

RKAG DEC/X11 SYSTEM EXERCISER MODULE
XRKAG0.P11 12-OCT-78 12:07

MACY11 30A(1052) 12-OCT-78 16:58 PAGE 2

SEQ 0001

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E676G-WC
PRODUCT NAME: CXRKAG0 DEC/X11 RK11 MODULE
DATE: SEPTEMBER 1978
MAINTAINER: DECX11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSIDERED 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 OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RKA IS AN IOMODX THAT EXERCISES RK02, RK03, RK04, RK05 DISK DRIVES ON AN RK11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RK DISK DRIVES WITH AN RK11 CONTROLLER

STORAGE:: RKA REQUIRES:

- 1. DECIMAL WORDS: 1057
- 2. OCTAL WORDS: 02041
- 3. OCTAL BYTES: 4102

3. PASS DEFINITION

ONE PASS OF THE RKA MODULE CONSISTS OF 512 CYCLES OF THE BASIC TEST SEQUENCE (WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

4. EXECUTION TIME

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

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177400, VECTOR: 220, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

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 DISK ADDRESS AND A FRESH BLOCK OF DATA
- D. GET A DRIVE ADDRESS
- 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
- J. IF END OF DRIVES, GO TO C ELSE GO TO D

8. OPERATION OPTIONS

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):
WILL NOT TYPE OUT DATA LATE ERRORS BUT WILL KEEP TRACK
OF THE NUMBER OF DATA LATE ERRORS

SR1 BIT 2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND KEEP TRACK OF THE NUMBER
OF DATA LATE ERRORS IN "DLTCNT"

9. NON-STANDARD PRINTOUTS

- A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN
THE DEC/X11 DOCUMENT
- B. ERROR MESSAGES DUMP THE CONTENTS OF THE 8 RK11 REGISTERS
IN THE FOLLOWING ORDER:

RKDS RKER RKCS RKWC RKBA RKDA RKMR RKDB

```

000000*  IOMODX <RKAG > 177400,220,5,0,0,512,5,BUFIN,256,,1024.
000000*  MODULE 150000,RKAG,177400,220,5,0,0,512,5,BUFIN,256,,1024.
; .TITLE RKAG DEC/X11 SYSTEM EXERCISER MODULE
; DOXCOM VERSION 6 23-MAY-78
*****
000000*  BEGIN: .ASCII /RKAG / ;MODULE NAME.
000000*  XFLAG: 0 BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000000*  ADDR: 177400+0 ;1ST DEVICE ADDR.
000000*  VECTOR: 220+0 ;1ST DEVICE VECTOR.
000000*  BR1: 0 BYTE PRTV5+0 ;1ST BR LEVEL.
000000*  BR2: 0 BYTE PRTV0+0 ;2ND BR LEVEL.
000000*  DIVD1: 0+1 ;DEVICE INDICATOR 1.
000000*  SR1: OPEN ;SWITCH REGISTER 1
000000*  SR2: OPEN ;SWITCH REGISTER 2
000000*  SR3: OPEN ;SWITCH REGISTER 3
000000*  SR4: OPEN ;SWITCH REGISTER 4
*****
000000*  STAT: 150000 ;STATUS WORD
000000*  START: 177400 ;MODULE START ADDR.
000000*  SPOINT: MODSP ;MODULE STACK POINTER.
000000*  PASCNT: 0 ;PASS COUNTER.
000000*  ICOUNT: 512. ;# OF ITERATIONS PER PASS=512.
000000*  SOFCNT: 0 ;LOC TO COUNT ITERATIONS.
000000*  HRDPCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000000*  SOFPCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000000*  HRDPCNT: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000000*  HRDPCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000000*  SVSVCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000000*  RANRNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000000*  CENFIG: 0 ;RESERVED FOR MONITOR USE
000000*  RES: 0 ;RESERVED FOR MONITOR USE
000000*  SVR0: OPEN ;LOC TO SAVE R0.
000000*  SVR1: OPEN ;LOC TO SAVE R1.
000000*  SVR2: OPEN ;LOC TO SAVE R2.
000000*  SVR3: OPEN ;LOC TO SAVE R3.
000000*  SVR4: OPEN ;LOC TO SAVE R4.
000000*  SVR5: OPEN ;LOC TO SAVE R5.
000000*  SVR6: OPEN ;LOC TO SAVE R6.
000000*  CSRA: OPEN ;ADDR OF CURRENT CSR.
000000*  SHADR: ;ADDR OF GOOD DATA, OR
000000*  ACSR: OPEN ;CONTENTS OF CSR
000000*  WBSADR: ;ADDR OF BAD DATA, OR
000000*  ASTAT: OPEN ;STATUS REG CONTENTS.
000000*  ERPTVP: ;TYPE OF ERROR
000000*  ASR: OPEN ;EXPECTED DATA.
000000*  AWAS: OPEN ;ACTUAL DATA.
000000*  RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
000000*  WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
000000*  WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000000*  INTD: OPEN ;# OF INTERRUPTS PER ITERATION
000000*  IDNUM: 5 ;MODULE IDENTIFICATION NUMBER=5

```

```

000124* 002614*
000125* 000000* ;READ BUFFER VIRTUAL ADDRESS
000130* 000000* ;READ BUFFER PHYSICAL ADDRESS
000132* 000400* ;READ BUFFER EA BITS
000133* 000400* ;SIZE OF THE READ BUFFER
000134* 000000* ;WRITE BUFFER PHYSICAL ADDRESS
000136* 000000* ;WRITE BUFFER EA BITS
000140* 002000* ;WRITE BUFFER SIZE REQUESTED
000142* 000000* ;WRITE BUFFER SIZE AVAILABLE
000144* 000000* ;CDATA/DATCK ERROR COUNT
000146* 000000* ;CDATA/DATCK WORD COUNT
000150* 000000* ;RESERVED FOR FUTURE USE
000040*
;REPT SPSIZ ;MODULE STACK STARTS HERE.
;LIST
;WORD 0
;LIST
;ENDR
000252*
MODSP:
*****

```

```
222 000252 005067 002300 177630 START: CLR CNT ; ZERO END OF PASS TESTER
223 000252 012767 002400 177630 MOV #256,WDTO ; WORDS TO MEM
224 000252 012767 002000 177624 MOV #1024,WDFR ; WORDS FROM MEM
225 000272 012767 000003 177620 MOV #3,INTR ; # OF INTERRUPTS/ITERATION
226 000300 005067 003570 CLR STDE ; CLEAR FLAGS AND SIDE INDICATOR
227 000300 005067 003570 CLR DLT CNT ; CLEAR DATA LATE ERROR COUNTER
228 000310 012767 177500 MOV DVID1, DVICE ; GET DRIVE INDICATOR
229 000316 016767 002244 MOV DVICE, DRIVE ; ALSO SAVE IT IN DRIVE
230 000324 012767 177775 MOV #3, BLK1 ; INITIALIZE BLOCK COUNTER
231 000336 012767 186000 CLR DRYVE ; ZERO UNIT NUMBER
232 000344 122737 000002 MOV #160000, DRVSFT ; INITIALIZE THE SHIFTED DRIVE #
233 000352 012767 000002 CMPR #BIT1, #41 ; IS RK UNIT 0 THE LOAD MEDIUM ?
234 000352 012767 000002 SNE #3 ; NO, CONTINUE
235 000360 001026 MOV #0, R2 ; INITIALIZE DRIVE COUNT
236 000360 113706 MOV #40, R0 ; GET LOAD MEDIUM COUNT
237 000364 012701 MOV #1, R1 ; LOAD UP R1 TO POINT TO DRIVE #0
238 000370 105700 1S: RFB ; IF R0 EQUAL TO 0 THEN
239 000374 008301 RSL ; GO TO 2S
240 000376 105300 RSI ; ELSE UPDATE DRIVE POINTER
241 000376 008301 RSL ; INCREMENT COUNT
242 000376 105300 DECP R0 ; UPDATE DRIVE NUMBER
243 000400 008301 BR #4 ; TRY AGAIN
244 000400 130167 002156 2S: BITR R1, DVICE ; IF DRIVE NOT SELECTED TO BE TESTED THEN
245 000410 001407 BEQ #3 ; GO TO 3S
246 000412 010267 MOV R2, DRVE ; ELSE LOAD DRIVE ADDRESS TO BE DROPPED
247 000412 001407 JSR PC, DROP ; GO DROP IT
248 000422 104403 MSCNS, REGIN, DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
249 000430 012767 177777 002134 3S: MOV #-1, DRVE ; INITIALIZE DRIVE COUNTER
250 000430 004767 001572 JSR PC, SETUP ; GENERATE REGISTER ADDRESSES
251 000442 004767 001770 JSR PC, REZET ; INITIALIZE RK REGS. AND ALL DRIVES
252 000446 005767 002114 TST DVICE ; DROP THE 400ULE ?
253 000452 001536 BEQ #1 ; YES
254 000452 001407 BR #1 ; NO
255 000456 005767 002074 RESTRT: TST CNT ; THIS IS SUPPORT
256 000462 001001 RNE RSTRT1 ; FOR DT03
257 000462 009672 BR #1 ; BUS SWITCH
258 000466 104415 000000 RSTRT1: GETPAS, REGIN, RBUFVA ; GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
259 000474 016767 177432 MOV #RBUFSZ, WCNT2 ; SAVE READ BUFFER SIZE
260 000502 005467 002102 NEG WCNT2 ; GET THE 2'S COMPLEMENT
261 000506 004767 000572 STRT: PC, BLOCK ; GET NEXT BLOCK NUMBER
262 000512 104414 000000 CMBUFS, REGIN ; GET WRITE BUFFER INFORMATION
263 000516 016767 177420 MOV #RBUFSZ, WCNT1 ; SAVE WRITE BUFFER SIZE
264 000524 005467 002058 NEG WCNT1 ; GET THE 2'S COMPLEMENT
265 000530 016700 002042 MOV #1, R0 ; LOAD BLOCK # FOR CONVRT
266 000534 004767 001574 JSR PC, CONVRT ; GENERATE DISK ADR. FROM BLOCK #
267 000540 004767 001016 NEXT: JSR PC, DRVADR ; GET A DRIVE ADDRESS
268 000544 005767 002016 TST DVICE ; ANY DRIVES LEFT ?
269 000550 001477 RNE FINI ; NO, GO DROP THE MODULE
270 000552 132767 000010 003315 BITR #BIT3, FLAG ; ALL DRIVES DONE ?
271 000560 001552 RNE STRT ; YES, GO GET ANOTHER BLOCK
272 000562 042767 160000 001774 BIC #160000, DSKADR ; CLEAR DRIVE ADDRESS
```

```
278 000570 056767 002000 001766 RIS DRVSFT, DSKADR ; LOAD DRIVE ADDRESS
279 000576 016777 001762 003024 MOV DSKADR, ARKDA ; LOAD DISK ADDRESS
280 000584 032777 000040 003004 BIT #BITS, ARKDS ; WRITE PROTECTED ?
281 000612 001406 BEQ #1 ; NO, CONTINUE
282 000614 004767 000526 JSR PC, DROP ; YES, DROP THE DRIVE
283 000620 104403 000000 MSCNS, REGIN, DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
284 000626 009744 BR #NEXT ; GO ON TO NEXT DRIVE
285 000630 032777 000100 002760 1S: BIT #BIT6, ARKDS ; DRIVE READY ?
286 000636 001003 RNE #2 ; YES, CONTINUE
287 000640 004767 JSR PC, NOTRDY ; NO, WAIT FOR READY
288 000644 000720 BR #TRY ; TRY AGAIN
289 000646 005067 003224 2S: CLR TRY1 ; ZERO RETRY COUNTERS
290 000652 105067 003222 CLRR TRY3 ;
```

```

301
302
303 000656 004567 000212 GO: JSR R5,WRITE ; WRITE SOME DATA
304 000664 000434 BR RETRY1 ; IF ERRORS, TRY IT AGAIN
305 000672 000434 BITR #R12,FLAG ; DID THE DISK OVERFLOW ?
306 000672 001407 REQ GOA ; NO, CONTINUE
307 000674 002767 000004 003173 RTCP #R12,FLAG ; YES, CLEAR THE OVERFLOW FLAG
308 000702 002767 177775 001666 MOV #3,RLK1 ; RESET THE BLOCK NUMBER
309 000710 000672 BR STRT ; START OVER AT BEGINNING OF DISK
300 000711 004567 000210 GOA: JSR R5,WRITCK ; WRITE-CHECK THE DATA
301 000711 000434 BR RETRY2 ; IF ERRORS, TRY AGAIN
302 000720 004567 000234 GOR: JSR R5,READ ; READ THE DATA WRITTEN
303 000720 000437 BR RETRY1 ; IF ERRORS, TRY AGAIN
304 000726 004412 000000 000126 CDATAS,REGIN,RRUFPA ; REQUEST FOR MONITOR TO CHECK DATA
305 000734 000438 .+2 ; IF ERROR, CONTINUE
306
307
308 000736 005267 001614 PASS: INC CNT ; COUNT A CYCLE
309 000742 000000 ENDITS,REGIN ; SIGNAL END OF ITERATION.
310 000742 104413 000000 PR NEXT ; MONITOR SHALL TEST END OF PASS
311
312 000746 000674 FINI: ENDS,REGIN ; DROP THE MODULE
313 ;
314 000750 104410 000000 ;
315
316
317 000754 105267 000116 RETRY1: INCR TRV1 ; COUNT THE RETRYS
318 000760 122767 000003 003110 CMPR #3,TRV1 ; LIMIT EXCEEDED ?
319 000760 001330 RNE GO ; NO, GO TRY IT AGAIN
320 000778 004423 MSGNS,REGIN,EXCED1 ; ASCII MESSAGE CALL WITH COMMON HEADER
321 BR NEXTA ; GO ON TO NEXT DRIVE
322
323
324 001000 105267 000073 RETRY2: INCR TRV2 ; COUNT RETRYS
325 001000 000003 003065 CMPR #3,TRV2 ; LIMIT EXCEEDED ?
326 001012 001330 RNE GO ; NO, TRY AGAIN
327 001014 104403 MSGNS,REGIN,EXCED2 ; ASCII MESSAGE CALL WITH COMMON HEADER
328 BR NEXTA ; GO ON TO NEXT DRIVE
329
330
331 001024 105267 000150 RETRY3: INCR TRV3 ; COUNT RETRYS
332 001030 122767 000003 003042 CMPR #3,TRV3 ; LIMIT EXCEEDED ?
333 001030 001330 RNE GO ; NO, GO TRY AGAIN
334 001040 104403 MSGNS,REGIN,EXCED3 ; ASCII MESSAGE CALL WITH COMMON HEADER
335
336
337 001046 032767 NEXTA: BIT #R10,SRI ; DROP THE DRIVE ?
338 001052 001405 REQ IS ; YES, SKIP TO NEXT DRIVE
339 001052 000264 JSR PC,DROP ; YES, DROP OPENING DRIVE
340 001066 104403 MSGNS,REGIN,DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
341 001070 000179 JMP NEXT ; GO ON TO NEXT DRIVE
342 ;
  
```

```

343
344
345
346
347 ; ----- RK11 DISK DRIVERS -----
348
349 001074 012767 000503 001456 WRITE: MOV #503,FUNC ; LOAD WRITE FUNCTION
350 001102 016777 001500 002514 MOV WCNT1,ARKWC ; LOAD WORD COUNT
351 001110 016777 177320 002510 MOV WRUFPA,ARKRA ; LOAD BUFFER ADDRESS
352 001124 000462 RR WRUFPA,XMEM ; LOAD EXTENDED MEMORY RITS
353 001126 012767 000507 001424 WRITCK: MOV #507,FUNC ; LOAD WRITE-CHECK FUNCTION
354 001134 016777 001446 002462 MOV WCNT1,ARKWC ; LOAD WORD COUNT
355 001142 016777 176766 002456 MOV WRUFPA,ARKBA ; LOAD BUFFER ADDRESS
356 001150 016767 176762 001404 RR WRUFPA,XMEM ; LOAD EXTENDED MEMORY RITS
357 001156 000445 RR GOGO ; CONTINUE
358 001160 012767 000505 001372 READ: MOV #505,FUNC ; LOAD READ FUNCTION
359 001168 016777 001416 002430 MOV WCNT2,ARKWC ; LOAD WORD COUNT
360 001174 016777 176726 002424 MOV RRUFPA,ARKRA ; LOAD BUFFER ADDRESS
361 001202 016767 176722 001352 MOV RRUFPA,XMEM ; LOAD EXTENDED MEMORY RITS
362 001210 000430 RR GOGO ; CONTINUE
363
364
365 001212 012777 000001 002402 CLEAR: MOV #1,ARKCS ; ISSUE A CONTROL RESET
366 001220 004767 001254 002376 JSR PC,WAIT1 ; GO WAIT FOR CONTROLLER READY
367 001224 016777 001344 002376 MOV DRVSPT,ARKDA ; RELOAD THE DRIVE ADDRESS
368 001246 001001 RIT #R16,ARKDS ; DRIVE READY ?
369 001248 000205 RTS #S ; YES, CONTINUE
370 001244 012777 000015 002350 2S: MOV #15,ARKCS ; NO, ABORT DRIVE RESET
371 001252 003767 000054 JSR PC,WAIT ; ISSUE A DRIVE RESET
372 001264 004767 000051 002336 JSR #1,ARKCS ; GIVE IT TIME TO COMPLETE
373 001270 000205 RTS PC,WAIT1 ; ISSUE ANOTHER CONTROLLER RESET
374 ; WAIT FOR CONTROLLER READY
375 ; RETURN
376
377 001272 012777 001326 176510 GOGO: MOV #NTRUPT,VECTOR ; SET INTERRUPT ENTRY POINTER
378 001300 016777 001260 002322 MOV OSKADR,ARKDA ; LOAD THE DISK ADDRESS
379 001306 016767 001250 001244 RIT XMEM,FUNC ; LOAD EXTENDED MEMORY RITS
380 001314 016777 001240 002300 MOV FUNC,ARKCS ; EXECUTE THE FUNCTION
381 EXITS,REGIN ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
382
383 001326 NTRUPT: ;
384 ; -----
385 001326 000304 000000 001334 FIRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI
386 ;
387
388 001334 004567 000512 1S: JSR R5,ERFORS ; GO CHECK FOR ERRORS
389 001340 000205 RTS #R5 ; ERRORS DETECTED, RETURN
390 001342 005725 TST (R5)+ ; NO ERRORS, SKIP RETRY
391 001344 000205 RTS #R5 ; RETURN OK
392 ;
  
```

```

391
392
393 001344 012701 000001 DROP: MOV #1,R1 ; INITIALIZE DROP PICKER
394 001345 016700 001214 MOV DRVVE,R0 ; GET THE DRIVE NUMBER
395 001346 001403 BFO #2 ; IF DRIVE 0 GO DROP IT
396 001360 006301 1S: ASL B1 ; NO, AIM AT THE NEXT DRIVE
397 001361 005300 DEC R0 ; IS THIS THE ONE ?
398 001362 001375 BNE B1 ; NO, LOOK AGAIN
399 001366 001174 2S: BFC #1,DRIVE ; DROP THE DRIVE
400 ;*****
401 ;CONVERT DRIVE TO ASCII AND
402 ;STORE AT ADRI
403 001372 104420 000000 002572 OTOAS,BEGIN,DRVVE,ADRI
404 001400 004064
405
406 001402 000207
407
408 ;*****
409 ;*****
410 ;*****
411 ;*****
412 ;*****
413 ;*****
414 ;*****
415 ;*****
416 ;*****
417 ;*****
418 ;*****
419 ;*****
420 ;*****
421 ;*****
422 ;*****
423 ;*****
424 ;*****
425 ;*****
426 ;*****
427 ;*****
428 ;*****
429 ;*****
430 ;*****
431 ;*****
432 ;*****
433 ;*****
434 ;*****
435 ;*****
436 ;*****
437 ;*****
438 ;*****
439 ;*****
440 ;*****
441 ;*****
442 ;*****
443 ;*****
444 ;*****
445 ;*****
446 ;*****

```

```

447
448
449 001562 005267 001004 DRVADR: INC DRVVE ; COUNT A DRIVE
450 001563 062767 001000 ADD #RIT3,DRVSFT ; DRIVE COUNT LINED UP WITH RKDA
451 001574 142767 000910 BICR #RIT3,FLAG ; CLEAR END OF DRIVES FLAG
452 001602 022767 000910 CMP #B,DRVVE ; ALL DRIVES CHECKED ?
453 001610 001404 1S: BFO #1 ; YES, GO FLAG END OF DRIVES
454 001612 006267 000752 ASR DRIVE ; NO, IS NEXT DRIVE CHOSEN ?
455 001620 000207 RTS DRVADR ; NO, GO TRY ANOTHER DRIVE
456
457
458 001622 152767 000010 002245 1S: BISR #RIT3,FLAG ; SET END OF DRIVES FLAG
459 001630 015767 177777 000704 MOV #1,DRIVE ; SET DRIVE COUNTER
460 001636 012767 160000 000730 MOV #160000,DRVSFT ; ZERO THE SHIFTED DRIVE #
461 001644 016767 000716 000716 MOV DVICE,DRIVE ; RESTORE CHOSEN DRIVES
462 001652 000207 RTS PC ; RETURN
463
464
465
466
467 001654 012767 177777 000710 NOTRDY: MOV #1,DRIVE ; START WITH FIRST DRIVE
468 001670 016767 000672 000672 MOV #160000,DRVSFT ; RESET DRIVE SELECT
469 001676 004767 177660 1S: JSR PC,DRVADR ; GET A DRIVE ADDRESS
470 001700 012767 000910 BICR #RIT3,FLAG ; ALL DRIVES CHECKED ?
471 001710 001171 2S: BFO #1 ; YES, RETURN
472 001712 016777 000656 001710 MOV DRVSFT,ARKDA ; NO, LOAD NEXT DRIVE ADDRESS
473 001720 032777 000100 001670 PIT #RIT6,ARKDS ; IS THIS DRIVE READY ?
474 001730 004767 000046 RNE #1 ; YES, CONTINUE
475 001734 000760 JSR #1,WAIT ; NO, WAIT FOR IT
476 001736 000207 BR 1 ; GO CHECK REST OF DRIVES
477 RTS PC ; RETURN
478
479
480
481 001740 014167 176142 ERSUR2: MOV -(R1),ASB ; LOAD THE DATA
482 001744 010167 176134 MOV -R1,ASADR ; LOAD ADDRESS OF DATA WRITTEN
483 001754 010267 176124 MOV R2,WASADR ; LOAD ADDRESS OF DATA READ
484 001760 005721 TST (R1)+ ; RESET REG. 1
485 001762 005722 TST (R2)+ ; RESET REG. 2
486
487
488 001764 016767 001632 176106 ERSUR1: MOV RKCS,CSRA ; LOAD ADR. OF CURRENT CSR
489 001772 017767 001624 176102 MOV #RKCS,ACSR ; LOAD CONTENTS OF CURRENT CSR
490 002000 000207 RTS PC ; RETURN
491

```

```

492
493
494
495
496 002002 012767 077777 001604 WAIT: MOV #77777,CLK ; SET THE TIMER
497 002010 000000 000000 1S: RTS ;
498 002014 104407 000000 BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR...
499 002014 104407 000000 BREAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
500 002020 032777 000100 001570 BIT #BIT6,ARKDS ; DRIVE READY ?
501 002020 001510 001560 BNE 2S ; YES, RETURN
502 002030 005369 001560 DEC CLK ; NO, WAIT SOME MORE ?
503 002034 001365 001560 BNE 1S ; YES, WAIT
504 002038 001767 177304 JSR PC,DROP ; TIME-OUT, DROP THE DRIVE
505 002042 104403 000000 MSGNS,BEGIN,DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
506 002050 000207 000000 RTS ; RETURN
507
508 002052 004767 177705 ERRORS: JSR PC,ERSUB1 ; LOAD ERROR INFORMATION
509 002056 032777 040090 BIT #BIT14,ARKCS ; HARD ERROR ?
510 002066 032777 000003 001524 BNE 4S ; YES, GO REPORT
511 002074 001041 000003 BIT #3,ARKER ; SOFT ERROR ?
512 002074 000723 000003 BNE 3S ; YES, GO REPORT
513 002074 000723 000003 RTS (R5)+ ; NO, SKIP RETRY
514 002100 032777 040000 001510 1S: BIT #BIT14,ARKER ; DISK OVERFLOW ?
515 002110 001403 000000 REQ 7S ; NO, CONTINUE
516 002114 004567 177320 JSR R5,ROOM ; YES, IS IT A REAL ERROR ?
517 002114 004567 000000 JSR R5,ROOM ; NO, CONTINUE
518 002120 032777 001000 001472 7S: BIT #BIT9,ARKER ; DATA LATE ERROR?
519 002126 001411 000000 REQ 2S ; NO
520 002130 005266 000420 INC DLTCNT ; INCREMENT ERROR COUNTER
521 002134 001013 000004 175654 BIT #BIT2,SR1 ; TYPE OUT ERROR?
522 002142 001013 000000 BNE 6S ; NO
523 002144 104403 000000 MSGNS,BEGIN,DLTERR ; ASCII MESSAGE CALL WITH COMMON HEADER
524 002150 000000 000000 2S:
525 002160 005067 175722 MSGNS,BEGIN,HARD ; ASCII MESSAGE CALL WITH COMMON HEADER
526 CLR ERRPT ;
527 *****
528 HDRS,BEGIN,TABLE *****
529
530 002172 104405 000000 003616 6S: JSR R5,CLEAR ; GO CLEAR OUT ERRORS
531 002176 000411 000000 BR 4S ; RETURN
532
533 002200 104403 000000 004022 3S: MSGNS,BEGIN,SOFT ; ASCII MESSAGE CALL WITH COMMON HEADER
534 002206 012767 000001 175672 MOV #1,ERRPT ; DATA ERROR
535 *****
536 SOFES,BEGIN,TABLE *****
537
538 002214 104406 000000 003616 4S: RTS ; RETURN
539 002224 000205 176762 5S: JSR R5,CLEAR ; CLEAR OUT ERRORS
540 002230 005723 000000 TST (R5)+ ; SKIP RETRY
541 002232 000205 000000 RTS ; RETURN OK
542

```

```

544
545
546 002234 016700 175546 SETUP: MOV ADDR,R0 ; GET DEVICE ADDRESS
547 002244 005720 001352 MOV R0,ARKDS ; GENERATE CONTROLLER REGS. ADDRESSES
548 002246 010067 001346 TST R0,ARKER ;
549 002252 005720 001342 MOV (R0),RKCS ;
550 002256 005720 001336 MOV (R0),RKWC ;
551 002266 005720 001332 TST (R0)+ ;
552 002270 010067 001332 MOV (R0),RKBA ;
553 002274 005720 001326 TST (R0)+ ;
554 002276 010067 001322 MOV (R0),RKDA ;
555 002302 005720 001322 TST (R0)+ ;
556 002304 010067 001316 MOV (R0),RKDB ;
557
558 002316 016700 175466 MOV VECTOR,R0 ; GET THE VECTOR ADDRESS
559 002320 016700 000306 MOV BRT,(R0)+ ; SET POINTER JUST IN CASE
560 002326 116710 175460 MOVB BRT,(R0) ; SET PRIORITY
561
562 002332 000207 2S: RTS PC ; RETURN
563
564
565 002334 005001 001532 CONVRT: CLR R1 ; ZERO REG. 1
566 002336 105067 001532 CLR R2 ; ZERO THE SIDE INDICATOR
567 002342 012703 177764 MOV #12,R3 ; LOAD REG. 3
568 002346 012704 000013 MOV #11,R4 ; LOAD REG. 4
569 002352 005700 004537 CMP #2399,R0 ; IS BLOCK ON SIDE 0 ?
570 002356 005700 000020 BCS 1S ; YES, CONTINUE
571 002360 152767 000020 BCS 1S ; NO, CONTINUE
572 002366 002700 000020 ADD #2400,R0 ; NORMALIZE BLOCK # FOR SIDE 1
573 002370 002700 173240 CMP R4,R0 ; FIND THE RIGHT CYLINDER ?
574 002374 002003 000020 BCS 1S ; YES, CONTINUE
575 002376 006030 000020 ADD R3,R0 ; NO, SUBTRACT 12 SECTORS (1 CYLINDER)
576 002400 005201 000020 INC R1 ; KEEP TRACK OF CYLINDER ADDRESS
577 002404 000723 000154 BR 1S ; GO TRY AGAIN
578 002410 156767 000146 MOV R0,DSKADR ; LOAD THE SECTOR ADDRESS
579 002416 012702 000005 BSR SIDE,DSKADR ; LOAD THE SIDE ADDRESS
580 002422 006301 000005 MOV #5,R2 ; SET UP FOR SHIFT
581 002426 003375 000130 ASL R1 ; LINE UP CVL. ADR. WITH DSKADR
582 002430 005167 000130 BGT R1,DSKADR ; DONE ?
583 002434 000207 000130 BSR R1,DSKADR ; YES, LOAD THE CYLINDER ADDRESS
584
585

```



```

593 002436 012777 000001 001156 REZET: MOV #1, @RKCS ; EXECUTE CONTROLLER RESET
594 002444 004767 000030 ; GO WAIT FOR CONTROLLER READY
595 002454 004767 177700 ; MAKE SURE ALL CHOSEN DRIVES ARE READY
596 002454 004767 177700 1S: JSR PC, @DRVADR ; GET A DRIVE ADDRESS
597 002460 132767 000010 001407 BITR #BIT3, FLAG ; ALL DRIVES DONE ?
598 002466 004809 176516 ; YES, RETURN
599 002474 000767 ; ISSUE DRIVE RESET AND CONTROLLER CLEAR
600 002474 000767 ; KEEP GOING
601 002476 000207 ; RETURN
602 ;
603 ;
604 ;
605 002500 012767 077777 001106 WAIT1: MOV #77777, CLK ; SET THE TIMER
606 002505 105777 001110 1S: TSTB @RKCS ; CONTROLLER READY ?
607 002514 104407 000000 ; YES, CONTINUE
608 002520 104407 000000 ; TEMPORARY RETURN TO MONITOR...
609 002520 104407 000000 ; THEN CONTINUE AT NEXT INSTRUCTION.
610 002524 005367 001064 DEC CLK ; WAIT SOME MORE ?
611 002530 001366 ; YES
612 002532 012767 000003 175346 MOV #3, @ERRTYP ; CONTROLLER NOT READY
613 ; *****
614 002540 104405 000000 003616 HRDERS, BEGIN, TABLE ; CONTROLLER NOT READY
615 ; *****
616 002546 000167 176176 JMP PC, FINI ; GO DROP THE MODULE
617 002552 000207 2S: RTS PC ; READY, RETURN
618 ;
619 ;
620 002554 000000 DLTCNT: 0
621 002556 000000 CNT: 0
622 002560 000000 FUNC: 0
623 002564 000000 XMEM: 0
624 002564 000000 DSKADR: 0
625 002566 000000 DVICE: 0
626 002570 000000 DRIVE: 0
627 002574 000000 DRVSFT: 0
628 002576 000000 BLK1: 0
629 002600 000000 BLK2: 0
630 002604 000000 BSIZE: 0
631 002606 000000 TBUF: 0
632 002610 000000 WCNT1: 0
633 002614 000000 WCNT2: 0
634 002618 000400 BUFLW: 256
635 002622 000400 BUFLN: 256
636 002626 000000 CLK: 0
637 002630 000000 TABLE: 0
638 002634 000000 RKDS: 0
639 002638 000000 RKERS: 0
640 002642 000000 RKCS: 0
641 002646 000000 RKWC: 0
642 002650 000000 RKBA: 0
643 002654 000000 RKMR: 0
644 002658 000000 RKDB: 0
645 002662 000000
646 002666 000000
647 002670 177777

```

```

648 003640 020040 044040 051101 MES1: .ASCIZ " HARD ERROR"
649 003644 020104 051105 047522
650 003654 000122 051440 043117 MES2: .ASCIZ " SOFT EROR"
651 003658 020040 051105 047522
652 003662 000122 051104 053111 MES4: .ASCIZ " DRIVE "
653 003666 000122 051104 050117 MES5: .ASCIZ " DROPPED*"
654 003670 020040 042522 051124 MES6: .ASCIZ " RETRY EXCEEDED*"
655 003674 020105 054105 042503
656 003678 042504 042504 042503
657 003682 000000 042504 042503
658 003686 000000 042504 042503
659 003690 000000 042504 042503
660 003694 000000 042504 042503
661 003698 000000 042504 042503
662 003702 000000 042504 042503
663 003706 000000 042504 042503
664 003710 000000 042504 042503
665 003714 000000 042504 042503
666 003718 000000 042504 042503
667 003722 000000 042504 042503
668 003726 000000 042504 042503
669 003730 000000 042504 042503
670 003734 000000 042504 042503
671 003738 000000 042504 042503
672 003742 000000 042504 042503
673 003746 000000 042504 042503
674 003750 000000 042504 042503
675 003754 000000 042504 042503
676 003758 000000 042504 042503
677 003762 000000 042504 042503
678 003766 000000 042504 042503
679 003770 000000 042504 042503
680 003774 000000 042504 042503
681 003778 000000 042504 042503
682 003782 000000 042504 042503
683 003786 000000 042504 042503
684 003790 000000 042504 042503
685 003794 000000 042504 042503
686 003798 000000 042504 042503
687 003802 000000 042504 042503
688 003806 000000 042504 042503
689 003810 000000 042504 042503
690 004064 000005 DLTERR: MESS9
691 004071 0000 NUMR: 5
692 004074 0000 STDE: .BVTB 0
693 004077 0000 FLAG: .BVTB 0
694 004080 0000 .EVEN
695 004083 0000 TRV1: .BVTB 0
696 004086 0000 TRV2: .BVTB 0
697 004089 0000 TRV3: .BVTB 0
698 004100 0000 .EVEN
699 004102 0000
700 000001 .END

```


