

.REM !

IDENTIFICATION

PRODUCT CODE: AC-E736E-MC
PRODUCT NAME: CXRXAEO DEC/X11 RX01 FLOPPY DISK MODULE
DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1975,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RXA IS AN IOMOD THAT EXERCISES TWO RX01 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 RX01 CONTROLLER

STORAGE:: RXA REQUIRES:

1. DECIMAL WORDS: 1091
2. OCTAL WORDS: 02103
3. OCTAL BYTES: 4206

3. PASS IDENTIFICATION

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

THREE CYCLES OF THE DISKETTE ARE MADE FOR EACH PASS:

1. STARTS AT SECTOR 1/TRACK 1
2. STARTS AT SECTOR 2/TRACK 1
3. STARTS AT SECTOR 3/TRACK 1

END OF PASS
RESTART SECTOR CYCLE AT SECTOR 1/TRACK 2

4. EXECUTION TIME

ONE PASS OF RXA RUNNING ALONE ON THE PDP-11/05 TAKES APPROXIMATELY .75 MINUTES FOR 2 DRIVES.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177170, VECTOR: 264, BR1: 5, DEVCNT: 2

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

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

7. MODULE OPERATION

TEST SEQUENCES:

- A. SETUP DRIVE REGISTER ADDRESSES AND MODULE VARIABLES
- B. SELECT DRIVES FOR TEST - IF NONE AVAILABLE, DROP MODULE
- C. INITIALIZE DRIVES
- D. SELECT A DRIVE
- E. WRITE DRIVE - IF ERROR, REPORT AND
RETRY UP TO RETRY LIMIT.
- F. READ DRIVE - IF ERROR, REPORT AND
RETRY UP TO RETRY LIMIT.
- G. DO DATA COMPARE FOR THE READS - IF ERROR, REPORT
- H. IF NOT DONE ALL DRIVES - GO TO D
- I. IF END OF PASS, REPORT AND GO TO B
- J. ELSE UPDATE STARTING ADDRESS, GO TO D

8. OPERATION OPTIONS

- SR1 BIT 0 CLEAR(0):
IF RETRY LIMIT IS EXCEEDED, CONTINUE WITH NEXT TEST.
- SR1 BIT 0 SET(1):
IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, REPORT A
HARD ERROR AND DROP THE MODULE.
- SR1 BIT 1 CLEAR(0):
USE ALTERNATING DATA PATTERN OF ONES AND ZEROS.
- SR1 BIT 1 SET(1):
USE DECREMENTING DATA PATTERN STARTING WITH
A RANDOM NUMBER.

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 RX01 REGISTER IN THE FOLLOWING ORDER.
 - RXCS (COMMAND REGISTER)
 - RXES (ERROR REGISTER)
 - RXSB (STATUS REGISTER)
 - RXTA (TRACK ADDRESS)
 - RXSA (SECTOR ADDRESS)
- C. DATA ERRORS REPORT UP TO 3 BAD WORDS, THEN DISCONTINUE CHECK
- D. RETRIES: EACH WRITE OR READ STATUS ERROR IS ACCOMPANIED BY A RETRY NUMBER:
 - RETRY 0: IS THE ORIGINAL ERROR
 - RETRY 1: IS THE FIRST RETRY OF THAT ERROR (SAME ADDRESS)
 - RETRY 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

!

```
000000- IOMOD <RXAE> 177170,264,5,0,2,40,67
000000- MODULE 140000,RXAE,177170,264,5,0,2,40,67
, TITLE RXAE DEC/X11 SYSTEM EXERCISER MODULE
DDCOM VERSION 6 23-MAY-78
.LIST BIN
*****
000000- BEGIN
000000- 054122 042501 040 MODNAM: .ASCII /RXAE / ;MODULE NAME
000005- 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006- 177170 ADDR: 177170+0 ;1ST DEVICE ADDR.
000010- 000264 VECTOR: 264+0 ;1ST DEVICE VECTCR.
000012- 240 BR1: .BYTE PRTY5+0 ;1ST BR LEVEL.
000013- 000 BR2: .BYTE PRTY0+0 ;2ND BR 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- 140000 STAT: 140000 ;STATUS WORD
000030- 000224 INIT: START ;MODULE START ADDR.
000032- 000224 SPOINT: MODSP ;MODULE STACK POINTER.
000034- 000000 PASCNT: 0 ;PASS COUNTER.
000036- 000040 ICNT: 40 ;# OF ITERATIONS PER PASS=40
000040- 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
000042- 000000 HROCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044- 000000 SDFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046- 000000 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050- 000000 SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052- 000000 RANUM: 0 ;# OF SYS ERRORS ACCUMULATED
000054- 000000 CONFID: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056- 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060- 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062- 000000 SVR0: OPEN ;LOC TO SAVE R0.
000064- 000000 SVR1: OPEN ;LOC TO SAVE R1.
000066- 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070- 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072- 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074- 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076- 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100- 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102- 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR
000104- 000000 BASADR: OPEN ;ADDR OF BAD DATA, OR
000104- 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
000106- 000000 ERRTP: OPEN ;TYPE OF ERROR
000106- 000000 ASI: OPEN ;EXPECTED DATA.
000110- 000000 AMAS: OPEN ;ACTUAL DATA.
000112- 000354 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
000114- 000000 WDRS: OPEN ;WORDS TO MEMORY PER ITERATION
000116- 000000 WDRP: OPEN ;WORDS FROM MEMORY PER ITERATION
000120- 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122- 000067 IDNUM: 67 ;MODULE IDENTIFICATION NUMBER=67
000040 .REPT SPSIZ ;MODULE STACK STARTS HERE.
.LIST
```

```
000224- .WORD 0
.LIST
MODSP: .ENDR
*****
```

```

245
246
247 000224 012767 000100 177662 START: MOV #64,WDTC ;64 WORDS TO NEW ITERATION
248 000224 012767 000100 177662 MOV #64,WDTC ;64 WORDS FROM NEW ITERATION
249 000240 012767 000002 177682 MOV #2,INTR ;2 INTERRUPTS/ITERATION
250 000246 012767 000001 002522 MOV #1,TASAV ;PRESET TRACK ADDRESS SAVE
251 000254 012767 000000 002516 MOV #1,TASAV ;PRESET SECTOR ADDRESS
252 000260 012767 000000 002516 MOV #1,TASAV,TA ;
253 000260 012767 000004 002476 MOV #1,SASAV,SA ;
254 000276 012767 000001 002476 MOV #1,PCNTR ;PRESET INTERNAL PASS COUNTER
255 000304 012767 000001 002474 MOV #1,WFLG ;ONE INTERNAL PASS WITH DATA = 0
256 000312 012767 177476 002430 MOV DVID1,DVIDIX ;GET DRIVE COUNT
257 000312 012767 000010 000041 BIT #10,BM41 ;CHECK IF FLCPY WAS LOAD MEDIUM
258 000326 000102 000000 000000 BNE RESTR1 ;BR IF NO
259 000330 042767 000001 002412 BIC #1,DVIDIX ;KILL TEST FOR DRIVE 0
260 000336 032767 000001 177450 BEQ RESTR1 ;HAS IT TO BE TESTED?
261 000346 104403 000000 003556 MSGNS,BEGIN,NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER
262
263 000354 016706 177452 RESTR1: MOV SPOINT,SP ;SET STACK POINTER
264 000354 012700 002723 MOV #UNITS,RO ;GET START OF TABLE
265 000364 012701 000010 MOV #10,R1 ;SET SIZE OF TABLE
266 000370 005020 CFC: CLR (R0)+ ;CLEAR TABLES
267 000374 005304 DBC R1 ;IF DONE
268 000376 004767 000646 BNE CFC ;IF NOT: BR
269 JSR PC,VSET ;GC SETUP ADDRESS AND VECTOR
270
271 000402 012777 040000 002400 MOV #40000,SRXCS ;INIT UNIT 0
272 000410 004767 002130 JSR PC,ANDN ;GC AWAIT DONE
273 000414 005777 002370 TST SRXCS ;SEE IF ERROR ON INIT
274 000420 100056 BPL TGO ;IF NOT: BR
275 000426 014403 000000 003530 MSGNS,BEGIN,DRP1 ;ASCII MESSAGE CALL WITH COMMON HEADER
276 000436 052767 040000 002354 TDPP: MOV #40000,SRXCS ;GET SRXCS
277 000444 017767 002342 BIS #40000,SRXCS ;SET SRXCS TC SHOW INIT COMMAND
278 000450 016767 002350 MOV SRXDB,SRXCS ;GET RATES
279 000456 016767 002350 CLR SRXSA ;SET TRACK ADDRESS
280 000464 016767 002330 MOV SRXCS,ACSR ;SET CONTENTS OF SRXCS
281 000472 016767 002312 MOV SRXCS,CSRA ;SET ADDRESS OF SRXCS
282 000500 016767 000119 MOV SRXCS,STAT ;GET ERROR
283 000514 004767 002024 JSR PC,ANDN ;READ STATUS B
284 000520 017767 002266 MOV SRXDB,SRXSB ;GC AWAIT DONE
285 000526 012767 000034 BEQ RESTR1 ;GET STATUS B
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
000534 104405 000000 003032 HDRERS,BEGIN,TABLE ;INITIALIZE ERROR
000542 000167 000306 JMP FINI ;GC DROP MODULE
LDROP:
000546 104403 000000 003556 MSGNS,BEGIN,NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER
000552 004767 000001 002164 TGO: BR ;GC Deselect UNIT 0
000564 001423 000000 002214 UNT0: MOV #1,SRXCS ;SEE IF INIT 0
000574 004767 000133 002214 JSR PC,ANDN ;IF NOT: BR
;READ STATUS OF UNIT 0
;GC AWAIT DONE

```

```

301 000600 017700 002206 MOV SRXDB,RO ;GET RXES
302 000604 032700 000200 BIT #20,RO ;IF SO: BR
303 000610 001014 000000 003556 UNT1: MOV #1,UNIT1 ;ASCII MESSAGE CALL WITH COMMON HEADER
304 000612 104403 000000 177260 MSGNS,BEGIN,NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER
305 000620 012767 000006 177260 MOV #6,ERRTYP ;NOT AVAILABLE
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****
312 *****
313 *****
314 *****
315 *****
316 *****
317 *****
318 *****
319 *****
320 000712 104405 000000 000000 HDRERS,BEGIN,NULL ;UNIT 0 NOT AVAILABLE
321 *****
322 *****
323 *****
324 *****
325 *****
326 *****
327 *****
328 *****
329 *****
330 *****
331 *****
332 *****
333 *****
334 *****
335 *****
336 *****
337 *****
338 *****
339 *****
340 *****
341 *****
342 *****
343 *****
344 *****
345 *****
346 *****
347 *****
348 *****
349 *****
350 *****
351 001060 012700 000100 DSUP: MOV #100,RO ;SET SIZE OF BUFFER
352 001064 003609 003609 MOV #BUF,R1 ;SET START OF BUFFER
353 001070 005767 001712 TST WFLG ;INTERNAL PASS DATA = ZERO
354 001074 001402 BEQ IS ;ALLREADY DONE:BR
355 001076 005003 CLR R3 ;YES LOAD BUFFER
356 001100 000417 BR #45 ;WITH ZEROS

```

```

RXAEO-P11 12-OCT-78 12:16
357 001102 032767 000002 176706 1S: BTT #2,SR1 ;+ USE RANDOM DATA?
358 001116 001411 ;+ NO: BRANCH
359 001116 001411
360 001116 016703 176732
361 001124 010321
362 001124 005303
363 001124 005300
364 001130 001374
365 001130 000207
366 001140 012703 125125 3S: MOV #1,25125,R3 ;+ WHEN FULL LEAVE
367 001140 012703 ;+ USE ALTERNATING
368 001144 005300 4S: MOV #1,(R1)+ ;+ ONES AND ZEROS TO
369 001144 001375 ;+ LOAD THE BUFFER
370 001146 000207 ;+ IF NOT: BR
371
372 ;SELECT UNIT FOR TEST*****
373
374 001150 005767 001602 SELD: TST DRVN ;SEE IF DRIVE 0
375 001150 001411 ;IF SO: BR
376 001156 005767 001572 TST UNIT1 ;SEE IF UNIT 1 AVAILABLE
377 001162 001403 ;IF SO: BR
378 001162 005087 001566 CLR DRVN ;ELSE SET RO 0
379 001172 001276 000020 001562 1S: MOV #20,UTT ;SET UNIT UNDER TEST
380 001200 012767 000261 002246 MOV #260,DNUM ;SET DRIVE NUMBER FOR PRINTS
381 001206 005087 001544 CLR DRVN ;CLEAR DRIVE NUMBER
382 001214 005767 001532 2S: TST UNIT0 ;SEE IF UNIT 0 AVAILABLE
383 001220 001403 ;IF SO: BR
384 001220 005167 001530 COM DRVN ;SWITCH UNITS
385 001224 005167 001522 3S: SELD ;SELECT UNIT
386 001230 005167 001522 COM DRVN ;SWITCH UNITS
387 001234 005087 001522 CLR UTT ;SELECT DRIVE 0
388 001240 002767 000260 002206 MOV #260,DNUM ;SET DRIVE NUMBER FOR PRINTS
389 001246 000207 ;RETURN
390
391 ;VECTOR SET UP
392
393 001250 016700 176534 VSET: MOV VECTOR,RO ;SET UP INTERRUPT HANDLER ADDRESS
394 001254 001770 001716 MOV #INTPT,(RO)+ ;SET BR LEVEL
395 001260 016710 176526 MOVB ADDR,RXCS ;SET ADDRESS OF RXCS
396 001270 005287 001516 MOV ADDR,RXDB ;SET ADDRESS OF RXCS
397 001272 016767 176510 001512 MOV ADDR,RXDB ;SET ADDRESS OF DATA BUFFER
398 001300 002767 000002 001504 ADD #2,RXDB ;SET ADDRESS OF DATA BUFFER
399 001306 000207
400
401 ;TRACK AND SECTOR UPDATE*****
402
403 001310 062767 000003 001456 TSS: ADD #3,SA ;BUMP SECTOR ADDRESS
404 001316 022767 000033 001450 CMP #35,SA ;SEE IF DONE SECTORS
405 001324 0101401 2S: BLOS #2 ;IF SO: BR
406 001324 000470 BR #106 ;RETURN
407 001330 005287 001444 001436 MOV #1,SAV,SA ;SET SA
408 001336 062767 000012 001426 ADD #12,TA ;INCREMENT TRACK ADDRESS

```

```

RXAEO-P11 12-OCT-78 12:16
413 001344 022767 000114 001420 CMP #114,TA ;SEE IF DONE TRACKS
414 001352 103056 BHS #106 ;IF NOT: BR
415 001354 016767 001416 001410 MOV TASAV,TA ;RESET TRACK ADDRESS
416 001362 005287 001414 INC PCNTR ;BUMP INTERNAL STARTING SECTOR
417 001372 001411 TST #FLG ;IS INTERNAL SPECIAL DATA PATTERN?
418 001372 001411 BEQ #15 ;+NO:CONTINUE
419 001374 005087 001406 4S: CLR #WFLG ;+YES:HAVE CYCLED THRU BOLTH DRIVES ONCE
420 001406 012767 000001 001372 MOV #1,SASV ;+NOW RESET POINTERS TO WRITE
421 001406 012767 000001 001362 MOV #1,TASV ;+SAME PLACES WITH REGULAR PATTERN
422 001414 000427 BR #9S
423 001416 022767 000004 001356 51S: CMP #4,PCNTR ;SEE IF DONE THREE
424 001426 001407 ;IF SO: BR
425 001426 001407 MOV PCNTR,SASV ;ELSE BUMP STARTING SECTOR ADDRESS
426 001434 000417 BR #9S ;SAVE TA AND SA AND LEAVE
427 001436 012767 000001 001336 6S: MOV #1,PCNTR ;RESET PASS COUNTER
428 001444 012767 000001 001326 MOV #1,SASV ;RESET STARTING SECTOR
429 001452 005287 001320 INC TASV ;RESET STARTING TRACK ADDRESS
430 001456 022767 000013 001312 CMP #13,TASV ;SEE IF DONE TRACKS
431 001464 001003 BNE #9S ;IF NOT: BR
432 001466 012767 000001 001302 MOV #1,TASAV ;ELSE RESET TRACK STARTING ADDRESS
433 001474 012767 001276 001270 MOV TASAV,TA ;SAVE TRACK ADDRESS
434 001502 016767 001272 001264 MOV SASAV,SA ;SAVE SECTOR
435 001510 000207 10S: RTS PC
436
437 ;WRITE SUBROUTINE*****
438
439 001512 012767 000001 001240 WRITE: MOV #1,CMD ;SET TO FILL BUFFER
440 001520 012767 000001 001240 MOV #16,RTF ;SET WRITE FLAG
441 001530 012701 003602 MOV #200,RO ;SET ADDRESS OF WRITE BUFFER
442 001532 012700 000200 MOV #200,RO ;SET SIZE OF BUFFER
443 001536 016777 001216 001244 1S: MOV CMD,BRXCS ;LOAD COMMAND
444 001544 004767 001076 JSR #ANTR ;GO AWAIT TRANSFER READY
445 001544 012177 MOVB (R1),BRXDB ;FILL BUFFER
446 001554 005300 DEC RO ;SEE IF DONE FILL
447 001556 001372 BNE #1S ;IF NOT: BR
448 001560 004767 000760 JSR #ANDN ;GO AWAIT DONE
449 001564 005767 001220 TST BRXCS ;SEE IF ERROR
450 001570 100025 BPL #1S ;IF NOT: BR
451 001572 016767 001212 176300 MOV RXCS,CSRA ;LOAD ADDRESS OF RXCS
452 001600 022767 001154 176274 MOV CMD,ACSR ;LOAD CURRENT COMMAND
453 001606 057767 001176 176266 BLS BRXCS,ACSR ;LOAD RXCS
454 001614 017767 001172 176262 MOV BRXDB,ASTAT ;LOAD RXES
455 001622 104403 000000 003520 MSGNS,BEGIN,BFER ;ASCII MESSAGE CALL WITH COMMON HEADER
456 001630 012767 000035 176250 MOV #35,BERTYPE
457
458 001636 104405 000000 000000 ;*****
459 ;*****
460 ;*****
461 ;*****
462 ;*****
463 ;*****
464 ;*****
465 ;*****
466 ;*****
467 ;*****
468 ;*****
469 ;*****
470 ;*****
471 ;*****
472 ;*****
473 ;*****
474 ;*****
475 ;*****
476 ;*****
477 ;*****
478 ;*****
479 ;*****
480 ;*****
481 ;*****
482 ;*****
483 ;*****
484 ;*****
485 ;*****
486 ;*****
487 ;*****
488 ;*****
489 ;*****
490 ;*****
491 ;*****
492 ;*****
493 ;*****
494 ;*****
495 ;*****
496 ;*****
497 ;*****
498 ;*****
499 ;*****
500 ;*****
501 ;*****
502 ;*****
503 ;*****
504 ;*****
505 ;*****
506 ;*****
507 ;*****
508 ;*****
509 ;*****
510 ;*****
511 ;*****
512 ;*****
513 ;*****
514 ;*****
515 ;*****
516 ;*****
517 ;*****
518 ;*****
519 ;*****
520 ;*****
521 ;*****
522 ;*****
523 ;*****
524 ;*****
525 ;*****
526 ;*****
527 ;*****
528 ;*****
529 ;*****
530 ;*****
531 ;*****
532 ;*****
533 ;*****
534 ;*****
535 ;*****
536 ;*****
537 ;*****
538 ;*****
539 ;*****
540 ;*****
541 ;*****
542 ;*****
543 ;*****
544 ;*****
545 ;*****
546 ;*****
547 ;*****
548 ;*****
549 ;*****
550 ;*****
551 ;*****
552 ;*****
553 ;*****
554 ;*****
555 ;*****
556 ;*****
557 ;*****
558 ;*****
559 ;*****
560 ;*****
561 ;*****
562 ;*****
563 ;*****
564 ;*****
565 ;*****
566 ;*****
567 ;*****
568 ;*****
569 ;*****
570 ;*****
571 ;*****
572 ;*****
573 ;*****
574 ;*****
575 ;*****
576 ;*****
577 ;*****
578 ;*****
579 ;*****
580 ;*****
581 ;*****
582 ;*****
583 ;*****
584 ;*****
585 ;*****
586 ;*****
587 ;*****
588 ;*****
589 ;*****
590 ;*****
591 ;*****
592 ;*****
593 ;*****
594 ;*****
595 ;*****
596 ;*****
597 ;*****
598 ;*****
599 ;*****
600 ;*****
601 ;*****
602 ;*****
603 ;*****
604 ;*****
605 ;*****
606 ;*****
607 ;*****
608 ;*****
609 ;*****
610 ;*****
611 ;*****
612 ;*****
613 ;*****
614 ;*****
615 ;*****
616 ;*****
617 ;*****
618 ;*****
619 ;*****
620 ;*****
621 ;*****
622 ;*****
623 ;*****
624 ;*****
625 ;*****
626 ;*****
627 ;*****
628 ;*****
629 ;*****
630 ;*****
631 ;*****
632 ;*****
633 ;*****
634 ;*****
635 ;*****
636 ;*****
637 ;*****
638 ;*****
639 ;*****
640 ;*****
641 ;*****
642 ;*****
643 ;*****
644 ;*****
645 ;*****
646 ;*****
647 ;*****
648 ;*****
649 ;*****
650 ;*****
651 ;*****
652 ;*****
653 ;*****
654 ;*****
655 ;*****
656 ;*****
657 ;*****
658 ;*****
659 ;*****
660 ;*****
661 ;*****
662 ;*****
663 ;*****
664 ;*****
665 ;*****
666 ;*****
667 ;*****
668 ;*****
669 ;*****
670 ;*****
671 ;*****
672 ;*****
673 ;*****
674 ;*****
675 ;*****
676 ;*****
677 ;*****
678 ;*****
679 ;*****
680 ;*****
681 ;*****
682 ;*****
683 ;*****
684 ;*****
685 ;*****
686 ;*****
687 ;*****
688 ;*****
689 ;*****
690 ;*****
691 ;*****
692 ;*****
693 ;*****
694 ;*****
695 ;*****
696 ;*****
697 ;*****
698 ;*****
699 ;*****
700 ;*****
701 ;*****
702 ;*****
703 ;*****
704 ;*****
705 ;*****
706 ;*****
707 ;*****
708 ;*****
709 ;*****
710 ;*****
711 ;*****
712 ;*****
713 ;*****
714 ;*****
715 ;*****
716 ;*****
717 ;*****
718 ;*****
719 ;*****
720 ;*****
721 ;*****
722 ;*****
723 ;*****
724 ;*****
725 ;*****
726 ;*****
727 ;*****
728 ;*****
729 ;*****
730 ;*****
731 ;*****
732 ;*****
733 ;*****
734 ;*****
735 ;*****
736 ;*****
737 ;*****
738 ;*****
739 ;*****
740 ;*****
741 ;*****
742 ;*****
743 ;*****
744 ;*****
745 ;*****
746 ;*****
747 ;*****
748 ;*****
749 ;*****
750 ;*****
751 ;*****
752 ;*****
753 ;*****
754 ;*****
755 ;*****
756 ;*****
757 ;*****
758 ;*****
759 ;*****
760 ;*****
761 ;*****
762 ;*****
763 ;*****
764 ;*****
765 ;*****
766 ;*****
767 ;*****
768 ;*****
769 ;*****
770 ;*****
771 ;*****
772 ;*****
773 ;*****
774 ;*****
775 ;*****
776 ;*****
777 ;*****
778 ;*****
779 ;*****
780 ;*****
781 ;*****
782 ;*****
783 ;*****
784 ;*****
785 ;*****
786 ;*****
787 ;*****
788 ;*****
789 ;*****
790 ;*****
791 ;*****
792 ;*****
793 ;*****
794 ;*****
795 ;*****
796 ;*****
797 ;*****
798 ;*****
799 ;*****
800 ;*****
801 ;*****
802 ;*****
803 ;*****
804 ;*****
805 ;*****
806 ;*****
807 ;*****
808 ;*****
809 ;*****
810 ;*****
811 ;*****
812 ;*****
813 ;*****
814 ;*****
815 ;*****
816 ;*****
817 ;*****
818 ;*****
819 ;*****
820 ;*****
821 ;*****
822 ;*****
823 ;*****
824 ;*****
825 ;*****
826 ;*****
827 ;*****
828 ;*****
829 ;*****
830 ;*****
831 ;*****
832 ;*****
833 ;*****
834 ;*****
835 ;*****
836 ;*****
837 ;*****
838 ;*****
839 ;*****
840 ;*****
841 ;*****
842 ;*****
843 ;*****
844 ;*****
845 ;*****
846 ;*****
847 ;*****
848 ;*****
849 ;*****
850 ;*****
851 ;*****
852 ;*****
853 ;*****
854 ;*****
855 ;*****
856 ;*****
857 ;*****
858 ;*****
859 ;*****
860 ;*****
861 ;*****
862 ;*****
863 ;*****
864 ;*****
865 ;*****
866 ;*****
867 ;*****
868 ;*****
869 ;*****
870 ;*****
871 ;*****
872 ;*****
873 ;*****
874 ;*****
875 ;*****
876 ;*****
877 ;*****
878 ;*****
879 ;*****
880 ;*****
881 ;*****
882 ;*****
883 ;*****
884 ;*****
885 ;*****
886 ;*****
887 ;*****
888 ;*****
889 ;*****
890 ;*****
891 ;*****
892 ;*****
893 ;*****
894 ;*****
895 ;*****
896 ;*****
897 ;*****
898 ;*****
899 ;*****
900 ;*****
901 ;*****
902 ;*****
903 ;*****
904 ;*****
905 ;*****
906 ;*****
907 ;*****
908 ;*****
909 ;*****
910 ;*****
911 ;*****
912 ;*****
913 ;*****
914 ;*****
915 ;*****
916 ;*****
917 ;*****
918 ;*****
919 ;*****
920 ;*****
921 ;*****
922 ;*****
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 ;*****
1000 ;*****

```

```
NTRUPT:-----
469 001716*
470
471 001716* 000004 000000* 001724*  ;PIRQ$,BEGIN,1$ ; QUEUE UP TO CONTINUE AT 1$ AND RTI
472
473
474
475 001724* 004767 000072 1$: JSR PC,SECK ;GO CHECK FOR ERROR
476 001730* 016700 176054 MOV VECTOR,R0 ;GET VECTOR ADDRESS
477 001736* 057720 TST R0 ;RUP POINTER
478 001742* 000207 176050 MOV BR1,(R0) ;ASSURE RESET TO BR LEVEL 5
479 RTS PC ;RETURN
480
481 ;READ SUBROUTINE*****
482 001744* 005067 001016 READ: CLR WTF ;CLEAR WRITE FLAG
483 001750* 016767 001006 MOV UTA,CMD ;SELECT DRIVE
484 001756* 052767 001077 BIS #107,CMD ;SET READ COMMAND + INT ENB
485 001764* 016777 000770 MOV CMD,SRXCS ;LOAD COMMAND
486 001772* 004767 000650 JSR PC,AWTR ;GC AWAIT TRANSFER READY
487 001776* 016777 000772 MOV SA,SRXDB ;LOAD SECTOR ADDRESS
488 002000* 047667 000936 JSR PC,AWTR ;GC AWAIT TRANSFER READY
489 002010* 016767 000936 MOV TA,SRXDB ;LOAD TRACK ADDRESS
490 002016* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
491
492 ;STATUS ERROR CHECK SUBROUTINE*****
493 002022* 005067 000736 SECK: CLR SERFL ;CLEAR STATUS ERROR FLAG
494 002026* 057767 000756 BIS SRXCS,CMD ;GET RXCS
495 002034* 005767 000720 TST CMD ;SEE IF ERROR
496 002040* 104401 000720 JSR PC,ERRTP ;IF SO: BR
497 002044* 000207 RTS PC ;ELSE RETURN
498 002044* 016767 000710 MOV CMD,SRXCS ;LOAD RXCS
499 002052* 016767 000702 MOV CMD,ACSR ;LOAD ACSE
500 002058* 016767 000704 MOV PC,AWTR ;GC AWAIT TRANSFER READY
501 002060* 016767 000700 MOV TA,SRXTA ;LOAD TRACK ADDRESS OF RXCS
502 002074* 016767 000674 MOV SA,SRXSA ;LOAD SECTOR ADDRESS
503 002102* 017767 000704 MOV SRXDB,SRXES ;GET RXES
504 002110* 016767 000706 MOV SRXES,STAT ;LOAD RXES STAT
505 002114* 016767 000640 MOV UTA,CMD ;SELECT DRIVE
506 002124* 052767 000017 BIS #17,CMD ;LOAD READ STATUS B COMMAND
507 002132* 016777 000622 MOV CMD,SRXCS ;EXECUTE COMMAND
508 002140* 016767 000650 JSR PC,AWTR ;GC AWAIT TRANSFER READY
509 002144* 016767 000650 MOV SRXDB,SRXSB ;LOAD STATUS B
510 002152* 005767 000610 TST RTF ;SEE IF WRITE ERROR
511 002156* 001013 BNE ZS ;IF SO: BR
512 002160* 014403 MSGNS,BEGIN,RTER ;ASCII MESSAGE CALL WITH COMMON HEADER
513 002166* 012767 000036 ;*****
514
515 002174* 104405 000000* 003032* HRDRS,BEGIN,TABLE ;READ ERROR
516 002202* 000167 000022 JMP CLEAR ;GO CLEAR ERROR
517
518 2$:
519 002206* 104403 000000* 003464* MSGNS,BEGIN,RTER ;ASCII MESSAGE CALL WITH COMMON HEADER
520 002214* 012767 000037 175664 ;*****
521
522 002222* 104405 000000* 003032* HRDRS,BEGIN,TABLE ;WRITE ERROR
523 ;*****
524 ;CLEAR ERRORS*****
```

```
525 002230* 005267 000530 CLEAR: INC SERFL ;SET STATUS ERROR FLAG
526 002234* 005767 000564 TST SRXSB ;SEE IF SB ERROR
527 002240* 001415 BEQ ZS ;IF NOT: BR
528 002240* 001415 MOV #240000,SRXCS ;INITIALIZE
529 002250* 004767 000270 1$: JSR PC,AWDN ;GO AWAIT DONE
530 002254* 005777 000530 TST SRXCS ;SEE IF ERROR CN INIT
531 002260* 100005 BEQ ZS ;IF NOT: BR
532 002270* 000000* 003544* MSGNS,BEGIN,DRP3 ;ASCII MESSAGE CALL WITH COMMON HEADER
533 002270* 000167 JMP IDRP ;GO DROP MODULE
534 2$: INC RTYN ;BUMP RETRY COUNTER
535 002274* 005267 001160 MOV RTYN,R3 ;GET COUNTER
536 002300* 016703 001154 BIC #1770,R3 ;MARK ASCII
537 002310* 027703 000003 CMP #3,R3 ;SEE IF DONE RETRIES
538 002314* 001011 BNE ZS ;IF NOT: BR
539 002316* 104403 MSGNS,BEGIN,HRDE ;ASCII MESSAGE CALL WITH COMMON HEADER
540 002324* 002767 000001 175464 BIT #1,SR1 ;SEE IF SHOULD DROP MODULE
541 002332* 001412 BEQ ZS ;IF NOT: BR
542 002334* 000167 176514 JMP FINI ;GC DROP MODULE
543 002340* 005726 000420 3$: TST (SP)+ ;RESET STACK
544 002342* 005767 WTF ;SEE IF WRITE TIME
545 002346* 001402 BEQ ZS ;IF NOT: BR
546 002350* 000167 177136 JMP WRITE ;RETRY WRITE
547 002354* 000167 177364 JMP READ ;RETRY READ
548 002360* 012767 000260 5$: MOV #260,RTYN ;RESET RETRY COUNTER
549 RTS PC ;RETURN
550
551 ;DATA CHECK SUBROUTINE*****
552 002370* 005767 000370 DCHK: TST SERFL ;SEE IF STATUS ERROR
553 002374* 001401 BEQ ZS ;IF NOT: BR
554 002376* 000207 TST PC ;SEE IF STATUS ERROR
555 002400* 005067 000364 1$: CLR BWCNT ;CLEAR BAD WORD COUNTER
556 002404* 012700 000200 MOV #200,R0 ;SET SIZE OF BUFFER
557 002410* 012701 004004* MOV #RBUF,R1 ;ADDRESS OF READ BUFFER
558 002414* 012777 000003 000366 2$: MOV #3,SRXCS ;LOAD READ BUFFER COMMAND
559 002422* 004767 000220 JSR PC,AWTR ;GC AWAIT TRANSFER READY
560 002426* 177721 000360 MOV#B SRXDB,(R1)+ ;EMPTY BUFFER
561 002432* 005300 RO ;SEE IF EMPTIED ENTIRE BUFFER
562 002434* 001372 BNE ZS ;IF NOT: BR
563 002436* 004767 JSR PC,AWDN ;GC AWAIT DONE
564 002442* 005000 000102 3$: CLR R0 ;CLEAR WORD COUNTER
565 002444* 012701 003602* MOV #RBUF,R1 ;SET ADDRESS OF WRITE DATA
566 002450* 012702 004004* MOV #RBUF,R2 ;SET ADDRESS OF READ DATA
567 002454* 001112 CMP (R1),(R2) ;TEST DATA
568 002456* 001006 BNE OS ;IF DATA BAD: BR
569 002460* 022122 5$: CMP (R1)+(R2)+ ;BUMP ADDRESS OF DATA POINTER
570 002462* 005200 INC R0 ;BUMP WORD COUNTER
571 002464* 012700 000100 6$: CMP #100,R0 ;SEE IF DONE ALL
572 002470* 001371 BNE ZS ;CHECK ALL
573 002472* 000207 RTS PC ;IF DONE: EXIT
574 002474* 010167 175402 MOV R1,SBADR ;LOAD GOOD ADDRESS
575 002476* 010167 175400 MOV R1,ASADR ;LOAD BAD ADDRESS
576 002500* 010167 175376 MOV (R1),ASB ;LOAD GOOD DATA
577 002510* 011267 175374 MOV (R2),AWAS ;LOAD BAD DATA
578 002514* 104403 000000* 003510* MSGNS,BEGIN,RTER ;ASCII MESSAGE CALL WITH COMMON HEADER
579 ;*****
580
```



```
581 002522* 104404 000000* DATER$,BEGIN ;DATA ERROR!!!
582 002526* 005267 000236 ;*****
583 002532* 022767 000003 000230 ;INC BWCNT ;BUMP BAD WORD COUNTER
584 002540* 001347 000000 ;CMP #3,BWCNT ;SEE IF 3 BAD WORDS
585 002542* 000207 000000 ;BNE $S ;IF NOT: BR
586 ;RTS PC ;RETURN
587 ;AWAIT DONE BIT SUBROUTINE*****
588
589
590 002544* 005067 000234 AMDN: CLR TOCNT ;PRESET TIME OUT COUNTER
591 002550* 032777 000040 000232 1$: BIT #40,@RXCS ;SEE IF DONE SET
592 002556* 001032 ;BNE $S ;IF SO: BR
593 002560* 104407 000000* ;BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...
594 002564* 104407 000000* ;BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
595 002570* 005267 000210 ;INC TOCNT ;BUMP TIME OUT COUNTER
596 002574* 001365 ;BNE $S ;IF NOT TIMED OUT: BR
597 002576* 005067 175302 CLR A$STAT ;CLEAR STATUS WORD
598 002602* 016767 000202 175270 MOV RXCS,CSRA ;SET ADDRESS OF RXCS
599 002610* 016767 000144 175264 MOV CMD,ACSR ;SET COMMAND WORD
600 002616* 104403 000000* 003566* MSGNS,BEGIN,DTND ;ASCII MESSAGE CALL WITH COMMON HEADER
601 002624* 012767 000040 175254 MOV #40,ERRTYP ;*****
602 002632* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;DCNE BIT TIME OUT
603 ;*****
604 ;*****
605 002640* 000167 176210 JMP FINI ;DROP MODULE
606 002644* 000207 ;RTS PC ;EXIT
607 ;AWAIT TRANSFER READY SUBROUTINE*****
608
609
610 002646* 005067 000132 AMDN: CLR TOCNT ;PRESET TIME OUT COUNTER
611 002652* 032777 000200 000130 1$: BIT #200,@RXCS ;SEE IF TRANSFER READY SET
612 002660* 001032 ;BNE $S ;IF SO: BR
613 002662* 104407 000000* ;BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...
614 002666* 104407 000000* ;BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
615 002672* 005267 000106 ;INC TOCNT ;BUMP TIME OUT COUNTER
616 002676* 001365 ;BNE $S ;IF NOT TIMED OUT: BR
617 002700* 005067 175200 CLR A$STAT ;CLEAR STATUS WORD
618 002704* 016767 000100 175166 MOV RXCS,CSRA ;SET ADDRESS OF RXCS
619 002712* 016767 000042 175162 MOV CMD,ACSR ;SET COMMAND WORD
620 002720* 104403 000000* 003574* MSGNS,BEGIN,TRTD ;ASCII MESSAGE CALL WITH COMMON HEADER
621 002726* 012767 000040 175152 MOV #40,ERRTYP ;*****
622 ;*****
623 002734* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;TRANSFER READY TIME OUT
624 ;*****
625 002742* 000167 176106 JMP FINI ;GO DROP MODULE
626 002746* 000207 ;RTS PC ;RETURN
627 ;FLAGS AND COUNTERS*****
628
629
630
631 002750* 000000 DVID1X: 0 ;HOLDS WHICH DRIVES TO TEST
632 002752* 000000 UNIT0: 0 ;UNIT 0 FLAG
633 002754* 000000 UNIT1: 0 ;UNIT 1 FLAG
634 002756* 000000 DRVN: 0 ;DRIVE NUMBER
635 002760* 000000 CMD: 0 ;COMMAND SAVE
636 002762* 000000 UFT: 0 ;UNIT UNDER TEST
```

```
637 002764* 000000 SERFL: 0 ;STATUS ERROR FLAG
638 002766* 000000 WTF: 0 ;WRITE FLAG
639 002770* 000000 BWCNT: 0 ;BAD WORD COUNTER
640 002772* 000000 TA: 0 ;CURRENT TRACK ADDRESS
641 002774* 000000 SA: 0 ;CURRENT SECTOR ADDRESS
642 002776* 000000 TASAV: 0 ;STARTING TRACK ADDRESS SAVE
643 003000* 000000 SASAV: 0 ;STARTING SECTOR ADDRESS SAVE
644 003002* 000000 PCNTR: 0 ;INTERNAL PASS COUNTER
645 003004* 000000 TOCNT: 0 ;TIME OUT COUNTER
646 003006* 000001 WVFLG: 1 ;WRITE VERIFY FLAG
647
648
649
650
651
652
653
654
655 ;+ NOT EQ ZERO [A] DISABLE INTERNAL PASS
;+ TRACK & SECTOR OFFSETTING.
;+ [B] USE DATA PATTERN OF ZERO.
;+ EQUAL ZERO [A] ALLOW INTERNAL PASS
;+ TRACK & SECTOR OFFSETTING.
;+ [B] USE DATA PATTERN DEFINED
;+ BY SRI.
;+ (RANDOM OR ONES AND ZEROS).
```

```
656 ;CONSTANTS*****
657
658 RXCS: 177170 ;RX01 COMMAND REGISTER
659 RXDB: 177172 ;RX01 DATA BUFFER
660 RXVC: 264 ;VECTOR
661 TPCNTR: 1 ;TOTAL PASS COUNTER
662
663 ;VARIABLES*****
664
665 SRXCS: 0
666 SRXES: 0
667 SRXSB: 0
668 SRXTA: 0
669 SRXSA: 0
670
671 ;MESSAGE TABLE*****
672
673 TABLE:
674 ARXCS: SRXCS
675 ARXES: SRXES
676 ARXSB: SRXSB
677 ARXTA: SRXTA
678 ARXSA: SRXSA
679 -1
680
681 003046 053440 044522 042524 MSG1: .ASCIZ " WRITE ERROR: RETRY:"
682 003054 042440 051122 051117
683 003062 020072 042522 051124
684 003070 035133 000 042101 MSG2: .ASCIZ " READ ERROR: RETRY:"
685 003078 042440 042522 051117
686 003106 020072 042522 051124
687 003114 035133 000 053111 MSG3: .ASCIZ " DRIVE "
688 003174 020105 051104 050117 MSG4: .ASCIZ " DROP MODULE%"
689 003174 020105 051104 046125
690 003174 020105 051104 044516 MSG5: .ASCIZ "% INITIALIZE ERROR:"
691 003174 020105 051104 046101
692 003174 020105 051105 047522
693 003174 020105 051105 047522
694 003174 020105 051105 047522
695 003174 020105 051105 047522
696 003174 020105 051105 047522
697 003174 020105 051105 047522
698 003174 020105 051105 047522
699 003174 020105 051105 047522
700 003174 020105 051105 047522
701 003174 020105 051105 047522
702 003174 020105 051105 047522
703 003174 020105 051105 047522
704 003174 020105 051105 047522
705 003174 020105 051105 047522
706 003174 020105 051105 047522
707 003174 020105 051105 047522
708 003174 020105 051105 047522
709 003174 020105 051105 047522
710 003174 020105 051105 047522
711 003174 020105 051105 047522
```

```
712 003266 052502 043106 051105
713 003274 042440 051122 051117
714 003302 000072 051101 020104 MSG10: .ASCIZ " HARD ERROR:"
715 003302 044040 047522 035122
716 003320 000 047125 052111 MSG11: .ASCIZ " UNIT 0 NOT AVAILABLE"
717 003320 000 047125 052111
718 003321 040 047125 052111
719 003342 030040 040524 046117
720 003342 041101 042514 000 MSG12: .ASCIZ " UNIT 1 NOT AVAILABLE"
721 003342 040 047125 052111
722 003342 030040 040524 046117
723 003342 041101 042514 000
724 003375 040 047504 042516 MSG13: .ASCIZ " DONE BIT TIME OUT:"
725 003375 040 047504 042516
726 003407 041040 052111 052040
727 003416 046511 020105 052517
728 003416 035124 000
729 003421 040 051124 047101 MSG14: .ASCIZ " TRANSFER READY TIME OUT:"
730 003421 043123 051105 051040
731 003421 040505 051504 052040
732 003442 046511 020105 052517
733 003450 035124 000
734 003454 000 000 DNUM: .EVEN
735 003454 000 000 .ASCIZ " 0"
736 003460 000 000 .EVEN
737 003460 000 000 .ASCIZ " 0"
738
739 WTER: .EVEN
740 003464 003117 MSG3
741 003466 003454 DNUM
742 003470 003046 MSG1
743 003474 000 RTYN
744 003474 000 -1
745 003476 003117 MSG3
746 003500 003454 DNUM
747 003504 003460 MSG2
748 003504 003460 RTYN
749
750 DTER: .EVEN
751 003510 003117 MSG3
752 003514 003454 DNUM
753 003514 003454 MSG8
754 003516 003117 -1
755 003520 003117 BFER: MSG3
756 003524 003454 DNUM
757 003524 003460 MSG9
758 003526 003117 -1
759 003530 003117 DRP1: MSG5
760 003534 003117 MSG4
761 003536 003117 -1
762 003540 003117 DRP2: MSG6
763 003544 003117 MSG4
764 003544 003117 -1
765 003546 003117 DRP3: MSG7
766 003550 003117 MSG4
767 003550 003117 -1
```


MODULE	CROSS REFERENCE	TABLE	USER SYMBOLS												
DVID1 = 000014R	197#	256	260												
DVID1X = 002750R	256#	259*	297	310	631#										
EMDIT = 104410	242#	341													
ERRRTP = 000106R	230#	288*	305*	318*	456*	513*	520*	601*	621*						
EXIT = 104400	245#	467	490												
FILE = 001054R	242#	326	346#	542	605	625									
GMBUF = 104415	242#														
GMBUF2 = 104414	242#														
HRDCNT = 000044R	210#														
HRDE = 003552R	232#														
HRDEF = 104405R	242#	768#	790#	307	320	458	515	522	603	623					
HRDFAS = 000050R	242#														
ICOUNT = 000036R	207#														
ICOUNT = 000040R	208#														
IDNUM = 001122R	242#														
INVT = 000030R	204#														
INTR = 000120R	236#														
LDRDP = 000546R	242#														
MAP2 = 104416	242#														
MODNAN = 000000R	191#														
MODSP = 000224R	205#	243#													
HSGNS = 104403	242#	276	295	304	317	325	455	512	519	532	539	579			
HSGSS = 104402	242#														
HSGS = 104401	245#														
HSG10 = 003046R	693#	743													
HSG10 = 003032R	718#	768													
HSG11 = 003321R	718#	770													
HSG12 = 003347R	722#	772													
HSG13 = 003375R	730#	777													
HSG14 = 003375R	730#	777													
HSG14 = 003073R	687#	748													
HSG3 = 003117R	691#	746	751	755	778										
HSG4 = 003127R	692#	760													
HSG4 = 003145R	692#	759													
HSG6 = 003171R	700#	762													
HSG7 = 003215R	704#	765													
HSG8 = 003243R	709#	753													
HSG9 = 003243R	709#	753													
INTRUPT = 001716R	397#	469#													
NULL = 000000R	245#	320	458	603	623										
NULLTO = 003556R	267#	304	307	770#											
NUMT1 = 003556R	267#	304	307	770#											
NUMT2 = 003556R	267#	304	307	770#											
OPEN = 000000R	192#	199	200	201	218	219	220	221	222	223	224	225			
OPEN = 000000R	227#	231	232	234	235	236	245#								
OTDAS = 104420	242#														
PASCNT = 000034R	454#														
PCNTR = 003002R	454#	416*	423	425	427*	644#									
PIROS = 000004R	245#	471													
POPP = 005726	245#														
POP2 = 006000	245#														
PRV = 000000	196#														
PRTY0 = 000000	196#	245#													
PRTY1 = 000040	245#														
PRTY2 = 000100	245#														

MODULE	CROSS REFERENCE	TABLE	USER SYMBOLS											
PRTY3 = 000140	245#													
PRTY4 = 000200	245#													
PRTY5 = 000240	195#	245#												
PRTY6 = 000300	445#													
PRTY7 = 000340	445#													
PS = 177776	245#													
PSW = 177776	242#													
PUSH = 057776	242#													
PUSH2 = 024646	245#													
RANDS = 104417	245#	359												
RANMUM = 000054R	360#	360												
RBF = 004804R	347#	785#												
READ = 001744R	353#	482#	547											
RESTRT = 000354R	233#	261	263#											
RES1 = 000056R	216#													
RES2 = 000060R	216#													
RESRT = 000112R	233#													
RTER = 003476R	517#	746#												
RTYN = 003460R	303*	336*	338*	534*	535*	548*	738#	744*	748					
RXCS = 003010R	407#	404	477	528*	530*	549*	748*	749	749					
RXCS = 003010R	407#	500	507*	530*	530*	553*	591	598	611	451	453	462*		
RXCS = 003010R	407#	301	314	400*	401*	445*	454	464*	466*	468*	489*	503		
RXDB = 003012R	279#	287	314	400*	401*	445*	454	464*	466*	468*	489*	503		
RXDB = 003012R	279#	287	314	400*	401*	445*	454	464*	466*	468*	489*	503		
SA = 002774R	509#	580	660#											
SASAV = 003000R	251*	511	408	411*	434*	464	487	502	641#					
SBADR = 000102R	226#	575#												
SECK = 020222R	472#	493#												
SLD = 001122R	323#	374#	379	387										
SERFL = 002764R	493*	553	553											
SOPCNT = 000042R	209#													
SOPERS = 104406	245#													
SOPERS = 000046R	241#													
SOPINT = 000032R	205#													
SPSIZ = 000040	238													
SRXCS = 003020R	277*	282	498*	667#	676									
SRXCS = 003022R	277*	503*	504	668#	677									
SRXCSA = 003030R	280*	502*	671#	680	678									
SRXSB = 003024R	287*	509*	669#	678										
SRXSTA = 003026R	281*	501*	670#	679										
SR3 = 000020R	198#	357												
SR4 = 000022R	200#													
SR4 = 000024R	201#													
TART = 000224R	204#	247#												
TART = 000026R	203#													
SVR0 = 000062R	218#													
SVR1 = 000064R	219#													
SVR2 = 000066R	220#													
SVR3 = 000068R	221#													
SVR4 = 000072R	222#													
SVR5 = 000074R	223#													
SVRG = 000076R	224#													
TSCHT = 000078R	225#													
TA = 002772R	222#	412*	413	415*	433*	466	489	501	640#					
TABLE = 003032R	290#	515	522	675#										
TASAV = 002776R	250*	252	415	421*	429*	430	432*	433	642#					

TDRP	000430R	277#	533				
TGD	000556R	275	297#				
TCNT	003004R	590*	595#	610*	615*	645#	
TPCTR	003016R	662	646#				
TRDPD=	000077R	620	777#				
TRTO	003574R	620	777#				
TSS	001310R	343	407#				
TST	007746R	316	324	327#	344		
UNIT0	002754R	322*	376	393*	384	632#	
UNIT1	002754R	322*	376	393*	384	632#	
UNIT	000566R	299#					
UNIT	000634R	296	298	309#			
UNITA	000634R	303	310#				
UTI	002762R	380*	389#	460	483	505	636#
VEC	003014R	661#					
VECTOR	000010R	194#	396	475			
VSET	001250R	170	396#				
WASADR	000104R	428#	576#				
WBUF	003602R	353	441	566	783#		
WDFR	000116R	235	248*				
WDT0	000114R	233	248*				
WRITE	001512R	331	317	439#	546		
WTR	003464R	513*	741#				
WTF	002766R	440	482#	510	544	638#	
WVFLG	004006R	255*	383	417	419*	646#	
XFLAG	000005R	192#					
.	= 004206R	735#	737#	740#	784#	786#	

. ABS. 000000 000
 004206 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XRXAEO,XRXAEO/SOL/CRF:SYN=DDXCOM,XRXAEO
 RUN-TIME: 1 2 .3 SECONDS
 RUN-TIME RATIO: 22/4=5.0
 CORE USED: 7K (13 PAGES)