80 002522 104403 000000 001476 MSGNS,BEGIN,DRPMSG ;ASCII MESSAGE CALL WITH COMMON HEADER
81 002534 3S: ;END
82
83 ;SET UP DEVICE REGISTERS - USE ADDRESS SPECIFIED BY ADDR IN HEADER
84
85 002534 016700 175246 1S: MOV ADDR,R0 ;GET DEVICE ADDRESS
86 002540 010067 176632 TST (R0)+ ;GENERATE CONTROLLER REG ADDRESSES
87 002544 005720 TST (R0)+
88 002546 010067 176626 MOV R0,RKWC
89 002552 005720 TST (R0)+
90 002554 010067 176622 MOV R0,RKRA
91 002560 005720 TST (R0)+
92 002562 010067 176616 MOV R0,RKDA
93 002566 005720 TST (R0)+
94 002570 010067 176612 MOV R0,RKCS2
95 002574 005720 TST (R0)+
96 002576 010067 176606 MOV R0,RKDS
97 002602 005720 TST (R0)+
98 002604 010067 176602 MOV R0,RKER
99 002610 005720 TST (R0)+
100 002612 010067 176576 MOV R0,RKASOF
101 002616 005720 TST (R0)+
102 002620 010067 176572 MOV R0,RKDC
103 002624 022020 CMP (R0)+,(R0)+ ;SKIP UNUSED REGISTER
104 002626 022020 CMP (R0)+,(R0)+ ;SKIP RKDP & RRMRI
105 002630 010067 MOV R0,RKCCPS
106 002636 010067 TST (R0)+
107 002642 016767 176560 MOV PC,RKRCPT
108 002650 012767 175500 MOV DVICE,DRIVE ;SAVE IT IS DRIVE
109 002654 177777 175474 MOV #1,DRIVE ;INITIALIZE DRIVE COUNT
110 002656 012767 177777 175460 MOV #1,DSKADR ;INITIALIZE SHIFTED DRIVE NUMBER
111 002664 016700 175120 MOV VECTOR,R0 ;GET VECTOR ADDRESS
112 002670 012720 024062 MOV #INTSRV,(R0)+ ;GET VECTOR ADDRESS
113 002674 115710 175120 MOVBR R1,(R0) ;SET PRIORITY
114 002700 032767 000200 175110 BIT #RIT,SP1 ;DUAL PORT?
115 002706 001403 BEQ 2S ;NO- BRANCH
116 002710 016767 175444 175440 MOV DPICNT,ICNT ;YES, LOAD DUAL PORT ITERATION COUNT
  
```

620 002716* 000207

25: RTS PC

```
621 ;CHECK SELECTED DRIVES TO SEE IF READY
622 ;IF NOT READY AFTER DELAY, DROP THE DRIVE
623 002720* 012777 000040 176460 CRKDRV: MOV #BIT5, @RKCS2 ;SUB-SYSTEM CLEAR
624 002726* 004767 002572 JSR PC, CNTRDY ;WAIT FOR CONTROLLER READY
625 002732* 005777 176440 TST @RKCS1 ;ERROR?
626 002736* 100013 RPL 1S ;NO-CONTINUE
627 002740* 104403 000000* 007725* MSGNS, BEGIN, UNCLR ;ASCII MESSAGE CALL WITH COMMON HEADER
628 002746* 012767 000047 175132 MOV #47, ERRTYP ;MONI CLEAR
629 *****
630 002754* 104405 000000* 001376* HRDERS, REGIN, RKCS1 ;
631 *****
632 002762* 000167 177026 JMP ENTST ;GO DROP MODULE
633 002766* 004767 003472 1S: JSR PC, DRVADR ;GET A DRIVE ADDRESS
634 002772* 000520 BR 9S ;DID ALL DRIVES
635 002774* 004767 002572 JSR PC, READY ;CHECK OUT RK06 OR RK07
636 003000* 009240 NOP ;DUMMY RETURN ADDRESS
637 ;THE ABOVE LINE ADDED 5-OCT-77
638 003002* 004767 003246 JSP PC, RDV1 ;DRIVE AVAILABLE
639 003006* 000167 177754 JMP 1S
640 003012* 016777 175326 176366 2S: MOV DSKADR, @RKCS2 ;NO, LOAD NEXT DRIVE ADDRESS
641 ;
642 003020* 016777 176562 176350 MOV #3, @RKCS1 ;DRIVE SELECT / PACK ACKNOWLEDGE
643 003026* 004767 002472 MOV PACKAC, @RKCS1 ;PACK ACKNOWLEDGE
644 003032* 005777 176340 JSR PC, CNTRDY
645 003036* 100472 TST @RKCS1 ;ERROR?
646 003040* 032777 100000 176342 HMI 8S ;YES-DROP DRIVE
647 003046* 001466 BIT #BIT15, @RKDS ;STATUS VALID
648 003050* 004767 REQ 8S
649 003054* 000167 177706 JSP PC, RDV1
650 003060* 016777 175260 176320 10S: MOV DSKADR, @RKCS2 ;
651 ;
652 003066* 016777 176502 176302 MOV #11, @RKCS1 ;START SPINDLE
653 003074* 005067 176266 MOV SPINDL, @RKCS1 ;START THE SPINDLE
654 003100* 004767 002420 CLR CLK ;CLEAR TIMER
655 003104* 005777 176266 JSP PC, CNTRDY ;WAIT FOR CONTROLLER READY
656 003110* 100427 TST @RKCS1 ;ERROR?
657 003112* 032777 100000 BMI 5S ;YES-DROP DRIVE
658 003120* 001010 BIT #BIT15, @RKDS ;ATTN SET-OPERATION COMPLETE?
659 003122* 104407 000000* BNE 20S ;YES-CONTINUE
660 003126* 104407 000000* BREAKS, BEGIN ;TEMPORARY RETURN TO MONITOR...
661 003132* 005367 176230 DEC CLK ;THEN CONTINUE AT NEXT INSTRUCTION.
662 003136* 100765 RMT 3S ;TIME LET?
663 003140* 000421 BR 2S ;YES-TRY AGAIN
664 003142* 004767 003106 20S: JSR PC, PDV1 ;NO
665 003146* 000167 177614 JMP 1S
666 ;
667 ;
668 003152* 016777 176420 176216 4S: MOV #5, @RKCS1 ;CLEAR ATTN BITS
669 003160* 004767 002172 MOV CLRDRV, @RKCS1 ;DRIVE CLEAR
670 003164* 000407 JSR PC, DRVRDY ;DRIVE READY
671 003166* 000412 BR 7S ;NO-ERROR
672 003170* 012767 007747* 176300 5S: MOV #DRVERR, DRPMSG ;SET UP ERROR MESSAGE
673 003176* 104403 000000* 001476* MSGNS, BEGIN, DRPMSG ;ASCII MESSAGE CALL WITH COMMON HEADER
674 ;
675 003204* 004767 003400 6S: JSR PC, DROP ;DROP DRIVE
676 003210* 010031* SPIND
```

677 003212 000665
 678 003214 003056
 680 003220 000240
 681 003222 000661
 682 003224 004767 003360
 683 003230 007502
 684 003232 000655
 685 003234 000207

75: BR 15
 JSR PC,RECAL ;TPV TO RECALIBRATE THE DRIVE
 NOP
 BP 15 ;DO NEXT DRIVE
 85: JSR PC,DROP
 UNVAL
 95: RR 15
 RTS PC ;RETURN

686
 687
 688
 689
 690 003236 004767 003222
 691 003242 000440
 692 003244 004767 003004
 693 003250 000187 177762
 694 003254 016777 175064 176124
 695
 696 003262 016777 176304
 697 003270 004767 002230 176106
 698 003274 005777 176076
 699 003300 100413
 700 003302 004767 002050
 701 003306 000410
 702 003310 032777 004000 176072
 703 003316 001410
 704 003320 004767 003264
 705 003324 007464
 706 003326 000743
 707 003330 004767 003254
 708 003334 007747
 709 003336 000737
 710
 711 003340 062716 000002
 712 003344 000207
 713

NXTDRV: JSR PC,DRVADR ;GET A DRIVE ADDRESS
 BP 55 ;PAD RETURN, NO DRIVE
 JSR PC,DRV1
 NXTDRV
 MOV DSKADR,@RKCS2 ;LOAD DISK ADDRESS
 ;
 MOV BIT0,@RKCS1 ;SELECT DRIVE
 SELDR,@RKCS1 ;SELECT DRIVE
 JSR PC,CNTRDY ;WAIT FOR CONTROLLER READY
 TST @RKCS1 ;ERROR?
 BMI 35 ;YES-GO DROP DRIVE
 JSR PC,DRVDRDY ;WAIT FOR DRIVE TO FINISH
 35
 BR
 25: BIT #BIT11,@RKDS ;WRITE PROTECTED ?
 BEQ 45
 JSR PC,DROP ;YES, DROP THE DRIVE
 PROT
 BR NXTDRV ;GO ON TO NEXT DRIVE
 35: JSR PC,DROP ;GO DROP DRIVE
 DRVDRDY
 BR NXTDRV ;GET NEXT DRIVE
 45: ADD #2,(SP)
 55: RTS PC ;GOOD RETURN

714	003346	062767	000001	175000
715	003354	032767	000100	174434
716	003362	001405		
717	003364	104417	000000	
718	003370	016767	174460	174756
719	003376	085767	174752	
720	003402	001761		
721	003404	026727	174744	064734
722	003412	002404		
723	003414	015767	000001	174732
724	003422	000751		
725	003424	012700	000252	
726	003430	012701	000020	
727	003434	026720	174714	
728	003440	001742		
729	003442	005301		
730	003444	001373		
731	003446	000207		

```

PICKBK: ADD #1,RLK1 ;DO NEXT BLOCK(SECTOR)
          BIT #BIT6,SRI ;DO RANDOM SEERS?
          BEQ 1S ;NO-CONTINUE
          RANDB,BEGIN
          MOV RANUM,RLK1 ;GET RANDOM NUMBER
          TST BLK1 ;BLOCK 0 SELECTED?
          BEQ PICKBK ;YES-GET ANOTHER BLOCK
          CMP BLK1,#27100. ;OUT OF ROUNDS
          BGT 2S ;NO-CONTINUE
          MOV #1,RLK1 ;GO BACK TO BLOCK 1
          RR PICKBK ;TRY AGAIN
          MOV #RADSPT,R0 ;GET RAD SPOT TABLE
          MOV #16,R1 ;LOOK FOR 16 ENTRIES
          CMP BLK1,(R0)+ ;IS THIS A RAD BLK1?
          BEQ PICKBK ;IF YES, GO PICK A NEW ONE
          DEC R1 ;COUNT A TABLE LOOK-UP
          BNE 3S ;RR BACK IF MORE TO GO
          RTS PC
  
```

732				
733				
734	003450	012767	000123	174662
735	003456	016767	174700	174644
736	003464	016767	174444	174642
737	003472	016767	174440	174642
738	003500	004767	000226	
739	003504	000207		
740				
741	003506	012767	000131	174624
742	003514	064767	000212	
743	003520	000207		
744				
745	003522	012767	000121	174610
746	003530	016767	174630	174572
747	003536	016767	174364	174570
748	003544	016767	174360	174570
749	003552	004767	000154	
750	003556	000207		
751				
752	003560	012767	000123	174552
753	003566	016767	174572	174534
754	003574	016767	174326	174532
755	003602	016767	174322	174532
756	003610	004767	000116	
757	003614	000207		
758				
759	003616	012767	000131	174514
760	003624	016767	174534	174476
761	003632	016767	174270	174474
762	003640	016767	174264	174474
763	003646	004767	000060	
764	003652	000207		
765				
766	003654	012767	000123	174456
767	003662	016767	174476	174440
768	003670	016767	174322	174438
769	003678	016767	174322	174438
770	003704	005067	174406	
771	003710	005067	174404	
772	003714	004767	000026	
773	003720	000207		
774				
775	003722	012767	000121	174410
776	003730	000754		

```

.SRRTL DRIVE COMMAND ROUTINES
WRITE: MOV #123,FUNC ;LOAD WRITE FUNCTION
        WCNT2,WDCNT ;LOAD WORD COUNT
        MOV BRUFA,BUFADR ;LOAD BUFFER ADDRESS
        MOV BRUFEA,XMEM ;LOAD EXTENDED MEMORY BITS
        JSR PC,EXCUTE
        RTS
WRITCK: MOV #131,FUNC ;LOAD WRITE-CHECK FUNCTION
        JSR PC,EXCUTE
        RTS
READ: MOV #121,FUNC ;LOAD READ FUNCTION
       WCNT2,WDCNT ;LOAD WORD COUNT
       MOV BRUFA,BUFADR ;LOAD BUFFER ADDRESS
       MOV BRUFEA,XMEM ;LOAD EXTENDED MEMORY BITS
       JSR PC,EXCUTE
       RTS
WRITEB: MOV #123,FUNC ;LOAD WRITE FUNCTION
         WCNT2,WDCNT ;LOAD WORD COUNT
         MOV BRUFA,BUFADR ;LOAD BUFFER ADDRESS
         MOV BRUFEA,XMEM ;LOAD EXTENDED MEMORY BITS
         JSR PC,EXCUTE
         RTS
WRITCR: MOV #131,FUNC ;LOAD WRITE CHECK FUNCTION
         WCNT2,WDCNT ;LOAD WORD COUNT
         MOV BRUFA,BUFADR ;LOAD BUFFER ADDRESS
         MOV BRUFEA,XMEM ;LOAD EXTENDED MEMORY BITS
         JSR PC,EXCUTE
         RTS
WRT00: MOV #123,FUNC ;LOAD WRITE FUNCTION
        WCNT2,WDCNT ;LOAD WORD COUNT
        MOV BRUFA,BUFADR ;LOAD BUFFER ADDRESS
        MOV BRUFEA,XMEM ;LOAD EXTENDED MEMORY BITS
        CLR TCYL ;DO CYL 0
        CLR TSEC ;SECTOR 0
        JSR PC,EXCUTE1
        RTS
RD00: MOV #121,FUNC ;LOAD READ COMMAND
       WRT00R
  
```

.SBTTL DEVICE DRIVER & INTERRUPT SERVICE

777						
778						
779						
780						
781	003732	016767	174366	174356	EXCUTE:	MOV CYLADR,TCYL ;GET CYL ADDR
782	003740	016767	174362	174352	MOV	SECADR,TSEC ;GET SECTOR ADDRESS
783	003746	012767	000003	175532	EXCUT1:	MOV #3,RETRY
784	003754	005067	175414		EXCUT2:	CLR INTFLG ;CLEAR EXPECTED INTERRUPT FLAG
785	003760	004767	002270		JSR	PC,CNTRDY
786	003764	000167	175370		JMP	RESTRT
787	003770	016777	174340	175404	1S:	MOV BUADR,@RKBA
788	003776	016777	174326	175374	MOV	WDCNT,@RKWC
789	004004	016777	174306	175404	MOV	TCV,@RKDC ;LOAD CYLINDER ADDR
790	004012	016777	174302	175364	MOV	TSEC,@RKDA ;LOAD THE ADDR OF SECTOR
791	004020	016700	174316		MOV	XMEM,RO ;EXTENDED MEM BITS
792	004024	006300			ASL	RO
793	004026	006300			ASL	RO
794	004030	006300			ASL	RO
795	004032	006300			ASL	RO
796	004034	056700	174300		BIS	FUNC,RO ;ADD FUNCTION CODE
797	004040	056700	175444		BIS	RKFLG,RO ;SET/RESET THE BIT10
798	004044	010077	175326		MOV	RO,@RKCS1 ;EXECUTE THE FUNCTION
799	004050	012767	000001	175316	MOV	#1,INTFLG ;TURN ON EXPECTED INTERRUPT FLAG
800	004056	104400	000000		EXIT\$,BEGTN	;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

801	004062	042777	100100	175306	INTSRV:	BIC #BIT15 BIT6,@RKCS1 ;DISABLE INTERRUPTS
802						-----
803	004070	000004	000000	004076		PIRQ\$,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
804						-----
805						
806	004076	005767	175272		1S:	TST INTFLG ;ARE WE REALLY EXPECTING AN INTERRUPT?
807	004102	001404			REQ	2S ;NO-MUST RE ERROR CONDITION
808	004104	032777	140000	175264	BIT	#BIT15 BIT14,@RKCS1 ;ERROR OR DRIVE INTERRUPT
809	004112	001523			REQ	6S ;NO-MUST RE OKAY
810						
811	004114	012700	000012		2S:	MOV #12,RO ;SAVE 11. REGISTERS
812	004120	016701	175252		MOV	RKCS1,R1 ;START OF SAVE REGISTERS
813	004124	013703	001542		MOV	#SVCS1,R2 ;START OF MEMORY TO PUT THEM IN
814	004130	012122			3S:	MOV (R1)+,(R2)+
815	004132	005300			DEC	RO
816	004134	002375			RGF	3S
817	004136	016767	175234	173734	MOV	RKCS1,CSRA ;SET UP CSR FOR ERROR CALL
818	004144	016767	175372	173730	MOV	SVCS1,ACSR ;SET UP CONTENTS OF SAME
819	004152	012777	100000	175216	MOV	#BIT15,@RKCS1 ;CLEAR THE ERROR
820	004160	004767	001340		JSR	PC,CNTRDY ;CONTROLLER READY
821	004170	000005	175206		BTL	@RKCS1 ;ERROR?
822	004172	004767	002412		JSL	4S ;NO-CONTINUE
823	004176	007725			JSP	PC,DROP ;YES-DROP DRIVE
824	004200	000167	175454		JMP	UNCLR ;START OVER
825	004204	005767	175332		TST	SVCS1 ;ERROR?
826	004210	100446			HMI	5S ;REPORT THE ERROR
827	004212	012767	007646	175256	MOV	#FXTRA,DRPMSG
828	004216	104403	000000	001476	MSGN\$,BEGIN,DRPMSG	;ASCII MESSAGE CALL WITH COMMON HEADER
829	004226	012767	000011	173652	MOV	#1,ERRTVP ;ILLEGAL INTERRUPT
830						*****
831	004234	104405	000000	001510		HDRERS,BEGIN,TABLE ;*****
832						*****
833	004242	012777	000040	175136	MOV	#BIT5,@RKCS2 ;SUBSYSTEM CLEAR
834	004250	004767	001250		JSR	PC,CNTRDY ;WAIT FOR CONTROLLER READY
835	004254	005777	175116		TST	@RKCS1 ;ERROR?
836	004260	100404			BMI	8S ;YES GO DROP TEST
837	004262	012706	000252		MOV	#MODSP,SP ;INITIALIZE THE SP
838	004266	000167	175466		JMP	LOOP3 ;JMP TO LOOP3
839	004272	012767	007725	175176	MOV	#UNCLR,DRPMSG ;YES-LOAD MESSAGE
840	004300	104403	000000	001476	MSGN\$,BEGIN,DRPMSG	;ASCII MESSAGE CALL WITH COMMON HEADER
841	004306	012767	000047	175572	MOV	#47,ERRTVP ;WONT CLEAR
842						*****
843	004314	104405	000000	001376		HDRERS,BEGIN,PKCS1 ;*****
844						*****
845	004322	000167	175466		JMP	ENTEST ;GO END IT ALL
846						
847	004326	004767	000036		5S:	JSR PC,ERRORS
848	004332	000414			BR	7S
849	004334	005367	175146		DEC	RETRY
850	004336	002205			BGE	EXCUT2
851	004340	032767	000001	173446	BIT	#BIT0,SR1
852	004350	001405			REQ	7S
853	004352	004767	002232		JSP	PC,DROP
854	004356	001506			JSR	PC,RETRY
855	004360	000401			BR	7S

857					
858	004362	000207		6S:	PTS PC
859					
860	004364	000167	175270	7S:	JMP RESTRT

```

      .SPTTL  ERROR HANDLER
861
862
863 004370 004767 000154      ERRORS:  JSR    PC,ERRGEN      ;GENERATE THE ASCII MESSAGE
864 004370
865
866 004374 032767 000200 175154      BIT    #R17,SVER      ;BAD SECTOR ERROR
867 004402 001410      BEQ    1S             ;SKIP THIS IF NOT
868 004404 004767 000362      JSR    PC,BSRD       ;READ & COMPARE BAD FILE SECTOR
869 004410 000432      BR     2S             ;NO MATCH - REPORT AS AN ERROR
870 004412 032767 000004 173376      BIT    #R12,SRI      ;TYPE ERROR MESSAGE?
871 004420 001450      BEQ    4S             ;NO JUST EXIT
872 004422 000425      BR     2S             ;TYPE IT ALL
873
874 004424 032767 100000 175124 1S:   BIT    #R15,SVER      ;DATA CHECK ERROR?
875 004432 001421      BEQ    2S             ;NO
876 004434 032767 000100 175114      BIT    #R16,SVER      ;ECC HARD ERROR?
877 004442 001015      RNF    2S             ;YES - GO REPORT IT
878
879 004444 016700 175072      MOV    SVCS1,RC
880 004450 042700 017741      RTC    #17741,PC      ;KEEP THE FUNCTION CODE
881 004454 022700 000020      CMP    #20,RC
882 004460 001046      RNF    2S
883
884 004462 004767 000554      JSR    PC,ECCOR      ;APPLY THE CORRECTION STUF
885 004466 032767 000010 173322      BIT    #R13,SRI      ;TYPE A MESSAGE
886 004474 001422      BEQ    4S             ;NO - JUST EXIT
887
888 004476 026727 175004 000003 2S:   CMP    RETRY,#3
889 004504 002410      BLT
890
891 004506 104403 000000 001430      MSGNS,REGIN,ERRUF    ;ASCII MESSAGE CALL WITH COMMON HEADER
892 004514 005067 173366      CLP    ERR1YP
893
894 004520 104405 000000 001510      ;*****
895 ;RDERS,BEGIN,TABLE ;
896 ;*****
897
898 004526 005767 174640      3S:   TST    SETERR        ;WAS IT A SOFT ERROR?
899 004530 001003      BNE    4S             ;YES
900 004534 004767 001536      JSR    PC,PECAL      ;NO - HARD - RECALIBRATE THE DRIVE
901 004540 000402      BR     5S
902
903 004542 062716 000002      4S:   ADD    #2,(SP)
904 004546 000207      5S:   RTS    PC
  
```

				.SBTTL	ERRKOR MESSAGE GENERATOR	
905						
906						
907	004550	005067	174616	ERRGEN:	CLP	SFTERR ;CLEAR SOFT ERROR FLAG
908	004554	012700	007114		MOV	#ERTAR,R0 ;ERROR TABLE PARAMETERS
909	004560	012700	001430		MOV	#ERRBUF,R5 ;PRINT POINTER BUFFER
910	004574	032767	137400		BIT	#SHARD,(R5) ;ASSUME THAT ITS HARD
911	004576	001020			BNE	IS ;ANY HARD ERRORS IN RKCS2?
912	004600	032767	036000		RIT	#SHARD,SVCS1 ;YES, REPORT
913	004606	001014	174734		RNE	IS ;ANY HARD ERRORS IN RKCS1?
914	004610	032767			BIT	#SHARD,SVDS ;YES REPORT
915	004616	001010	004070		BNE	IS ;ANY HARD ERRORS IN RKDS?
916	004620	032767	053057		BIT	#RHARD,SVER ;REPORT
917	004630	001004			BNE	IS ;ANY HARD ERRORS IN RKER
918	004634	005269	007370		INC	#SOFT,(R5) ;MUST BE SOFT BY NOW
919	004640	005275	174532		SFTERR	;(R5)+ ;SET SOFT ERROR FLAG
920	004640	005275		1\$:	TST	(R5)+
921	004642	012767	000004		MOV	#4,CNTR1 ;COUNTER FOR 4 REGISTERS
922	004650	012700	007114		MOV	#ERTAR,R0 ;START OF ERROR TABLE PARAMETERS
923	004654	012001		2\$:	MOV	(R0)+,R1 ;GET ADDRESS OF POINTER TO REGISTER
924	004656	012004			MOV	(R0)+,R4 ;ADDRESS OF ERROR MESSAGE TABLE
925	004660	012003			MOV	(R0)+,R3 ;GET MASK
926	004662	011102			MOV	(R1)+,R2 ;GET CONTENTS OF REGISTER
927	004664	040302			BIC	R3,R2 ;KEEP ONLY THOSE IN BIT MASK
928	004666	001424			BEQ	R3,R3 ;GO TO NEXT REGISTER IF ZERO
929	004670	005163			COM	R3
930	004672	012767	000020	174566	MOV	#16.,CNTR2
931						
932						
933	004700	000241		3\$:	CLC	;CLEAR THE CARRY
934	004702	006003			ROR	R3 ;ROTATE THE MASK WORD
935	004704	103402			BCS	4\$
936	004706	006002			ROR	R2 ;ROTATE THE VALUE - DUMMY ROTATE
937	004710	000413			BR	6\$
938						
939	004712	000241		4\$:	CLC	
940	004714	006002			ROR	R2 ;IS BIT SET?
941	004716	103006			BCC	5\$;NO
942	004720	012725	007350		MOV	#PLNKS,(R5)+ ;TWO SPACES
943	004724	010425			MOV	R4,(R5)+ ;PUT ADDRESS OF MESSAGE IN TABLE
944	004726	020527	001454		CMP	R5,#ERRBUF+24 ;END OF BUFFER SPACE
945	004732	002010			BGE	7\$;YES - GO PRINT THEM
946						
947	004734	062704	000004	5\$:	ADD	#4,R4 ;ADDRESS OF NEXT MESSAGE
948	004740	005367	174522	6\$:	DEC	CNTR2 ;DONE WITH THIS WORD?
949	004744	001355			BNE	3\$
950						
951	004746	005367	174512		DEC	CNTR1 ;DONE WITH ALL REGISTERS?
952	004752	001340			BNE	2\$
953						
954	004754	012725	007404	7\$:	MOV	#DRVE,(R5)+ ;
955	004760	012725	001504		MOV	#NUMB,(R5)+ ;
956	004764	012715	177777		MOV	#-1,(R5) ;END OF MESSAGES
957	004770	000207			RTS	PC

				.SBTTL	BAD SECTOR CHECK	
958						
959	004772	012767	000001	174502	BSRD:	MOV #1,HSRD ;FLAG TO INDICATE READING HARDWARE SECTORS
960	005006	012700	000776		MOV	#776,R0 ;INITIAL SECTOR COUNT
961						
962						
963	005004	062700	000002	1\$:	ADD	#2,R0
964	005010	020027	001012		CMP	R0,#1012 ;DONE WITH HARDWARE FILE?
965	005014	001002		2\$:	BNE	2\$
966	005016	005067	174460		CLP	HSRD ;DOING SOFTWARE FROM NOW ON
967						
968	005022	020027	001026	2\$:	CMP	R0,#1026 ;END OF SOFTWARE FILE?
969	005030	002104			RGE	6\$;END OF FILE AND NO MATCH
970	005034	004767	001220		JSR	PC,PDV1
971	005034	000167	174620		JMP	RESTR1
972						
973	005040	016777	173300	10\$:	MOV	DSKADR,@RKCS2 ;LOAD DISK ADDRESS
974	005046	012777	177400		MOV	#400,@RKWC ;ONE SECTOR TRANSFER
975	005054	016777	173046		MOV	PRUFP@,@RKBA ;BUFFER ADDRESS
976	005062	012777	000632	174326	MOV	#410,@RKDC ;DESIRED CYLINDER
977	005070	005767	174514		TST	RKFLG ;BRANCH IF IT IS A RK06
978	005074	001403			BEQ	#8 ;SKIP THE NEXT MOV INSTRUCTION
979	005076	012777	001456	174312	MOV	#1456,@RKDC ;LAST CYLINDER FOR RK07
980	005104	010077	174274		MOV	R0,@RKDA ;DISK SECTOR/TRACK ADDRESS
981	005110	016705	173014		MOV	R0,PEA,R5 ;GET READ BUFF EA BITS
982	005114	006305			ASL	R5 ;ALIGN BITS
983	005116	006305			ASL	R5
984	005120	006305			ASL	R5
985	005122	006305			ASL	R5
986	005124	052705	000021		BIS	#21,R5 ;LOADFUNCTION
987	005130	056705	174454		BIS	RKFLG,R5 ;SET THE RK06/RK07 FLAG
988	005134	010577	174236		MOV	R5,@RKCS1 ;ISSUE THE FUNCTION
989	005140	004767	000212		JSR	PC,DRVDY ;WAIT FOR DRIVE READY
990	005144	000435			BR	6\$
991	005146	005777	174224		TST	@RKCS1 ;ANY ERRORS?
992	005152	100714			BMI	IS ;YES IF SET
993						
994	005154	012701	000177		MOV	#177,R1 ;MAXIMUM NUMBER OF BAD SECTOR ENTRIES
995	005160	012702	000376		MOV	#RBUF+10,R2 ;START OF BAD SECTOR INFO
996	005164	005301		3\$:	DEC	R1
997	005166	001414			BEQ	4\$;THIS IS THE END
998						
999	005170	012203			MOV	(R2)+,R3 ;CYLINDER INFOR
1000	005172	012204			MOV	(R2)+,R4 ;SECTOR/TRACK INFO
1001	005174	020327	177777		CMP	R3,#-1 ;END OF SECTOR INFO
1002	005200	001407			BEQ	4\$
1003	005202	026703	174354		CMP	SWDC,R3 ;DESIRED CYLINDER?
1004	005206	001366			RNE	3\$;NO - KEEP LOOKING
1005	005210	026704	173112		CMP	SECADR,R4 ;DESIRED SECTOR?
1006	005214	001363			BNE	3\$;NO - KEEP LOOKING
1007	005216	000406			BR	5\$;MATCH RETURN
1008						
1009	005220	005767	174256	4\$:	TST	HSRD ;DOING THE HARDWARE GUVS?
1010	005224	001403			BEQ	5\$;NO - ALL DONE THEN
1011	005226	012700	001010		MOV	#1010,R0 ;SET UP FOR SOFTWARE FILE
1012	005232	000664			RR	IS
1013						

1014 005234 006270 000002
 1015 005240 006207

53: ADD #2,(SP)
 63: FTS PC

1016
 1017
 1018 005242 016702 174300
 1019 005246 162702 001000
 1020 005252 016701 174306
 1021 005256 005301
 1022
 1023
 1024 005260 010100
 1025 005264 006201
 1026 005268 006201
 1027 005266 006201
 1028 005270 042701 C00001
 1029 005274 060102
 1030
 1031 005276 016703 174264
 1032 005302 005004
 1033 005304 042700 177760
 1034 005310 001404
 1035
 1036
 1037 005312 006303
 1038 005314 006304
 1039 005316 005300
 1040 005320 001374
 1041
 1042 005322 011200
 1043 005324 010301
 1044 005326 040311
 1045 005330 040001
 1046 005334 050102 174206
 1047 005340 001405
 1048
 1049 005342 011200
 1050 005344 010401
 1051 005346 040412
 1052 005350 040001
 1053 005352 050112
 1054 005354 000207

```

.SBTTL  EPROR CORRECTION ROUTINE
ECCOR:  MOV  SVBA,R2      ;BEGINNING OF SECTOR ADDRESS
        SUB  #1000,R2    ;START OF BAD DATA BUFFER
        MOV  SVECP5,R1   ;POSITION OF ERROR
        DEC  R1

        MOV  R1,P0
        ASP  R1          ;DETERMINE WORDS THAT ARE BAD
        ASR  R1
        ASR  R1
        BIC  #1,R1
        ADD  R1,R2
        MOV  SVECP7,R3   ;ERROR CORRECTION PATTERN
        CLR  R4
        BIC  #177760,R0 ;BIT POSITION FOR START OF ECC CORRECTION
        BEQ  Z5          ;CORRECTION STARTS ON WORD BOUNDRY

1$:     ZSL  R3
        ROL  R4
        DEC  R0
        BNE  1$         ;CHECK IF IN POSITION

2$:     MOV  (R2),R0     ;CORRECT 1ST WORD WITH EXCLUSIVE OR
        MOV  R3,R1
        BIC  R3,(R2)
        BIC  R0,R1
        BIS  R1,(R2)
        CMP  R2,SVBA
        BEQ  3$

        MOV  (R2),R0   ;CORRECT SECOND WORD
        MOV  R4,R1
        BIC  R4,(R2)
        BIC  R0,R1
        BIS  R1,(R2)

3$:     RTS  PC        ;RETURN
  
```



```

;ROUTINE DETERMINES IF DRIVE IS AVAILABLE, READY, VOLUME
;VALID, AND NOT WRITE PROTECTED
READY:
1097 JSP PC,RKCHK ;CHECK IT IS A RK06/RK07
1098 RP 15 ;ERROR EXIT FROM RKCHK
1099 MOV DSKADR,RRKCS2 ;LOAD DRIVE NUMRER
1100 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1101 JSR SELDRV,RRKCS1 ;SELECT THE DRIVE
1102 MOV PC,CNTRDY ;WAIT FOR CONTROLLER READY
1103 JSP PC,CNTRDY ;WAIT FOR CONTROLLER READY
1104 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1105 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1106 JSP PC,CNTRDY ;WAIT FOR CONTROLLER READY
1107 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1108 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1109 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1110 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1111 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1112 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1113 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1114 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1115 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1116 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1117 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1118 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1119 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1120 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1121 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1122 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1123 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1124 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1125 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1126 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1127 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1128 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1129 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1130 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1131 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1132 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1133 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1134 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1135 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1136 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1137 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1138 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1139 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1140 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1141 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1142 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1143 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1144 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1145 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1146 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1147 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE
1148 JSP SELDRV,RRKCS1 ;SELECT THE DRIVE
1149 MOV #RIT0,RRKCS1 ;SELECT THE DRIVE

```

```

;ROUTINE WAITS FOR DRIVE READY, IF NOT READY
NOTRDY:
1150 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1151 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1152 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1153 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1154 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1155 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1156 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1157 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1158 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1159 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1160 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1161 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1162 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1163 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1164 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1165 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1166 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1167 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1168 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1169 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1170 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1171 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1172 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1173 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1174 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1175 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1176 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1177 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1178 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1179 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1180 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1181 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1182 MOV #RIT0,RRKCS1 ;SELECT DRIVE
1183 MOV #RIT0,RRKCS1 ;SELECT DRIVE

```

```

1184
1185
1186
1187
1188 006170 016777 172150 173210
1189
1190 006176 016777 173374 173172
1191 006164 004767 177314
1192 006210 000240
1193 006212 000240
1194 006214 000240
1195 006216 016777 172122 173162
1196
1197 006224 016777 173356 173144
1198 006232 004767 177256
1199 006236 012777 100000 173132
1200 006244 000240
1201 006246 000240
1202 006250 000240
1203 006252 000207
1204
1205
1206
1207
1208 006254 004767 177312
1209 006260 000403
1210 006266 004767 177542
1211 006268 000207
1212 006270 062716 000004
1213 006274 000207
1214
  
```

```

;ROUTINE ISSUES A DRIVE CLEAR TO SELECTED DRIVE
;AND THEN ISSUES A DRIVE SELECT AND PACK ACKNOWLEDGE.
;WILL ALSO CLEAR ANY CONTROLLER ERRORS BEFORE RETURNING.
CLEAR:  MOV     DSKADR,@RKCS2 ;GET DRIVE #
;        MOV     #5,@RKCS1 ;ISSUE DRIVE CLEAR
;        MOV     CLRDRV,@RKCS1 ;CLEAR THE DRIVE
;        JSP     PC,CNTRDY
;
;        MOV     DSKADR,@RKCS2
;        MOV     #3,@RKCS1 ;ISSUE SELECT DRIVE AND PACK ACK.
;        MOV     PACKAC,@RKCS1 ;ISSUE PACK ACKNOWLEDGE
1S:     JSR     PC,CNTRDY ;WAIT FOR CU READY
;        MOV     #BIT15,@RKCS1 ;CLEAR CU ERRORS
;
;        NOP
;        NOP
;        RTS     PC
  
```

;ROUTINE CALLS READY AND NOTRDY SUBROUTINES

```

RDY1:   JSR     PC,READY ;CHECK FOR DRIVE AVAILABLE
;        RF     1S ;GOOD RETURN
;        JSR     PC,NOTRDY ;WAIT FOR DRIVE AVAILABLE
;        RTS     ;DRIVE NOT AVAILABLE
1S:     ADD     #4,(SP)
;        RTS     PC
  
```

```

1215 006276 004767 177752
1216 006302 000167 000154
1217 006306
1218
1219 006306 016777 173264 173062
1220 006312 004767 177204
1221 006314 005777 173032
1222 006324 100447
1223 006326 004767 177722
1224 006332 000167 000124
1225 006336
1226
1227 006336 016777 173242 173032
1228 006344 005267 173916
1229 006350 004767 177150
1230 006354 005777 173016
1231 006360 100431
1232 006362 032777 040000 173020
1233 006370 001010
1234 006372 104407 000000
1235 006376 104407 000000
1236 006402 005367 172780
1237 006406 100765
1238 006410 000421
1239 006412 004767
1240 006416 000167 000040
1241 006422
1242
1243 006422 016777 173150 172746
1244 006430 004767 176722
1245 006434 000403
1246 006436 062716 000002
1247 006442 000407
1248 006444 004767 000140
1249 006450 007747
1250 006452 009403
1251
1252 006454 004767 000130
1253 006460 007675
1254 006462 000207
  
```

```

RECAL:  JSR     PC,RDY1
;        JMP     5S
19S:    MOV     #5,@RKCS1 ;CLR ATTN BIT
;        MOV     CLRDRV,@RKCS1 ;CLEAR THE ATT BIT
;        JSR     PC,CNTRDY ;CONTROLLER READY
;        TST     @RKCS1 ;ANY ERRORS?
;        BMI     3S ;YES-GO REPORT
;        JSR     PC,RDY1
;        JMP     5S
20S:    MOV     #13,@RKCS1 ;RECALIBRATE THE DRIVE
;        MOV     RECALY,@RKCS1 ;RECALIBRATE THE DRIVE
;        CLR     CLK ;CLEAR TIMER
;        JSF     PC,CNTRDY ;WAIT FOR CONTROLLER READY
;        TST     @RKCS1 ;ANY ERRORS?
;        BMI     3S ;YES-GO REPORT
1S:     BIT     #BIT14,@RKDS ;ATTN SET-OPERATION COMPLETE?
;        RNE     2S ;YES-CONTINUE
;        BPEAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
;        BPEAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
;        DEC     CLK
;        BMI     1S ;TIME LEFT?
;        RR     4S ;NO
;        JSR     PC,RDY1
;        JMP     5S
30S:    MOV     #5,@RKCS1 ;CLEAR ATTN BITS
;        MOV     CLRDRV,@RKCS1 ;CLEAR ATTN BITS
;        JSR     PC,DRVRDY ;DRIVE READY?
;        BR     3S ;NO - ERROR
;        ADD     #2,(SP) ;ADJUST STACK
3S:     JSR     PC,DROP ;GO DROP DRIVE
;        DRVERP
;        BR     5S
4S:     JSR     PC,DROP ;DROP THE DRIVE IF NOT READY
;        CALIB
5S:     RTS     PC ;RETURN
  
```

1255	006464	005267	171662		DRVADR: INC	DPYVE		;COUNT A DRIVE
1256	006470	005267	171650		INC	DSKADR		;NEXT DRIVE
1257	006474	005767	171646		TST	DVICE		;ANY DRIVES LEFT?
1258	006500	001005			RNE	1S		;YES
1259	006502	012767	007550	172766	MOV	#GONE,DRPMSG		
1260	006510	000167	173300		JMP	ENTEST		;STOP THE RUN
1261	006514	042767	177770	171622	1S: RJC	#177770,DSKADR		;UP TO ADDR 7 ONLY
1262	006522	022767	000010	171622	CMP	#A.,DPYVE		;ALL DRIVES CHECKED ?
1263	006530	001406			REQ	2S		;YES, GO FLAG END OF DRIVES
1264	006532	006267	171612		ASF	DVICE		;NO, IS NEXT DRIVE CHOSEN ?
1265	006534	103352			RCC	DPVADR		;NO, GO TRY ANOTHER DRIVE
1266	006540	062716	000002		ADD	#2,(SP)		;ADJUST STACK POINTER
1267	006544	000412			BR	3S		
1268								
1269	006546	012767	177777	171576	2S: MOV	#-1,DPYVE		;RESET DRIVE COUNTER
1270	006554	012767	177777	171562	MOV	#-1,DSKADR		;ZERO THE SHIFTED DRIVE #
1271	006562	016767	171560	171560	MOV	DVICE,DRIVE		;RESTORE CHOSEN DRIVES
1272	006570	000406			RP	4S		
1273								
1274	006572	016767	171554	172704	3S: MOV	DPYVE,NUMR		;CURRENT DRIVE NUMBER IN MESSAGE STRING
1275	006600	052767	000060	172676	RTS	#0,NUMR		;MAKE IT ASCII
1276	006606	000267			4S: RTS	PC		
1277								
1278								

1279	006610	017667	000000	172660	DROP:	MOV	#0(SP),DRPMSG	
1280	006610	017667	000000	001470	MSGNS,REGIN,DRP	;ASCII MESSAGE CALL WITH COMMON HEADER		
1281	006616	104403	000006	171254	MOV	#6,ERRTYP		;OFF LINE
1282	006624	012767	000006	171254				
1283								
1284	006632	10440E	000000	001376				
1285								
1286	006640	012767	000001		MOV	#1,R1		;INITIALIZE DROP PICKER
1287	006644	016767	171502		MOV	DPYVE,RC		;GET THE DRIVE NUMBER
1288	006650	001403			REQ	2S		;IF DRIVE 0 GO DROP IT
1289	006652	006301			1S: ASL	R1		;NO, AIM AT THE NEXT DRIVE
1290	006654	00530A			DEC	R0		;IS THIS THE ONE ?
1291	006656	001375			BNE	1S		;NO, LOOK AGAIN
1292	006660	040167	171462		2S: RJC	R1,DVICE		;DROP THE DRIVE
1293	006664	012777	100000	172504	MOV	#BIT15,#RKCS1		;ISSUE CONTROLLER CLEAR
1294	006672	004767	176626		JSR	PC,CNTRDY		;GO WAIT FOR CU READY
1295	006676	005777	172474		TST	#RKCS1		;ANY ERRORS?
1296	006702	100005			RPL	3S		
1297	006704	012767	007725	172564	MOV	#UNCLR,DRPMSG		;SET ERROR MESSAGE
1298	006712	006167	173076		JMP	ENTEST		;DROP MODULE
1299	006716	062716	000002		3S: ADD	#2,(SP)		
1300	006722	000267			RTS	PC		;RETURN
1301								
1302								
1303								

1304 006724 016700 171424
 1305 006730 042700 100000
 1306 006734 012701 064734
 1307 006740 0050C2
 1308 006742 16701 171175
 1309 006746 16201
 1310 006750 020001
 1311 006752 003401
 1312 006754 160100
 1313 006756
 1314
 1315 006756 005067 171342
 1316 006762 005067 171340
 1317 006766 022700 000102
 1318 006772 101005
 1319 006774 005267 171324
 1320 007000 162700 000102
 1321 007004 006770
 1322
 1323 007006 022700 000026
 1324 007012 101005
 1325 007014 005267 171306
 1326 007020 162700 000026
 1327 007024 000770
 1328
 1329 007026 000367 171274
 1330 007032 000067 171270
 1331 007036 000207
 1332

BLKADR: MOV BLK1,PC ;GET CURRENT BLOCK NUMBER
 BIC #100000,R0 ;CLEAR THE SIGN BIT
 MOV #27100,R1 ;MAX BLOCK ADDRESS
 CLP R2
 MOVBR WRUFSZ+1,R2 ;GET NUMBER OF SECTORS
 SUB R2,P1 ;SUBTRACT NUMBER OF SECTORS TRANSFERED
 CMP R0,R1 ;MAKE SURE XFER WON'T RUN OFF END
 BLS ;ITS OK
 STR R1,PC ;MAKE IT SOME SMALLER NUMBER
 1S:
 CLP CYLADR
 CLP SECADR
 2S: CMP #66,PC ;GET CYLINDER NUMBR
 BHI 3S ;ON THIS TRACK
 INC CYLADR ;NO - INCREMENT CYLINDER ADDRESS
 SUB #66,PC ;SECTORS PER CYLINDER
 BR ;FIND RIGHT CYLINDER
 3S: CMP #22,R0 ;FIND TRACK
 BHI 4S ;ON RIGHT TRACK
 INC SECADR ;RUMP TRACK COUNTER
 SUB #22,PC ;22 SECTORS PER TRACK
 BR ;CONTINUE
 4S: SWAP SECADR ;ALIGN TRACK BITS
 BIS PC,SECADR ;COMBINE WITH SECTOR ADDRESS
 RTS PC ;RETURN

1333 007040 012700 000366
 1334 007044 016701 171062
 1335 007050 005067
 1336 007052 005301
 1337 007054 001375
 1338 007056 000207
 1339
 1340
 1341
 1342
 1343 007060 016700 171260
 1344 007064 052700 000010
 1345 007070 004767 177160
 1346 007074 000167 172560
 1347
 1348 007100 010077 172302
 1349
 1350 007104 016777 172462 172264
 1351 007112 000207

CLRRB: MOV #RBUF,R0 ;CLEAR RBUF BUFFER
 MOV RRUFSZ,R1 ;GET ITS ADDR AND SIZE
 CLRCOM (R0)+ ;CLEAR ANOTHER
 DEC R1 ;COUNT ANOTHER
 BNE CLRRCOM ;BR BACK TILL DONE
 RTS PC
 RELEASE: MOV DSKADR,R0
 BIS #10,PC ;PUT RELEASE BIT IN
 JSR PC,RDY1
 JMP RESTRT
 1S: MOV R0,@RKCS2 ;WRITE THE COMMAND
 MOV #1,@RKCS1 ;DO IT
 MOV SELDRV,@RKCS1 ;DO THE COMMAND
 RTS PC

1352	007114	001542		ERTAB:	SVC51		; POINTER TO ADDRESS OF REGISTER
1353	007116	007324			FMC51		; START OF ERROR MESSAGES FOR THIS REGISTER
1354	007120	141777			141777		; BIT MASK
1355							
1356	007122	001552			SVC52		
1357	007124	007264			FMC52		
1358	007126	000377			000377		
1359							
1360	007130	001554			SVDS		
1361	007132	007244			EMDS		
1362	007134	173707			173707		
1363							
1364	007136	001556			SVFR		
1365	007140	007144			EMFR		
1366	007142	000000			000000		
1367							
1368	007144	046111	000106	EMER:	.ASCIZ	ILF	; ILLEGAL FUNCTION
1369	007150	045523	000111		.ASCIZ	SKI	; SEEK INCOMPLETE
1370	007154	054116	000106		.ASCIZ	NXF	; NON-EXECUTABLE FUNCTION
1371	007160	051104	000120		.ASCIZ	DRP	; DRIVE PARITY ERROR
1372	007164	046506	000124		.ASCIZ	FMT	; FORMAT
1373	007170	0521C4	000131		.ASCIZ	DTV	; DRIVE TYPE ERROR
1374	007174	041505	000110		.ASCIZ	ECH	; ERROR CORRECTION HARD
1375	007200	051502	000105		.ASCIZ	RSE	; RAD SECTOR ERROR
1376	007204	053110	000125		.ASCIZ	HVP	; HEADER VRC ERROR
1377	007210	047503	000105		.ASCIZ	COE	; CYLINDER OVERFLOW ERROR
1378	007214	042111	000101		.ASCIZ	IDA	; INVALID DISK ADDRESS ERROR
1379	007220	046127	000105		.ASCIZ	WLF	; WRITE LOCK
1380	007224	052104	000105		.ASCIZ	DTE	; DRIVE TIMMING ERROR
1381	007230	050117	000111		.ASCIZ	OPI	; OPERATION INCOMPLETE
1382	007234	047125	000123		.ASCIZ	UNS	; DRIVE UNSAFE
1383	007240	041504	000113		.ASCIZ	DCK	; DATA CHECK
1384							
1385	007244	041501	000114	EMDS:	.ASCIZ	ACL	; DRIVE AC POWER LOW
1386	007248	042504	000124		.ASCIZ	DCI	; DRIVE DC POWER LOW
1387	007254	042504	000124		.ASCIZ	DOT	; DRIVE OFF TRACK
1388	007260	051127	000114		.ASCIZ	WRL	; WRITE LOCK
1389							
1390	007264	043125	000105	EMCS2:	.ASCIZ	UPE	; UNIT FIELD ERROR
1391	007270	042115	000123		.ASCIZ	MDS	; MULTIPLE DRIVE SELECT
1392	007274	043520	000105		.ASCIZ	PGF	; PROGRAMMING ERROR
1393	007300	042515	000115		.ASCIZ	NEM	; NON EXISTANT MEMORY
1394	007304	042516	000104		.ASCIZ	NED	; NON EXISTANT DRIVE
1395	007310	041557	000105		.ASCIZ	WPE	; WRITE PARITY ERROR
1396	007314	041557	000105		.ASCIZ	WCE	; WRITE CHECK ERROR
1397	007320	046104	000124		.ASCIZ	DLT	; DATA LATE
1398							
1399	007324	042103	000124	EMCS1:	.ASCIZ	CDT	; CONTROLLER DRIVE TYPE
1400	007330	052103	000117		.ASCIZ	CTO	; CONTROLLER TIMEOUT
1401	007334	043103	000115		.ASCIZ	CFM	; CONTROLLER FORMAT
1402	007340	050123	000104		.ASCIZ	SPA	; SERIAL PARITY ERROR
1403	007344	042503	000122		.ASCIZ	CFP	; CONTROLLER ERROR
1404							
1405	007350	020040	000	BLNKS:	.ASCIZ	" "	; TWO BLANKS
1406	007354	007354			.FVEN		

1407							
1408	007354	044040	051101	020104	HARD:	.ASCIZ	" HARD ERROR"
1409	007362	051105	047517	000124			
1410	007370	051446	043117	020124	SOFT:	.ASCIZ	" SOFT ERROR"
1411	007376	051105	043117	000124			
1412	007404	042040	044522	042526	DRVE:	.ASCIZ	" DRIVE "
1413	007412	000040					
1414	007414	042040	047522	050120	DRPFD:	.ASCIZ	" DROPPED "
1415	007422	042105	000040				
1416	007426	042040	044522	042526	LOAD:	.ASCIZ	" DRIVE 0 DROPEED LOAD MEDIUM"
1417	007434	030040	042040	047522			
1418	007442	042520	042105	046040			
1419	007450	046517	020104	042514			
1420	007456	046504	046504	000045			
1421	007456	053440	044522	042524	PROT:	.ASCIZ	" WRITE PROT.*"
1422	007472	050040	047522	027124			
1423	007500	000045					
1424	007502	042040	044522	042526	UNAVAIL:	.ASCIZ	" DRIVE UNAVAILABLE*"
1425	007510	052440	040516	040526			
1426	007516	046111	041101	042514			
1427	007524	000045					
1428	007526	042040	044522	042526	NOTRED:	.ASCIZ	" DRIVE NOT READY*"
1429	007534	047040	052117	051040			
1430	007542	040505	045014	000045			
1431	007550	047040	020117	053101	GONE:	.ASCIZ	" NO AVAILABLE DRIVES*"
1432	007556	044501	040514	046102			
1433	007564	020105	051104	053111			
1434	007572	051505	000045				
1435	007576	051040	052105	054522	RTRY:	.ASCIZ	" RETRY EXCEEDED*"
1436	007604	042440	041530	042505			
1437	007612	042504	022504	000			
1438	007617	040140	047503	052116	CNTRNV:	.ASCIZ	" CONTROLLER NOT READY*"
1439	007624	047040	046114	051040			
1440	007632	047040	052117	051040			
1441	007640	040505	044504	000045			
1442	007646	042440	052130	040522	EXTRA:	.ASCIZ	" FYTRANEOUS INTERRUPT*"
1443	007654	042516	046517	020124			
1444	007662	047111	042524	051122			
1445	007670	050125	022524	000			
1446	007675	040	047125	041101	CALIB:	.ASCIZ	" UNABLE TO RECALIBRATE*"
1447	007702	042514	052040	020117			
1448	007710	042522	046503	044514			
1449	007716	051102	052101	022505			
1450							
1451	007725	040	047125	041101	UNCLP:	.ASCIZ	" UNABLE TO CLEAR*"
1452	007732	042514	052040	020117			
1453	007740	046103	040505	022522			
1454	007746	000					
1455	007747	040	051104	053111	DRVERR:	.ASCIZ	" DRIVE ERROR*"
1456	007754	020105	051105	047522			
1457	007762	022522	000				
1458	007765	117	044124	051105	PORTHC:	.ASCIZ	" OTHER PORT NOT UPDATING BLOCK ZERO*"
1459	007775	050240	051114	020124			
1460	010006	040504	044524	043516			
1461	010006	040504	044524	043516			
1462	010014	041040	047514	045503			

RKBD DEC/X11 SYSTEM EXERCISEP MODULE
IRKBD0.P11 12-OCT-78 12:08

MACV11 3CA(1052) 12-OCT-78 16:50 PAGE 49
CROSS REFERENCE TABLE -- USER SYMBOLS

RUN-TIME RATIO: 29/6=3.7
CORE USED: 7K (13 PAGES)