

1

.NLIST SEQ,LD,BIN
.REPT 0

.NLIST TTM

IDENTIFICATION

PRODUCT CODE: AC-F098C-MC
PRODUCT NAME: CXRXBC0 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

FXR 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/TRACK #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
BRI:5
DEVcnt: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 F., F. & G. FOR NEW DENSITY
- J. UPDATE STARTING ADDRESS, GO TO D

8. OPERATION OPTIONS

SR1 BIT 0 SET(1):
IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, REPORT
HARD ERROR AND DROP THE MODULE.

SR1 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.

PXCS (COMMAND REGISTER)
 RXES (ERROR REGISTER)
 FXTA (TRACK ADDRESS)
 FXSA (SECTOR ADDRESS)
 PXSB (WORD COUNT REG.) / (DEFINITIVE ERROR CODE)->----
 PXSBI (DRV1 TRACK ADR) / (DRV0 TRACK ADR) !
 PXSPI (TARGET SECTOR) / (TARGET TRACK) !
 PXSPI (TRACK ADR SELECTED DRV*) / (MICROCODE STATUS)->-- !

(UPPER 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 = HEAD 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
KERTRK= 40	/TRIED TO ACCESS A TRACK GREATER THAN 76.
KHOMERR= 50	/HOME WAS FOUND BEFORE DESIRED TRACK WAS REACHED.
KSELFERR= 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.
KNXPBAM= 120	/A PREAMBLE COULD NOT BE FOUND.
KNXIDAM= 130	/PREAMBLE FOUND BUT NO ID MARK FOUND WITHIN ALLOWABLE TIME.
KNCHCER= 140	/CRC ERROR ON WHAT APPEARED TO BE A HEADER. ERROR IS NOT ASSERTED.
KTKSKER= 150	/TRACK ADDRESS OF GOOD HEADER DOES NOT COMPARE WITH DESIRED TRACK.
KXSTRYS= 160	/TOO MANY TRIES FOR AN IDAM.
KMODAM= 170	/DATA AM NOT FOUND IN ALLOTTED TIME.
KDCRCER= 200	/CRC ERROR ON READING THE SECTOR FROM THE DISK.
KMANER= 220	/R/W ELECTRONICS FAILED MAINTENANCE MODE TEST.
KWCNOV= 230	/WORD COUNT OVERFLOW.
KSTDER= 240	/WRONG KEY WORD FOR SET MEDIA DENSITY COMMAND.

C. RETRIES; EACH WRITE OR 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 SR1=1 OR
 CONTINUE TO NEXT ADDRESS IF SR1=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
 -

!.....

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCS:	EPR	INT	XM	XM	RX2	ISID	DEN	TR	IE	DON	DRV	FUN	FUN	FUN	GO	
WC:	X	X	X	X	X	X	X	X								WORD COUNT
BA:	BASE ADDRESS															
RXES:	X	X	X	X	MXM	WC	ISID	DRV	DEL	DSK	DEN	AC	INT	ISID	CRC	
							OVF	#1	#1	RDY	DAT	DEN	ERR	LOW	DON	RDY
RXTA:	X	X	X	X	X	X	X	X	X	X	0					TRACK ADDRESS
RXSA:	X	X	X	X	X	X	X	X	X	X	0	0	0			SECTOR ADDRESS

[0 -> 76. LEGAL]

[1 -> 26. LEGAL]

.ENDP

```

IOMODX <RXRC>,177170,264,5,0,2,202,137,RBUF,64,,66.
MODULE 150000,RXRC,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
;*****
REGIN:
MODNAM: .ASCII /RXRC / MODULE NAME.
XFLAG: .RYTF OPEN ;USED TO KEEP TRACK OF WBUF USAGE
ADDR: 177170+0 ;1ST DEVICE ADDR.
VECTOP: 264+0 ;1ST DEVICE VECTOR.
BR1: .BYTE PRTY5+0 ;1ST RR LEVEL.
BR2: .BYTE PRTY0+0 ;2ND RR LEVEL.
DVID1: 2+1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1
SR2: OPEN ;SWITCH REGISTER 2
SR3: OPEN ;SWITCH REGISTER 3
SR4: OPEN ;SWITCH REGISTER 4
;*****
STAT: 150000 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SPOINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICONT: 202 ;# OF ITERATIONS PER PASS=2V2
ICOUNT: 0 ;LOC TO COUNT ITERATIONS
SOFcnt: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRCPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
RANUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG:
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVR0: OPEN ;LOC TO SAVE R0.
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;ADDR OF CURRENT CSR.
SRADR: ;ADDR OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: ;ADDR OF BAD DATA, OR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTPY: ;TYPE OF ERROR
ASB: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSTR: RSTR ;RESTART ADDRESS AFTER END OF PASS
WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
WDFP: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 137 ;MODULE IDENTIFICATION NUMBER=137
RBUFVA: RBUF ;READ BUFFER VIRTUAL ADDRESS
RBUFPA: OPEN ;READ BUFFER PHYSICAL ADDRESS
  
```

```

RBUFPA: OPEN ;READ BUFFER PA BITS
RBUFSZ: 64. ;SIZE OF THE READ BUFFER
WBUFPA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
WBUFSZ: 66. ;WRITE BUFFER SIZE REQUESTED
WBUFQ: 66. ;WRITE BUFFER SIZE AVAILABLE
WBUFSZ: OPEN ;COATA/DATCK ERROR COUNT
CORRECT: OPEN ;COATA/DATCK WOPD COUNT
CONDCT: OPEN ;RESERVED FOR FUTURE USE
FREE: OPEN ;MODULE STACK STARTS HERE.
,REPT EPSIZ
,LIST
,MOD 0
,LIST
,ENDR
MODSP:
;*****
  
```

```

322
323
324 000257* 000000
325 000254* 000000
326 000256* 000000
327 000260* 000000
328 000262* 000000
329 000264* 000000
330 000266* 000000
331 000270* 000000
332 000272* 000000
333 000274* 000000
334 000276* 000000
335 000300* 000000
336 000302* 000100
337 000304* 000000
338 000306* 000000
339 000310* 000000
340 000312* 000000
341 000314* 000000
342 000316* 000000
343 000320* 000000
344 000322* 000000
345 000324* 000000
346 000326* 000000
347 000330* 000000
348 000332* 000000
349 000334* 000000
350 000336* 000000
351
352
353
354
355
356 000340* 000000
357 000342* 000000
358 000344* 000000
359 000346* 000000
360 000350* 000000
361 000352* 000000
362 000354* 000000
363 000356* 000000
364 000360* 000350*
365 000362* 000364*
366 000364* 000366*
367
368
369
370 000366* 177170
371 000370* 177172
372 000372* 000264
373 000374* 000111
374

```

```

;---FLAGS AND COUNTERS ---
DVIDIX: 0 ;HOLDS WHICH DRIVES TO TEST
UNIT0: 0 ;UNIT 0 FLAG
UNIT1: 0 ;UNIT 1 FLAG
DRVN: 0 ;DRIVE NUMBER
CMD: 0 ;COMMAND SAVE
UTTI: 0 ;UNIT UNDER TEST
UTI: 0 ;UNIT UNDER COMMAND
WTF: 0 ;WRITE FLAG
TAI: 0 ;CURRENT TRACK ADDRESS
SAI: 0 ;CURRENT SECTOR ADDRESS
LASAV: 0 ;STARTING TRACK ADDRESS SAVE
SASAV: 0 ;STARTING SECTOR ADDRESS SAVE
WDCNT: 100 ;WORD COUNTER
DENSITY: 0 ;DENSITY
DEN: 0 ;DENSITY FLAG
TOMLT: 0 ;TIME OUT MULTIPLIER
TOCNT: 0 ;TIME OUT COUNTER
INT: 0 ;INITIALIZE FLAG
UTDONE: 0 ;UNIT DONE FLAG
TSSFLG: 0 ;TSS DONE WITH UNIT FLAG
INLOOP: 0 ;IN LOOP FLAG
ITBYP: 0 ;RESTART INIT BYPASS FLAG
RKBYPS: 0 ;BREAK BYPASS FLAG
RTRYFL: 0 ;RETRY FLAG
EXERCT: 0 ;NON EXISTENT MEMORY ERROR COUNTER
SERFLI: 0 ;STATUS ERROR FLAG
FIN: 0 ;FINI FLAG
;---TEMPORARY STORAGE REGISTERS ---
SPXCS: 0
SRXES: 0
SRXTA: 0
SRXSA: 0
SPXSH: 0
SRXSH1: 0
SRXSH2: 0
SRXSP: 0
CECYA: SPXSP ;DEFINITIVE ERROR CODE VIRTUAL ADR
;+2 ;DEFINITIVE ERROR CODE PHYSICAL ADR
;+2 ;DEFINITIVE ERROR CODE EXTENDED ADM BITS
;---CONSTANTS ---
HXCS: 177170 ;HX02 COMMAND REGISTER - ADDRESS
RXDB: 177172 ;RX02 DATA BUFFER - ADDRESS
VECI: 264 ;VECTOR
VARIFY: 111 ;VARIFY WD FOR SET DENSITY=ASCII "I"

```

```

375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430

```

```

;BEGINROUTINE (MODULE 0.0 -- CONTROL)
; IF START
; ; INITIALIZE
; ; CALL MOD 1.0 - INITIALIZE DEVICE
; ; CALL MOD 6.0 - CHECK STATUS
; RESTART
; BEGINCO
; ; IF INLOOP NOT SET
; ; THEN
; ; ; IF INITIALIZE_BYPASS NOT SET
; ; ; THEN-INITIALIZE-TRACK & SECTOR
; ; ; ENDF
; ; ; IF FINI_FLAG NOT SET
; ; ; THEN
; ; ; ; CALL MOD 2.0 - OUTPUT (WRITE)
; ; ; ; IF FINI_FLAG NOT SET
; ; ; ; THEN
; ; ; ; ; CALL MOD 3.0 - INPUT (READ)
; ; ; ; ; IF FINI_FLAG NOT SET
; ; ; ; ; THEN
; ; ; ; ; CALL MOD 4.0 - PROCESS SECTOR
; ; ; ; ; SET INLOOP_FLAG & CALL ITERATION
; ; ; ; ENDF
; ; ; ENDF
; ; ; ENDF
; ; ENDF
; ; CLEAR INLOOP
; ; SET INITIALIZE_BYPASS_FLAG
; ; IF SECTOR_DONE_FLAG SET
; ; ; THEN
; ; ; CLEAR SECTOR_DONE_FLAG
; ; ; CALL MOD 8.0-UNITS DONE CHECK
; ; ; IF UNIT_DONE SET
; ; ; ; THEN
; ; ; ; SET TRACK_ADR=TRACK_ADR+10,
; ; ; ; IF TRACK_ADR > 76
; ; ; ; THEN-SET TRACK_ADR=TRACK_ADR SAVE
; ; ; ; INCREMENT SECTOR_ADDRESS_SAVE & CLEAR INITIALIZE_BYPASS_FLAG
; ; ; ; IF SECTOR_ADDRESS_SAVE=4
; ; ; ; ; THEN
; ; ; ; ; SET SECTOR_ADDRESS_SAVE=1
; ; ; ; ; INCREMENT TRACK_ADDRESS_SAVE
; ; ; ; ; IF TRACK_ADDRESS_SAVE=11,
; ; ; ; ; THEN
; ; ; ; ; SET TRACK_ADDRESS_SAVE=1
; ; ; ; ; COMMENT DENSITY_FLAG
; ; ; ; ; CALL MOD 7.0 - SET DENSITY
; ; ; ; ENDF
; ; ; ENDF
; ; ; ENDF
; ; ENDF
; ; ENDF
; ; CALL MOD 5.0-SELECT UNIT FOR TEST
; ; DOUNTIL FINI=1
; ; FINI-->END
;ENDROUTINE

```



```
431 ;MOD P,0 CONTROL -- PROCESS SECTORS -----
432 START: NOP
433 MOV SPOINT,SP ;INITIALIZE STACK
434 MOV #1,TASAV ;PRESET TRACK ADDRESS SAVE
435 MOV #1,SASAV ;PRESET SECTOR ADDRESS
436 MOV #100,WDNCT ;SET WORD COUNT FOR SINGLE DENSITY
437 CLR FIN ;CLEAR FINI FLAG
438 CLR DEN ;CLEAR DENSITY FLAG
439 CLR DFNSTY ;CLEAR DENSITY
440 CLR UNIT0 ;CLEAR UNIT#0 AVAIL FLAG
441 CLR UNIT1 ;CLEAR UNIT#1 AVAIL FLAG
442 CLR UTT ;CLEAR UNIT UNDER TEST
443 CLR DRYN ;CLEAR DRIVE #
444 CLR SFRFL ;CLEAR STATUS ERR FLAG
445 CLR NXERCT ;CLEAR NON-EXISTENT MEMORY ERROR COUNT
446 MOV DVID1,DVID1X ;GET DRIVE COUNT
447 CMPR #15,#41 ;IF RX02 FLOPPY WAS LOAD MEDIUM
448 RNE FIRST ;THEN
449 TSTR #40 ;IF DRIVE #
450 RNE D1 ;THEN
451 RIC #1,DVID1X ;KILL TEST FOR DRIVE 0
452 RIF #1,DVID1 ;WAS IT TO BE TESTED?
453 BFC FIRST ;IF NO
454 MOV #260,DNUM ;SET DRIVE # FOR PRINT
455 RR NNAV ;RR TO NOT AVAIL YSG
456 RIC #2,DVID1X ;KILL TEST FOR DRIVE 1
457 RIT #2,DVID1 ;IF WAS TO BE TESTED
458 BEQ FIRST ;THEN
459 MOV #261,DNUM ;SET DRIVE #1 FOR PRINT
460 NUAV:
461 MSGN0,BEGIN,NUNT ;FASCT MESSAGE CALL WITH COMMON HEADER
462 FIPST: MOV #1,INT ;SET INIT FLAG
463 JSR PC,VSET ;GO SETUP ADDRESS & VECTOR
464 BIT #4000,0RXCS ;IF DEVICE IS
465 RNF GO ;RX01 THEN
466 MSGN0,BEGIN,DRP4 ;FASCT MESSAGE CALL WITH COMMON HEADER
467 JMP FINI ;GO DROP MODULE
468 JSR PC,INITIAL ;INITIALIZE DEVICE---DO MOD 1.0
469 TST FIN ;IF FINI FLAG
470 BNE FINI ;NOT SET THEN
471 JSR PC,STATUS ;CHECK STATUS-----DO MOD 6.0
472 TST FIN ;IF FINI FLAG
473 BNE FINI ;NOT SET, THEN
474 CLR INT ;RESET INIT FLAG
475 CALL SFLD ;GO SELECT UNIT-----DO MOD 5.0
476 RESTRT: NOP
477 RGNDO: TST INLOOP ;IF INLOOP
478 RNE BKLOOP ;NOT SET, THEN
479 TST ITRYPS ;IF INITIALIZE BYPASS FLAG
480 BNE EDRYPS ;NOT SET, THEN
481 MOV SASAV,SA ;PRESET SECTOR STARTING ADDRESS
482 MOV TASAV,TA ;PRESET TRACK STARTING ADDRESS
483 EDRYPS: TST FIN ;IF FINI FLAG
484 BNE BKLOOP ;NOT SET, THEN
485 GETPAS,BEGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT BRUFVA
486 GRWUF8,BEGIN ;GET WRITE BUFFER INFORMATION
```

```
487 WRITE: JSR PC,OUTPUT ;GO WRITE-----DO MOD 2.0
488 TST FIN ;TEST FINI FLAG
489 BNE BKLOOP ;IF SET THEN BR
490 READ: JSR PC,INPUT ;GO READ-----DO MOD 3.0
491 TST FIN ;TEST FINI FLAG
492 BNE BKLOOP ;IF SET THEN BR
493 CALL TSS ;GO UPDATE SECTOR - DO MOD 4.0
494 MOV #1,INLOOP ;SET INLOOP FLAG
495 ENDTSS,BEGIN ;SIGNAL END OF ITERATION.
496 ;MONITOR SHALL TEST END OF PASS
497 BKLOOP: CLR INLOOP ;CLEAR INLOOP FLAG
498 MOV #1,ITRYP5 ;SET INITIALIZE BYPASS FLAG
499 TST TSSFLG ;IF UNIT TSS
500 BFC DOUNTL ;IS DONE, THEN
501 CLR TSSFLG ;CLEAR FLAG FOR TSS DONE
502 CALL UDONCK ;CALL UNIT DONE CHECK - DO MOD 6.0
503 TST UTDONE ;IF UNIT DONE
504 BEQ 18 ;NOT SET, THEN
505 ADD #12,TA ;INCREMENT TRACK ADDRESS
506 CMP #114,TA ;SEE IF DONE TRACKS
507 BHS 18 ;IF NOT: BR
508 MOV TASAV,TA ;RESET TRACK ADDRESS
509 CLR ITRYPS ;CLEAR INITIALIZE BYPASS FLAG
510 INC SASAV ;BUMP STARTING SECTOR ADDRESS
511 CMP #4,SASAV ;IF SECTORS = DONE (THREE)
512 BNE 18 ;THEN
513 MOV #1,SASAV ;PRESET STARTING SECTOR
514 INC TASAV ;BUMP STARTING TRACK ADDRESS
515 CMP #11,TASAV ;IF TRACKS=DONE (TEN)
516 BNE 18 ;THEN
517 MOV #1,TASAV ;RESET TRACK STARTING ADDRESS
518 COM DEN ;CHANGE DENSITY
519 JSR PC,DENCH ;SET DEVICE DENSITY
520 CALL SFLD ;SELECT UNIT-----DO MOD 5.0
521 DOUNTL: TST FIN ;DO UNTIL FINI FLAG
522 BEQ RESTRT ;SET
523 FINI:
524 ENDS,BEGIN ;DROP THE MODULE
525 ;-----
```

```

526 ;MOD 1,0 INITIALIZE DEVICE -----
527
528 001162* 012767 000000 177072 INITIAL: MOV #40000,CMD ;SET INT COMMAND
529 001170* 016777 177066 177170 MOV CMD,ARXCS ;INIT UNIT 0
530 001176* 024767 003142 JSP 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 ARXCS ;SET IF ERROR ON INIT
534 001214* 100005 RPL ENDINT ;IF NOT! BR
535 001216* 104403 000000* 005450* MSGN8,AFGIN,DRP1 ;ASCII MESSAGE CALL WITH COMMON HEADER
536 001224* 004767 002564 JSP PC,TDPR ;DO TEST DROP CLEAN-UP
537 001230* 000207 ENDINT: RTS PC ;RETURN
538 ;-----
539 ;MOD 2,0 WRITE SUBROUTINE -----
540
541 001232* 012767 000001 177030 OUTPUT: MOV #1,WTF ;SET WRITE FLAG
542 001240* 004767 000134 JSP PC,OURUF2 ;THEN O/P RX02 BUFFER - DO MOD 2.1
543 001244* 005767 177066 TST FIN ;IF FINI FLAG
544 001250* 001050 BNE ENDOUT ;EQUALS ZERO THEN
545 001252* 012767 000260 004150 MOV #260,RTYN ;RESET PETRY COUNTER-ASCII "0"
546 001260* 016767 177000 176774 181 MOV UTT,CMD ;SELECT DRIVE
547 001266* 052767 000105 176766 BIS #105,CMD ;SET TO WRITE SECTOR + INT ENABLE
548 001274* 056767 177004 176760 BIS DENSITY,CMD ;SET DENSITY
549 001302* 016777 176754 177056 MOV CMD,ARXCS ;LOAD COMMAND
550 001310* 004767 003140 JSP PC,AWTR ;GO AWAIT TRANSFER READY
551 001314* 005767 177016 TST FIN ;IF FINI FLAG
552 001320* 001024 BNE ENDOUT ;EQUALS ZERO THEN
553 001322* 016777 176746 177040 MOV SA,ARXDR ;LOAD SECTOR ADDRESS
554 001330* 004767 003120 JSP PC,AWTR ;GO AWAIT TRANSFER READY
555 001334* 005767 176776 TST FIN ;IF FINI FLAG
556 001340* 001014 BNE ENDOUT ;EQUALS ZERO THEN
557 001342* 016777 176724 177020 MOV TA,ARXDR ;LOAD TRACK ADDRESS
558 001350* 004767 001764 JSP PC,INTER ;WAIT FOR INTERRUPT
559 001354* 005767 176756 TST FIN ;IF FINI FLAG
560 001360* 001004 BNE ENDOUT ;NOT SET, THEN
561 001362* 022767 000003 176740 CMP #3,PTRYFL ;IF RETRY FLAG
562 001370* 001333 BNE 10 ;EQUALS 3 THEN
563 001372* 005067 176672 ENDOUT: CLR WTF ;CLEAR WRITE FLAG
564 001376* 000207 RTS PC ;RETURN
565 ;-----
566

```

```

567 ;MOD 2,1 OUTPUT RX02 BUFFER -----
568
569 001400* 012767 000001 176654 OURUF2: MOV #1,CMD ;SET FILL BUFFER COMMAND
570 001406* 056767 176672 176646 BIS DENSITY,CMD ;SET DENSITY
571 001414* 016701 176516 MOV WRUFEA,P1 ;GET EXT. ADX. BITS
572 001420* 000301 SWAB R1 ;
573 001422* 050167 176634 BIS R1,CMD ;SET EXT. ADX. BITS
574 001426* 016777 176630 176732 MOV CMD,ARXCS ;LOAD COMMAND
575 001434* 004767 003014 JSP PC,AWTR ;WAIT FOR "TR"
576 001440* 005767 176672 TST FIN ;IF FINI FLAG
577 001444* 001044 BNE ENDOUT2 ;EQUALS ZERO THEN
578 001446* 016777 176630 176714 MOV WDCNT,ARXDB ;LOAD WORD COUNT FOR OUTPUT BUFFER
579 001454* 005267 176646 JNC BRKYP5 ;SET BREAK BYPASS FLAG
580 001460* 004767 002770 JSP PC,AWTR ;WAIT FOR "TR"
581 001464* 005067 176636 CLR BRKYP5 ;CLEAR BREAK BYPASS FLAG
582 001470* 005767 176642 TST FIN ;IF FINI FLAG
583 001474* 001030 BNE ENDOUT2 ;EQUALS ZERO THEN
584 001476* 016777 176432 176604 MOV WRUFPA,ARXDR ;LOAD BASE ADDRESS FOR OUTPUT BUFFER
585 001504* 004767 002634 JSP PC,AWDN ;WAIT FOR "DONE"
586 001510* 005767 176622 TST FIN ;IF FINI FLAG
587 001514* 001020 BNE ENDOUT2 ;EQUALS ZERO THEN
588 001516* 004767 002442 CALL CKNXP ;CALL CHECK NON-EXISTENT MEMORY ERRORS
589 001522* 005777 176640 TST ARXCS ;IF DEVICE ERROR BIT
590 001526* 100013 RPL ENDOUT2 ;IS SET THEN
591 001530* 004767 002526 CALL SUPTRG ;CALL SETUP PRINT REGISTERS
592 001534* 012767 000035 176344 MOV #35,FPRTYP ;SETUP FILL BUFFER ERROR
593 ;*****
594 001542* 104405 000000* 000000 HRDRS,BEGIN,NULL ;BUFFER FILL ERROR
595 ;*****
596 001550* 104402 000000* 005442* MSGSS,BEGIN,BFEP2 ; ASCII MESSAGE CALL WITH NO HEADER
597 001556* 000207 ENDOUT2: RTS PC ;RETURN
598 ;-----

```

```

599 ;MOD 3.0 READ SUBROUTINE -----
600
601 001560 012767 000260 003642 INPUT: MOV #260,RTYM ;RESET RETRY COUNTER=ASCII "0"
602 001566 016767 176472 176466 18: MOV UTT,CMD ;SELECT DRIVE
603 001574 052767 000107 176460 RIS #107,CMD ;SET READ COMMAND + INT ENB
604 001602 056767 176476 176452 RIS DFNSTY,CMD ;SET DENSITY
605 001610 016777 176446 176550 MOV CMD,0RXCS ;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 RNE ENDIN ;NOT SET, THEN
609 001630 016777 176440 176532 MOV SA,0RXDB ;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 RNE ENDIN ;NOT SET, THEN
613 001650 016777 176416 176512 MOV TA,0RXDB ;LOAD TRACK ADDRESS
614 001656 004767 001456 JSR PC,INTERR ;WAIT FOR INTERRUPT
615 001662 022767 000003 176440 CMP #3,RTNYFL ;IF RETRY FLAG
616 001670 001336 RNE 18 ;EQUALS 3, THEN
617 001672 004767 000002 JSR PC,INRUF2 ;THEN GET WX02 BUFFER - DO MOD 3.1
618 001676 000207 ENDIN: RTS PC ;RETURN
619 ;-----
  
```

```

620 ;MOD 3.1 INPUT RX02 BUFFER -----
621
622 001700 005767 176430 INRUF2: TST SERFL ;IF STATUS ERROR NOT SET
623 001704 001077 RNE ENDIN2 ;THEN
624 001706 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 PRUFEA,R1 ;GET EXT. ADDR. BITS
627 001726 000301 SWAB R1 ;
628 001730 050167 176326 BIS R1,CMD ;SET EXT. ADDR. BITS
629 001734 016777 176322 176424 MOV CMD,0RXCS ;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 RNE ENDIN2 ;EQUALS ZERO
633 001754 016777 176322 176406 MOV WDCNT,0RXDB ;THEN LOAD WORD COUNT FOR INPUT BUFFER
634 001762 005267 176340 INC BXBYP ;SET BREAK BYPASS FLAG
635 001766 004767 002462 JSR PC,AWTR ;WAIT FOR "TR" DO MOD U,TR
636 001772 005067 176330 CLR BXBYP ;CLEAR BREAK BYPASS FLAG
637 001776 005767 176334 TST FIN ;IF FINI FLAG
638 002002 001040 RNE ENDIN2 ;EQUALS ZERO
639 002004 016777 176116 176356 MOV R0UFA,0RXDB ;THEN LOAD BASE ADDRESS FOR INPUT BUFFER
640 002012 004767 002326 JSR PC,AWDN ;WAIT FOR "DONE"
641 002016 005767 176314 TST FIN ;IF FINI FLAG
642 002022 001030 RNE ENDIN2 ;EQUALS ZERO THEN
643 002024 005777 176336 TST 0RXCS ;IF DEVICE ERROR BIT
644 002030 100021 BPL 18 ;SET, THEN
645 002032 004767 002126 CALL CKNXER ;CALL CHECK NON-EXISTENT MEMORY ERRORS
646 002036 005777 176324 TST 0RXCS ;IF DEVICE ERROR BIT
647 002042 100020 BPL ENDIN2 ;IS STILL SET, THEN
648 002044 004767 002212 CALL SUPTRG ;CALL SETUP PRINT REGISTERS
649 002050 012767 000030 176030 MOV #30,ERRTY ;SETUP DATA TRANSFER ERROR
650 ;*****
651 002056 104405 000000 000000 HRFDR$,REGIN,NULL ;EMPTY BUFFER ERROR
652 ;*****
653 002064 104402 000000 005434 MSG$,REGIN,BFER1 ;ASCII MESSAGE CALL WITH NO HEADER
654 002072 000404 ER ENDIN2 ;RR TO EXIT
655 002074 18:
656 002074 104412 000000 000126 CPDATA$,BEGIN,PRUFPA ;REQUEST FOR MONITOR TO CHECK DATA
657 002102 002104 .+2 ;IF ERROR, CONTINUE
658 002104 000207 ENDIN2: RTS PC ;RETURN
659 ;-----
  
```

```

660 ;MOD 4.0 SECTOR UPDATE -----
661
662 002106* 062767 000003 176160 TSS: AND #3,SA ;RUMP SECTOR ADDRESS
663 002114* 022767 000033 176152 CMP #33,SA ;SEE IF DONE SECTORS
664 002122* 001401 BR 18 ;IF SO: BR
665 002124* 000406 BR ENDTSS ;RETURN
666 002126* 016767 176146 176140 18: MOV SASAV,SA ;RESET SA
667 002134* 012767 000001 176156 MOV #1,TSSFLG ;SET TSS FLAG- (UNIT DONE WITH TSS)
668 002142* 000207 ENDTSS: RTS PC
669
670 ;MOD 5.0 SELECT UNIT FOR TEST -----
671
672
673 002144* 005767 176110 SELD: TST DRVN ;SEE IF DRIVE 0
674 002150* 001415 BEQ 28 ;IF SO: BR
675 002152* 005067 176102 CLR DRVN ;ELSE SET TO 0
676 002156* 005767 176074 TST UNIT1 ;SEE IF UNIT 1 AVAILABLE
677 002162* 001401 BEQ 18 ;IF SO: BR
678 002164* 000767 BR SELD ;GO SEE IF UNIT IS AVAILABLE
679 002166* 012767 000020 176070 18: MOV #20,UTT ;SET UNIT UNDER TEST
680 002174* 012767 000261 003222 MOV #261,DNUM ;SET DRIVE NUMBER FOR PRINTS
681 002202* 000413 BR ENDSL ;RETURN
682 002204* 005167 176050 28: COM DRVN ;SWITCH UNITS
683 002210* 005767 176040 TST UNIT0 ;SEE IF UNIT 0 AVAILABLE
684 002214* 001401 BEQ 36 ;IF SO: BR
685 002216* 000752 BR SFLD ;SELECT NEXT
686 002220* 005067 176040 36: CLR UTT ;SELECT DRIVE 0
687 002224* 012767 000260 003172 MOV #260,DNUM ;SET DRIVE NUMBER FOR PRINTS
688 002232* 000207 ENDSL: RTS PC ;RETURN
689

```

```

690 ;MOD 6.0 DEVICE STATUS -----
691
692 002234* 000200 STATUS: NOP
693 002236* 004767 000120 00: JSR PC,UNIT0 ;CHECK STATUS UNIT 0 - DO MOD 6.1
694 002242* 005767 176006 TST UNIT0 ;IF UNIT 0 IS
695 002246* 001012 BNE U1 ;AVAIL, THEN
696 002250* 005767 176040 TST INT ;IF INIT FLAG
697 002254* 001407 BEQ U1 ;IS SET, THEN
698 002256* 005067 176002 CLR UTT ;SET UNIT=0
699 002262* 012767 000260 003134 MOV #260,DNUM ;SET DRIVE # FOR PRINT
700 002270* 004767 000322 JSR PC,DNSET ;SETUP DENSITY UNIT 0 - DO MOD 6.3
701 002274* 004767 000200 01: JSR PC,UNIT1 ;CHECK STATUS UNIT 1 - DO MOD 6.2
702 002300* 005767 175752 TST UNIT1 ;IF UNIT 1 IS
703 002304* 001014 BNE 18 ;AVAIL, THEN
704 002306* 005767 176002 TST INT ;IF INIT FLAG
705 002312* 001422 BEQ ENDST ;IS SET, THEN
706 002314* 012767 000020 175742 MOV #20,UTI ;SET UNIT=1
707 002322* 012767 000261 003074 MOV #261,DNUM ;SET DRIVE # FOR PRINT
708 002330* 004767 000262 JSR PC,DNSET ;SETUP DENSITY UNIT 1 - DO MOD 6.3
709 002334* 000411 BR ENDSL ;END TO END STATUS
710 002336* 005767 175712 18: TST UNIT0 ;IF UNIT 0
711 002342* 001406 BNE ENDSL ;NOT AVAIL, THEN
712 002344* 104403 000000* 005460 MSGS,REGIN,DRP2 ;ASCII MESSAGE CALL WITH COMMON HEADER
713 002352* 012767 000001 175756 MOV #1,FIN ;GO DROP MODULE
714 002360* 000207 ENDSL: RTS PC ;RETURN
715
716
717 ;MOD 6.1 UNIT 0 STATUS -----
718
719
720 002362* 032767 000001 175667 UNIT: BIT #1,DVIDIX ;IF UNIT 0
721 002370* 001437 BEQ 18 ;SELECTED, THEN
722 002372* 012767 000011 175662 MOV #13,CMD ;READ STATUS UNIT 0
723 002400* 005767 175700 175654 RTS DENSITY,CMD ;SET DENSITY
724 002406* 016777 175650 175752 MOV CMD,ORACS ;SEND COMMAND
725 002414* 004767 001724 JSR PC,AWDN ;GO AWAIT DONE
726 002420* 005767 175712 TST FIN ;IF FINI FLAG
727 002424* 001024 BNE ENDSL ;EQUALS ZERO THEN
728 002426* 017700 175736 MOV RXDR,R0 ;GET RXES
729 002432* 032700 000200 BIT #200,R0 ;IF DRIVE RDY
730 002436* 001017 BNE ENDSL ;NOT SET, THEN
731 002440* 012767 000260 002756 MOV #260,DNUM ;SET DRIVE # FOR PRINT
732 002446* 012767 000000 175432 MOV #0,FRPTYP
733
734 002454* 104405 000000* 000000 ;*****
735 ;HDRS,REGIN,NULL ;UNIT 0 NOT AVAILABLE
736 002462* 104402 000000* 005510 MSGS,REGIN,NUNT ;*****
737 002470* 012767 177777 175556 18: MOV #1,UNIT0 ;ASCII MESSAGE CALL WITH NO HEADER
738 002476* 000207 ENDSL: RTS PC ;RESELECT UNIT 0
739

```

```
740 ;MOD 6.2 UNIT 1 STATUS -----
741
742 002500* 032767 000002 175544 UNIT1: BIT #2, DVID1X ;IF UNIT 1
743 002500* 001437 REQ 18 ;SELECTED, THEN
744 002510* 012767 000033 175544 MOV #13, CMD ;READ STATUS UNIT 1
745 002516* 056767 175562 175536 RIS DENSITY, CMD ;SET DENSITY
746 002524* 016777 175532 175634 MOV CMD, PRXCS ;SEND COMMAND
747 002532* 004767 001606 JSR PC, AWDN ;GO AWAIT DONE
748 002536* 005767 175574 TST FIN ;IF FINI FLAG
749 002542* 001024 BNE ENDUT1 ;EQUALS ZERO THEN
750 002544* 017767 175620 MOV PRXCS, R0 ;READ RXES
751 002550* 032700 000200 BIT #200, R0 ;IF DRIVE HDY
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, ERPTYP
755 ;*****
756 002572* 104405 000000* 000000 ;UNIT 1 NOT AVAILABLE
757 ;*****
758 002600* 104402 000000* 005510* MSGS6, BEGIN, NUNT ;ASCII MESSAGE CALL WITH NO HEADER
759 002606* 012767 177777 175442 18: MOV #=1, UNIT1 ;DEFLECT 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 UTT, CMD ;SET UNIT
768 002632* 016777 175424 175526 MOV CMD, PRXCS ;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 UTT, CMD ;SET UNIT
774 002666* 016777 175372 175472 MOV CMD, PRXCS ;SEND COMMAND
775 002674* 004767 001554 JSR PC, AWDN ;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 VERIFY, PRXDE ;SEND VERIFY WORD
779 002714* 104403 000000* 005716* MSGS6, 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 ERROR
782 002732* 000207 XDNSET: RTS PC ;RETURN
783 ;-----
```

```
784 ;MOD 7.0 DENSITY CHANGE -----
785
786 002734* 000240 DNCH: NOP
787 002736* 005767 175314 TST UNIT1 ;IF UNIT 1
788 002742* 001007 HNE 18 ;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 26 ;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 56 ;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, PRXCS ;SEND COMMAND
799 003024* 004767 001424 JSR PC, AWDN ;WAIT FOR "TR"
800 003030* 005767 175302 TST FIN ;IF FINI FLAG
801 003034* 001003 BNE ENDNC ;EQUALS ZERO THEN
802 003036* 016777 175332 175324 MOV VERIFY, PRXDE ;SEND VERIFY COMMAND
803 003044* 104403 000000* 005731* MSGS6, BEGIN, SDBLDN ;ASCII MESSAGE CALL WITH COMMON HEADER
804 003052* 004767 000262 ;WAIT FOR "DONE" & INTERRUPT
805 003056* 005767 175254 JSR PC, INTER
806 003062* 001070 TST FIN ;IF FINI FLAG
807 003064* 005767 175176 BNE ENDNC ;EQUALS ZERO THEN SET
808 003070* 001411 TST UT ;IF
809 003072* 005067 175170 HFO 45 ;UNIT 1 WAS DONE
810 003076* 012767 000260 002320 CLR UT ;SET TO DO UNIT 0
811 003104* 005767 175144 MOV #260, DNUM ;SET DRIVE # FOR PRINT
812 003110* 001001 TST UNIT0 ;IF UNIT 0
813 003112* 000733 HNE 48 ;IS AVAIL
814 003114* 012767 000000 175162 48: MOV #0, DENSITY ;SET DENSITY=SINGLE
815 003122* 000450 BR 38 ;BRANCH TO END
816 003124* 012767 000511 175130 56: 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, PRXCS ;SEND COMMAND
819 003146* 004767 001302 JSR PC, AWDN ;WAIT FOR "TR"
820 003152* 005767 175160 TST FIN ;IF FINI FLAG
821 003156* 001032 BNE ENDNC ;EQUALS ZERO THEN
822 003160* 016777 175210 175202 MOV VERIFY, PRXDE ;SEND VERIFY COMMAND
823 003166* 104403 000000* 005750* MSGS6, BEGIN, SDBLDN ;ASCII MESSAGE CALL WITH COMMON HEADER
824 003174* 004767 000140 ;WAIT FOR "DONE" & INTERRUPT
825 003200* 005767 175132 JSR PC, INTER
826 003204* 001017 TST FIN ;IF FINI FLAG
827 003206* 005767 175054 BNE ENDNC ;EQUALS ZERO THEN
828 003212* 001411 TST UT ;IF
829 003214* 005067 175046 HFO 68 ;UNIT 1 WAS DONE
830 003220* 012767 000260 002176 CLR UT ;SET TO DO UNIT 0
831 003226* 005767 175022 MOV #260, DNUM ;SET DRIVE # FOR PRINT
832 003232* 001001 TST UNIT0 ;IF UNIT 0
833 003234* 000733 HNE 58 ;IS AVAIL
834 003236* 012767 000400 175040 68: MOV #400, DENSITY ;SET DENSITY = DOUBLE
835 003244* 000207 ENDNC: RTS PC ;RETURN
836 ;-----
```

```

837 ;MOD 0,0 - UNIT DONE CHECK -----
838
839 UDACK: TST UTT ;IF UNIT 0
840 BNE 10 ;UNDER TEST, THEN
841 TST UNIT1 ;IF UNIT 1
842 BNE 10 ;AVAILABLE, THEN
843 CLR UTDONE ;CLEAR UNITS DONE FLAG
844 BR XUDNCK ;R TO MOD EXIT
845 MOV #1,UTDONE ;SET UNITS DONE FLAG
846 XUDNCK: RETURN
847
848 ;MOD U,V6 VECTOR SET UP -----
849
850 VSET: MOV VVECTOR,R0 ;SET UP INTERRUPT HANDLER ADDRESS
851 MOV #INTUPT,(R0)+ ;SET RR LEVEL
852 MOVRR RRI,(R0) ;SET ADDRESS OF RXCS
853 MOV ADDR,RXCS ;SET ADDRESS OF RXCS
854 MOV ADDR,RADR ;SET ADDRESS OF DATA BUFFER
855 ADD #2,RADR
856 RTS PC
857
858 ;MOD U,INTR INTERRUPT HANDLER -----
859
860 INTER: EXIT0,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
861
862 NTRUPT: ;
863 FIPOS,BEGIN,10 ;QUEUE UP TO CONTINUE AT 10 AND RTI
864 ;
865 JSP PC,SECK ;GO CHECK FOR ERROR
866 MOV VVECTOR,R0 ;GET VECTOR ADDRESS
867 TST (R0)+ ;RUMP POINTER
868 MOV BRI,(R0) ;ASSURE RESET TO BR LEVEL 5
869 RTS PC ;RETURN
870
871
872
873

```

```

874 ;MOD U,EC STATUS ERROR CHECK SUBROUTINE -----
875
876 SECK: CLR SEFPL ;CLEAR STATUS ERROR FLAG
877 RTS ;GET RXCS
878 TST CMD ;SEE IF ERROR
879 BMI 10 ;IF SOI BR
880 MOV #3,RTYFL ;SET RETRY FLAG, SO NO RETRY
881 ENDEC ;ELSE RETURN
882 CALL SUPTRG ;CALL SETUP PRINT REGISTERS
883 MOV UTT,CMD ;SELECT DPIPE
884 CALL DVEC ;GET DEVICE ERROR CODE
885 CALL CKDNPR ;CHECK FOR DENSITY ERROR
886 TST WTF ;SEE IF WRITE ERROR
887 BNE 20 ;IF SOI BR
888 CALL CKRDER ;CHECK READ ERROR
889 BR CL ;GO CLEAR ERROR
890 CALL CWTWP ;CALL CHECK WRITE ERROR
891 CL1 JSR PC,CLEAR ;CLEAR ERRORS
892 ENDEC: RTS PC ;RETURN
893
894 ;MOD U,CKDNPR - CHECK DENSITY ERRORS -----
895
896 CKDNPR: MOV SPXCS,R5 ;SETUP RXCS FOR TEST
897 BIC #177761,R5 ;CLEAR ALL BUT COMMAND
898 CMP #10,R5 ;IF COMMAND WAS
899 BNE XDNER ;SET DENSITY, THEN
900 MOV #1,FINI ;SET FINI FLAG
901 TST DEN ;IF
902 BNE 10 ;SINGLE DENSITY, THEN
903 MSGN0,BEGIN,SETSD ;ASCII MESSAGE CALL WITH COMMON HEADER
904 BR XINFR ;CONTINUE
905
906
907
908
909
910
911
912
913 ;MOD U,CKRDER - CHECK READ ERRORS -----
914
915 CKRDER: MOV #0,ERRTP
916 ;*****
917 SOFEX0,BEGIN,TABLE ;READ ERROR
918 ;*****
919 TST DEN ;IF DENSITY FLAG IS
920 BNE 10 ;EQUAL TO ZERO (SINGLE) THEN
921 MSGS0,BEGIN,RTSRD ;ASCII MESSAGE CALL WITH NO HEADER
922 BR 20 ;GO PRINT ERRORS
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

925 ;MOD U,CKWTER - CHECK WRITE ERRORS -----
926
927 003606* 012767 000000 174272 CKWTER: MOV #0,ERRTYP
928 ;*****
929 003614* 104406 000000* 004626* SGTFRS,REGIN,TABIE ;WRITE ERROR
930 ;*****
931 003622* 005767 174461* TST DFN ;IF DENSITY FLAG IS
932 003626* 001004 WNF 18 ;EQUAL TO ZERO (SINGLE) THEN
933 003630* 104402 000000* 005630* MSGSS,REGIN,WTERED ;ASCII MESSAGE CALL WITH NO HEADER
934 003636* 000403 RR 28 ;GO PRINT ERRORS
935 003640* 181 MSGSS,REGIN,WTERED ;ASCII MESSAGE CALL WITH NO HEADER
936 003640* 104402 000000* 005606* 281 RETURN ;RETURN
937 003646* 000207 ;*****
938
939 ;MOD U,CLR CLFAP ERRORS -----
940
941
942 003650* 005267 174460 CLEAR: INC SRFL ;SET STATUS ERROR FLAG
943 003654* 105767 174470 TSTB SPXSB ;SET IF P.E.C. ERROR
944 003660* 001421 REQ 28 ;IF NOT: RR
945 003662* 012767 000000 174372 MOV #0000,CMD ;SET COMMAND= INIT
946 003670* 016777 174366 174470 MOV CMD,PRXCS ;REINITIALIZE
947 003676* 004767 000442 181 JSP PC,AWDN ;GO AWAIT DONE
948 003702* 005777 174460 TST PRXCS ;SET IF ERROR ON INIT
949 003706* 100006 RPL 28 ;IF NOT: RR
950 003710* 104403 000000* 005466* MSGSS,REGIN,DRP3 ;ASCII MESSAGE CALL WITH COMMON HEADER
951 003716* 004767 000072 JSR PC,DRP ;GO DROP MODULE
952 003722* 000431 RR ENDCLR
953 003724* 005267 001500 281 INC RTYN ;BUMP RETRY COUNTER
954 003730* 016703 001474 MOV RTYN,R3 ;GET COUNTER
955 003734* 042703 177770 FIC #17770,R3 ;MARK ASCII
956 003740* 010367 174364 MOV R3,RTYFL ;SET RETRY FLAG
957 003744* 022703 000003 CMP #3,R3 ;SEE IF DONE RETRIES
958 003750* 001016 RNE ENDCLR ;IF NOT: RR
959 ;*****
960 003752* 104405 000000* 004626* HRDPRS,REGIN,TABIE ;RECLASSIFICATION OF SOFT ERR
961 ;*****
962 003760* 032767 000001 174030 BIT #1,SR1 ;SEE IF SHOULD DROP MODULE
963 003766* 001404 REQ 38 ;IF NOT: RR
964 003770* 012767 000001 174340 MOV #1,FIN ;GO DROP MODULE
965 003776* 000403 RR ENDCLR
966 004000* 012767 000260 001422 381 MOV #260,RTYN ;RESET RETRY COUNTER
967 004006* 005067 174336 ENDCLR: CLF SRXSH ;CLFAP ERROR
968 004012* 000207 RTS PC ;RETURN
969 ;*****
  
```

```

970 ;MOD U,DRP TEST DROP CLEAN-UP -----
971
972 004014* 004767 000242 DRP: CALL SHPTRG ;CALL SETUP PRINT REGISTERS
973 004020* 005067 174236 CIR CMD ;CLEAR COMMAND WORD
974 004024* 004767 000032 CALL DVEC ;GET DEVICE ERROR CODE
975 004030* 005767 174302 TST FIN ;IF FINI FLAG
976 004034* 010111 BNE ENTDOR ;EQUALS ZERO THEN
977 004036* 012767 000034 174042 281 MOV #34,ERRTYP
978 ;*****
979 004044* 104405 000000* 004626* HRDPRS,REGIN,TABIE ;INITIALIZE ERROR
980 ;*****
981 004052* 012767 000001 174256 MOV #1,FIN ;GO DROP MODULE
982 004060* 000207 ENTDOR: RTS PC
983 ;*****
984
985 ;MOD U,DVEC ----- GET DEVICE ERROR CODE -----
986
987 004062* 000240 DVEC: NOP
988 004064* 052767 000017 174170 RTS #17,CMD ;FOR ERROR CODE COMMAND
989 004072* 104415 000000* 000360* GETPAS,REGIN, DECVA ;GET PHYSICAL ADDRESS FROM 16-BIT DECVA
990 004100* 016701 174256 MOV DECVA+2,R1 ;GET DEF F.C. PHYS ADDR
991 004104* 016702 174254 MOV DECVA+4,R2 ;GET DEF F.C. XTEND ADDR BITS
992 004110* 000302 SWAP R2
993 004112* 050267 174144 RIS P2,CMD ;SET EXTEND ADDR BITS
994 004116* 016777 174140 174242 MOV CMD,PRXCS ;SEND COMMAND
995 004124* 005267 174176 INC RKPYP5 ;SET PREFAX BYPASS FLAG
996 004130* 004767 000320 JSR PC,AWTR ;THEN GO AWAIT TR
997 004134* 005067 174166 CLR RKPYP5 ;CLEAR HREAX BYPASS FLAG
998 004140* 005767 174172 TST FIN ;IF FINI FLAG
999 004144* 001006 RNE ENDVEC ;EQUALS ZERO THEN
1000 004146* 010177 174216 MOV R1,PRXDR ;SET BASE ADDR FOR LOAD EXT ERR RFG.
1001 004152* 004767 000166 JSR PC,AWDN ;GO AWAIT DONE
1002 004156* 004767 000002 CALL CMXER ;CALL CHECK NON-EXIST MEMOPY ERROR
1003 004162* 000207 EDDVFC: RTS PC ;RETURN
1004 ;*****
  
```

```

1005
1006
1007
1008 004164* 032777 004000 174176 CKNXER: BIT #4000,0RXDB ;IF NONEXISTENT MEMORY
1009 004172* 001432 BEQ XNXFR ;ERROR, THEN
1010 004174* 004767 000062 CALL SUPTRG ;CALL SETUP PRINT REGISTERS
1011 004200* 012777 040000 174160 MOV #40000,0RXCS ;INITIALIZE DEVICE (TO CLEAR ERROR)
1012 004206* 012767 000010 173672 MOV #10,ERRTYP ;SET NON-EXISTENT MEMORY ERROR
1013
1014 004214* 104405 000000* 000000 HDRERS,BEGIN, NULL ;NON-EXISTENT MEMORY ERROR
1015
1016 004222* 104402 005764* 000000* MSGS0,0XERMS,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 XNXER ;ERROR COUNT#3, THEN
1020 004244* 012767 000001 174064 101 MOV #1,FIN ;SET FINI FLAG
1021 004252* 104402 000000* 005504* MSGS0,BEGIN,DPP5 ;ASCII MESSAGE CALL WITH NO HEADER
1022 004260* 000207 XNXFR: RETURN ;RETURN
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
;MOD U,SUPG - SETUP PRINT REGISTERS -----
SUPTRG: MOV 0RXCS,0RXCS ;GET RXCS
BIT CMD,0RXCS ;SET 0RXCS TO SHOW COMMAND
MOV 0RXDB,0RXES ;GET RXES
MOV SA,0RXSA ;GET SECTION ADDRESS
MOV TA,0RXTA ;SET TRACK ADDRESS
MOV 0RXCS,0ACSR ;SET CONTENTS OF RXCS
MOV 0RXCS,0CSRA ;SET ADDRESS OF RXCS
MOV 0RXES,0ASTAT ;GET RXES
RETURN
;-----

```

```

1037
1038
1039 004344* 005067 173740 AWDN: CLR TOMLT ;PRESET TIME OUT MULTIPLIER
1040 004350* 005067 173736 101 CLR TOCNT ;PRESET TIME OUT COUNTER
1041 004354* 032777 000000 174004 201 BIT #40,0RXCS ;SEE IF DONE SET
1042 004362* 001033 RNE 30 ;IF 001 BR
1043 004364* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
1044 004370* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1045 004374* 005267 173712 INC TOCNT ;BUMP TIME OUT COUNTER
1046 004400* 001365 BNE 20 ;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 10 ;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 HDRERS,BEGIN, NULL ;DONE BIT TIME OUT
1054
1055 004436* 104402 000000* 005554* MSGS0,BEGIN,DPP5 ;ASCII MESSAGE CALL WITH NO HEADER
1056 004444* 012767 000001 173664 MOV #1,FIN ;DROP MODULE
1057 004452* 000207 RTS PC ;EXIT
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
;MOD U,WTR AWAIT TRANSFER READY SUBROUTINE -----
AWTR: CLR TOMLT ;PRESET TIMEOUT MULTIPLIER
101 CLR TOCNT ;PRESET TIME OUT COUNTER
201 BIT #40,0RXCS ;IF DONE BIT NOT
RNE 40 ;SET, THEN
BIT #200,0RXCS ;SEE IF TRANSFER READY SET
BNE 60 ;IF 001 BR
TST 0RKBYP5 ;IF BREAK BYPASS FLAG
RNE 30 ;NOT SET, THEN
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
INC TOCNT ;BUMP TIME OUT COUNTER
BNE 20 ;IF NOT TIMED OUT BR
INC TOMLT ;INCREMENT TIMEOUT MULTIPLIER
CMP #2,TOMLT ;IF ON 2ND
BHI 10 ;TIMEOUT PASS, THEN
CALL SUPTRG ;CALL SETUP PRINT REGISTERS
MOV #40,ERRTYP
;-----
HDRERS,BEGIN, NULL ;TRANSFER READY TIME OUT
;-----
MOV #1,FIN ;GO DROP MODULE
BIT #40,0RXCS ;IF DONE BIT
BEQ 50 ;SET, THEN
MSGS0,BEGIN,TRONER ;ASCII MESSAGE CALL WITH NO HEADER
RR 60 ;EXIT
;-----
MSGS0,BEGIN,TRIO ;ASCII MESSAGE CALL WITH NO HEADER
101 CLR 0RKBYP5 ;CLEAR BREAK BYPASS FLAG
601 RTS PC ;RETURN
;-----

```



```

1092                                     ;---- MESSAGE TABLE -----
1093                                     TABLE:
1094 004626* 000340* ARXCS: SPXCS
1095 004630* 000342* ARXES: SPXFS
1096 004632* 000344* ARXTA: SPXTA
1097 004634* 000346* ARXSA: SPXSA
1098 004636* 000350* ARXSB: SPXSB
1099 004640* 000352* ARXSB1: SPXSB1
1100 004642* 000354* ARXSB2: SPXSB2
1101 004644* 000356* ARXSB3: SPXSB3
1102 004646* 177777 -1
1103                                     ;-----
1104                                     ;---- MESSAGES -----
1105                                     MSG1: .ASCIZ 'RXG1 SUBSYSTEM'
1106
1107
1108 004650* 054122 030460 051440 MSG1: .ASCIZ 'TIME OUT'
1109 004656* 041125 054523 052123 MSG2: .ASCIZ 'DRIVE '
1110 004660* 046505 000 MSG3: .ASCIZ 'DROP MODULE'
1111 004667* 040 044524 042515 MSG4: .ASCIZ 'INITIALIZE'
1112 004674* 047440 052125 000 MSG5: .ASCIZ 'NO UNITS TO TEST'
1113 004701* 040 051104 053111 MSG6: .ASCIZ '% REINITIALIZE'
1114 004706* 020105 000 MSG7: .ASCIZ '% SET -> WAITING ON'
1115 004711* 040 037055 051104 MSG8: .ASCIZ 'FILL BUFFER'
1116 004716* 050117 046440 042117 MSG9: .ASCIZ 'SINGLE'
1117 004724* 046125 022505 000 MSG10: .ASCIZ 'DOUBLE'
1118 004731* 040 047111 052111 MSG11: .ASCIZ 'DONE BIT'
1119 004736* 040511 044514 042537 MSG12: .ASCIZ 'TRANSFER READY'
1120 004744* 000 MSG13: .ASCIZ 'DENSITY'
1121 004745* 040 047516 052440 MSG14: .ASCIZ 'ERROR- RETRY'
1122 004752* 044516 051524 052040 MSG15: .ASCIZ '% IF INTERRUPTED-> MAY NEED REFORMAT'
1123 004760* 020117 042524 052123 MSG16: .ASCIZ '% NONEXISTENT MEMORY ERROR - INITIALIZE DEVICE'
1124 004766* 000 MSG17: .ASCIZ 'EVEN'
1125 004767* 045 051040 044505 MSG18: .ASCIZ 'O'
1126 004774* 044516 044524 046101 MSG19: .ASCIZ 'EVEN'
1127 005002* 055111 000105 MSG20: .ASCIZ 'O'
1128 005006* 051440 052105 026440 MSG21: .ASCIZ 'EVEN'
1129 005014* 020076 040527 052111 MSG22: .ASCIZ 'EVEN'
1130 005022* 047111 020107 047117 MSG23: .ASCIZ 'EVEN'
1131 005030* 000 MSG24: .ASCIZ 'EVEN'
1132 005031* 040 044506 046114 MSG25: .ASCIZ 'EVEN'
1133 005036* 041040 043125 042506 MSG26: .ASCIZ 'EVEN'
1134 005044* 000122 047111 046107 MSG27: .ASCIZ 'EVEN'
1135 005046* 051440 047111 046107 MSG28: .ASCIZ 'EVEN'
1136 005054* 000105 MSG29: .ASCIZ 'EVEN'
1137 005056* 042040 052517 046102 MSG30: .ASCIZ 'EVEN'
1138 005064* 000105 MSG31: .ASCIZ 'EVEN'
1139 005066* 042040 047117 020105 MSG32: .ASCIZ 'EVEN'
1140 005074* 044502 000124 MSG33: .ASCIZ 'EVEN'
1141 005100* 052040 040527 051516 MSG34: .ASCIZ 'EVEN'
1142 005106* 042506 020122 042522 MSG35: .ASCIZ 'EVEN'
1143 005114* 042101 000131 MSG36: .ASCIZ 'EVEN'
1144 005120* 042040 047105 044523 MSG37: .ASCIZ 'EVEN'
1145 005126* 054524 000 MSG38: .ASCIZ 'EVEN'
1146 005131* 040 051105 047522 MSG39: .ASCIZ 'EVEN'
1147 005136* 026522 051040 052105 MSG40: .ASCIZ 'EVEN'

```

```

1148 005144* 054522 000072 MSG41: .ASCIZ 'O'
1149 005150* 020045 042523 000124 MSG42: .ASCIZ 'O'
1150 005156* 020045 051127 052111 MSG43: .ASCIZ 'O'
1151 005164* 000105 MSG44: .ASCIZ 'O'
1152 005166* 020045 042522 042101 MSG45: .ASCIZ 'O'
1153 005174* 000 MSG46: .ASCIZ 'O'
1154 005175* 040 046505 052120 MSG47: .ASCIZ 'O'
1155 005202* 020131 052502 043106 MSG48: .ASCIZ 'O'
1156 005210* 051105 000 MSG49: .ASCIZ 'O'
1157 005213* 040 047516 020124 MSG50: .ASCIZ 'O'
1158 005220* 053101 040511 040514 MSG51: .ASCIZ 'O'
1159 005226* 046102 000105 MSG52: .ASCIZ 'O'
1160 005232* 042440 051122 051117 MSG53: .ASCIZ 'O'
1161 005240* 000 MSG54: .ASCIZ 'O'
1162 005241* 045 026440 051476 MSG55: .ASCIZ 'O'
1163 005246* 052105 044524 043516 MSG56: .ASCIZ 'O'
1164 005254* 042040 051511 042513 MSG57: .ASCIZ 'O'
1165 005262* 052124 020105 047524 MSG58: .ASCIZ 'O'
1166 005270* 000 MSG59: .ASCIZ 'O'
1167 005271* 045 020040 044440 MSG60: .ASCIZ 'O'
1168 005276* 020106 047111 042524 MSG61: .ASCIZ 'O'
1169 005304* 051122 050125 042524 MSG62: .ASCIZ 'O'
1170 005312* 026504 020076 040515 MSG63: .ASCIZ 'O'
1171 005320* 020131 042516 042105 MSG64: .ASCIZ 'O'
1172 005326* 051040 043105 051117 MSG65: .ASCIZ 'O'
1173 005334* 040515 022524 000 MSG66: .ASCIZ 'O'
1174 005341* 045 000 MSG67: .ASCIZ 'O'
1175 005343* 055 020076 047516 MSG68: .ASCIZ 'O'
1176 005350* 042516 044530 052123 MSG69: .ASCIZ 'O'
1177 005356* 047105 020124 042515 MSG70: .ASCIZ 'O'
1178 005364* 047515 054522 042440 MSG71: .ASCIZ 'O'
1179 005372* 051122 051117 026440 MSG72: .ASCIZ 'O'
1180 005400* 044440 044516 044524 MSG73: .ASCIZ 'O'
1181 005406* 046101 055111 020105 MSG74: .ASCIZ 'O'
1182 005414* 042504 044526 042503 MSG75: .ASCIZ 'O'
1183 005422* 000 MSG76: .ASCIZ 'O'
1184 005424* 030040 000 MSG77: .ASCIZ 'O'
1185 005430* 000 MSG78: .ASCIZ 'O'
1186 005430* 030040 000 MSG79: .ASCIZ 'O'
1187 005434* 000 MSG80: .ASCIZ 'O'
1188
1189

```

1190
1191 005434 005175
1192 005436 005232
1193 005440 177777
1194 005442 005031
1195 005444 005232
1196 005446 177777
1197 005450 004731
1198 005452 005232
1199 005454 004711
1200 005456 177777
1201 005460 004745
1202 005462 004711
1203 005464 177777
1204 005466 004767
1205 005470 005232
1206 005472 004711
1207 005474 177777
1208 005476 004650
1209 005500 004711
1210 005502 177777
1211 005504 004711
1212 005506 177777
1213 005510 004701
1214 005512 005424
1215 005514 005213
1216 005516 177777
1217 005520 004701
1218 005522 005424
1219 005524 005150
1220 005526 005046
1221 005530 005120
1222 005532 005232
1223 005534 177777
1224 005536 004701
1225 005540 005424
1226 005542 005150
1227 005544 005056
1228 005546 005120
1229 005550 005232
1230 005552 177777
1231 005554 005066
1232 005556 004667
1233 005560 004711
1234 005562 177777
1235 005564 005100
1236 005566 004667
1237 005570 004711
1238 005572 177777
1239 005574 005066
1240 005576 005006
1241 005600 005100
1242 005602 004711
1243 005604 177777
1244 005606 004701
1245 005610 005424

----- FORMATTED MESSAGES -----
BFR1: MSG21 ;"EMPTY BUFFER"
MSG23 ;"ERROR"
-1
BFR2: MSG9 ;"FILL BUFFER"
MSG23 ;"ERROR"
-1
DRP1: MSG5 ;"INITIALIZE"
MSG23 ;"ERROR"
MSG4 ;"DROP MODULE"
-1
DRP2: MSG6 ;"NO UNITS TO TEST"
MSG4 ;"DROP MODULE"
-1
DRP3: MSG7 ;"REINITIALIZE"
MSG23 ;"ERROR"
MSG4 ;"DROP MODULE"
-1
DRP4: MSG1 ;"RX01 SUBSYSTEM"
MSG4 ;"DROP MODULE"
-1
DRP5: MSG4 ;"DROP MODULE"
-1
MIINT: MSG3 ;"DRIVE"
DNUM ;" "
MSG22 ;"NOT AVAILABLE"
-1
SETSDE: MSG3 ;"DRIVE"
DNUM ;" "
MSG18 ;"SET"
MSG11 ;"SINGLE"
MSG15 ;"DENSITY"
MSG23 ;"ERROR"
-1
SEIDDE: MSG3 ;"DRIVE"
DNUM ;" "
MSG18 ;"SET"
MSG12 ;"DOUBLE"
MSG15 ;"DENSITY"
MSG23 ;"ERROR"
-1
DNT0: MSG13 ;"DONE BIT"
MSG2 ;"TIME OUT"
MSG4 ;"DROP MODULE"
-1
TRT0: MSG14 ;"TRANSFER READY"
MSG2 ;"TIME OUT"
MSG4 ;"DROP MODULE"
-1
TPDNEP: MSG13 ;"DONE BIT"
MSG9 ;"SET -> WAITING ON"
MSG14 ;"TRANSFER READY"
MSG4 ;"DROP MODULE"
-1
WTERDD: MSG3 ;"DRIVE"
DNUM ;" "

1246 005612 005156
1247 005614 005056
1248 005616 005120
1249 005620 005131
1250 005622 005430
1251 005624 005341
1252 005626 177777
1253 005630 004701
1254 005632 005424
1255 005634 005156
1256 005636 005046
1257 005640 005120
1258 005642 005131
1259 005644 005430
1260 005646 005341
1261 005650 177777
1262 005652 004701
1263 005654 005424
1264 005656 005166
1265 005660 005056
1266 005662 005120
1267 005664 005131
1268 005666 005430
1269 005670 005341
1270 005672 177777
1271 005674 004701
1272 005676 005424
1273 005700 005166
1274 005702 005046
1275 005704 005120
1276 005706 005131
1277 005710 005430
1278 005712 005341
1279 005714 177777
1280 005716 004701
1281 005720 005424
1282 005722 005241
1283 005724 005046
1284 005726 005120
1285 005730 005271
1286 005732 177777
1287 005734 004701
1288 005736 005424
1289 005740 005241
1290 005742 005046
1291 005744 005120
1292 005746 177777
1293 005750 004701
1294 005752 005424
1295 005754 005241
1296 005756 005056
1297 005760 005120
1298 005762 177777
1299 005764 005343
1300 005766 177777
1301

MSG19 ;"WRITE"
MSG12 ;"DOUBLE"
MSG15 ;"DENSITY"
MSG17 ;"ERROR-RETRY:"
RTYN ;" "
MSG26 ;"CRLF"
-1
WTEPSD: MSG3 ;"DRIVE"
DNUM ;" "
MSG19 ;"WRITE"
MSG11 ;"SINGLE"
MSG15 ;"DENSITY"
MSG17 ;"ERROR-RETRY:"
RTYN ;" "
MSG26 ;"CRLF"
-1
PTEPDD: MSG3 ;"DRIVE"
DNUM ;" "
MSG20 ;"READ"
MSG12 ;"DOUBLE"
MSG15 ;"DENSITY"
MSG17 ;"ERROR-RETRY:"
RTYN ;" "
MSG26 ;"CRLF"
-1
WTERSDD: MSG3 ;"DRIVE"
DNUM ;" "
MSG20 ;"READ"
MSG11 ;"SINGLE"
MSG15 ;"DENSITY"
MSG17 ;"ERROR-RETRY:"
RTYN ;" "
MSG26 ;"CRLF"
-1
SETSDN: MSG3 ;"DRIVE"
DNUM ;" "
MSG24 ;"SETTING DISKETTE TO"
MSG11 ;"SINGLE"
MSG15 ;"DENSITY"
MSG25 ;"IF INTERRUPTED-MAY NEED REFORMAT"
-1
SSGLDN: MSG3 ;"DRIVE"
DNUM ;" "
MSG24 ;"SETTING DISKETTE TO"
MSG11 ;"SINGLE"
MSG15 ;"DENSITY"
-1
SDBLDN: MSG3 ;"DRIVE"
DNUM ;" "
MSG24 ;"SETTING DISKETTE TO"
MSG12 ;"DOUBLE"
MSG15 ;"DENSITY"
-1
NXERMS: MSG27 ;"NON EXISTENT MEMORY ERROR"
-1

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

----- DATA BUFFERS -----
 ,EVEN
 BRUF: 0 READ BUFFER
 ,=,+400

 ,END

ACSR	000102R	293*	1032*																
ADDR	000000R	250*	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*																
AKAS	000110R	298*																	
ANDN	004344R	530	585	64*	725	747	769	780	947	1001	1039*								
AWTP	001454R	551	555	575	580	606	610	630	635	775	794	819	996	1062*					
REGIN	000000R	256*	461	466	485	486	495	524	535	594	596	651	653	656					
		712	734	736	756	779	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*																	
BKXPS	000326R	346*	579*	581*	634*	636*	995*	997*	1068	1089*									
BKLOOP	001002R	478	484	489	492	497*													
BREAKS	104407	322*	1043	1044	1070	1071													
BR1	000012R	261*	853	871															
BR2	000013R	262*																	
BTODS	104421	322*																	
CDATAS	104412	322*	656																
CDERCT	000144R	312*																	
CDWDCT	000146R	313*																	
CKDNFR	003472R	885*	897*																
CKNXXE	004164R	588*	645*	1002*	1008*														
CKRDER	003544R	880*	913*																
CKWTER	003606R	890*	927*																
CL	003464R	889	891*																
CLEAR	003654R	891	942*																
CMD	000262R	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*					

SEU 5033

[illegible]

[illegible][illegible]

WDIO	000114P	300#				
WPIF	000740P	407#				
WTERDD	005600P	936	1244#			
WTERSD	005630P	933	1253#			
WTF	000270P	331#	542#	564#	886	
XDNER	003542P	904	905	908#		
XDNSET	002732P	771	777	782#		
XFLAG	000005P	258#				
XNXER	004260P	1009	1019	1022#		
XUDNCK	003276P	844	846#			
.	0006372P	365	366	657	1184#	1186# 1188# 1306#

. ABS. 000000 000
 006372 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

RXRAC0, RXRBC0/SOL/CRF:SYN=DDXCOM, RXRAC0
 RUN=TIME: 7 12 1 SECONDS
 RUN=TIME RATIO: 90/21=4.2
 CORE USED: 9K (17 PAGES)