

1

.NLIST TTM

.NLIST SEQ,LD,BIN
.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-F098C-MC
PRODUCT NAME: CXRXE0 RX02 MODULE
PRODUCT DATE: APRIL 1979
MAINTAINER: TAPE DIAGNOSTICS

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) 1978, 1979 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RXB IS AN IOMODX THAT EXERCISES TWO RX02 FLOPPY DISKS ON THE UNIBUS. IT EXERCISES BOTH DRIVES BY WRITING AND READING ALL AVAILABLE DRIVES.

ERRORS ARE CHECKED FOR BUFFER FILL, WRITE, READ, AND DATA COMPARE. TWO RETRIES ARE DONE FOR EACH WRITE OR READ STATUS ERROR. ALL ERRORS ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 OR 2 DISKETTES WITH AN RX02 CONTROLLER

STORAGE: RXB REQUIRES:

- 1. DECIMAL WORDS: 1661
- 2. OCTAL WORDS: 03175
- 3. OCTAL BYTES: 6372

3. PASS IDENTIFICATION

ONE PASS OF THE RXB MODULE CONSISTS OF 130. WRITE AND READ PASSES ON AVAILABLE DRIVES. THE TEST SEQUENCE WRITES THEN READS EVERY THIRD SECTOR OF EVERY TENTH TRACK STARTING AT TRACK 1 SECTOR 1.

THE ENTIRE DISKETTE IS DONE, EACH PASS OF THE DISKETTE STARTS AS SHOWN:

- 1. STARTS AT SECTOR #1/TRACK #1
END OF PASS
- 2. STARTS AT SECTOR #2/TPACK #1
END OF PASS
- 3. STARTS AT SECTOR #3/TRACK #1
END OF PASS
- 4. STARTS AT SECTOR #1/TRACK #2
...ETC.

4. EXECUTION TIME

ONE PASS OF RXB RUNNING ALONE ON THE PDP-11/05 AVERAGES
.75 MINUTES FOR 2 DRIVES.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS: DEVADR:177170
VECTOR:264
ERI:5
DEV CNT:2

REQUIRED PARAMETERS: NONE

6. DEVICE/OPTION SETUP

ASSURE ALL DRIVES ARE POWERED UP, DISKETTES INSTALLED,
AND READY.

7. MODULE OPERATION

- A. SETUP DRIVE REGISTER ADDRESSES AND MODULE VARIABLES
- B. SELECT DRIVES FOR TEST - IF NONE AVAILABLE, DROP MODULE
- C. INITIALIZE DRIVES
- D. SET AVAILABLE DRIVES TO SINGLE DENSITY
- E. WRITE AND READ ALL AVAILABLE DRIVES - IF ERROR, REPORT AND
RETRY FAILING FUNCTION UP TO RETRY LIMIT.
- F. DO DATA COMPARE FOR ALL READS - IF ERROR, REPORT
- G. UPDATE TRACK + SECTOR - DO E. + F. UNTIL ALL SECTORS
OF AVAILABLE DRIVES ARE DONE.
- H. CHANGE DENSITY OF AVAILABLE DRIVES
- I. DO E., F. & G. FOR NEW DENSITY
- J. UPDATE STARTING ADDRESS, GO TO D

8. OPERATION OPTIONS

SRI BIT 0 SET(1):
IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, REPORT
HARD ERROR AND DROP THE MODULE.

SRI BIT 0 CLEAR(0):
IF RETRY LIMIT IS EXCEEDED, CONTINUE WITH NEXT TEST.

9. NON-STANDARD PRINTOUTS

- A. ALL PRINTOUTS HAVE THE STANDARD FORMAT DESCRIBED IN THE DEC/X11 DOCUMENT.
- B. ERROR MESSAGES DUMP THE CONTENTS OF THE RX02 REGISTERS IN THE FOLLOWING ORDER.

```

RXCS (COMMAND REGISTER)
RXES (ERROR REGISTER)
RXTA (TRACK ADDRESS)
RXSA (SECTOR ADDRESS)
RXSB (WORD COUNT REG.)
RXSB1 (DRV1 TRACK ADR)
RXSB2 (TARGET SECTOR)
RXSB3 (TRACK ADR SELECTED DRV*)
      / (DEFINITIVE ERROR CODE)->
      / (DRV# TRACK ADR)
      / (TARGET TRACK)
      / (MICROCODE STATUS)->
      / (LOWER BYTE-ODD)
      / (LOWER BYTE-EVEN)
  
```

* = ONLY MEANINGFUL ON A CODE 150 ERROR, SEEK ERROR.

```

MICROCODE STATUS <-----
-----
  
```

```

BIT #7 = UNIT SELECT
BIT #6 = DENSITY DRIVE #1
BIT #5 = HEAC LOAD
BIT #4 = DENSITY DRIVE #0
  
```

TABLE OF DEFINITIVE ERROR CODES <-----

```

KNXDVM= 10 /DRIVE 0 FAILED TO SEE HOME ON INITIALIZE. NO ERROR BIT
KNXDVI= 20 /DRIVE 1 FAILED TO SEE HOME ON INITIALIZE. NO ERROR BIT
KERTKE 40 /TRIED TO ACCESS A TRACK GREATER THAN 76.
KHOMERR= 50 /HOME WAS FOUND BEFORE DESIRED TRACK WAS REACHED.
KSELEFR= 60 /SELF DIAGNOSTIC ERROR.
KNXHDR= 70 /DESIRED SECTOR COULD NOT BE FOUND AFTER LOOKING AT 52 HEADERS.
KWPROT= 100 /WRITE FUNCTION ATTEMPTED ON A WRITE PROTECTED DISK.
KTIMERR=110 /MORE THAN 40 MICROSECONDS AND NO SEPCLOCK SEEN.
KNXPRA=120 /A PREAMBLE COULD NOT BE FOUND.
KNXIDAM=130 /PREAMBLE FOUND BUT NO ID MARK FOUND WITHIN ALLOWABLE TIME.
KNCHCFR=140 /CRC ERROR ON WHAT APPEARED TO BE A HEADER. ERROR IS NOT ASSERTED.
KIKSKLR=150 /TRACK ADDRESS OF GOOD HEADER DOES NOT COMPARE WITH DESIRED TRACK.
KXSTRYS=160 /TOO MANY TRIES FOR AN IDAM.
KNODAM= 170 /DATA AM NOT FOUND IN ALLOTTED TIME.
KDCRCFR=200 /CRC ERROR ON READING THE SECTOR FROM THE DISK.
KNANER= 220 /R/W ELECTRONICS FAILED MAINTENANCE MODE TEST.
KNCHNOV= 230 /WORD COUNT OVERFLOW.
KSTDER= 240 /WRONG KEY WORD FOR SET MEDIA DENSITY COMMAND.
  
```

C. RETRIES: EACH WRITE OP READ STATUS ERROR IS ACCOMPANIED BY A RETRY NUMBER:

PETRY 0: IS THE ORIGINAL ERROR
 PETRY 1: IS THE FIRST PETRY OF THAT ERROR (SAME ADDRESS)
 PETRY 2: IS THE SECOND RETRY OF THAT ERROR (SAME ADDRESS)
 NOW DROP THE MODULE IF SRI=1 OR
 CONTINUE TO NEXT ADDRESS IF SRI=0

10. DEVICE REGISTERS

CODE	FUNCTION
0	FILL BUFFER
1	EMPTY BUFFER
2	WRITE SECTOR
3	READ SECTOR
4	SET DENSITY ** TAKES 15 SECONDS **
5	READ MAINTENANCE STATUS
6	WRITE SECTOR WITH DELETED DATA
7	READ ERROR CODE

115	114	13	12	11	10	09	08	07	06	05	04	03	02	01	00				
RXCS:	EPR	INT	X	M	IP	X	2	IS	ID	EN	IR	IE	ID	ON	DR	V	FUN	FUN	GO
WC:	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
BA:	BASE ADDRESS																		
RXES:	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RXTA:	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RXSA:	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[0 -> 76. LEGAL]
 [1 -> 26. LEGAL]

.ENDP

```

000000* IOMODX <RXBC >,177170,264,5,0,2,202,137,RBUF,64,,66.
000000* MODULE 150000,RXBC ,177170,264,5,0,2,202,137,RBUF,64,,66.
      .TITLE RXBC DEC/X11 SYSTEM EXERCISER MODULE
      ; DDXCOM VERSION 6 23-MAY-78
      .LIST RIN
;*****
000000* BEGIN:
000000* 054122 041502 040 MODNAM: .ASCII /RXBC / ;MODULE NAME.
000005* 000 XFLAG: .BYTE OPEN ;USED TO KEEP IPACK OF WBUFF USAGE
000006* 177170 ADDR: 177170+0 ;1ST DEVICE ADDR.
000010* 000264 VECTOR: 264+0 ;1ST DEVICE VECTOR.
000012* 240 BR1: .BYTE PRTY5+0 ;1ST PR LEVEL.
000013* 000 BR2: .BYTE PRTY0+0 ;2ND PR LEVEL.
000014* 000003 DVID1: 2+1 ;DEVICE INDICATOR 1.
000016* 000000 SR1: OPEN ;SWITCH REGISTER 1
000020* 000000 SR2: OPEN ;SWITCH REGISTER 2
000022* 000000 SR3: OPEN ;SWITCH REGISTER 3
000024* 000000 SR4: OPEN ;SWITCH REGISTER 4
;*****
000026* 150000 STAT: 150000 ;STATUS WORD.
000030* 000376 INIT: START ;MODULE START ADDR.
000032* 000252 SPOINT: MODSP ;MODULE STACK POINTER.
000036* 000000 PASCNT: 0 ;PASS COUNTER.
000040* 000000 ICNT: 202 ;# OF ITERATIONS PER PASS=202
000042* 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
000044* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000046* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000050* 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000052* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000054* 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000056* 000000 RANRND: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000060* 000000 CONF1: 0 ;RESERVED FOR MONITOR USE
000062* 000000 CONF2: 0 ;RESERVED FOR MONITOR USE
000064* 000000 SVR0: OPEN ;LOC TO SAVE R0.
000066* 000000 SVR1: OPEN ;LOC TO SAVE R1.
000070* 000000 SVR2: OPEN ;LOC TO SAVE R2.
000072* 000000 SVR3: OPEN ;LOC TO SAVE R3.
000074* 000000 SVR4: OPEN ;LOC TO SAVE R4.
000076* 000000 SVR5: OPEN ;LOC TO SAVE R5.
001000* 000000 CSRA: OPEN ;ADDR OF CURRENT CSP.
001002* 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR
001004* 000000 ACSRP: OPEN ;CONTENTS OF CSR.
001006* 000000 WASADDR: OPEN ;ADDR OF BAD DATA, OR
001008* 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
001010* 000000 ERPTYP: OPEN ;TYPE OF ERROR
001012* 000000 ASB: OPEN ;EXPECTED DATA.
001014* 000000 AKAS: OPEN ;ACTUAL DATA.
001016* 000000 RSTRT: RSTRT ;RESTART ADDRESS AFTER FWD OF PASS
001018* 000666 WDM0: OPEN ;WORDS TO MEMORY PER ITERATION
001020* 000000 WDM1: OPEN ;WORDS FROM MEMORY PER ITERATION
001022* 000000 INTR1: OPEN ;# OF INTERRUPTS PER ITERATION
001024* 000137 IDNUM: 137 ;MODULE IDENTIFICATION NUMBER=137
001026* 000770 PRUFVA: RBUF ;READ BUFFER VIRTUAL ADDRESS
001028* 000000 PRUFPA: OPEN ;READ BUFFER PHYSICAL ADDRESS
  
```

```

000130* 000000 RBUFEA: OPEN ;READ BUFFER EA BITS
000132* 000100 PRUFSZ: 64 ;SIZE OF THE READ BUFFER
000134* 000000 WRUFPA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
000136* 000000 WRUFEA: OPEN ;WRITE BUFFER EA BITS
000140* 000102 WRUFQ: 66 ;WRITE BUFFER SIZE REQUESTED
000142* 000000 WRUFSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE
000144* 000000 CDWDCCT: OPEN ;COATA/DATCK ERORR COUNT
000146* 000000 CDWDCT: OPEN ;COATA/DATCK WORD COUNT
000150* 000000 FREE: OPEN ;RESERVED FOR FUTURE USE
      .REPT 8PSIZ
      .LIST
      .WORD 0
      .LIST
      .ENDR
000252*
;*****
  
```

```

322 ;---FLAGS AND COUNTERS ---
323
324 000252* 000000 DVIDIX: 0 ;HOLDS WHICH DRIVES TO TEST
325 000254* 000000 UNIT0: 0 ;UNIT 0 FLAG
326 000256* 000000 UNIT1: 0 ;UNIT 1 FLAG
327 000260* 000000 DRVN: 0 ;DRIVE NUMBER
328 000262* 000000 CMD: 0 ;COMMAND SAVE
329 000264* 000000 UIT: 0 ;UNIT UNDER TEST
330 000266* 000000 UI: 0 ;UNIT UNDER COMMAND
331 000270* 000000 WTF: 0 ;WRITE FLAG
332 000272* 000000 TA: 0 ;CURRENT TRACK ADDRESS
333 000274* 000000 SA: 0 ;CURRENT SECTOR ADDRESS
334 000276* 000000 TASAV: 0 ;STARTING TRACK ADDRESS SAVE
335 000300* 000000 SASAV: 0 ;STARTING SECTOR ADDRESS SAVE
336 000302* 000100 WDCNT: 100 ;WORD COUNTER
337 000304* 000000 DENSTY: 0 ;DENSITY
338 000306* 000000 DEN: 0 ;DENSITY FLAG
339 000310* 000000 TOMLT: 0 ;TIME OUT MULTIPLIER
340 000312* 000000 TOCNT: 0 ;TIME OUT COUNTER
341 000314* 000000 INT: 0 ;INITIALIZE FLAG
342 000316* 000000 UTDONE: 0 ;UNIT DONE FLAG
343 000320* 000000 TSSFLG: 0 ;TSS DONE WITH UNIT FLAG
344 000322* 000000 INLOOP: 0 ;IN LOOP FLAG
345 000324* 000000 ITRBYP: 0 ;RESTART INIT BYPASS FLAG
346 000326* 000000 BRKBYPS: 0 ;BREAK BYPASS FLAG
347 000330* 000000 RTRYFL: 0 ;RETRY FLAG
348 000332* 000000 NEXRCT: 0 ;NON EXISTENT MEMORY ERROR COUNTER
349 000334* 000000 SERFL: 0 ;STATUS ERROR FLAG
350 000336* 000000 FIN: 0 ;FINI FLAG
351
352
353 ;---TEMPORARY STORAGE REGISTERS ---
354
355
356 000340* 000000 SPXCS: 0
357 000342* 000000 SPXLS: 0
358 000344* 000000 SRXTA: 0
359 000346* 000000 SRXSA: 0
360 000350* 000000 SRXSH: 0
361 000352* 000000 SRXSB1: 0
362 000354* 000000 SRXSB2: 0
363 000356* 000000 SRXSB3: 0
364 000360* 000350 DECVA: 0 ;DEFINITIVE ERROR CODE VIRTUAL ADR
365 000362* 000364* +2 ;DEFINITIVE ERROR CODE PHYSICAL ADR
366 000364* 000366* +2 ;DEFINITIVE ERROR CODE EXTENDED ADR BITS
367
368 ;---CONSTANTS ---
369
370 000366* 177170 RXCS: 177170 ;RX02 COMMAND REGISTER - ADDRESS
371 000370* 177172 RXDB: 177172 ;RX02 DATA BUFFER - ADDRESS
372 000372* 000264 VEC: 264 ;VECTOR
373 000374* 000111 VARIFY: 111 ;VARIFY WD FOR SET DENSITY=ASCII "I"
374

```

```

375 ;BEGINROUTINE (MODULE 0,0 -- CONTROL)
376 ; IF START
377 ; ; INITIALIZE
378 ; ; CALL MOD 1,0 - INITIALIZE DEVICE
379 ; ; CALL MOD 6,0 - CHECK STATUS
380 ; RESTART
381 ; BEGINDO
382 ; ; IF INLOOP NOT SET
383 ; ; THEN
384 ; ; ; IF INITIALIZE_BYPASS NOT SET
385 ; ; ; THEN=INITIALIZE-TRACK & SECTOR
386 ; ; ; ENDF
387 ; ; ; IF FINI_FLAG NOT SET
388 ; ; ; THEN
389 ; ; ; ; CALL MOD 2,0 - OUTPUT (WRITE)
390 ; ; ; ; IF FINI_FLAG NOT SET
391 ; ; ; ; THEN
392 ; ; ; ; ; CALL MOD 3,0 - INPUT (READ)
393 ; ; ; ; ; IF FINI_FLAG NOT SET
394 ; ; ; ; ; THEN
395 ; ; ; ; ; ; CALL MOD 4,0 - PROCESS SECTOR
396 ; ; ; ; ; ; SET INLOOP_FLAG & CALL ITERATION
397 ; ; ; ; ; ENDF
398 ; ; ; ; ENDF
399 ; ; ; ENDF
400 ; ; ENDF
401 ; ; CLEAR INLOOP
402 ; ; SET INITIALIZE_BYPASS FLAG
403 ; ; IF SECTOR_DONE_FLAG SET
404 ; ; THEN
405 ; ; ; CLEAR SECTOR_DONE_FLAG
406 ; ; ; CALL MOD 8,0=UNITS DONE CHECK
407 ; ; ; IF UNIT_DONE SET
408 ; ; ; THEN
409 ; ; ; ; SET TRACK_ADR=TRACK_ADR+10,
410 ; ; ; ; IF TRACK_ADR > 76
411 ; ; ; ; THEN=SET TRACK_ADR=TRACK_ADR SAVE
412 ; ; ; ; INCREMENT SECTOR_ADDRESS_SAVE & CLEAR INITIALIZE_BYPASS_FLAG
413 ; ; ; ; IF SECTOR_ADDRESS_SAVE=4
414 ; ; ; ; THEN
415 ; ; ; ; ; SET SECTOR_ADDRESS_SAVE=1
416 ; ; ; ; ; INCREMENT TRACK_ADDRESS_SAVE
417 ; ; ; ; ; IF TRACK_ADDRESS_SAVE=11,
418 ; ; ; ; ; THEN
419 ; ; ; ; ; ; SET TRACK_ADDRESS_SAVE=1
420 ; ; ; ; ; ; COMMENT DENSITY_FLAG
421 ; ; ; ; ; ; CALL MOD 7,0 - SET DENSITY
422 ; ; ; ; ; ENDF
423 ; ; ; ; ENDF
424 ; ; ; ENDF
425 ; ; ENDF
426 ; ; ENDF
427 ; ; CALL MOD 5,0=SELECT UNIT FOR TEST
428 ; ; DOUNTIL FINI=1
429 ; ; FINI-->END
430 ;ENDROUTINE

```

```

431 ;MOD 0,0 CONTROL -- PROCESS SECTORS -----
432 START: NOP
433 000376 000240 MOV SPOINT,SP ;INITIALIZE STACK
434 000404 016706 177426 MOV #1,TASAV ;PRESET TRACK ADDRESS SAVE
435 000412 012767 000001 177664 MOV #1,SASAV ;PRESET SECTOR ADDRESS
436 000420 012767 000100 177650 MOV #100,WDCNT ;SET WORD COUNT FOR SINGL DENSITY
437 000426 005067 177704 CLR FIN ;CLEAR FINI FLAG
438 000432 005067 177650 CLR DEN ;CLEAR DENSITY FLAG
439 000436 005067 177642 CLR DFNSTY ;CLEAR DENSITY
440 000442 005067 177606 CLR UNIT0 ;CLEAR UNIT#0 AVAIL FLAG
441 000446 005067 177604 CLR UNIT1 ;CLEAR UNIT#1 AVAIL FLAG
442 000452 005067 177606 CLR UTT ;CLEAR UNIT UNDER TEST
443 000456 005067 177576 CLR DRVN ;CLEAR DRIVE #
444 000462 005067 177646 CLR SFRFL ;CLEAR STATUS ERR FLAG
445 000466 005067 177640 CLR NXERCT ;CLEAR NON-EXISTENT MEMORY ERROR COUNT
446 000472 016767 177316 177552 MOV DVID1,DVID1X ;GET DRIVE COUNT
447 000500 122737 000015 000041 CMPL #15,#41 ;IF RX02 FLOPPY WAS LOAD MEDIUM
448 000506 001033 BNE FIRST ;THEN
449 000510 105737 000040 TSTR #40 ;IF DRIVE 0
450 000514 001013 BNE D1 ;THEN
451 000516 042767 000001 177526 BIC #1,DVID1X ;KILL TEST FOR DRIVE 0
452 000524 032767 000001 177262 BIT #1,DVID1 ;WAS IT TO BE TESTED?
453 000532 001421 BFG FIRST ;IF NO
454 000534 012767 000260 004662 MOV #260,DNUM ;SET DRIVE # FOR PRINT
455 000542 000412 RR #AV ;IF TO NOT AVAIL MSG
456 000544 042767 000002 177500 D1: BIC #2,DVID1X ;KILL TEST FOR DRIVE 1
457 000552 032767 000002 177234 BIT #2,DVID1 ;IF WAS TO BE TESTED
458 000560 001406 BEQ FIRST ;THEN
459 000562 012767 000261 004634 MOV #261,DNUM ;SET DRIVE #1 FOR PRINT
460 000570 NUAV:
461 000570 104403 000000 005510 MSGNS,BEGIN,NUNT ;ASCII MESSAGE CALL WITH COMMON HEADER
462 000576 012767 000001 177510 FIFST: MOV #1,INT ;SET INIT FLAG
463 000604 004767 002470 JSR PC,VSET ;GO SETUP ADDRESS & VECTOR
464 000610 032777 004000 177550 BIT #4000,RPXCS ;IF DEVICE IS
465 000616 001005 BNF GO ;RX01 THEN
466 000620 104403 000000 005476 MSGNS,BEGIN,DRP4 ;ASCII MESSAGE CALL WITH COMMON HEADER
467 000626 000167 000324 JMP FINI ;GO DROP MODULE
468 000632 004767 000324 JSF PC,INITIAL ;INITIALIZE DEVICE---DO MOD 1,0
469 000636 005767 177474 TST FIN ;IF FINI FLAG
470 000642 001145 BNE FINI ;NOT SET THEN
471 000644 004767 001364 JSR PC,STATUS ;CHECK STATUS-----DO MOD 6,0
472 000650 005767 177462 TST FIN ;IF FINI FLAG
473 000654 001140 BNE FINI ;NOT SET, THEN
474 000656 005067 177432 CLR INT ;RESET INIT FLAG
475 000662 004767 001256 CALL SFLD ;GO SELECT UNIT-----DO MOD 5,0
476 000666 000240 RESTRT: NOP
477 000670 005767 177426 RGND0: TST I%LOOP ;IF INLOOP
478 000674 001042 BNE B%LOOP ;NOT SET, THEN
479 000676 005767 177422 TST ITRYP5 ;IF INITIALIZE BYPASS FLAG
480 000702 001006 BNE BTRYP5 ;NOT SET, THEN
481 000704 016767 177370 177362 MOV SASAV,SA ;PRESET SECTOR STARTING ADDRESS
482 000712 016767 177360 177352 MOV TASAV,TA ;PRESET TRACK STARTING ADDRESS
483 000720 005767 177412 EDRYFS: TST FIN ;IF FINI FLAG
484 000724 001026 BNE B%LOOP ;NOT SET, THEN
485 000726 104415 000000 000124 GFTPAS,BEGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
486 000734 104414 000000 GKBIFS,BFGIM ;GET WRITE BUFFER INFORMATION

```

```

487 000740 004767 000266 WRITE: JSR PC,OUTPUT ;GO WRITE-----DO MOD 2,0
488 000744 005767 177366 TST FIN ;TEST FINI FLAG
489 000750 001014 BNE B%LOOP ;IF SET THEN BR
490 000752 004767 000602 READ: JSR PC,INPUT ;GO READ-----DO MOD 3,0
491 000756 005767 177354 TST FIN ;TEST FINI FLAG
492 000762 001007 BNE B%LOOP ;IF SET THEN BR
493 000764 004767 001116 CALL TSS ;GO UPDATE SECTOR - DO MOD 4,0
494 000770 012767 000001 177324 MOV #1,INLOOP ;SET INLOOP FLAG
495 000776 104413 000000 ENDTSS,BEGIN ;SIGNAL END OF ITERATION,
496 001002 005067 177314 BKLOOP: CLR I%LOOP ;MONITOR SHALL TEST END OF PASS
497 001006 012767 000001 177310 MOV #1,ITRYP5 ;CLEAR INLOOP FLAG
498 001014 005767 177300 TST TSSFLG ;SET INITIALIZE BYPASS FLAG
499 001020 001453 BNE BTRYP5 ;IF UNIT TSS
500 001022 005067 177272 BFG DOUNTL ;IS DONE, THEN
501 001026 004767 002214 CLR TSSFLG ;CLEAR FLAG FOR TSS DONE
502 001032 005767 177264 CALL UDONCK ;CALL UNIT DONE CHECK - DO MOD 8,0
503 001036 001442 TST UTDONE ;IF UNIT DONE
504 001040 001042 BEQ 1$ ;NOT SET, THEN
505 001044 062767 000012 177224 ADD #12,TA ;INCREMENT TRACK ADDRESS
506 001046 022767 000114 177216 CMP #114,TA ;SEE IF DONE TRACKS
507 001054 103033 BHIS 1$ ;IF NOT: BR
508 001056 016767 177214 177206 MOV TASAV,TA ;RESET TRACK ADDRESS
509 001064 005067 177234 CLP ITRYP5 ;CLEAR INITIALIZE BYPASS FLAG
510 001070 005267 177204 INC SASAV ;BUMP STARTING SECTOR ADDRESS
511 001074 022767 000004 177176 CVP #4,SASAV ;IF SFCTORS = DONE (THREE)
512 001082 001020 BNE 1$ ;THEN
513 001084 012767 000001 177166 MOV #1,SASAV ;RESET STARTING SECTOR
514 001086 005267 177160 INC TASAV ;BUMP STARTING TRACK ADDRESS
515 001094 022767 000013 177152 CMP #11,,TASAV ;IF TRACKS=DONE (TEN)
516 001096 001007 BNE 1$ ;THEN
517 001098 012767 000001 177142 MOV #1,TASAV ;RESET TRACK STARTING ADDRESS
518 001104 005167 177146 COM DEN ;CHANGE DENSITY
519 001110 004767 001570 JSR PC,DENCH ;SET DEVICE DENSITY
520 001114 004767 000774 1$: CALL SFLD ;SELECT UNIT-----DO MOD 5,0
521 001116 005767 177162 DOUNTL: TST FIN ;DO UNTIL FINI FLAG
522 001118 001644 REQ RESTRT ;SET
523 001156 104410 000000 FINI: ENDS,BEGIN ;DROP THE MODULE
524 001156 104410 000000
525 -----

```



```

526 ;MOD 1,0 INITIALIZE DEVICE -----
527
528 001162 012767 040000 177072 INITIAL: MOV #40000,CMD ;SET INT COMMAND
529 001170 016777 177066 177170 MOV CMD,0RXCS ;INIT UNIT #
530 001176 024767 003142 JSR PC,AWDN ;GO AWAIT DONE
531 001202 005767 177130 TST FIN ;IF FINI FLAG
532 001206 001010 BNE ENDINT ;EQUALS ZERO THEN
533 001210 005777 177152 TST 0RXCS ;IF IF ERROR ON INIT
534 001214 100005 BPL ENDINT ;IF NOT RR
535 001216 104403 000000 005450 MSGNS,AFGIN,DRP1 ;ASCII MESSAGE CALL WITH COMMON HEADER
536 001224 004767 002564 JSR PC,TURP ;DO TEST DROP CLEAN-UP
537 001230 000207 ENDINT: RTS PC ;RETURN
538 ;-----
539
540 ;MOD 2,0 WRITE SUBROUTINE -----
541
542 001232 012767 000001 177030 OUTPUT: MOV #1,WTF ;SET WRITE FLAG
543 001240 004767 000134 JSR PC,OURUF2 ;THEN O/P RX02 BUFFER = DO MOD 2.1
544 001244 005767 177066 TST FIN ;IF FINI FLAG
545 001250 001050 BNE ENDOUT ;EQUALS ZERO THEN
546 001252 012767 000260 004150 MOV #260,RTYN ;RESET PTRY COUNTER-ASCII "0"
547 001260 016767 177000 176774 10: MOV UTT,CMD ;SELECT DRIVE
548 001266 052767 000105 176766 BIS #105,CMD ;SET TO WRITE SECTOR + INT ENABLE
549 001274 056767 177004 176760 BIS DENSITY,CMD ;SET DENSITY
550 001302 016777 176754 177056 MOV CMD,0PXCS ;LOAD COMMAND
551 001310 004767 003140 JSR PC,AWTR ;GO AWAIT TRANSFER READY
552 001314 005767 177016 TST FIN ;IF FINI FLAG
553 001320 001024 BNE ENDOUT ;EQUALS ZERO THEN
554 001322 016777 176746 177040 MOV SA,0RXDR ;LOAD SECTOR ADDRESS
555 001330 004767 003120 JSR PC,AWTR ;GO AWAIT TRANSFER READY
556 001334 005767 176776 TST FIN ;IF FINI FLAG
557 001340 001010 BNE ENDOUT ;EQUALS ZERO THEN
558 001342 016777 176724 177020 MOV TA,0RXDR ;LOAD TRACK ADDRESS
559 001350 004767 001764 JSR PC,INTER ;WAIT FOR INTERRUPT
560 001354 005767 176756 TST FIN ;IF FINI FLAG
561 001360 001004 BNE ENDOUT ;NOT SET, THEN
562 001362 022767 000003 176740 CMP #3,PTRYFL ;IF RETRY FLAG
563 001370 001333 BNE IS ;EQUALS 3 THEN
564 001372 005067 176672 ENDOUT: CLR WTF ;CLEAR WRITE FLAG
565 001376 000207 RTS PC ;RETURN
566 ;-----
  
```

```

567 ;MOD 2,1 OUTPUT RX02 BUFFER -----
568
569 001400 012767 000001 176654 OUBUF2: MOV #1,CMD ;SET FILL BUFFER COMMAND
570 001406 056767 176672 176646 BIS DENSITY,CMD ;SET DENSITY
571 001414 016701 176516 MOV WBUFEA,F1 ;GET EXT. ADK. BITS
572 001420 000301 SWAB R1 ;
573 001422 050167 176634 BIS R1,CMD ;SET EXT. ADK. BITS
574 001426 016777 176630 176732 MOV CMD,0RXCS ;LOAD COMMAND
575 001434 004767 003014 JSR PC,AWTR ;WAIT FOR "TR"
576 001440 005767 176672 TST FIN ;IF FINI FLAG
577 001444 001044 BNE ENDOUT ;EQUALS ZERO THEN
578 001446 016777 176630 176714 MOV WNCNT,0PXD6 ;LOAD WORD COUNT FOR OUTPUT BUFFER
579 001454 005267 176646 JMC BRBYPS ;SET BREAK BYPASS FLAG
580 001460 004767 002770 JSR PC,AWTR ;WAIT FOR "TR"
581 001464 005067 176636 CLR BRBYPS ;CLEAR BREAK BYPASS FLAG
582 001470 005767 176642 TST FIN ;IF FINI FLAG
583 001474 001030 BNE ENDOUT ;EQUALS ZERO THEN
584 001476 016777 176432 176664 MOV WBUFEA,0RXDR ;LOAD BASE ADDRESS FOR OUTPUT BUFFER
585 001504 004767 002634 JSR PC,AWDN ;WAIT FOR "DONE"
586 001510 005767 176622 TST FIN ;IF FINI FLAG
587 001514 001020 BNE ENDOUT ;EQUALS ZERO THEN
588 001516 004767 002442 CALL CNXNXP ;CALL CHECK NON-EXISTENT MEMORY ERRORS
589 001522 005777 176640 TST 0RXCS ;IF DEVICE ERROR BIT
590 001526 100013 BPL ENDOUT ;IS SET THEN
591 001530 004767 002526 CALL SUPTRG ;CALL SETUP PRINT REGISTERS
592 001534 012767 000035 176344 MOV #35,FRRTYP ;SETUP FILL BUFFER ERROR
593 ;*****
594 001542 104405 000000 000000 HRDERS,RCIN,NULL ;BUFFER FILL ERROR
595 ;*****
596 001550 104402 000000 005442 MSGNS,BEGIN,BFEP2 ;ASCII MESSAGE CALL WITH NO HEADER
597 001556 000207 ENDOUT: RTS PC ;RETURN
598 ;-----
  
```

```

599                                     ;MOD 3,0 READ SUBROUTINE -----
600
601 001560 012767 000260 003642 INPUT: MOV #260,RTYN ;RESET RETRY COUNTER=ASCII "0"
602 001566 016767 176472 176466 IS: MOV UTT,CMD ;SELECT DRIVE
603 001574 052767 000107 176460 BIS #107,CMD ;SET READ COMMAND + INT ENB
604 001602 056767 176476 176452 BIS DFNSTY,CMD ;SET DENSITY
605 001610 016777 176446 176550 MOV CMD,RRXCS ;LOAD COMMAND
606 001616 004767 002632 JSR PC,AWTR ;GO AWAIT TRANSFER READY
607 001622 005767 176510 TST FIN ;IF FINI FLAG
608 001626 001023 BFE ENDIN ;NOT SET, THEN
609 001630 016777 176440 176532 MOV SA,RRXDP ;LOAD SECTOR ADDRESS
610 001636 004767 002612 JSR PC,AWTR ;GO AWAIT TRANSFER READY
611 001642 005767 176470 TST FIN ;IF FINI FLAG
612 001646 001013 BFE ENDIN ;NOT SET, THEN
613 001650 016777 176416 176512 MOV TA,RRXDB ;LOAD TRACK ADDRESS
614 001656 004767 001456 JSR PC,INTER ;WAIT FOR INTERRUPT
615 001662 022767 000003 176440 CMP #3,RTNYFL ;IF RETRY FLAG
616 001670 001336 BFE 16 ;EQUALS 3, THEN
617 001672 004767 000002 JSR PC,INRUF2 ;THEN GET HX02 BUFFER = DO MOD 3,1
618 001676 000207 ENDIN: RTS PC ;RETURN
619
  
```

```

620                                     ;MOD 3,1 INPUT HX02 BUFFER -----
621
622 001700 005767 176430 INRUF2: TST SRRFL ;IF STATUS ERROR NOT SET
623 001706 001077 BFE ENDIN2 ;THEN
624 001710 012767 000003 176346 MOV #3,CMD ;SET EMPTY BUFFER COMMAND
625 001714 056767 176364 176340 BIS DFNSTY,CMD ;SET DENSITY
626 001722 016701 176202 MOV PDUFEA,R1 ;GET EXT. ADP. HITS
627 001726 000301 SVAR P1 ;
628 001730 050167 176326 BIS R1,CMD ;SET EXT. ADP. HITS
629 001734 016777 176322 176424 MOV CMD,RRXCS ;ELSE LOAD COMMAND
630 001742 004767 002506 JSR PC,AWTR ;WAIT FOR "TR" DO MOD U,TR
631 001746 005767 176364 TST FIN ;IF FINI FLAG
632 001752 001054 BFE ENDIN2 ;EQUALS ZERO
633 001754 016777 176372 176406 MOV WDCNT,RRXDB ;THEN LOAD WORD COUNT FOR INPUT BUFFER
634 001762 005267 176340 INC RKBYPB ;SET BREAK BYPASS FLAG
635 001766 004767 002462 JSR PC,AWTR ;WAIT FOR "TR" DO MOD U,TR
636 001772 005067 176330 CLR RKBYPB ;CLEAR BREAK BYPASS FLAG
637 001776 005767 176334 TST FIN ;IF FINI FLAG
638 002002 001040 BFE ENDIN2 ;EQUALS ZERO
639 002004 016777 176116 176356 MOV RBUFEA,RRXDB ;THEN LOAD BASE ADDRESS FOR INPUT BUFFER
640 002012 004767 002326 JSR PC,AWTR ;WAIT FOR "DONE"
641 002016 005767 176314 TST FIN ;IF FINI FLAG
642 002022 001030 BFE ENDIN2 ;EQUALS ZERO THEN
643 002024 005777 176336 TST RRXCS ;IF DEVICE ERROR BIT
644 002030 100021 BPL 16 ;SET, THEN
645 002032 004767 002126 CALL CKNXER ;CALL CHECK NON-EXISTENT MEMORY ERRORS
646 002036 005777 176324 TST RRXCS ;IF DEVICE ERROR BIT
647 002042 100020 BPL ENDIN2 ;IS STILL SET, THEN
648 002044 004767 002212 CALL SUBTRG ;CALL SETUP PRINT REGISTERS
649 002050 012767 000030 176030 MOV #30,ERRTP ;SETUP DATA TRANSFER ERROR
650
651 002056 104405 000000 000000 ;*****
652 HDRS,REGIN,NULL ;EMPTY BUFFER ERROR
653 002064 104402 000000 005434 ;*****
654 MSGSS,REGIN,BFEP1 ;ASCII MESSAGE CALL WITH NO HEADER
655 002072 000404 EP ENDIN2 ;RR TO EXIT
656 002074 104412 000000 000126 IS1 ;
657 002102 002104 000000 000126 CPDATA,BEGIN,BRUEPA ;REQUEST FOR MONITOR TO CHECK DATA
658 002104 000207 +2 ;IF ERROR, CONTINUE
659 ENDIN2: RTS PC ;RETURN
  
```

```

660 ;MOD 4,0 SECTOR UPDATE -----
661
662 TSS: ADD #1,SA ;RUMP SECTOR ADDRESS
663 CMP #33,SA ;SEE IF DONE SECTORS
664 PLOS 15 ;IF SO1 BR
665 BR ENDISS ;RETURN
666 MOV SASAV,SA ;RESET SA
667 MOV #1,TSSFLG ;SET TSS FLAG- (UNIT DONE WITH TSS)
668 ENDTSS: RTS PC
669 ;-----
670 ;MOD 5,0 SELECT UNIT FOR TEST -----
671
672
673 SELD: TST DRVN ;SEE IF DRIVE 0
674 REQ 26 ;IF SO1 BR
675 CLR DRVN ;ELSE SET TO 0
676 TST UNIT1 ;SEE IF UNIT 1 AVAILHLF
677 RFO 15 ;IF SO1 BR
678 BP SELD ;GO SEE IF UNIT IS AVAILHLF
679 MOV #20,UTT ;SET UNIT UNDER TEST
680 MOV #261,DNUM ;SET DRIVE NUMBER FOR PRINTS
681 BR FNDSEL ;RETURN
682 COM DRVN ;SWITCH UNITS
683 TST UNIT0 ;SEE IF UNIT 0 AVAILHLF
684 RFO 35 ;IF SO1 BR
685 BR SELD ;SELECT NEXT
686 CLR UTT ;SELECT DRIVE 0
687 MOV #260,DNUM ;SET DRIVE NUMBER FOR PRINTS
688 ENDTSS: RTS PC ;RETURN
689 ;-----

```

```

690 ;MOD 6,0 DEVICE STATUS -----
691
692 STATUS: NOP
693 JSR PC,UNTO ;CHECK STATUS UNIT 0 - DO MOD 6.1
694 TST UNIT0 ;IF UNIT 0 IS
695 BNE U1 ;AVAIL, THEN
696 TST INT ;IF INIT FLAG
697 RFO U1 ;IS SET, THEN
698 CLR UTT ;SET UNIT=0
699 MOV #260,DNUM ;SET DRIVE # FOR PRINT
700 JSR PC,DNSET ;SETUP DENSITY UNIT 0 - DO MOD 6.3
701 JSR PC,UNTI ;CHECK STATUS UNIT 1 - DO MOD 6.2
702 TST UNIT1 ;IF UNIT 1 IS
703 RFO 15 ;AVAIL, THEN
704 TST INT ;IF INIT FLAG
705 RFO ENDST ;IS SET, THEN
706 MOV #20,UTT ;SET UNIT=1
707 MOV #261,DNUM ;SET DRIVE # FOR PRINT
708 JSR PC,DNSET ;SETUP DENSITY UNIT 1 - DO MOD 6.3
709 BR ENDST ;RR TO END STATUS
710 TST UNIT0 ;IF UNIT 0
711 BEQ ENDST ;NOT AVAIL, THEN
712 MSGS,REGIN,DRP2 ;ASCII MESSAGE CALL WITH COMMON HEADER
713 MOV #1,FIN ;GO DROP MODULE
714 ENDTSS: RTS PC ;RETURN
715 ;-----
716 ;MOD 6.1 UNIT 0 STATUS -----
717
718
719 UNTO: BIT #1,DVIDIX ;IF UNIT 0
720 RFO 15 ;SELECTED, THEN
721 MOV #13,CMD ;READ STATUS UNIT 0
722 BIS DENSITY,CMD ;SET DENSITY
723 MOV CMD,ORXCS ;SEND COMMAND
724 JSR PC,AWDN ;GO AWAIT DONE
725 TST FIN ;IF FINI FLAG
726 BNE ERDIT0 ;EQUALS ZERO THEN
727 MOV #200,R0 ;GET ROPS
728 BIT #200,R0 ;IF DRIVE R0Y
729 BNE ERDIT0 ;NOT SET, THEN
730 MOV #260,DNUM ;SET DRIVE # FOR PRINT
731 RFO ERPTYP
732 *****
733 HDRS,REGIN,NULL ;UNIT 0 NOT AVAILABLE
734 *****
735 MSGS,REGIN,NUNT ; ASCII MESSAGE CALL WITH NO HEADER
736 MOV #1,UNIT0 ;DESELECT UNIT 0
737 ENDTSS: RTS PC ;RETURN
738 ;-----

```

```

740 ;MOD 6.2 UNIT 1 STATUS -----
741
742 002500 032767 000002 175544 UNIT1: BIT #2,DVIDIX ;IF UNIT 1
743 002505 001437 REQ 16 ;SELECTED, THEN
744 002510 012767 000033 175544 MOV #33,CMD ;READ STATUS UNIT 1
745 002516 056767 175562 175536 RIS DENSITY,CMD ;SET DENSITY
746 002524 016777 175532 175634 MOV CMD,ORXCS ;SEND COMMAND
747 002532 004767 001606 JSH PC,AWDN ;GO AWAIT DONE
748 002536 005767 175574 TST FIN ;IF FINI FLAG
749 002542 001024 BNE ENDUT1 ;EQUALS ZERO THEN
750 002544 017700 175620 MOV ORXDB,R0 ;READ RXES
751 002550 032700 000200 RIT #200,R0 ;IF DRIVE RDY
752 002554 001017 BNE ENDUT1 ;NOT SET, THEN
753 002556 012767 000261 002640 MOV #261,DNUM ;SET DRIVE # FOR PRINT
754 002564 012767 000006 175314 MOV #6,ERTYX
755 ;*****
756 002572 104405 000000 000000 HDRPG,BEGIN,NULL ;UNIT 1 NOT AVAILABLE
757 ;*****
758 002600 104402 000000 005510 MSGS,BEGIN,NUNT ;ASCII MESSAGE CALL WITH NO HEADER
759 002606 012767 177777 175442 18: MOV #-1,UNIT1 ;DESELECT UNIT 1
760 002614 000207 ENDUT1: RTS PC ;RETURN
761 ;-----
762
763 ;MOD 6.3 DEVICE DENSITY SETUP -----
764
765
766 002616 012767 000013 175436 DNSET: MOV #13,CMD ;SET READ STATUS
767 002624 056767 175434 175430 RIS UT,CMD ;SET UNIT
768 002632 016777 175424 175526 MOV CMD,ORXCS ;SEND COMMAND
769 002640 004767 001500 JSR PC,AWDN ;GO AWAIT DONE
770 002644 005767 175466 TST FIN ;IF FINI FLAG
771 002650 001030 BNE XDNSET ;IS ZERO, THEN
772 002652 012767 000011 175402 MOV #11,CMD ;SET SINGLE DENSITY
773 002660 056767 175400 175374 RIS UT,CMD ;SET UNIT
774 002666 016777 175370 175472 MOV CMD,ORXCS ;SEND COMMAND
775 002674 004767 001554 JSR PC,AWTR ;GO AWAIT "TR"
776 002700 005767 175432 TST FIN ;IF FINI FLAG IS
777 002704 001012 BNE XDNSET ;ZERO
778 002706 016777 175462 175454 MOV VARYFY,ORXDE ;SEND VARYFY WORD
779 002714 104403 000000 005716 MSGS,BEGIN,SETSDN ;ASCII MESSAGE CALL WITH COMMON HEADER
780 002722 004767 001416 JSR PC,AWDN ;WAIT FOR "DONE"
781 002726 004767 000440 JSR PC,SECK ;GO CHECK FOR EPROR
782 002732 000207 XDNSET: RTS PC ;RETURN
783 ;-----

```

```

784 ;MOD 7.0 DENSITY CHANGE -----
785
786 002734 000240 DENCH: NOP
787 002736 005767 175314 TST UNIT1 ;IF UNIT 1
788 002742 001007 RME 16 ;IS AVAIL
789 002744 012767 000020 175314 MOV #20,UT ;SET TO DO UNIT 1
790 002752 012767 000261 002444 MOV #261,DNUM ;SET DRIVE # FOR PRINT
791 002760 000405 BR 25 ;CONTINUE
792 002762 005067 175300 18: CLR UT ;ELSE SET TO DO UNIT 0
793 002766 012767 000260 002430 MOV #260,DNUM ;SET DRIVE # FOR PRINT
794 002774 005767 175306 28: TST DFN ;IF DENSITY FLAG
795 003000 001051 BNE 55 ;EQUALS ZERO (SINGLE DENSITY) THEN
796 003002 012767 000111 175252 38: MOV #111,CMD ;SETUP SET DENSITY = SINGLE COMMAND
797 003010 056767 175252 175244 RIS UT,CMD ;SETUP UNIT UNDER TEST
798 003016 016777 175240 175342 MOV CMD,ORXCS ;SEND COMMAND
799 003024 004767 001424 JSR PC,AWTR ;WAIT FOR "TR"
800 003030 005767 175302 TST FIN ;IF FINI FLAG
801 003034 001103 BNE ENDDNC ;EQUALS ZERO THEN
802 003036 016777 175332 175324 MOV VARYFY,ORXDR ;SEND VARYFY COMMAND
803 003044 104403 000000 005734 MSGS,BEGIN,SSGLDN ;ASCII MESSAGE CALL WITH COMMON HEADER
804 003052 004767 000262 JSR PC,INTER ;WAIT FOR "DONE" & INTERRUPT
805 003056 005767 175254 TST FIN ;IF FINI FLAG
806 003062 001070 BNE ENDDNC ;EQUALS ZERO THEN SET
807 003064 005767 175176 TST UT ;IF
808 003070 001411 BRQ 45 ;UNIT 1 WAS DONE
809 003072 005067 175170 CLR UT ;SET TO DO UNIT 0
810 003076 012767 000260 002320 MOV #260,DNUM ;SET DRIVE # FOR PRINT
811 003104 005767 175144 TST UNIT0 ;IF UNIT 0
812 003110 001001 BNE 45 ;IS AVAIL
813 003112 000733 BR 38 ;SET UNIT 0 DENSITY
814 003114 012767 000000 175162 45: MOV #0,DENSITY ;SET DENSITY=SINGLE
815 003122 000450 BR ENDDNC ;BRANCH TO END
816 003124 012767 000511 175130 55: MOV #511,CMD ;ELSE SET DENSITY=DOUBLE COMMAND
817 003132 056767 175130 175122 RIS UT,CMD ;SETUP UNIT UNDER TEST
818 003140 016777 175116 175220 MOV CMD,ORXCS ;SEND COMMAND
819 003146 004767 001302 JSR PC,AWTR ;WAIT FOR "TR"
820 003152 005767 175160 TST FIN ;IF FINI FLAG
821 003156 001032 BNE ENDDNC ;EQUALS ZERO THEN
822 003160 016777 175210 175207 MOV VARYFY,ORXDR ;SEND VARYFY COMMAND
823 003166 104403 000000 005750 MSGS,BEGIN,SDBLDN ;ASCII MESSAGE CALL WITH COMMON HEADER
824 003174 004767 000140 JSR PC,INTER ;WAIT FOR "DONE" & INTERRUPT
825 003200 005767 175132 TST FIN ;IF FINI FLAG
826 003204 001017 BNE ENDDNC ;EQUALS ZERO THEN
827 003206 005767 175054 TST UT ;IF
828 003212 001411 BRQ 68 ;UNIT 1 WAS DONE
829 003214 005067 175046 CLR UT ;SET TO DO UNIT 0
830 003220 012767 000260 002176 MOV #260,DNUM ;SET DRIVE # FOR PRINT
831 003226 005767 175022 TST UNIT0 ;IF UNIT 0
832 003232 001001 BNE 68 ;AVAIL
833 003234 000733 BR 55 ;CONTINUE
834 003236 012767 000400 175040 68: MOV #400,DENSITY ;SET DENSITY = DOUBLE
835 003244 000207 ENDDNC: RTS PC ;RETURN
836 ;-----

```

```

837 ;MOD 8,0 - UNIT DONE CHECK -----
838
839 003246* 005767 175012 UNDNCK: TST UTT ;IF UNIT 0
840 003252* 001006 BNE 18 ;UNDER TEST, THEN
841 003254* 005767 174776 TST UNIT1 ;IF UNIT 1
842 003260* 001003 BNE 18 ;AVAILABLE, THEN
843 003262* 005067 175031 CLR UTDONE ;CLEAR UNITS DONE FLAG
844 003266* 000403 BR XDNCK ;RR TO MOD EXIT
845 003270* 012767 000001 175020 18: MOV #1,UTDONE ;SET UNITS DONE FLAG
846 003276* 000207 XDNCK: RETURN ;RETURN
847
848 ;MOD U,VS VECTOR SET UP -----
849
850
851 003300* 016700 174504 VSET: MOV VVECTOR,R0
852 003304* 012720 003344* MOV #INTRUPT,(R0)+ ;SET UP INTERRUPT HANDLER ADDRESS
853 003310* 116710 174476 MOV R1,(R0) ;SET RR LEVEL
854 003314* 016767 174466 175044 MOV ADDR,RXCS ;SET ADDRESS OF RXCS
855 003322* 016767 174466 175040 MOV ADDR,RXDR ;SET ADDRESS OF RXCS
856 003330* 062767 000002 175032 ADD #2,RXDR ;SET ADDRESS OF DATA BUFFER
857 003336* 000207 RTS PC
858
859 ;MOD U,INTR INTERRUPT HANDLER -----
860
861
862 003340* INTER:
863 003340* 104400 000000* EXITS,REGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
864 003344* NTRUPT:
865 ;-----
866 003344* 000004 000000* 003352* FIPOS,REGIN,18 ; QUEUE UP TO CONTINUE AT 16 AND RTI
867 ;-----
868 003352* 004767 000014 18: JSR PC,SECK ;GO CHECK FOP ERROR
869 003356* 016700 174426 MOV VVECTOR,R0 ;GET VECTOR ADDRESS
870 003362* 005720 TST (R0)+ ;RUMP POINTER
871 003364* 016710 174422 MOV R1,(R0) ;ASSURE RESET TO BR LEVEL 5
872 003370* 000207 RTS PC ;RETURN
873 ;-----
  
```

```

874 ;MOD U,EC STATUS ERROR CHECK SUBROUTINE -----
875
876 003372* 005067 174736 SECK: CLR SFPFL ;CLEAR STATUS ERROR FLAG
877 003376* 057767 174764 174656 RIS #RXCS,CMD ;GET RXCS
878 003404* 005767 174652 TST CMD ;SEE IF ERROR
879 003410* 100404 BMI 18 ;IF SO: BR
880 003412* 012767 000003 174710 MOV #1,PTRYFL ;SET RETRY FLAG, SO NO RETRY
881 003420* 000423 BR EIDEC ;ELSE RETURN
882 003422* 004767 000634 18: CALL SUPTRG ;CALL SETUP PRINT REGISTERS
883 003426* 016767 174632 174626 MOV UTT,CMD ;SELECT DRIVE
884 003434* 004767 000422 CALL DVEC ;GET DEVICE ERROR CODE
885 003440* 004767 000026 CALL CKDNER ;CHECK FOR DENSITY ERROR
886 003444* 005767 174620 TST WTF ;SEE IF WRITE ERROR
887 003450* 001003 BNE 28 ;IF SO: BR
888 003452* 004767 000066 CALL CKRDER ;CHECK READ ERROR
889 003456* 000402 BR CL ;GO CLEAR ERROR
890 003460* 004767 000122 28: CALL CKWTEP ;CALL CHECK WRITE ERROR
891 003464* 004767 000160 CLI JSR PC,CLFAR ;CLEAR ERRORS
892 003470* 000207 ENDEC: RTS PC ;RETURN
893
894 ;MOD U,CKDNER - CHECK DENSITY ERRORS -----
895
896
897 003472* 016705 174642 CKDNER: MOV SRXCS,R5 ;SETUP RXCS FOR TEST
898 003476* 042705 177761 R1C #177761,R5 ;CLEAR ALL HUT COMMAND
899 003502* 022705 000010 CMP #10,R5 ;IF COMMAND WAS
900 003506* 001015 BNE XDNCK ;SET DENSITY, THEN
901 003510* 012767 000001 174620 MOV #1,FINI ;SET FINI FLAG
902 003516* 005767 174564 TST DEN ;IF
903 003522* 001004 BNE 18 ;SINGLE DENSITY, THEN
904 003524* 104403 000000* 005520* MSGS6,REGIN,SETSPR ;ASCII MESSAGE CALL WITH COMMON HEADER
905 003532* 000403 BR XDNCK ;CONTINUE
906
907 003534* 104403 000000* 005536* 18: MSGS6,REGIN,SETDUE ;ASCII MESSAGE CALL WITH COMMON HEADER
908 003542* 000207 XDNCK: RETURN ;RETURN
909
910 ;MOD U,CKRDER - CHECK READ ERRORS -----
911
912
913 003544* 012767 000000 174334 CKRDER: MOV #0,EPRTYP
914 ;*****
915 003552* 104406 000000* 004626* SOFERS,REGIN,TABLE ;READ ERROR
916 ;*****
917 003560* 005767 174522 TST DEN ;IF DENSITY FLAG IS
918 003564* 001004 BNE 18 ;EQUAL TO ZERO (SINGLE) THEN
919 003566* 104402 000000* 005674* MSGS6,REGIN,RTERRSD ; ASCII MESSAGE CALL WITH NO HEADER
920 003574* 000403 BR 28 ;GO PRINT ERRORS
921
922 003576* 104402 000000* 005652* 18: MSGS6,REGIN,RTERRDD ; ASCII MESSAGE CALL WITH NO HEADER
923 003604* 000207 RETURN ;RETURN
924 ;-----
  
```

```
925 ;MOD U,CKWTER - CHECK WRITE ERRORS -----
926
927 003606* 012767 000000 174272 CKWTER: MOV #0,ERRTP
928 ;*****
929 003614* 104406 000000* 004626* SOFFPS,BEGIN,TABLE ;WRITE ERROR
930 ;*****
931 003622* 005767 174460 TST DEFN ;IF DENSITY FLAG IS
932 003626* 001004 BNE 15 ;EQUAL TO ZERO (SINGLE) THEN
933 003630* 104402 000000* 005630* MSGS0,BEGIN,WTERSD ; ASCII MESSAGE CALL WITH NO HEADER
934 003636* 000403 BR 28 ;GO PRINT ERRORS
935 003640*
936 003640* 104402 000000* 005606* 16: MSGS0,BEGIN,WTEPDD ; ASCII MESSAGE CALL WITH NO HEADER
937 003646* 000207 28: RETURN ;RETURN
938 ;-----
939
940 ;MOD U,CLF CLFAP ERRORS -----
941
942 003650* 005267 174460 CLEAR: INC SFRFL ;SET STATUS ERROR FLAG
943 003654* 105767 174470 TSTB SRXSB ;SEE IF R.E.C. ERROP
944 003660* 001421 BEQ 26 ;IF NOT: BR
945 003662* 012767 040000 174372 MOV #40000,CMD ;SET COMMAND= INIT
946 003670* 016777 174366 174470 MOV CMD,0RXCS ;REINITIALIZE
947 003676* 004767 000442 16: JSR PC,AWDN ;GO AWAIT DONE
948 003702* 005777 174460 TST 0RXCS ;SET IF ERROR ON INIT
949 003706* 100006 RPL 28 ;IF NOT: BR
950 003710* 104403 000000* 005466* MSGN0,BEGIN,DRP3 ;ASCII MESSAGE CALL WITH COMMON HEADER
951 003716* 004767 000072 BR PC,DRP ;GO DROP MODULE
952 003722* 000431 BR ENDCLR
953 003724* 005267 001500 26: INC RTYN ;RUMP RETRY COUNTER
954 003730* 016703 001474 MOV RTYN,R3 ;GET COUNTER
955 003734* 042703 177700 PIC #17770,R3 ;MARK ASCII
956 003740* 010367 174364 MOV R1,RTRYFL ;SET RETRY FLAG
957 003744* 022703 000003 CMP #3,R3 ;SEE IF DONE RETRIES
958 003750* 001016 BNE ENDCLR ;IF NOT: BR
959 ;*****
960 003752* 104405 000000* 004626* HRDPS,BEGIN,TABLE ;RECLASSIFICATION OF SOFT ERK
961 ;*****
962 003760* 032767 000001 174030 BIT #1,SR1 ;SEE IF SHOULD DROP MODULE
963 003766* 001404 BEQ 36 ;IF NOT: BR
964 003770* 012767 000001 174340 MOV #1,FIN ;GO DROP MODULE
965 003776* 000403 BR ENDCLR
966 004000* 012767 000260 001422 36: MOV #260,RTYN ;RFSET RETRY COUNTER
967 004006* 005067 174336 ENDCLR: CLF SRXSB ;CLFAP ERROR
968 004012* 000207 RTS PC ;RETURN
969 ;-----
```

```
970 ;MOD U,DRP TEST DROP CLEAN-UP -----
971
972 004014* 004767 000242 DRP: CALL SUPTRG ;CALL SETUP PRINT REGISTERS
973 004020* 005067 174236 CLR CMD ;CLEAR COMMAND WORD
974 004024* 004767 000032 CALL DVEC ;GET DEVICE EPROR CODE
975 004030* 005767 174302 TST FIN ;IF FINI FLAG
976 004034* 001011 BNE ENDTDR ;EQUALS ZERO THEN
977 004036* 012767 000034 174042 28: MOV #34,ERRTP
978 ;*****
979 004044* 104405 000000* 004626* HRDPS,BEGIN,TABLE ;INITIALIZE ERROP
980 ;*****
981 004052* 012767 000001 174256 MOV #1,FIN ;GO DROP MODULE
982 004060* 000207 ENDTDR: RTS PC
983 ;-----
984
985 ;MOD U,DVEC ----- GET DEVICE ERROP CODE -----
986
987 004062* 000240 DVEC: NOP ;
988 004064* 052767 000017 174170 RIS #17,CMD ;OR ERROR CODE COMMAND
989 004072* 104415 000000* 000360* GETPAS,BEGIN, DECVA ;GET PHYSICAL ADDRESS FROM 16-BIT DECVA
990 004100* 016701 174256 MOV DECVA+2,R1 ;GET DEF E.C. PHYS ADP
991 004104* 016702 174254 MOV DECVA+4,R2 ;GET DEF E.C. XTEND ADP BITS
992 004110* 000302 SWAR R2 ;
993 004112* 050267 174144 BIS R2,CMD ;SET EXTEND ADP BITS
994 004116* 016777 174140 174242 MOV CMD,0RXCS ;SEND COMMAND
995 004124* 005267 174176 INC BKRYPS ;SET BREAK BYPASS FLAG
996 004130* 004767 000320 JSR PC,AWTR ;THEN GO AWAIT TR
997 004134* 005067 174166 CLR BKRYPS ;CLEAR BREAK BYPASS FLAG
998 004140* 005767 174172 TST FIN ;IF FINI FLAG
999 004144* 001006 BNE EDDVEC ;EQUALS ZERO THEN
1000 004146* 010177 174216 MOV R1,0RXDB ;SET BASE ADDR FOR LOAD EXT ERR REG.
1001 004152* 004767 000166 JSR PC,AWDN ;GO AWAIT DONE
1002 004156* 004767 000002 CALL CKXER ;CALL CHECK NON-EXIST MEMOPY ERROP
1003 004162* 000207 EDDVEC: RTS PC ;RETURN
1004 ;-----
```

```
1005  
1006 ;MOD U,CKNXER - CHECK NON-EXISTENT MEMORY ERRORS -----  
1007  
1008 004164* 032777 004000 174176 CKNXER: BIT #4000,0PXDB ;IF NONEXISTENT MEMORY  
1009 004172* 001432 BEQ XXNFR ;ERROR, THEN  
1010 004174* 004767 000062 CALL SUPTRG ;CALL SETUP PRINT REGISTERS  
1011 004200* 012777 000000 174160 MOV #4000,0RXCS ;INITIALIZE DEVICE (TO CLEAR ERROR)  
1012 004206* 012767 000010 173672 MCV #10,ERRTYP ;SET NON-EXISTENT MEMORY ERROR  
1013 ;*****  
1014 004214* 104405 000000* 000000 HRDRS,BEGIN,NULL ;NON-EXISTENT MEMORY ERROR  
1015 ;*****  
1016 004222* 104402 005764* 000000* MSGS$,NXERMS,BEGIN  
1017 004230* 005267 174076 INC NXERCT ;INCREMENT NONEXISTENT ERROR COUNT MEMORY  
1018 004234* 022767 000004 174070 CMP #4,NXERCT ;IF NON-EXISTENT MEMORY  
1019 004242* 101006 BHI XXNFR ;ERROR COUNT=3, THEN  
1020 004244* 012767 000001 174064 1$; MOV #1,FIN ;SET FINI FLAG  
1021 004252* 104402 000000* 005504* MSGS$,BEGIN,DRP5 ;ASCII MESSAGE CALL WITH NO HEADER  
1022 004260* 000207 XNFR: RETURN ;RETURN  
1023 ;-----  
1024 ;MOD U,SURG - SETUP PRINT REGISTERS -----  
1025  
1026  
1027 004262* 017767 174100 174050 SUPTRG: MOV 0RXCS,SRXCS ;GET RXCS  
1028 004270* 056767 173766 174042 1$; CLR TOCNT ;SET SRXCS TO SHOW COMMAND  
1029 004276* 017767 174066 174036 MOV 0RXDB,SRXES ;GET PXES  
1030 004304* 016767 173764 174034 MOV SA,SRXSA ;GET SECTOR ADDRESS  
1031 004312* 016767 173754 174024 MOV TA,SRXTA ;SET TRACK ADDRESS  
1032 004320* 016767 174014 173554 MOV SRXCS,ACSR ;SET CONTENTS OF RXCS  
1033 004326* 016767 174034 173544 MOV RXCS,CSRA ;SET ADDRESS OF RXCS  
1034 004334* 016767 174002 173542 MOV SRXES,ASTAT ;GET PXES  
1035 004342* 000207 RETURN ;RETURN  
1036 ;-----
```

```
1037 ;MOD U,WDN AWAIT DONE BIT SUBROUTINE -----  
1038  
1039 004344* 005067 173740 AWDN: CLR TOMLT ;PRESET TIME OUT MULTIPLIER  
1040 004350* 005067 173736 1$; CLR TOCNT ;PRESET TIME OUT COUNTER  
1041 004354* 032777 000040 174004 2$; BIT #40,0RXCS ;SEE IF DONE SET  
1042 004362* 001033 BNE 3$ ;IF 501 BR  
1043 004364* 104407 000000* BPEAKS,BEGIN ;TEMPORARY RETURN TO MONITOR, ...  
1044 004370* 104407 000000* BPEAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
1045 004374* 005267 173712 INC TOCNT ;BUMP TIME OUT COUNTER  
1046 004400* 001365 BNE 2$ ;IF NOT TIMED OUT: BR  
1047 004402* 005267 173702 INC TOMLT ;INCREMENT TIMEOUT MULTIPLIER  
1048 004406* 022767 000002 173674 CMP #2,TOMLT ;IF ON 2ND  
1049 004414* 101355 BHI 1$ ;TIMEOUT PASS, THEN  
1050 004416* 004767 177640 CALL SUPTRG ;CALL SETUP PRINT REGISTERS  
1051 004422* 012767 000011 173456 MOV #11,ERRTYP  
1052 ;*****  
1053 004430* 104405 000000* 000000 HRDRS,BEGIN,NULL ;DONE BIT TIME OUT  
1054 ;*****  
1055 004436* 104402 000000* 005554* MSGS$,BEGIN,DRP5 ;ASCII MESSAGE CALL WITH NO HEADER  
1056 004444* 012767 000001 173664 MOV #1,FIN ;PROP MODULE  
1057 004452* 000207 3$; RTS PC ;EXIT  
1058 ;-----  
1059 ;MOD U,WTR AWAIT TRANSFER READY SUBROUTINE -----  
1060  
1061  
1062 004454* 005067 173630 AWR: CLR TOMLT ;PRESET TIMEOUT MULTIPLIER  
1063 004460* 005067 173626 1$; CLR TOCNT ;PRESET TIME OUT COUNTER  
1064 004464* 032777 000040 173674 2$; BIT #40,0RXCS ;IF DONE BIT NOT  
1065 004472* 001026 BNE 4$ ;SET, THEN  
1066 004474* 032777 000200 173664 EIT #200,0RXCS ;SEE IF TRANSFER READY SET  
1067 004502* 001046 BNE 5$ ;IF 501 BR  
1068 004504* 005767 173616 TST BKBYPS ;IF BREAK BYPASS FLAG  
1069 004510* 001004 BNE 3$ ;NOT SET, THEN  
1070 004512* 104407 000000* BPEAKS,BEGIN ;TEMPORARY RETURN TO MONITOR, ...  
1071 004516* 104407 000000* BPEAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
1072 004522* 005267 173564 3$; INC TOCNT ;BUMP TIME OUT COUNTER  
1073 004526* 001356 BNE 2$ ;IF NOT TIMED OUT: BR  
1074 004530* 005267 173554 INC TOMLT ;INCREMENT TIMEOUT MULTIPLIER  
1075 004534* 022767 000002 173546 CMP #2,TOMLT ;IF ON 2ND  
1076 004542* 101346 BHI 1$ ;TIMEOUT PASS, THEN  
1077 004544* 004767 177512 CALL SUPTRG ;CALL SETUP PRINT REGISTERS  
1078 004550* 012767 000040 173330 4$; MOV #40,ERRTYP  
1079 ;*****  
1080 004556* 104405 000000* 000000 HRDRS,BEGIN,NULL ;TRANSFER READY TIME OUT  
1081 ;*****  
1082 004564* 012767 000001 173544 MOV #1,FIN ;GO DROP MODULE  
1083 004572* 032777 000040 173566 BIT #40,0RXCS ;IF DONE BIT  
1084 004600* 001404 BEQ 5$ ;SET, THEN  
1085 004602* 104402 000000* 005574* MSGS$,BEGIN,TRDNER ;ASCII MESSAGE CALL WITH NO HEADER  
1086 004610* 000403 RR 6$ ;EXIT  
1087 004612* 5$;  
1088 004612* 104402 000000* 005564* MSGS$,BEGIN,TRTO ;ASCII MESSAGE CALL WITH NO HEADER  
1089 004620* 005067 173502 6$; CLR BKHYPS ;CLEAR BREAK BYPASS FLAG  
1090 004624* 000207 RTS PC ;RETURN  
1091 ;-----
```

```
1092 ;--- MESSAGE TABLE -----
1093
1094 004626* TABLE:
1095 004626* 000340* ARXCS: SRXCS
1096 004630* 000342* ARXES: SPXFS
1097 004632* 000344* ARXTA: SRXTA
1098 004634* 000346* ARXSA: SPXSA
1099 004636* 000350* ARXSB: SRXSB
1100 004640* 000352* ARXSB1: SPXSB1
1101 004642* 000354* ARXSB2: SRXSB2
1102 004644* 000356* ARXSB3: SPXSB3
1103 004646* 177777 -1
1104 ;-----
1105 ;--- MESSAGES -----
1106
1107
1108 004650* 054122 030460 051440 MSG1: ,ASCIZ 'RX01 SUBSYSTEM'
1109 004656* 041125 054523 052123
1110 004664* 046505 000
1111 004667* 040 044524 042515 MSG2: ,ASCIZ ' TIME OUT'
1112 004674* 047440 052125 000
1113 004701* 040 051104 053111 MSG3: ,ASCIZ ' DRIVE '
1114 004706* 020105 000
1115 004711* 040 037055 051104 MSG4: ,ASCIZ ' ->DROP MODULES'
1116 004716* 050117 046440 042117
1117 004724* 046125 022505 000
1118 004731* 040 047111 052111 MSG5: ,ASCIZ ' INITIALIZE'
1119 004736* 040511 044514 042532
1120 004744* 000
1121 004745* 040 047516 052440 MSG6: ,ASCIZ ' NO UNITS TO TEST'
1122 004752* 044516 051524 052040
1123 004760* 020117 042524 052123
1124 004766* 000
1125 004767* 045 051040 044505 MSG7: ,ASCIZ '% REINITIALIZE'
1126 004774* 044516 044524 046101
1127 005002* 055111 000105
1128 005006* 051440 052105 026440 MSG8: ,ASCIZ ' SET -> WAITING ON'
1129 005014* 020076 040527 052111
1130 005022* 047111 020107 047117
1131 005030* 000
1132 005031* 040 044506 046114 MSG9: ,ASCIZ ' FILL BUFFER'
1133 005036* 041040 043125 042506
1134 005044* 000122
1135 005046* 051440 047111 046107 MSG11: ,ASCIZ ' SINGLE'
1136 005054* 000105
1137 005056* 042040 052517 046102 MSG12: ,ASCIZ ' DOUBLE'
1138 005064* 000105
1139 005066* 042040 047117 020105 MSG13: ,ASCIZ ' DONE BIT'
1140 005074* 044502 000124
1141 005100* 052040 040522 051516 MSG14: ,ASCIZ ' TRANSFER READY'
1142 005106* 042506 020122 042522
1143 005114* 042101 000131
1144 005120* 042040 047105 044523 MSG15: ,ASCIZ ' DENSITY'
1145 005126* 054524 000
1146 005131* 040 051105 047522 MSG17: ,ASCIZ ' ERROR- RETRY;'
1147 005136* 026522 051040 052105
```

```
1148 005144* 054522 000072
1149 005150* 020045 042523 000124 MSG18: ,ASCIZ '% SFT'
1150 005156* 020045 051127 052111 MSG19: ,ASCIZ '% WRITE'
1151 005164* 000105
1152 005166* 020045 042522 042101 MSG20: ,ASCIZ '% READ'
1153 005174* 000
1154 005175* 040 046505 052120 MSG21: ,ASCIZ ' EMPTY BUFFER'
1155 005202* 020131 052502 043106
1156 005210* 051105 000
1157 005213* 040 047516 020124 MSG22: ,ASCIZ ' NOT AVIALABLE'
1158 005220* 053101 040511 040514
1159 005226* 046102 000105
1160 005232* 042440 051122 051117 MSG23: ,ASCIZ ' ERROR'
1161 005240* 000
1162 005241* 045 026440 051476 MSG24: ,ASCIZ '% ->SETTING DISKETTE TO'
1163 005246* 052105 044524 043516
1164 005254* 042040 051511 042513
1165 005262* 052174 020105 047524
1166 005270* 000
1167 005271* 045 020040 044440 MSG25: ,ASCIZ '% IF INTERRUPTED-> MAY NEED REFORMAT%'
1168 005276* 020106 047111 042524
1169 005304* 051122 050125 042524
1170 005312* 026504 020076 040515
1171 005320* 020131 042516 042105
1172 005326* 051040 043105 051117
1173 005334* 040515 022524 000
1174 005341* 045 000
1175 005343* 055 020076 047516 MSG26: ,ASCIZ '% '
1176 005350* 042516 044530 052123 MSG27: ,ASCIZ '%-> NONEXISTENT MEMORY ERROR = INITIALIZE DEVICE'
1177 005356* 047105 020124 042515
1178 005364* 047515 054522 042440
1179 005372* 051122 051117 026440
1180 005400* 044440 044516 044524
1181 005406* 046101 055111 020105
1182 005414* 042504 044526 042503
1183 005422* 000
1184 005424* 030040 000 DNUM: ,EVFN
1185 005430* 030040 000 ,ASCIZ ' 0'
1186 005430* 030040 000 ,EVEN
1187 005430* 030040 000 RTYN: ,ASCIZ ' 0'
1188 005434* ,EVEN
1189 ;-----
```



```
1190 ;--- FORMATTED MESSAGES -----  
1191 005434 005175 RFR1: MSG21 ;"EMPTY BUFFER"  
1192 005436 005232 MSG23 ;"ERROR"  
1193 005440 177777 -1  
1194 005442 005031 RFR2: MSG9 ;"FILL BUFFER"  
1195 005444 005232 MSG23 ;"ERROR"  
1196 005446 177777 -1  
1197 005450 004731 DPP1: MSG5 ;"INITIALIZE"  
1198 005452 005232 MSG23 ;"ERROR"  
1199 005454 004711 MSG4 ;"DROP MODULE"  
1200 005456 177777 -1  
1201 005460 004745 DRP2: MSG6 ;"NO UNITS TO TEST"  
1202 005462 004711 MSG4 ;"DROP MODULE"  
1203 005464 177777 -1  
1204 005466 004767 DRP3: MSG7 ;"REINITIALIZE"  
1205 005470 005232 MSG23 ;"ERROR"  
1206 005472 004711 MSG4 ;"DROP MODULE"  
1207 005474 177777 -1  
1208 005476 004650 DRP4: MSG1 ;"RX01 SUBSYSTEM"  
1209 005500 004711 MSG4 ;"DROP MODULE"  
1210 005502 177777 -1  
1211 005504 004711 DRP5: MSG4 ;"DROP MODULE"  
1212 005506 177777 -1  
1213 005510 004701 MUNT: MSG3 ;"DRIVE"  
1214 005512 005424 DNUM ;"#"  
1215 005514 005213 MSG22 ;"NOT AVAILABLE"  
1216 005516 177777 -1  
1217 005520 004701 SETSDE: MSG3 ;"DRIVE"  
1218 005522 005424 DNUM ;"#"  
1219 005524 005150 MSG18 ;"SET"  
1220 005526 005046 MSG11 ;"SINGLE"  
1221 005530 005120 MSG15 ;"DENSITY"  
1222 005532 005232 MSG23 ;"ERROR"  
1223 005534 177777 -1  
1224 005536 004701 SETDDE: MSG3 ;"DRIVE"  
1225 005540 005424 DNUM ;"#"  
1226 005542 005150 MSG18 ;"SET"  
1227 005544 005056 MSG12 ;"DOUBLE"  
1228 005546 005120 MSG15 ;"DENSITY"  
1229 005550 005232 MSG23 ;"ERROR"  
1230 005552 177777 -1  
1231 005554 005066 DNT0: MSG13 ;"DONE BIT"  
1232 005556 004667 MSG2 ;"TIME OUT"  
1233 005560 004711 MSG4 ;"DROP MODULE"  
1234 005562 177777 -1  
1235 005564 005100 TRT0: MSG14 ;"TRANSFER READY"  
1236 005566 004667 MSG2 ;"TIME OUT"  
1237 005570 004711 MSG4 ;"DROP MODULE"  
1238 005572 177777 -1  
1239 005574 005066 TRDNEP: MSG13 ;"DONE BIT"  
1240 005576 005006 MSG8 ;"SET -> WAITING ON"  
1241 005600 005100 MSG14 ;"TRANSFER READY"  
1242 005602 004711 MSG4 ;"->DROP MODULE"  
1243 005604 177777 -1  
1244 005606 004701 WTERDD: MSG3 ;"DRIVE"  
1245 005610 005424 DNUM ;"#"
```

```
1246 005612 005156 MSG19 ;"WRITE"  
1247 005614 005056 MSG12 ;"DOUBLE"  
1248 005616 005120 MSG15 ;"DENSITY"  
1249 005620 005131 MSG17 ;"ERROR-RETRY:"  
1250 005622 005430 PTYN ;"#"  
1251 005624 005341 MSG26 ;"CRLF"  
1252 005626 177777 -1  
1253 005630 004701 WTEPSD: MSG3 ;"DRIVE"  
1254 005632 005424 DNUM ;"#"  
1255 005634 005156 MSG19 ;"WRITE"  
1256 005636 005046 MSG11 ;"SINGLE"  
1257 005640 005120 MSG15 ;"DENSITY"  
1258 005642 005131 MSG17 ;"ERROR-RETRY:"  
1259 005644 005430 PTYN ;"#"  
1260 005646 005341 MSG26 ;"CRLF"  
1261 005650 177777 -1  
1262 005652 004701 PTERDD: MSG3 ;"DRIVE"  
1263 005654 005424 DNUM ;"#"  
1264 005656 005166 MSG20 ;"READ"  
1265 005660 005056 MSG12 ;"DOUBLE"  
1266 005662 005120 MSG15 ;"DENSITY"  
1267 005664 005131 MSG17 ;"ERROR-RETRY:"  
1268 005666 005430 PTYN ;"#"  
1269 005670 005341 MSG26 ;"CRLF"  
1270 005672 177777 -1  
1271 005674 004701 WTERSD: MSG3 ;"DRIVE"  
1272 005676 005424 DNUM ;"#"  
1273 005700 005166 MSG20 ;"READ"  
1274 005702 005046 MSG11 ;"SINGLE"  
1275 005704 005120 MSG15 ;"DENSITY"  
1276 005706 005131 MSG17 ;"ERROR-RETRY:"  
1277 005710 005430 PTYN ;"#"  
1278 005712 005341 MSG26 ;"CRLF"  
1279 005714 177777 -1  
1280 005716 004701 SETSDN: MSG3 ;"DRIVE"  
1281 005720 005424 DNUM ;"#"  
1282 005722 005241 MSG24 ;"SETTING DISKETTE TO"  
1283 005724 005046 MSG11 ;"SINGLE"  
1284 005726 005120 MSG15 ;"DENSITY"  
1285 005730 005271 MSG25 ;"IF INTERRUPTED-MAY NEED REFORMAT"  
1286 005732 177777 -1  
1287 005734 004701 SSGLDN: MSG3 ;"DRIVE"  
1288 005736 005424 DNUM ;"#"  
1289 005740 005241 MSG24 ;"SETTING DISKETTE TO"  
1290 005742 005046 MSG11 ;"SINGLE"  
1291 005744 005120 MSG15 ;"DENSITY"  
1292 005746 177777 -1  
1293 005750 004701 SDBLDN: MSG3 ;"DRIVE"  
1294 005752 005424 DNUM ;"#"  
1295 005754 005241 MSG24 ;"SETTING DISKETTE TO"  
1296 005756 005056 MSG12 ;"DOUBLE"  
1297 005760 005120 MSG15 ;"DENSITY"  
1298 005762 177777 -1  
1299 005764 005343 NXERMS: MSG27 ;"->NON EXISTENT MEMORY ERROR"  
1300 005766 177777 -1  
1301 ;-----
```

1302
 1303
 1304
 1305 005770* 000000
 1306 006372*
 1307
 1308 000001

```

----- DATA BUFFERS -----
      .EVEN
RBUF: 0          ;READ BUFFER
      .+400
-----
      .END
  
```

ACSR	000102R	293*	1032*																	
ADDR	000006R	259*	854	855																
ADDR22=	001000	322*																		
ARXCS	004626R	1095*																		
ARXES	004630R	1096*																		
ARXSA	004634R	1098*																		
ARXSB	004636R	1099*																		
ARXSB1	004640R	1100*																		
ARXSB2	004642R	1101*																		
ARXSB3	004644R	1102*																		
ARXTA	004632R	1097*																		
ASB	000106R	297*																		
ASTAT	000104R	295*	1034*																	
AWAS	000110R	298*																		
AWDN	004344R	530	585	640	725	747	769	780	947	1001	1039*									
AWTF	004454R	551	555	575	580	606	610	630	635	775	799	819	996	1062*						
BEGIN	000000R	256*	461	466	485	486	495	524	535	594	596	651	653	656						
		712	734	736	756	758	779	803	823	863	866	904	907	915						
		919	922	929	933	936	950	960	979	989	1014	1016	1021	1043						
		1044	1053	1055	1070	1071	1080	1085	1088											
BFER1	005434R	653	1191*																	
BFER2	005442R	596	1194*																	
BGND0	000670R	477*																		
BIT0 =	000001	322*																		
BIT1 =	000002	322*																		
BIT10 =	002000	322*																		
BIT11 =	004000	322*																		
BIT12 =	010000	322*																		
BIT13 =	020000	322*																		
BIT14 =	040000	322*																		
BIT15 =	100000	322*																		
BIT2 =	000004	322*																		
BIT3 =	000010	322*																		
BIT4 =	000020	322*																		
BIT5 =	000040	322*																		
BIT6 =	000100	322*																		
BIT7 =	000200	322*																		
BIT8 =	000400	322*																		
BIT9 =	001000	322*																		
BK0YPS	000326R	346*	579*	581*	634*	636*	995*	997*	1068	1089*										
BKLOOP	001002R	478	484	489	492	497*														
BREAK6=	104407	322*	1043	1044	1070	1071														
BR1	000012R	261*	853	871																
BR2	000013R	262*																		
BTOD6 =	104421	322*																		
CDATAS=	104412	322*	656																	
CDERCT	000144R	312*																		
CDWDCT	000146R	313*																		
CKDNER	003472R	885*	897*																	
CKNXEF	004164R	588*	645*	1002*	1008*															
CKRDER	003544R	888*	913*																	
CKWTER	003606R	890*	927*																	
CL	003464R	889	891*																	
CLEAR	003650R	891	942*																	
CMD	000762R	328*	528*	529	547*	548*	549*	550	569*	570*	573*	574	602*	603*						
		604*	605	624*	625*	628*	629	722*	723*	724	744*	745*	746	766*						

Table with columns for module names and cross-reference values. Includes entries like CONFIG, CSRA, DATEK, etc., with values ranging from 281# to 1185#.

Table with columns for module names and cross-reference values. Includes entries like HRDPAS, ICONF, ICONF, etc., with values ranging from 278# to 312#.

WDTO	000114R	300#							
WRITE	000740R	487#							
WTERDD	005606R	936	1244#						
WTERSD	005630R	933	1253#						
WTF	000270R	331#	542#	564#	886				
XDNER	003542R	900	905	900#					
XDNSSET	002732R	771	777	782#					
XFLAG	000005R	258#							
XNXER	004260R	1009	1019	1022#					
XUDNCK	003276R	844	846#						
.	= 006372R	365	366	657	1184#	1186#	1188#	1306#	

. ABS. 000000 000
006372 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XRXBC0, XRXBC0/SOL/CRF; SYM=DDXCOM, XRXBC0
RUN-TIME: 7 12 1 SECONDS
RUN-TIME RATIO: 90/21=4.2
CORE USED: 9K (17 PAGES)