

# DM11A

DATA TEST  
MD-11-DZDMB-B

EP-DZDMB-B-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames. The leftmost column of frames contains text, likely test instructions or labels. The remaining frames contain data, including numerical values, tables, and possibly small graphs or plots. The data is organized in a structured, grid-like format across the card.



.REM :

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDMS-B-D  
 PRODUCT NAME: DM11 DATA TESTS  
 DATE RELEASED: MAY, 1976  
 MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT 1972, 1976 BY DIGITAL EQUIPMENT CORPORATION  
 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  
 DOCUMENT.  
 THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A  
 LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH  
 THE TERMS OF SUCH LICENSE.  
 DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY  
 FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT  
 THAT IS NOT SUPPLIED BY DIGITAL.

## 1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11 (ASYNCHRONOUS DATA MULTIPLEXER), MAINDEC-11-DZDMA (DM11 LOGIC TESTS), AND MAINDEC-11-DZDMB (DM11 MULTIPLE LINE DATA TESTS). THE LOGIC TESTS INDIVIDUALLY TEST EACH OF THE 16 DM11 LINES AND ALL COMMON LOGIC. THE MULTIPLE LINE DATA TESTS RUN SEVERAL LINES CONCURRENTLY AND ARE USED TO TEST LINE INTERACTION AND DATA TRANSMISSION/RECEPTION RELIABILITY. THIS DOCUMENT DESCRIBES THE MULTIPLE LINE DATA TESTS. THE AVAILABLE TESTS ARE:

PRG0 - DATA TESTS  
 PRG1 - DATA TEST (ALL LINES SIMULTANEOUSLY)  
 PRG2 - TRANSMIT TO TERMINALS  
 PRG3 - ECHO RECEIVED DATA

## 2. REQUIREMENTS

## 2.1 EQUIPMENT

- A. PDP 11 FAMILY PROCESSOR
- B. DM11
- C. JUMPERS CONNECTING 16 TRANSMITTERS TO THEIR RESPECTIVE RECEIVERS.
- D. TERMINALS (IF AVAILABLE)
- E. DM11 DISTRIBUTION PANEL

## 2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (4K) EXCEPT THAT AREA RESERVED FOR THE LOADERS.

2.3 PREREQUISITE PROGRAMS  
 MAINDEC-11-DZDMA (DM11 LOGIC TESTS)

## 3. LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.









169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202

5.0 PROGRAM DESCRIPTIONS

5.1 PRGO - LOGIC TESTS

PRGO CONSISTS OF 34(8) INDEPENDENT ROUTINES WHICH TRANSMIT VARIOUS DATA PATTERNS ON ALL LINES WITH A DECREASING DELAY BEFORE STARTING SUCCESSIVE LINES. THE DATA IS CHECKED WHEN ALL TRANSMITTERS HAVE COMPLETED TRANSMITTING. IF A DATA ERROR OCCURS THE ERROR TYPEOUT WILL SHOW THE DATA FAILURE AND THE LINE NUMBER.

5.2 PRG1 - DATA TEST (ALL LINES SIMULTANEOUSLY)

PROGRAM 1 TRANSMITS ' A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890' ON ALL LINES SIMULTANEOUSLY. WHEN ALL LINES HAVE FINISHED RECEIVED DATA IS VERIFIED. AN ERROR TYPEOUT IS AS IN PRGO.

5.3 PRG2 - TRANSMIT TO TERMINALS

PROGRAM 2 IS THE SAME AS PROGRAM 1 EXCEPT THAT THE RECEIVED DATA IS NOT CHECKED.

5.4 PRG3 - ECHO RECEIVED DATA

NOTE: THIS PROGRAM MAY ONLY BE RUN IF USING AN ASR 33 NOT MODIFIED BY DEC.

PROGRAM 3 ECHOES BACK DATA RECEIVED FROM A TERMINAL.

NOTE: PROGRAM 3 SHOULD BE RUN AND DATA TYPED AT ALL AVAILABLE DM11 TERMINALS. IT IS THE ONLY TEST THAT INSURE CORRECT OPERATION OF THE DM11 DISTRIBUTION PANEL LOGIC. IF THE TERMINALS ARE ASR-33 WITH A PAPER TAPE READER/PUNCH I SUGGEST THAT INDIVIDUAL TAPES BE MADE UP FOR EACH LINE. THIS CAN BE DONE BY RUNNING PROGRAM 2 WITH THE PUNCH TURNED ON. PROGRAM 2 WILL THEN PUNCH A TAPE ON EACH TERMINAL WITH THE LINE NUMBER IDENTIFIER AT THE BEGINNING OF EACH TAPE. PROGRAM 3 CAN BE RUN WITH THESE TAPES IN THE PAPER TAPE READERS.







```

259 000000
260 000000
261 000001
262 000002
263 000003
264 000004
265 000005
266 000006
267 000007
268
269 104000
270 104001
271 104002
272 104003
273 104004
274 104006
275 104007
276 104010
277 104012
278 104013
279 104014
280 104015
281 104016
282 104017
283 104020
284
285 000007
286 177777
287 125252
288 052525
289 000000
290 177777
291 000000
292 000000
293 000000
294 000000
295 000002
296 000004
297 000006
298 000010
299 000012
300 000014
301 000016
302 000020
303 000022
304 000024
305 000026
306 000030
307 000032
308 000034
309 000036
310 000040
311 000042
312 000044
313 000046
314 000050

```

```

PRTY0=0
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7
;EMT CALLS
TYPE=EMT+0
TYPES=EMT+1
STALL=EMT+2
ERROR=EMT+3
DATCHK=EMT+4
STRXV=EMT+6
STTXV=EMT+7
EHALT=EMT+10
SCOPE=EMT+12
SAVREG=EMT+13
RSTREG=EMT+14
ERROR1=EMT+15
SUSWR=EMT+16
KBDIN=EMT+17
CNTLU=EMT+20

BELL=007
ATLAST=-1
ALTO=125252
ALT1=052525
Y=0
X=-1
A=0
.=0
HALT
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
EMTINT
PRTY7
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT

```

```

;ALTERNATING 0'S PATTERN
;ALTERNATING 1'S PATTERN

;SP OVERFLOW, BUS ERROR TRAP
;RESERVED INSTRUCTION TRAP
;TRACE TRAP
;TRAP TO CALL IOX
;POWER FAIL TRAP
;EMT TRAP

;TRAPPED TO PREVIOUS ADDRESS.
;TRAPPED TO PREVIOUS ADDRESS.
;TRAPPED TO PREVIOUS ADDRESS.

```



315	000054	000056	.+2	
316	000056	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
317	000060	000062	.+2	
318	000062	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
319	000064	000056	.+2	
320	000066	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
321	000070	000072	.+2	
322	000072	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
323	000074	000076	.+2	
324	000076	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
325	000100	000102	.+2	
326	000102	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
327	000104	000106	.+2	
328	000106	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
329	000110	000112	.+2	
330	000112	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
331	000114	000116	.+2	
332	000116	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
333	000120	000122	.+2	
334	000122	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
335	000124	000126	.+2	
336	000126	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
337	000130	000132	.+2	
338	000132	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
339	000134	000136	.+2	
340	000136	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
341	000140	000142	.+2	
342	000142	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
343	000144	000146	.+2	
344	000146	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
345	000150	000152	.+2	
346	000152	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
347	000154	000156	.+2	
348	000156	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
349	000160	000162	.+2	
350	000162	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
351	000164	000166	.+2	
352	000166	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
353	000170	000172	.+2	
354	000172	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
355	000174	000176	.+2	
356	000176	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
357	000200	000202	.+2	
358	000202	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
359	000204	000206	.+2	
360	000206	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
361	000210	000212	.+2	
362	000212	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
363	000214	000216	.+2	
364	000216	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
365	000220	000222	.+2	
366	000222	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
367	000224	000226	.+2	
368	000226	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
369	000230	000232	.+2	
370	000232	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.

371	000234	000236	.+2	
372	000236	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
373	000240	000242	.+2	
374	000242	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
375	000244	000246	.+2	
376	000246	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
377	000250	000252	.+2	
378	000252	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
379	000254	000256	.+2	
380	000256	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
381	000260	000262	.+2	
382	000262	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
383	000264	000266	.+2	
384	000266	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
385	000270	000272	.+2	
386	000272	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
387	000274	000276	.+2	
388	000276	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
389	000300	000302	.+2	
390	000302	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
391	000304	000306	.+2	
392	000306	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
393	000310	000312	.+2	
394	000312	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
395	000314	000316	.+2	
396	000316	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
397	000320	000322	.+2	
398	000322	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
399	000324	000326	.+2	
400	000326	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
401	000330	000332	.+2	
402	000332	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
403	000334	000336	.+2	
404	000336	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
405	000340	000342	.+2	
406	000342	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
407	000344	000346	.+2	
408	000346	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
409	000350	000352	.+2	
410	000352	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
411	000354	000356	.+2	
412	000356	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
413	000360	000362	.+2	
414	000362	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
415	000364	000366	.+2	
416	000366	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
417	000370	000372	.+2	
418	000372	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
419	000374	000376	.+2	
420	000376	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
421				



```

422          000046 000046          =46          ;ACT11 HOOKS
423          000046 002452          $ENDAD
424          000052 000052          =52
425          000052 020000          020000
426
427
428          000174 000174          =174
429          000174 000000          DISPREG:0
430          000176 000000          SWREG: 0
431
432          000200 000200          =200
433          000200 000137 002076          JMP      @#START      ;GO TO START OF DIAGNOSTIC.
434          000204 000137 002114          JMP      @#RSTAT1     ;GO GET PROGRAM # & RESTART PROGRAM
435                                     ;USING PREVIOUS DM11 PARAMETERS
436          000210 000137 002154          JMP      @#RSTAT2     ;RESTART PREVIOUS PROGRAM USING
437                                     ;PREVIOUS DM11 PARAMETERS
438
439          001100 001100          =1100
440
441          001100 000000          SPBOT: 0
442          001102 177570          SWR: 177570
443          001104 177570          DISPLAY:177570
444          001106 000000          CAT: OPEN
445                                     ;STARTING ADDRESS OF
446          001146 000000          WCT: OPEN             ;CURRENT ADDRESS TABLE
447                                     ;STARTING ADDRESS OF
448          001206 000000          BAT: OPEN             ;WORD COUNT TABLE
449                                     ;STARTING ADDRESS OF
450          001246 000000          VAC: OPEN             ;BIT ASSEMBLY TABLE
451          001250 175000          CSR: 175000           ;32. SPARE WORDS
452          001252 175002          BAR: 175002           ;ADDRESS OF CLOCK STATUS REGISTER
453          001254 175004          BKCSR: 175004         ;ADDRESS OF BUFFER ACTIVE REGISTER
454          001256 175006          BASREG: 175006        ;ADDRESS OF BREAK STATUS REGISTER
455          001260 000000          CLKINT: OPEN          ;ADDRESS OF BASE REGISTER
456          001262 000240          CLKLVL: PRTY5         ;DM11 VECTOR ADDRESS (RECEIVER)
457          001264 000000          XMTINT: OPEN          ;PRIORITY LEVEL
458          001266 000240          XMTLVL: PRTY5         ;DM11 VECTOR ADDRESS (TRANSMITTER)
459          001270 000000          BARIM: OPEN           ;TRANSMITTER PRIORITY LEVEL
460          001272 000000          TTDAT: OPEN           ;PROGRAM BAR IMAGE
461          001274 000000          LINBIT: OPEN          ;TUMBLE TABLE DATA
462          001276 000000          BARDAT: OPEN          ;LINE BIT (FOR BAR)
463          001300 000000          TTPTR: OPEN           ;BAR DATA
464          001306 001306          =VAC+32.              ;PROGRAM TUMBLE TABLE POINTER
465          001306 000000          TUMTAB: OPEN          ;STARTING ADDRESS OF
466          001506 001506          =TUMTAB+128.          ;TUMBLE TABLE
467          001506 000060          TKVTR: 60             ;LSR INTERRUPT VECTOR
468          001510 000200          TKLVL: PRTY4          ;LSR PRIORITY LEVEL
469          001512 000064          TPVTR: 64             ;LSP INTERRUPT VECTOR
470          001514 000200          TPLVL: PRTY4          ;LSP PRIORITY LEVEL
471          001516 000000          KSTART: OPEN          ;CURRENT PROGRAM START ADDRESS.
472          001520 000000          CURTST: OPEN          ;CONTAINS ADDR OF CURRENT TEST.
473          001522 000000          RTNNO: OPEN           ;CONTAINS CURRENT TEST #.
474          001524 000000          NXTST: OPEN           ;CONTAINS ADDR OF NEXT TEST.
475          001526 000000          ICTR: OPEN            ;CONTAINS CURRENT ITERATION COUNT
476          001530 000000          SCOPTR: OPEN          ;CONTAINS CURRENT SCOPE POINTER.
477          001532 177774          PRGLIM: -4

```

478 001534 005120  
479 001536 006100  
480 001540 006264  
481 001542 006274  
482 001544 005142  
483 001546 006116  
484 001550 006270  
485 001552 006312  
486 001554 002650  
487 001556 000000  
488 001560 000000  
489 001562 001726  
490 001564 001660  
491 001566 000000  
492 001570 000000  
493 001572 000000  
494 001574 000000  
495 001576 000000  
496 001600 002322  
497 001602 002546  
498 001604 002606  
499 001606 001744  
500 001610 003652  
501 001612 003526  
502 001614 003572  
503  
504 001616 000000  
505 001620 177560  
506 001622 177562  
507 001624 177564  
508 001626 177566  
509 001630 000000  
510 001632 000000  
511 001634 000000  
512 001636 000000  
513 001640 000000  
514 001642 000000  
515 001644 000000  
516 001646 000000  
517 001650 000000

PRGTAB: PRG0  
PRG1  
PRG2  
PRG3  
RSTART: PRG0R  
PRG1R  
PRG2R  
PRG3R  
EMTTAB: TYP  
OPEN  
OPEN  
ERR  
DTCHK  
OPEN  
OPEN  
OPEN  
OPEN  
OPEN  
OPEN  
ESCOPE  
SAVRG  
RSTRG  
ERR1  
SUSWRR  
KBDINTT  
CNTLUU  
  
SRT: OPEN  
TKCSR: 177560  
TKDBR: 177562  
TPCSR: 177564  
TPDBR: 177566  
RCVDAT: OPEN  
XMTDAT: OPEN  
CARMSK: OPEN  
TEMP: OPEN  
PCADD: OPEN  
APCADD: OPEN  
PRVCNT: OPEN  
LINE: OPEN  
LINBUF: OPEN

:PRG0 START ADDRESS  
:PRG1 START ADDRESS  
:PRG2 START ADDRESS  
:PRG3 START ADDRESS  
:PRG0 RESTART ADDRESS  
:PRG1 " "  
:PRG2 " "  
:PRG3 " "  
:POINTER TO TYPEOUT ROUTINE  
:POINTER TO CHAINED MESSAGES ROUTINE  
:POINTER TO RANDOM STALL ROUTINE  
:POINTER TO ERROR ROUTINE



```

518
519 001652 104000          INCRTN: TYPE
520 001654 012540          M1          ;TYPE INCORRECT ROUTINE SELECTED.
521 001656 000207          RTS          %7          ;EXIT.
522
523          ;DATA CHECK ROUTINE.
524 001660 123737 001630 001632 DTCHK:  CMPB  RCVDAT,XMTDAT ;COMPARE EXPECTED AND RECEIVED
525 001666 001416          BEQ    1$          ;CHARS. BRANCH IF SAME.
526 001670 004737 002050          JSR    7,CNVDAT   ;CONVERT RCVDAT & XMTDAT TO ASCII
527 001674 032777 020000 177200 BIT    #BIT13,ASWR ;ERROR TYPEOUT DESIRED?
528 001702 001010          BNE    1$          ;BRANCH IF NO TYPEOUT DESIRED
529 001704 004537 004352          JSR    5,ASOACNV  ;CONVERT LINE
530 001710 001646          LINE          ;NUMBER
531 001712 012514          ALINE          ;TO ASCII
532 001714 000002          2
533 001716 104015          ERROR1
534 001720 104000          TYPE          ;TYPE LINE # AS PART
535 001722 012505          LINEM        ;OF ERROR MESSAGE
536 001724 000002          1$: RTI          ;EXIT.
537
538          ;ERROR SERVICE ROUTINE CALLED BY TRAP (HLT)
539 001726 012737 000402 002026 ERR:   MOV    #402,ERRB ;MOV BR .+6 TO ERRB
540 001734 013737 001640 001642          MOV    ASPCADD,ASAPCADD ;GET PC WHERE ERROR OCCURRED
541 001742 000410          BR    ERRA
542 001744 012737 000240 002026 ERR1:  MOV    #240,ERRB ;MOVE NOP TO ERRB
543 001752 013737 001640 001642          MOV    ASPCADD,ASAPCADD ;GET PC WHERE ERROR OCCURRED
544 001760 004737 002050          JSR    7,ASCNVAT  ;CONVERT RCVDAT & XMTDAT TO ASCII
545 001764 104017          ERR2:  KBDIN
546 001766 032777 020000 177106 BIT    #BIT13,ASWR ;CHECK FOR IG
547 001774 001017          BNE    ERRC      ;ERROR PRINTOUT DESIRED
548 001776 004537 004352          JSR    5,ASOACNV ;BRANCH IF NO PRINTOUT
549 002002 001642          APCADD      ;CONVERT
550 002004 013174          APC          ;DATA
551 002006 000006          6          ;TO
552 002010 004537 004352          JSR    5,ASOACNV ;ASCII
553 002014 001522          RTNNO
554 002016 013164          ATNUMB
555 002020 000003          3          ;FOR
556 002022 104000          TYPE          ;PRINTOUT
557 002024 013161          EMO          ;TYPE ERROR
558 002026 000000          ERRB:  OPEN    ;MESSAGE
559 002030 104000          TYPE          ;NOP IF ERROR1, BR .+6 IF ERROR
560 002032 012440          ERDAT      ;TYPE ANOTHER MESSAGE
561 002034 005777 177042          ERRC:  TST    ASWR  ;IF ERROR 1
562 002040 100001          BPL    1$        ;HALT ON ERROR
563 002042 000000          HALT        ;GO TO EXIT IF NO HALT ON ERROR
564 002044 104017          1$:  KBDIN      ;HALT
565 002046 000002          RTI          ;CHECK FOR IG
566          ;RETURN
567
568          ;SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE
569          ;IN MESSAGE.
570 002050 004537 004352          CNVDAT: JSR    5,OACNV
571 002054 001632          XMTDAT
572 002056 012460          AASB
573 002060 000006          6
574 002062 004537 004352          JSR    5,OACNV

```

574 002066 001630  
575 002070 012475  
576 002072 000006  
577 002074 000207  
578  
579

RCVDAT  
AWAS  
6  
RTS 7 ;EXIT



```

00000000 00000000 012706 001100 START: MOV #SPBOT,%6 ;INITIALIZE STACK
00000001 00000001 104016 SUSWR ;CHECK FOR HARDWARE SWITCH REGISTER
00000002 00000002 004737 003764 JSR 7,%DMPAR ;GET DM11 PARAMETERS
00000003 00000003 004737 003312 JSR 7,%OVLAY ;PUT HALT,+2 IN VECTOR AREA
00000004 00000004 012706 001100 RSTAT1: MOV #SPBOT,%6 ;INITIALIZE STACK
00000005 00000005 104000 TYPE
00000006 00000006 012520 NO
00000007 00000007 004737 003346 JSR 5,RECD ;GET THE PRGNUM &
00000008 00000008 000000 PRGNUM: 0 ;PUT IT HERE
00000009 00000009 043737 001532 002130 BIC PRGLIM,PRGNUM ;MASK OFF UNUSED BITS
00000010 00000010 006327 002130 ASL PRGNUM ;SHIFT PROGRAM #
00000011 00000011 012700 002130 MOV PRGNUM,%D ;GET PROGRAM #
00000012 00000012 000170 001534 JMP @PRGTAB(0) ;GO START PROGRAM
00000013 00000013 012706 001100 RSTATE2: MOV #SPBOT,%6 ;INITIALIZE STACK
00000014 00000014 013700 002130 MOV PRGNUM,%D ;GET PROGRAM #
00000015 00000015 000170 001544 JMP @RSTART(0) ;GO RESTART PROGRAM
00000016 00000016 022727 000176 001102 SRSET: CMP #SWREG,SWR
00000017 00000017 001404 BEQ 1$
00000018 00000018 104000 TYPE ;TYPE OPTIONS MESSAGE
00000019 00000019 012570 M3
00000020 00000020 000000 HALT ;WAIT FOR USER TO SET OPTIONS
00000021 00000021 000401 BR GETRDY
00000022 00000022 104020 1$: CNTLU
00000023 00000023 013737 001516 001524 GETRDY: MOV KSTART,NXTST ;ADDR OF 1ST ROUTINE TO NXTST
00000024 00000024 012737 000006 000004 GTRDYX: MOV #6,%EARRVEC ;RESET ERROR TRAP.
00000025 00000025 005037 177776 CLR PSW
00000026 00000026 012706 001100 MOV #SPBOT,%6 ;SET BOTTOM OF STACK.
00000027 00000027 000005 RESET ;ISSUE RESET.
00000028 00000028 004737 002462 GTRDYA: JSR %7,FORMD ;ROLL FORWARD TO "NEXT" ROUTINE.
00000029 00000029 032777 001000 176630 BIT #BIT9,%SWR ;CHECK SELECT ROUTINE SWITCH
00000030 00000030 001003 BNE GTRDYC ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
00000031 00000031 000177 177240 JMP @COURTST ;GO RUN CURRENT ROUTINE.
00000032 00000032 000452 BR ;NO GO, MANUAL RTN BYPASSED.
00000033 00000033 017700 176614 GTRDYC: MOV %SWR,%D ;(SR) TO RD
00000034 00000034 042700 177600 BIC #177600,%D ;MASK UNDESIRED BITS
00000035 00000035 123700 001522 CMPB RTNNO,%D ;COMPARE RTNNO TO (RD)
00000036 00000036 001002 BNE GTRDYD ;BRANCH IF ROUTINE NOT FOUND YET.
00000037 00000037 000177 177214 JMP @COURTST ;GO RUN ROUTINE.
00000038 00000038 022737 177777 001524 GTRDYD: CMP #-1,NXTST ;NO, CHECK FOR LAST ROUTINE.
00000039 00000039 001352 BNE GTRDYA ;BRANCH IF NOT LAST ROUTINE.
00000040 00000040 004737 001652 JSR %7,INCRTN ;YES, INCORRECT ROUTINE SELECTED.
00000041 00000041 000734 BR GETRDY ;START OVER.

;SCOPE ROUTINE (CALLED BY EMT INST.)
ESCOPE: NOP
00000042 00000042 005077 176720 CLR %CSR ;INITIALIZE
00000043 00000043 005077 176720 CLR %BKCSR ;THE
00000044 00000044 005077 176712 CLR %BAR ;DM11
00000045 00000045 104017 KBDIN
00000046 00000046 012777 001106 176706 MOV #CAT,%BASREG
00000047 00000047 032777 040000 176524 BIT #BIT14,%SWR ;CHECK FOR SCOPE OPTION.
00000048 00000048 001403 BEQ SCOPEB ;BRANCH IF SCOPE SW NOT SET.
00000049 00000049 013716 001530 SCOPEA: MOV SCOPEB ;SET UP TO RETURN TO ROUTINE.
00000050 00000050 000002 RTI ;RETURN TO ROUTINE.
00000051 00000051 032777 004000 176506 SCOPEB: BIT #BIT11,%SWR ;TEST INHIBIT ITERATION SWITCH

```

636	002374	001003		BNE	SCOPEC	:BRANCH IF INHIBIT ITERATION SW SET.
637	002376	005337	001526	DEC	ICTR	:DECREMENT ITERATION COUNT.
638	002402	001366		BNE	SCOPEA	:BRANCH IF COUNT NOT 0.
639	002404	022626		SCOPEC: POPSP2		:POP STACK TWICE
640	002406	032777	001000 176466	SCOPED: BIT	%BIT9,%SWR	:CHECK SELECT ROUTINE SWITCH
641	002414	001276		BNE	GETRDY	:BRANCH IF SELECT RTN SW SET
642	002416	022737	177777 001524	CMP	%-1,%NXTST	:LAST TEST?
643	002424	001275		BNE	GTCDYX	:BRANCH IF NOT LAST TEST.
644	002426	104000		TYPE		:TYPE
645	002430	012543		M2		:PRGEND
646	002432	013702	000042	MOV	%42,%2	:CHECK DDP/ACT11 MONITOR HOOK
647	002436	001665		BEQ	GETRDY	
648	002440	000005		RESET		
649	002442	000240		NOP		
650	002444	000240		NOP		
651	002446	000240		NOP		
652	002450	000240		NOP		
653	002452	004712		SENDAD: JSR	7,(2)	:RETURN TO DDP/ACT11 MONITOR
654	002454	000240		NOP		
655	002456	000240		NOP		
656	002460	000240		NOP		
657	002462	013705	001524	FORWD: MOV	NXTST,%5	:ADDR OF NEXT ROUTINE TO RS.
658	002466	012537	001520	MOV	(5)+,%RTNNO	:GET NEXT ROUTINE NUMBER.
659	002470	012537	001524	MOV	(5)+,%NXTST	:GET ADDR OF NEXT "NEXT" ROUTINE.
660	002476	012537	001526	MOV	(5)+,%ICTR	:GET ITERATION COUNT.
661	002502	012537	001530	MOV	(5)+,%SCOPTA	:GET SCOPE LOOP ENTRY POINTER.
662	002506	010537	001520	MOV	%5,%CURTST	:ADDR OF NOW CURRENT TEST TO CURTST.
663	002512	000207		RTS	%7	:EXIT FORWD SUBROUTINE.
664	002514	011646		:EMT TRAP INTERPRETER		
665	002516	162716	000002	EMTINT: MOV	(6)-,%6	:GET PC OF NEXT INSTRUCTION
666	002522	011637	001640	SUB	%2,%6	:POINT SP TO PC OF EMT
667	002526	017616	000000	MOV	(6),%PCADD	:GET PC OF EMT CALL
668	002532	105066	000001	MOV	3(6),%6	:GET EMT CALL
669	002536	006316		CLRB	1(6)	:STRIP EMT & SAVE IDENTIFIER
670	002540	062716	001554	ASL	(6)	:SHIFT IDENTIFIER LEFT
671	002544	013607		ADD	%EMTTAB,%6	
672				MOV	3(6)+,%7	:GO TO PROPER EMT
673				:SAVE REGS 0 TO 4 SUBROUTINE.		
674	002546	012637	002602	SAVRG: MOV	(6)+,%1	:SAVE PC AND PSW.
675	002552	012637	002604	MOV	(6)+,%2	
676	002556	010446		MOV	%4,-,%6	:SAVE REGS 0 - 4
677	002560	010346		MOV	%3,-,%6	:IN STACK.
678	002562	010246		MOV	%2,-,%6	
679	002564	010146		MOV	%1,-,%6	
680	002566	010046		MOV	%0,-,%6	
681	002570	013746	002604	MOV	%5,-,%6	:RESTORE PC AND PSW.
682	002574	013746	002602	MOV	15,-,%6	
683	002600	000002		RTI		:EXIT.
684	002602	000000		15: OPEN		:CONTAINS SAVED PC
685	002604	000000		25: OPEN		:CONTAINS SAVED PSW
686				:RESTORE REGS 0 TO 4 SUBROUTINE.		
687	002606	000240		RSTRG: NOP		



```

699 002610 012637 002644      MOV      (6)+,1$      ;SAVE PC AND PSW.
699 002614 012637 002646      MOV      (6)+,2$
699 002620 012600      MOV      (6)+,%0      ;RESTORE REGS 0 - 4
699 002622 012601      MOV      (6)+,%1      ;FROM STACK.
699 002624 012602      MOV      (6)+,%2
699 002626 012603      MOV      (6)+,%3
699 002630 012604      MOV      (6)+,%4
699 002632 013746 002646      MOV      2$,-(6)      ;RESTORE PC AND PSW.
700 002636 013746 002644      MOV      1$,-(6)
701 002642 000002      RTI
702 002644 000000      1$:      OPEN          ;CONTAINS SAVED PC
703 002646 000000      2$:      OPEN          ;CONTAINS SAVED PSW

;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
706 002650 000240      TYP:      NOP
707 002652 011600      MOV      (SP),%0      ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
708 002654 062716 000002      ADD      #2,(SP)      ;SET UP EXIT.
709 002660 011000      MOV      3%0,%0      ;ADDRESS OF MESSAGE TO RD.
710 002662 112037 002762      1$:      MOV      (0)+,5$      ;GET CHARACTER
711 002666 122737 000100 002762      CMP      #100,5$      ;CHECK FOR"3"CHARACTER
712 002674 001001      BNE      2$          ;BRANCH IF NOT"3".
713 002676 000002      RTI          ;TERMINATOR CHAR. DONE. EXIT.
714 002700 122737 000045 002762      2$:      CMP      #45,5$      ;CHECK FOR"%".
715 002706 001412      BEQ      4$          ;BRANCH IF"%".
716 002710 004737 002716      JSR      %7,3$      ;TYPE CHAR IN 5$
717 002714 000762      BR
718 002716 113777 002762 176702      3$:      MOV      5$,JTPDBR      ;OUTPUT CHARACTER TO PRINTER
719 002724 105777 176674      TST      JTPCSR      ;WAIT FOR DONE FLAG.
720 002730 100375      BPL      -4
721 002732 000207      RTS      %7          ;EXIT
722 002734 112737 000015 002762      4$:      MOV      #15,5$      ;MOVE CARRIAGE RETURN CODE TO 5$
723 002742 004737 002716      JSR      %7,J#3$      ;GO TYPE CHAR.
724 002746 112737 000012 002762      MOV      #12,5$      ;MOVE LF CODE TO 5$.
725 002754 004737 002716      JSR      %7,3$      ;GO TYPE CHAR.
726 002760 000740      BR      1$
727 002762 000000      5$:      OPEN

;SUBROUTINE TO GET DM11 PARAMETERS
730 002764 000240      DMPAR:  ;VECTOR ADDRESS
731 002766 004737 003312      NOP
732 002772 104000      JSR      7,OVRLAY      ;BEGIN
733 002774 012310      TYPE      WHERE      ;ASK USER FOR RECEIVER INT. VECTOR
734 002776 004537 003346      JSR      5,RECD      ;OF UNIT UNDER TEST
735 003002 000000      VECTOR:  ;GET THE VECTOR &
736 003004 005737 003002      D          ;PUT IT HERE
737 003010 001003      TST      VECTOR
738 003012 012737 000300 003002      BNE      1$
739 003020 023727 003002 000300      1$:      MOV      #300,VECTOR      ;SET VECTOR = TO 0300
740 003026 103003      CMP      VECTOR,#300      ;IS VECTOR HIGHER OR
741 003030 104000      BHS      2$          ;EQUAL TO 0300
742 003032 012540      TYPE      ;TYPE "??"
743 003034 000753      MI
744 003036 023727 003002 000770      3$:      BR      DMPAR      ;ASK FOR ANOTHER VECTOR
745 003044 101371      CMP      VECTOR,#770      ;IS VECTOR = TO OR
746 003046 032737 000007 003002      BHI      2$          ;LESS THAN 770
747 003046 032737 000007 003002      BIT      #7,VECTOR      ;LSB OF VECTOR MUST BE ALL 0'S

```

```

748 003054 001365          BNE      2$
749 003056 013737 003002 001260      MOV     VECTOR,3#CLKINT
750 003064 062737 000004 003002      ADD     #4,VECTOR
751 003072 013737 003002 001254      MOV     VECTOR,3#XMTINT
752
753
754 003100 104000          :UNIT NUMBER
755 003102 012403      DMPARB: TYPE
756 003104 004537 003346          WHICH
757 003110 000000      JSR     5,RECD          :GET THE UNIT 3
758 003112 023727 003110 000017      UNIT: 0              :PUT IT HERE
759 003120 101403      CMP     UNIT,#17
760 003122 104000      BLOS   1$
761 003124 012540      MI
762 003126 000764      BR     DMPARB
763 003130 006337 003110      1$: ASL   UNIT
764 003134 006337 003110      ASL   UNIT
765 003140 006337 003110      ASL   UNIT
766 003144 012702 000004      MOV     #4,%2
767 003150 012701 001250      MOV     #CSR,%1
768 003154 042711 000370      2$: BIC   #370,(1)
769 003160 063721 003110      ADD     UNIT,(1)+
770 003164 005302      DEC    %2
771 003166 001372      BNE    2$
772
773          :CALCULATE CHARACTER LENGTH
774 003170 012777 001106 176060      MOV     #CAT,3BASREG
775 003176 005077 176046      CLR     3CSR
776 003202 012737 177777 006622      MOV     #-1,OUTBUF      :LOAD OUTBUF WITH CHAR TO BE TRANSMITTED
777 003210 012737 177777 001146      MOV     #-1,WCT         :SET UP TO TRANSMIT 1 CHAR
778 003216 012737 177777 001634      MOV     #-1,3#CARMSK    :PRE SET THE CHARACTER MASK
779 003224 012737 006622 001106      MOV     #OUTBUF,CAT     :1 CHARACTER ON LINE 0
780 003232 012777 003270 176020      MOV     #3$,3CLKINT    :LOAD RECEIVER INTERRUPT
781 003240 012777 000340 176014      MOV     #340,3CLKLVL   :AND PRIORITY LEVEL
782 003246 005037 001306      CLR     TUMTAB
783 003252 012777 000001 175772      MOV     #1,3BAR        :START TRANSMITTING
784 003260 012777 000105 175762      MOV     #BIT6+BIT2+BIT0,3CSR :SET IE,MAINT AND GO BITS
785 003266 000001          :WAIT FOR RECEIVER INTERRUPT
786 003270 005077 175754      3$: CLR   3CSR
787 003274 143737 001306 001634      BICB   TUMTAB,CARMSK   :LOAD CHARACTER LENGTH MASK
788 003302 005037 177776      CLR     PSW            :RESTORE PROCESSOR TO PRIORITY 0
789 003306 022626          :RESTORE THE STACK POINTER
790 003310 000207          :EXIT PARAMETERS ROUTINE
791
792          :ROUTINE TO LOAD TRAP/INTERRUPT VECTOR AREA WITH HALT,..+2. HALTS PROGRAM
793          :AT ADDRESS OF TRAP/INTERRUPT VECTOR +2.
794 003312 012701 000300      OVLAY: MOV    #300,%1
795 003316 012702 000302      MOV    #302,%2
796 003322 010221      1$: MOV    %2,(1)+
797 003324 005021      CLR    (1)+
798 003326 020227 000776      CMP    %2,#776
799 003332 001403      BEQ    2$
800 003334 062702 000004      ADD    #4,%2
801 003340 000770      BR     1$
802 003342 000240      2$: NOP
803 003344 000207      RTS    7              :EXIT

```



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

003346 010046  
003350 005015  
003352 012737 000007 003524  
003353 105777 176234  
003364 100375  
003366 117700 176230  
003372 142700 000200  
003376 110077 176224  
003402 122700 000025  
003406 001443  
003410 122700 000015  
003414 001415  
003416 142700 000060  
003422 132700 000110  
003426 001031  
003430 006315  
003432 006315  
003434 006315  
003436 150015  
003440 005337 003524  
003444 001422  
003446 000744  
003450 105777 176150  
003454 100375  
003456 012777 000012 176142  
003464 105777 176134  
003470 100375  
003472 005077 176130  
003476 105777 176122  
003502 100375  
003504 005725  
003506 012600  
003510 000205  
003512 104000  
003514 012540  
003516 104000  
003520 012400  
003522 000712  
003524 000000

:SUBROUTINE TO RECEIVE DATA  
:THIS SUBROUTINE RECEIVES DATA FROM THE KEYBOARD (UP TO SIX OCTAL  
:DIGITS AND PLACES THEM INTO THE ADDRESS FOLLOWING THE SUBROUTINE  
:CALL (JSR 5,RECD). NO REGISTER CONTENTS ARE DISTURBED.

:SUBROUTINE TO INPUT DATA FROM TTY

RECD: MOV R0, -(SP)  
1\$: CLR (5) ;CLEAR OLD DATA  
MOV #7, CNT ;SET CHAR COUNT  
2\$: TSTB @TKCSR ;WAIT FOR CHAR  
BPL 2\$  
MOVB @TKDBR, R0  
BICB #20C, R0 ;STRIP OFF PARITY  
MOVB R0, @TPDBR ;ECHO CHARACTER  
CMPB #25, R0 ;IS IT A ^U  
BEQ 5\$ ;BRANCH IF YES  
CMPB #15, R0 ;IS IT A ^CR  
BEQ 6\$ ;BRANCH IF YES  
BICB #60, R0  
BITB #110, R0 ;CHECK FOR 0-7 (8)  
BNE 7\$ ;BRANCH IF NOT  
ASL (5)  
ASL (5)  
ASL (5) ;SHIFT DATA  
BISB R0, (5) ;INSET NEW CHAR  
DEC CNT  
BEQ 7\$ ;ONLY 6 CHAR'S PLEASE  
BR 2\$ ;NEXT CHARACTER  
6\$: TSTB @TPCSR  
BPL 6\$ ;WAIT FOR READY  
MOV #12, @TPDBR ;TYPE ^LF  
8\$: TSTB @TPCSR  
BPL 8\$ ;WAIT FOR READY  
CLR @TPDBR ;NEXT CHARACTER  
9\$: TSTB @TPCSR  
BPL 9\$ ;WAIT FOR READY  
TST (R5)+ ;ADJUST R5  
MOV (SP)+, R0 ;RESTORE R0  
RTS R5  
7\$: TYPE  
M1  
5\$: TYPE  
\$CTLU  
BR 1\$ ;START OVER  
CNT: 0

:ROUTINE TO CHECK FOR ^G BEING TYPED

003526 022737 000176 001102 KBDINTT: CMP #SWREG, SWR  
003534 001015 BNE 1\$  
003536 005037 003630 CLR TMP1 ;CLEAR TEMP AREA  
003542 117737 176054 003630 MOVB @TKDBR, TMP1 ;FETCH THE BUFFER

```

860 003550 142737 000200 003630 BICB #200,TMP1 ;STRIP OFF PARITY
861 003556 122737 000007 003630 CMPB #7,TMP1 ;WAS IT 1G
862 003564 001001 BNE 1$ ;NOP
863 003566 104020 CNTLU ;GO CHANGE IT
864 003570 000002 1$: RTI ;EXIT
865
866
867 ;ROUTINE TO CHANGE CONTENTS OF SWREG(LOC 176)
868
869 003572 022737 000176 001102 CNTLUU: CMP #SWREG,SWR
870 003600 001023 BNE FAJAG
871 003602 104000 TYPE
872 003604 012351 $SWREG
873 003606 004537 004352 JSR RS,0ACNV ;CONVERT TO ASCII
874 003612 000176 SWREG
875 003614 012360 $VALUE
876 003616 000006 6
877 003620 104000 TYPE
878 003622 012360 $VALUE
879 003624 004537 003346 JSR 5,RECD ;GET THE TMP1 &
880 003630 000000 TMP1: 0 ;PUT IT HERE
881 003632 022737 000007 003524 CMP #7,CNT
882 003640 001403 BNE FAJAG
883 003642 013777 003630 175232 MOV TMP1,2SWR ;CHANGE CONTENTS OF SWREG
884 003650 000002 FAJAG: RTI
885
886
887 003652 013746 000006 SUSWRR: MOV 2#6,-(SP) ;SAVE VECTORS
888 003656 013746 000004 MOV 2#4,-(SP)
889 003662 012737 003702 000004 MOV #1$,2#4 ;SET UP FOR TIMEOUT
890 003670 022777 177777 175204 CMP #-1,2SWR ;REFERENCE HARDWARE SWITCH REGISTER
891 003676 001402 BEQ 2$
892 003700 000407 BR 3$
893 003702 022626 1$: CMP (SP)+,(SP)+ ;ADJUST STACK
894 003704 012737 000176 001102 2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
895 003712 012737 000174 001104 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
896 003720 012637 000004 3$: MOV (SP)+,2#4 ;RESTORE VECTORS
897 003724 012637 000006 MOV (SP)+,2#6
898 003730 000002 RTI

```



```

900      :SUBROUTINE TO TRANSMIT ON ALL LINES WITH A DELAY BETWEEN TRANSMITTING
901      :ON SUCCESSIVE LINES. THE DELAY FOR THE TEST IS SUPPLIED BY THE
902      :CALLING JSR INSTRUCTION. DATA IS CHECKED AFTER ALL
903      :LINES HAVE FINISHED TRANSMITTING.
904
905      003732 000240      DLYXMT: NOP      ;BEGIN TEST
906      003734 012777 001106 175314  MOV      #CAT,@BASREG ;SET UP BASE REGISTER
907      003742 004737 004264      JSR      7,@#IDENT ;TRANSMIT LINE # ON EACH LINE
908      003746 000240      NOP      ;NOP
909      003750 005077 175274      CLR      @CSR      ;GET MESSAGE ADDRESS
910      003754 012537 003764      MOV      (5)+,10$ ;LOAD OUTPUT BUFFER
911      003760 004537 004440      JSR      5,@#BMOVE ;WITH DATA TO
912      003764 000000      10$: OPEN ;BE TRANSMITTED
913      003766 006622      OUTBUF
914      003770 000100      64.
915      003772 005037 001306      CLR      @#TUMTAB ;CLEAR TUMBLE
916      003776 004537 004440      JSR      5,@#BMOVE ;TABLE (200
917      004002 001306      TUMTAB ;BYTES)
918      004004 001307      TUMTAB+1
919      004006 000177      177
920      004010 004537 004440      JSR      5,@#BMOVE ;CLEAR CHARACTER COUNT TABLE
921      004014 001306      TUMTAB
922      004016 012126      CNTTAB
923      004020 000020      16.
924      004022 005037 006766      CLR      @#LNOBUF ;CLEAR ALL
925      004026 004537 004440      JSR      5,@#BMOVE ;LINE'S INPUT
926      004032 006766      LNOBUF ;BUFFERS
927      004034 006767      LNOBUF+1 ;(16. BUFFERS OF 100. CHARS. EACH)
928      004036 003077      1599.
929      004040 022737 000006 002130  CMP      #6,PRGNUM
930      004046 001002      BNE      +6
931      004050 000137 006322      JMP      PRG3A
932      004054 012504      MOV      (5)+,%4 ;GET # OF CHARACTERS TO TRANSMIT BEFORE
933      :TRANSMITTING ON NEXT LINE
934      004056 012737 001306 001300  MOV      #TUMTAB,@#TTPTR ;INITIALIZE TUMBLE TABLE POINTER
935      004064 013701 001260      MOV      @#CLKINT,%1 ;GET RECEIVER VECTOR ADDRESS
936      004070 012721 004666      MOV      #RINT,(1)+ ;LOAD RECEIVER VECTOR
937      004074 013721 001262      MOV      @#CLKLVL,(1)+ ;AND PRIORITY LEVEL
938      004100 012721 005060      MOV      #TINT,(1)+ ;LOAD TRANSMITTER VECTOR
939      004104 013721 001266      MOV      @#XMTLVL,(1)+ ;AND PRIORITY LEVEL
940      004110 005737 002130      TST      PRGNUM ;RUNNING PROGRAM 0?
941      004114 001402      BEQ      +6
942      004116 000137 006126      JMP      PRG1A ;RETURN TO PROGRAM 1 CODE
943      004122 012777 010101 175120  MOV      #BIT12+BIT6+BIT0,@CSR ;SET IE & GO BITS
944      004130 012737 000001 001274  MOV      #1,@#LINBIT
945      004136 005037 001646      CLR      @#LINE
946      004142 013700 001646      1$: MOV      LINE,%0 ;LINE # X2 TO R0
947      004146 000240      NOP      ;NOP
948      004150 004537 004462      JSR      5,@#XMITD ;TRANSMIT 64 CHARACTERS
949      004154 177700      -64. ;ON LINE # AS SPECIFIED IN ADDRESS LINE
950      004156 020460 001146      2$: CMP      %4,WCT(0) ;WAIT FOR THE WORD COUNT TO DEC TO THE
951      004162 001375      BNE      2$ ;CORRECT VALUE BEFORE STARTING NEXT LINE
952      004164 062737 000002 001646  ADD      #2,LINE ;FORM NEXT LINE NUMBER
953      004172 006337 001274      ASL      LINBIT ;SHIFT LINE BIT
954      004176 103361      BCC      1$ ;START NEXT LINE
955      004200 005760 001146      3$: TST      WCT(0) ;WAIT FOR LAST LINE TO FINISH

```

```

956 004204 001375          BNE      3$
957 004206 042777 177400 175034    BIC      #177400, @CSR      ;CLEAR ODD BYTE OF CSR
958 004214 062700 000001    31$:    ADD      #1, R0          ;WAIT FOR RECEIVER TO RECEIVE
959 004220 001375          BNE      31$              ;ALL TRANSMITTED DATA
960 004222 017737 175024 001630    MOV      @BAR, RCVDAT      ;GET AND TEST BAR CONTENTS
961 004230 001410          BEQ      4$              ;BRANCH IF IS CLEAR
962
963 004232 005037 001632          CLR      XMTDAT
964 004236 005077 175006          CLR      @CSR
965 004242 005077 175004          CLR      @BAR
966 004246 104015          ERROR1   ;ERROR! BAR DID NOT CLEAR IN SUFICIENT TIME
967 004250 000403          BR       5$              ;EXIT TEST
968 004252 000240    4$:    NOP
969 004254 004737 004560    JSR      7, @CHKDAT        ;GO TEST DATA
970 004260 022626    5$:    CMP      (6)+, (6)+      ;RESET THE STACK
971 004262 104012          SCOPE      ;SCOPE
972
973
974 ;SUBROUTINE TO TRANSMIT ON EACH LINE ITS LINE NUMBER (CRLF YX CRLF).
975 004264 005037 001646    IDENT:  CLR      @#LINE      ;GET LINE NUMBER 0
976 004270 012737 000001 001274    MOV      #1, @#LINBIT      ;GET LINE BIT
977 004276 013702 001646    1$:    MOV      LINE, %2
978 004302 016262 012146 001106    MOV      ID(2), CAT(2)      ;LOAD CAT
979 004310 012762 177772 001146    MOV      #-6, WCT(2)      ;LOAD WORD COUNT
980 004316 053777 001274 174726    BIS      LINBIT, @BAR      ;SET BAR BIT
981 004324 062737 000002 001646    ADD      #2, LINE          ;FORM NEXT LINE NUMBER
982 004332 006337 001274          ASL      LINBIT           ;FORM NEXT LINE BIT
983 004336 103357          BCC      1$              ;BRANCH IF NOT DONE
984 004340 005777 174706    2$:    TST      @BAR          ;WAIT FOR BAR TO CLEAR
985 004344 001375          BNE      2$
986 004346 000240          NOP
987 004350 000207          RTS      7              ;EXIT SUBROUTINE
988
989 ;OCTAL TO ASCII CONVERT ROUTINE
990 004352 104013    OACNV:  SAVREG
991 004354 013537 004436          MOV      @ (5)+, 2$      ;SAVE REGISTERS ON THE STACK
992 004360 012501          MOV      (5)+, %1        ;GET OCTAL VALUE.
993 004362 012502          MOV      (5)+, %2        ;GET DESTINATION ADDR.
994 004364 060201          ADD      %2, %1          ;GET CONVERT COUNT.
995 004366 013703 004436    1$:    MOV      2$, %3          ;DEVELOP ADDR TO STORE 1ST CHAR.
996 004372 042703 177770          BIC      #177770, %3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
997 004376 062703 000060          ADD      #60, %3         ;CONVERT DIGIT TO ASCII.
998 004402 110341          MOVB    %3, -(1)         ;STORE ASCII CHARACTER.
999 004404 042737 000007 004436    BIC      #7, 2$
1000 004412 006037 004436          ROR      2$
1001 004416 006037 004436          ROR      2$
1002 004422 006037 004436          ROR      2$
1003 004426 005302          DEC      %2
1004 004430 001356          BNE      1$              ;DONE ALL DIGITS?
1005 004432 104014          RSTREG
1006 004434 000205          RTS      %5              ;BRANCH IF NOT DONE.
1007 004436 000000    2$:    OPEN
1008
1009
1010
1011 ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.

```



1012	004440	104013
1013	004442	012501
1014	004444	012502
1015	004446	012503
1016	004450	112122
1017	004452	005303
1018	004454	001375
1019	004456	104014
1020	004460	000205

BMOVE:	SAVREG		:SAVE REGS.
	MOV	(5)+,%1	:GET"FROM"ADDRESS
	MOV	(5)+,%2	:GET"TO"ADDRESS
	MOV	(5)+,%3	:GET COUNT
1\$:	MOVB	(1)+,(2)+	:MOVE BYTE
	DEC	%3	:DECREMENT COUNT
	BNE	1\$	:BRANCH IF NOT DONE.
	RSTREG		:RESTORE REGS.
	RTS	%5	:DONE EXIT

```

1021
1022
1023      ;SUBROUTINE TO TRANSMIT DATA.  SUBROUTINE CALLED BY
1024      ;JSR 5,XMITD
1025      XMITD:  NOP
1026      MOV     %0,-(SP)      ;SAVE RO ON THE STACK
1027      MOV     @#LINE,%0    ;GET LINE
1028      MOV     #OUTBUF,CAT(0) ;LOAD FIRST CHAR ADDRESS IN CAT
1029      MOV     (5)+,WCT(0)   ;LOAD WORD COUNT INTO LINE'S TABLE ADDRESS
1030      BIS     @#LINBIT,@#BARIM ;LOAD LINE POSITION INTO BAR IMAGE
1031      BIS     LINBIT,@BAR   ;START TRANSMITTING ON LINE SPECIFIED
1032      ;IN LINBIT
1033      MOV     (SP)+,%0     ;RESTORE RO
1034      NOP
1035      RTS     5            ;EXIT
1036
1037      ;SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
1038      GTLINB: MOV    %0,-(SP) ;SAVE RO ON THE STACK
1039      CLR     @#LINBIT
1040      MOV     @#LINE,%0     ;GET LINE
1041      SEC
1042      ;SET CARRY
1043      1$:  ROL     LINBIT    ;SHIFT LINE BIT
1044      SUB     #2,%0        ;SUBTRACT 2 FROM LINE NUMBER
1045      BPL     1$
1046      MOV     (SP)+,%0     ;RESTORE RO
1047      RTS     7            ;EXIT
1048
1049      ;SUBROUTINE TO CHECK TRANSMITTED DATA
1050      CHKDAT: SAVREG      ;SAVE THE REGISTERS ON THE STACK
1051      NOP
1052      CLR     %1          ;CLEAR CHARACTER COUNT
1053      MOV     #INTAB,%2   ;GET ADDRESS OF LINE'S INPUT BUFFER
1054      CLR     %3          ;ADDRESS ;GET LINE COUNT
1055      1$:  MOV     %3,@#LINE ;MOVE LINE # TO LINE
1056      MOV     (2)+,@#LINBUF ;GET LINE'S INPUT BUFFER ADDRESS
1057      DEC     LINBUF      ;SUBTRACT 1 FROM LINE'S INPUT BUFFER ADDRESS
1058      2$:  INC     LINBUF   ;INCREMENT LINE'S INPUT BUFFER ADDRESS
1059      MOVB   @LINBUF,@#RCV DAT ;GET RECEIVED CHARACTER
1060      MOVB   OUTBUF(1),XMTDAT ;GET TRANSMITTED CHARACTER
1061      BIC   @#CARMSK,XMTDAT ;CLEAR UNTRANSMITTED BITS
1062      DATCHK
1063      INC     %1          ;INCREMENT CHARACTER COUNT
1064      CMP     %1,#64.    ;ALL CHARACTERS BEEN COMPARED
1065      BNE     2$        ;GO CHECK NEXT CHAR. IF NOT
1066      CLR     %1          ;CLEAR CHARACTER COUNT
1067      INC     %3          ;INCREMENT LINE COUNT
1068      CMP     %3,#16.   ;ALL LINES CHECKED?
1069      BLT     1$        ;BRANCH IF ALL LINES NOT CHECKED
1070      RSTREG
1071      RTS     7            ;EXIT SUBROUTINE
1072
1073
1074      ;RECEIVER INTERRUPT SERVICE ROUTINE
1075      RINT:  NOP          ;BEGIN
1076      SAVREG ;SAVE THE REGISTERS ON THE STACK

```



```

1077 004672 013701 001300      MOV      @#TTPTR,%1      ;GET TUMBLE TABLE POINTER
1078 004676 011137 001272      MOV      (1),TTDAT      ;GET TUMBLE TABLE ENTRY
1079 004702 100410                BMI      2$              ;BRANCH IF VALID DATA ENTRY
1080 004704 104003                ERROR    ;ERROR! FALSE INTERRUPT
1081 004706 000454                BR       6$              ;EXIT
1082 004710 011137 001272      1$:     MOV      (1),@#TTDAT ;GET TUMBLE TABLE ENTRY
1083 004714 001451                BEQ      6$              ;GO TO EXIT IF NO DATA ENTRY
1084 004716 100402                BMI      2$              ;BRANCH IF VALID DATA ENTRY
1085 004720 104003                ERROR    ;ERROR! NO VALID DATA ENTRY INDICATOR
1086 004722 000425                BR       3$              ;EXIT
1087 004724 005011                CLR      (1)             ;CLEAR TUMBLE TABLE ENTRY
1088 004726 042737 160400 001272    2$:     BIC      #160400,@#TTDAT ;CLEAR ALL BUT CHAR. & LINE #
1089 004734 113702 001273        MOVVB   TTDAT+1,%2      ;PUT LINE # IN R2 (LINE WILL BE IN LSH)
1090 004740 010204                MOV      %2,%4
1091 004742 016237 012066 001650    MOV      INTAB(2),@#LINBUF ;GET LINE'S INPUT BUFFER ADDRESS
1092 004750 006202                ASR      %2              ;SHIFT LINE #
1093 004752 005003                CLR      %3
1094 004754 116203 012126        MOVVB   CNTTAB(2),%3    ;GET LINE'S RECEIVED CHAR. COUNT
1095 004760 105262 012126        INCB    CNTTAB(2)      ;INCREMENT CHARACTER COUNT
1096 004764 060337 001650        ADD     %3,LINBUF       ;FORM ADDRESS WHERE CHAR. IS TO BE STORED
1097 004770 113777 001272 174652    MOVVB   TTDAT,@LINBUF  ;STORE CHAR. IN LINE'S INPUT BUFFER
1098 004776 000240                NOP
1099 005000 016437 001146 001630    3$:     MOV      WCT(4),RCVDAT  ;GET TRANSMITTERS WORD COUNT
1100 005006 003405                BLE     4$              ;BRANCH IF WORD COUNT IS 0 OR NEGATIVE
1101 005010 010437 001632        MOV      %4,XMTDAT      ;GET LINE # OF FAILING LINE
1102 005014 104015                ERROR1  ;ERROR! INCORRECT WORD COUNT IN
1103                ;TYPE OUT SHOWS FAILING LINE #, AND FAILING LINE'S WORD COUNT
1104 005016 000005                RESET
1105 005020 104012                SCOPE
1106                ;STOP THE DM11
1107                ;EXIT TEST
1107 005022 022701 001504      4$:     CMP      #TUMTAB+176,%1 ;IS THE TUMBLE TABLE POINTER AT THE
1108 005026 001002                BNE     5$              ;THE END OF THE TABLE
1109 005030 012701 001304        MOV      #TUMTAB-2,%1  ;RESET POINTER
1110 005034 005721                5$:     TST     (1)+           ;INCREMENT POINTER
1111 005036 000724                BR      1$              ;GO CHECK NEXT ENTRY
1112 005040 042777 000200 174202    6$:     BIC      #BIT7,@CSR    ;CLEAR RECEIVER DONE FLAG
1113 005046 010137 001300        MOV      %1,TTPTR      ;SAVE POINTER
1114 005052 104014                RSTREG  ;RESTORE THE REGISTERS
1115 005054 000240                NOP
1116 005056 000002                RTI
1117                ;EXIT SERVICE ROUTINE
1118                ;TRANSMITTER INTERRUPT SERVICE ROUTINE
1119 005060 000240                †INT:  NOP
1120 005062 032777 060000 174160    BIT     #BIT14+BIT13,@CSR ;TEST ERROR FLAGS
1121 005070 001404                BEQ     1$              ;BRANCH IF NO ERROR FLAGS
1122 005072 104003                ERROR   ;ERROR! ERROR FLAG IS SET
1123 005074 042777 060000 174146    BIC     #BIT14+BIT13,@CSR ;CLEAR ERROR FLAGS
1124 005102 005777 174142      1$:     TST     @CSR          ;TEST READY FLAG
1125 005106 100003                BPL     2$              ;BRANCH IF READY IS CLEAR
1126 005110 042777 100000 174132    BIC     #BIT15,@CSR    ;CLEAR READY FLAG
1127 005116 000002                2$:     RTI
1128
1129

```

```

1130
1131 005120 104000
1132 005122 012661
1133 005124 012737 005160 0C1516 PRGO: TYPE
1134 005132 005037 001522 PRGOM
1135 005136 000137 002170 PRGOA: MOV #RTO,KSTART ;GET ADDRESS OF FIRST TEST
1136 005142 012737 005160 001516 PRGOR: CLR RTNNO ;CLEAR ROUTINE #
1137 005150 005037 001522 PRGOR: JMP SRSET ;GET ADDRESS OF FIRST TEST
1138 005154 000137 002212 PRGOR: MOV #RTO,KSTART ;GET ADDRESS OF FIRST TEST
1139 *****
1140 005160 000000 RT0: 0 ;ROUTINE # 0 *
1141 005162 005200 RT1 ;ADDR OF NEXT ROUTINE. *
1142 005164 000002 2 ;ITERATION COUNT *
1143 005166 005170 RTOA ;SCOPE ENTRY POINT. *
1144 000000 X=X+1
1145 *****
1146 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1147 ;NEXT LINE.
1148 005170 004537 003732 RTOA: JSR 5,DLYXMT ;GO DO TEST.
1149 005174 013203 MSG1 ;TRANSMIT THIS MESSAGE &
1150 005176 000000 0 ;DELAY THIS MUCH BETWEEN LINES
1151 *****
1152 005200 000001 RT1: 1 ;ROUTINE # 1 *
1153 005202 005220 RT2 ;ADDR OF NEXT ROUTINE. *
1154 005204 000002 2 ;ITERATION COUNT *
1155 005206 005210 RT1A ;SCOPE ENTRY POINT. *
1156 000001 X=X+1
1157 *****
1158 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1159 ;NEXT LINE.
1160 005210 004537 003732 RT1A: JSR 5,DLYXMT ;GO DO TEST.
1161 005214 013203 MSG1 ;TRANSMIT THIS MESSAGE &
1162 005216 177740 -32. ;DELAY THIS MUCH BETWEEN LINES
1163 *****
1164 005220 000002 RT2: 2 ;ROUTINE # 2 *
1165 005222 005240 RT3 ;ADDR OF NEXT ROUTINE. *
1166 005224 000002 2 ;ITERATION COUNT *
1167 005226 005230 RT2A ;SCOPE ENTRY POINT. *
1168 000002 X=X+1
1169 *****
1170 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1171 ;NEXT LINE.
1172 005230 004537 003732 RT2A: JSR 5,DLYXMT ;GO DO TEST.
1173 005234 013203 MSG1 ;TRANSMIT THIS MESSAGE &
1174 005236 177720 -48. ;DELAY THIS MUCH BETWEEN LINES
1175 *****
1176 005240 000003 RT3: 3 ;ROUTINE # 3 *
1177 005242 005260 RT4 ;ADDR OF NEXT ROUTINE. *
1178 005244 000002 2 ;ITERATION COUNT *
1179 005246 005250 RT3A ;SCOPE ENTRY POINT. *
1180 000003 X=X+1
1181 *****
1182 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1183 ;NEXT LINE.
1184 005250 004537 003732 RT3A: JSR 5,DLYXMT ;GO DO TEST.
1185 005254 013203 MSG1 ;TRANSMIT THIS MESSAGE &

```





005430  
005434  
005436  
005440  
005444  
005448  
005452  
005456  
005460  
005464  
005468  
005472  
005476  
005480  
005484  
005488  
005492  
005496  
005500  
005504  
005508  
005512  
005516  
005520

004537  
013304  
177710  
000013  
005460  
000014  
005500  
000002  
005470  
000014  
004537  
013304  
177704  
000014  
004537  
013304  
177702  
000015  
005520

003732  
003732  
003732  
003732  
003732  
003732  
003732  
003732

```

:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT10A: JSR      5,DLYXMT      :GO DO TEST.
      MSG2      :TRANSMIT THIS MESSAGE &
      -32.      :DELAY THIS MUCH BETWEEN LINES
*****
RT11:      11      :ROUTINE # 11
      RT12      :ADDR OF NEXT ROUTINE.
      RT11A     :ITERATION COUNT
      X=X+1     :SCOPE ENTRY POINT.
*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT11A: JSR      5,DLYXMT      :GO DO TEST.
      MSG2      :TRANSMIT THIS MESSAGE &
      -48.      :DELAY THIS MUCH BETWEEN LINES
*****
RT12:      12      :ROUTINE # 12
      RT13      :ADDR OF NEXT ROUTINE.
      RT12A     :ITERATION COUNT
      X=X+1     :SCOPE ENTRY POINT.
*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT12A: JSR      5,DLYXMT      :GO DO TEST.
      MSG2      :TRANSMIT THIS MESSAGE &
      -56.      :DELAY THIS MUCH BETWEEN LINES
*****
RT13:      13      :ROUTINE # 13
      RT14      :ADDR OF NEXT ROUTINE.
      RT13A     :ITERATION COUNT
      X=X+1     :SCOPE ENTRY POINT.
*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT13A: JSR      5,DLYXMT      :GO DO TEST.
      MSG2      :TRANSMIT THIS MESSAGE &
      -60.      :DELAY THIS MUCH BETWEEN LINES
*****
RT14:      14      :ROUTINE # 14
      RT15      :ADDR OF NEXT ROUTINE.
      RT14A     :ITERATION COUNT
      X=X+1     :SCOPE ENTRY POINT.
*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT14A: JSR      5,DLYXMT      :GO DO TEST.
      MSG2      :TRANSMIT THIS MESSAGE &
      -62.      :DELAY THIS MUCH BETWEEN LINES
*****
RT15:      15      :ROUTINE # 15
      RT16      :ADDR OF NEXT ROUTINE.

```



```

1298 005504 000002          2          : ITERATION COUNT          *
1299 005506 005510          RT15A          : SCOPE ENTRY POINT.      *
1300 000016          X=X+1
1301 *****
1302 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1303 : NEXT LINE.
1304 RT15A: JSR      5,DLYXMT          : GO DO TEST.
1305 005510 004537 003732          : TRANSMIT THIS MESSAGE 3
1306 005514 013304          MSG2          : DELAY THIS MUCH BETWEEN LINES
1307 005516 177701          -63.
1308 *****
1309 RT16:  16          : ROUTINE # 16            *
1310 005520 000016          RT17          : ADDR OF NEXT ROUTINE.  *
1311 005522 005540          2          : ITERATION COUNT        *
1312 005524 000002          RT16A         : SCOPE ENTRY POINT.     *
1313 005526 005530          X=X+1
1314 *****
1315 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1316 : NEXT LINE.
1317 RT16A: JSR      5,DLYXMT          : GO DO TEST.
1318 005530 004537 003732          : TRANSMIT THIS MESSAGE 3
1319 005534 013304          MSG2          : DELAY THIS MUCH BETWEEN LINES
1320 005536 177700          -64.
1321 *****
1322 RT17:  17          : ROUTINE # 17            *
1323 005540 000017          RT20          : ADDR OF NEXT ROUTINE.  *
1324 005542 005560          2          : ITERATION COUNT        *
1325 005544 000002          RT17A         : SCOPE ENTRY POINT.     *
1326 005546 005550          X=X+1
1327 *****
1328 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1329 : NEXT LINE.
1330 RT17A: JSR      5,DLYXMT          : GO DO TEST.
1331 005550 004537 003732          : TRANSMIT THIS MESSAGE 3
1332 005554 013404          MSG3          : DELAY THIS MUCH BETWEEN LINES
1333 005556 177720          -48.
1334 *****
1335 RT20:  20          : ROUTINE # 20            *
1336 005560 000020          RT21          : ADDR OF NEXT ROUTINE.  *
1337 005562 005600          2          : ITERATION COUNT        *
1338 005564 000002          RT20A         : SCOPE ENTRY POINT.     *
1339 005566 005570          X=X+1
1340 *****
1341 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1342 : NEXT LINE.
1343 RT20A: JSR      5,DLYXMT          : GO DO TEST.
1344 005570 004537 003732          : TRANSMIT THIS MESSAGE 3
1345 005574 013404          MSG3          : DELAY THIS MUCH BETWEEN LINES
1346 005576 177704          -60.
1347 *****
1348 RT21:  21          : ROUTINE # 21            *
1349 005600 000021          RT22          : ADDR OF NEXT ROUTINE.  *
1350 005602 005620          2          : ITERATION COUNT        *
1351 005604 000002          RT21A         : SCOPE ENTRY POINT.     *
1352 005606 005610          X=X+1
1353 *****
1354 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1355 : NEXT LINE.
1356 RT21A: JSR      5,DLYXMT          : GO DO TEST.
1357 005610 004537 003732          : TRANSMIT THIS MESSAGE 3
1358 005614 013404          MSG3

```

```

1354 005616 177701
1355
1356 005620 000022
1357 005622 005640
1358 005624 000002
1359 005626 005630
1360 000022
1361
1362
1363
1364 005630 004537 003732
1365 005634 013404
1366 005636 177700
1367
1368 005640 000023
1369 005642 005660
1370 005644 000002
1371 005646 005650
1372 000023
1373
1374
1375
1376 005650 004537 003732
1377 005654 013504
1378 005656 177740
1379
1380 005660 000024
1381 005662 005700
1382 005664 000002
1383 005666 005670
1384 000024
1385
1386
1387
1388 005670 004537 003732
1389 005674 013504
1390 005676 177710
1391
1392 005700 000025
1393 005702 005720
1394 005704 000002
1395 005706 005710
1396 000025
1397
1398
1399
1400 005710 004537 003732
1401 005714 013504
1402 005716 177702
1403
1404 005720 000026
1405 005722 005740
1406 005724 000002
1407 005726 005730
1408 000026
1409

```

```

-53. ;DELAY THIS MUCH BETWEEN LINES
*****
RT22: 22 ;ROUTINE # 22 *
RT23 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT22A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT22A: JSR 5,DLYXMT ;GO DO TEST.
MSG3 ;TRANSMIT THIS MESSAGE &
-64. ;DELAY THIS MUCH BETWEEN LINES
*****
RT23: 23 ;ROUTINE # 23 *
RT24 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT23A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT23A: JSR 5,DLYXMT ;GO DO TEST.
MSG4 ;TRANSMIT THIS MESSAGE &
-32. ;DELAY THIS MUCH BETWEEN LINES
*****
RT24: 24 ;ROUTINE # 24 *
RT25 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT24A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT24A: JSR 5,DLYXMT ;GO DO TEST.
MSG4 ;TRANSMIT THIS MESSAGE &
-56. ;DELAY THIS MUCH BETWEEN LINES
*****
RT25: 25 ;ROUTINE # 25 *
RT26 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT25A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT25A: JSR 5,DLYXMT ;GO DO TEST.
MSG4 ;TRANSMIT THIS MESSAGE &
-62. ;DELAY THIS MUCH BETWEEN LINES
*****
RT26: 26 ;ROUTINE # 26 *
RT27 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT26A ;SCOPE ENTRY POINT. *
X=X+1
*****

```



```

1410
1411
1412 005730 004537 003732
1413 005734 013504
1414 005736 177700
1415
1416 005740 000027
1417 005742 005760
1418 005744 000002
1419 005746 005750
1420 000027
1421
1422
1423
1424 005750 004537 003732
1425 005754 013604
1426 005756 177720
1427
1428 005760 000030
1429 005762 006000
1430 005764 000002
1431 005766 005770
1432 000030
1433
1434
1435
1436 005770 004537 003732
1437 005774 013604
1438 005776 177710
1439
1440 006000 000031
1441 006002 006021
1442 006004 000002
1443 006006 006010
1444 000031
1445
1446
1447
1448 006010 004537 003732
1449 006014 013604
1450 006016 177704
1451
1452 006020 000032
1453 006022 006040
1454 006024 000002
1455 006026 006030
1456 000032
1457
1458
1459
1460 006030 004537 003732
1461 006034 013604
1462 006036 177702
1463
1464 006040 000033
1465 006042 006060

```

```

:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT26A: JSR      5,DLYXMT      ;GO DO TEST.
      MSG4      ;TRANSMIT THIS MESSAGE &
      -64.      ;DELAY THIS MUCH BETWEEN LINES
:*****
RT27:   27      ;ROUTINE # 27 *
      RT30      ;ADDR OF NEXT ROUTINE. *
      2         ;ITERATION COUNT *
      RT27A     ;SCOPE ENTRY POINT. *
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT27A: JSR      5,DLYXMT      ;GO DO TEST.
      MSG5      ;TRANSMIT THIS MESSAGE &
      -48.      ;DELAY THIS MUCH BETWEEN LINES
:*****
RT30:   30      ;ROUTINE # 30 *
      RT31      ;ADDR OF NEXT ROUTINE. *
      2         ;ITERATION COUNT *
      RT30A     ;SCOPE ENTRY POINT. *
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT30A: JSR      5,DLYXMT      ;GO DO TEST.
      MSG5      ;TRANSMIT THIS MESSAGE &
      -56.      ;DELAY THIS MUCH BETWEEN LINES
:*****
RT31:   31      ;ROUTINE # 31 *
      RT32      ;ADDR OF NEXT ROUTINE. *
      2         ;ITERATION COUNT *
      RT31A     ;SCOPE ENTRY POINT. *
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT31A: JSR      5,DLYXMT      ;GO DO TEST.
      MSG5      ;TRANSMIT THIS MESSAGE &
      -60.      ;DELAY THIS MUCH BETWEEN LINES
:*****
RT32:   32      ;ROUTINE # 32 *
      RT33      ;ADDR OF NEXT ROUTINE. *
      2         ;ITERATION COUNT *
      RT32A     ;SCOPE ENTRY POINT. *
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT32A: JSR      5,DLYXMT      ;GO DO TEST.
      MSG5      ;TRANSMIT THIS MESSAGE &
      -62.      ;DELAY THIS MUCH BETWEEN LINES
:*****
RT33:   33      ;ROUTINE # 33 *
      RT34      ;ADDR OF NEXT ROUTINE. *

```

1466 006044 000002  
 1467 006046 006050  
 1468 000033  
 1469  
 1470  
 1471  
 1472 006050 004537 003732  
 1473 006054 013604  
 1474 006056 177701  
 1475  
 1476 006060 000034  
 1477 006062 177777  
 1478 006064 000002  
 1479 006066 006070  
 1480 000034  
 1481  
 1482  
 1483  
 1484 006070 004537 003732  
 1485 006074 013604  
 1486 006076 177700  
 1487 177777

```

      2                               : ITERATION COUNT          *
      RT33A                           : SCOPE ENTRY POINT.      *
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT33A: JSR      5,DLYXMT             : GO DO TEST.
      MSG5                               : TRANSMIT THIS MESSAGE &
      -63.                               : DELAY THIS MUCH BETWEEN LINES
:*****
RT34:  34                               : ROUTINE # 34            *
      RT35                               : ADDR OF NEXT ROUTINE.  *
      2                                   : ITERATION COUNT        *
      RT34A                              : SCOPE ENTRY POINT.    *
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT34A: JSR      5,DLYXMT             : GO DO TEST.
      MSG5                               : TRANSMIT THIS MESSAGE &
      -64.                               : DELAY THIS MUCH BETWEEN LINES
RT35=-1

```



```

1488
1489
1490
1491 006100 104000
1492 006102 012707
1493 006104 022737 000176 001102
1494 006112 001001
1495 006114 104020
1496 006116 004537 003732
1497 006122 013203
1498 006124 177700
1499 006126 012737 006622 001106
1500 006134 004537 004440
1501 006140 001106
1502 006142 001110
1503 006144 000040
1504 006146 012737 177700 001146
1505 006154 004537 004440
1506 006160 001146
1507 006162 001150
1508 006164 000040
1509 006166 012777 010100 173054
1510 006174 023727 002130 000004
1511 006202 001403
1512 006204 052777 000001 173036
1513 006212 012777 177777 173032
1514 006220 005777 173026
1515 006224 001375
1516 006226 005205
1517 006230 001376
1518 006232 005077 173012
1519 006236 023727 002130 000004
1520 006244 001402
1521 006246 004737 004560
1522 006252 104000
1523 006254 012543
1524 006256 012706 001100
1525 006262 000715
1526
1527
1528
1529
1530
1531
1532 006264 104000
1533 006266 012767
1534 006270 000137 006104

```

```

:PRG1- DATA TESTS ALL LINES SIMULTANEOUSLY. DATA TRANSMITTED IS 'THE
:QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
PRG1: TYPE :TYPE
PRG1M :PROGRAM TITLE
PRGX: CMP #SWREG,SWR :SEE IF SWITCH-LESS
BNE PRG1R :BRANCH IF NOT
CNTLU :GET SWREG SETTINGS
PRG1R: JSR 5,DLYXMT :GO TO DLYXMT TO SET UP DM11
MSG1 :MSG1 WILL BE THE DATA TRANSMITTED
-64. :DO NOT DELAY
PRG1A: MOV #OUTBUF,CAT :LOAD CURRENT
JSR 5,BMOVE :ADDRESS TABLE
CAT :TO POINT TO
CAT+2 :OUTBUF
32.
MOV #-64.,WCT :LOAD WORD COUNT
JSR 5,BMOVE :TO -64.
WCT
WCT+2
32.
MOV #BIT12+BIT6,DCSR :SET TRANSMITTER & RECIEVER TO BITS
CMP PRGNUM,#4 :RUNNING PROGRAM #2?
BEQ +10
BIS #BIT0,DCSR :SET THE GO BIT
MOV #-1,DBAR :START TRANSMITTING ON ALL LINES
TST DBAR :WAIT FOR ALL LINES TO COMPLETE
BNE -4
INC %5
BNE -2
CLR DCSR
PRG1C: CMP PRGNUM,#4 :DO NOT CHECK DATA IF RUNNING
BEQ PRG1D :PROGRAM # 2
JSR 7,CHKDAT :GO CHECK RECEIVED DATA
PRG1D: TYPE :TYPE
M2 :'PRGEND'
PRG1EX: MOV #SPBOT,SP :RESET THE STACK POINTER
BR PRG1R :GO RESTART TEST

```

```

:PRG2-PROGRAM 2 RUNS PROGRAM 1 EXCEPT FOR THE DATA CHECKING
:WHEN ALL LINES ARE FINISHED TRANSMITTING. THIS ALLOWS THE DATA
:TRANSMITTED TO BE SENT TO TERMINALS. BEFORE STARTING THIS PROGRAM
:REMOVE THE JUMPERS CONNECTING THE TRANSMITTERS TO THE RECEIVERS.
PRG2: TYPE :TYPE PROGRAM TITLE
PRG2M :AND INSTRUCTIONS
PRG2R: JMP PRGX :GO RUN PRG1

```

```

1535
1536
1537
1538
1539
1539
1540
1541 006274 104000
1542 006276 013016
1543 006300 022737 000176 001102
1544 006306 001001
1545 006310 104020
1546 006312 004537 003732
1547 006316 013203
1548 006320 000240
1549 006322 012737 001306 001300
1550 006330 013701 001260
1551 006334 012721 006372
1552 006340 013721 001262
1553 006344 012721 006576
1554 006350 013721 001266
1555 006354 012777 010101 172666
1556 006362 012700 000001
1557 006366 005200
1558 006370 000776
1559
1560
1561 006372 000240
1562 006374 000240
1563 006376 013701 001300
1564 006402 011137 001272
1565 006406 001463
1566 006410 005011
1567 006412 032737 040000 001272
1568 006420 001047
1569 006422 042737 160400 001272
1570 006430 113702 001273
1571 006434 010237 001646
1572 006440 004737 004526
1573 006444 033777 001274 172600
1574 006452 001414
1575 006454 033777 001274 172570
1576 006462 001374
1577 006464 032777 060000 172556
1578 006472 001401
1579 006474 104003
1580 006476 042777 100000 172544
1581 006504 113762 001272 006622
1582 006512 012762 177777 001146
1583 006520 010203
1584 006522 062703 006622
1585 006526 010362 001106
1586 006532 053777 001274 172512
1587 006540 022701 001504
1588 006544 001002
1589 006546 012701 001304
1590 006552 005721

:PRG3-ECHO TEST THIS PROGRAM ECHOS BACK DATA RECEIVED FROM ANY DM11
:TERMINAL(S)
:NOTE: THIS TEST IS THE ONLY TEST THAT INSURES PROPER OPERATION
:OF THE DM11 DISTRIBUTION PANEL LOGIC.

PRG3: TYPE PRG3M ;TYPE PROGRAM
      CMP #SWREG,SWR ;SEE IF SWITCH-LESS
      BNE PRG3R ;BRANCH IF NOT
      CNTLU ;GET SWREG SETTINGS
PRG3R: JSR 5,DLYXMT ;USE PART OF THE
      MSG1 ;DLYXMT ROUTINE TO
      NOP ;SET UP DM11
PRG3A: MOV #TUMTAB, TTPTR ;INITIALIZE SOFTWARE POINTER
      MOV CLKINT,%1 ;LOAD RECEIVER
      MOV #RINT3,(1)+ ;AND TRANSMITTER
      MOV CLKLVL,(1)+ ;VECTORS AND PRIORITY
      MOV #TINT3,(1)+ ;LEVELS
      MOV XMTLVL,(1)+
      MOV #BIT12+BIT6+BIT0,DCSR ;SET IE AND GO BITS
      MOV #1,%0
      INC %0
      BR -2

RINT3: NOP
      NOP
      MOV TTPTR,%1 ;GET SOFTWARE POINTER
RINT3A: MOV (1),TTDAT ;GET TUMBLE TABLE ENTRY
      BEQ RINT3X ;EXIT IF NO ENTRY
      CLR (1) ;CLEAR ENTRY
      BIT #BIT14,TTDAT ;WAS BREAK RECEIVED
      BNE RINT3B ;DO NOTHING ABOUT IT
      BIC #160400,TTDAT ;CLEAR ALL BUT LINE # AND DATA
      MCVB TTDAT+1,%2 ;GET LINE NUMBER
      MOV %2,LINE ;FETCH LINE NUMBER
      JSR 7,GTLINEB ;FORM LINE BIT FOR BAR
      BIT LINBIT,ABAR ;IS THIS LINE ACTIVE
      BEQ NONACT ;LINE NOT ACTIVE
      BIT LINBIT,ABAR ;WAIT FOR LINE
      BNE -6
      BIT #BIT14+BIT13,DCSR
      BEQ .+4 ;BRANCH IF NO ERRORS
      ERROR
      BIC #BIT15,DCSR ;CLEAR TRANSMIT DONE
NONACT: MOVB TTDAT,OUTBUF(2) ;STORE RECEIVED CHARACTER
      MOV #-1,WCT(2) ;LOAD LINE'S WORD COUNT
      MOV %2,%3
      ADD #OUTBUF,%3
      MOV %3,CAT(2) ;AND CURRENT ADDRESS
RINT3B: BIS LINBIT,ABAR ;ECHO RECEIVED CHARACTER
      CMP #TUMTAB+176,%1 ;CHECK TUMBLE
      BNE .+6 ;TABLE POINTER
      MOV #TUMTAB-2,%1
      TST (1)+

```



1591	006554	000712			BR	RINT3A	
1592	006556	042777	000200	172464	RINT3X: BIC	#BIT7,DCSR	;CLEAR CHARACTER DONE FLAG
1593	006564	010137	001300		MOV	%1,TTPTR	;RESTORE POINTER
1594	006570	000240			NOP		
1595	006572	000240			NOP		
1596	006574	000002			RTI		;EXIT
1597							
1598	006576	000240			TINT3: NOP		
1599	006600	032777	060000	172442	BIT	#BIT14+BIT13,DCSR	;ANY ERROR FLAGS SET
1600	006606	001401			BEQ	.+4	
1601	006610	104003			ERROR		
1602	006612	042777	160000	172430	BIC	#BIT15+BIT14+BIT13,DCSR	;CLEAR ALL FLAGS
1603	006620	000002			RTI		;EXIT

1604 006622 000000  
1605 006766 006766  
1606 006766 000000  
1607 007132 007132  
1608 007132 000000  
1609 007276 007276  
1610 007276 000000  
1611 007442 007442  
1612 007442 000000  
1613 007606 007606  
1614 007606 000000  
1615 007752 007752  
1616 007752 000000  
1617 010116 010116  
1618 010116 000000  
1619 010262 010262  
1620 010262 000000  
1621 010426 010426  
1622 010426 000000  
1623 010572 010572  
1624 010572 000000  
1625 010736 010736  
1626 010736 000000  
1627 011102 011102  
1628 011102 000000  
1629 011246 011246  
1630 011246 000000  
1631 011412 011412  
1632 011412 000000  
1633 011556 011556  
1634 011556 000000  
1635 011722 011722  
1636 011722 000000  
1637 012066 012066  
1638 012066 006766  
1639 012070 007132  
1640 012072 007276  
1641 012074 007442  
1642 012076 007606  
1643 012100 007752  
1644 012102 010116  
1645 012104 010262  
1646 012106 010426  
1647 012110 010572  
1648 012112 010736  
1649 012114 011102  
1650 012116 011246  
1651 012120 011412  
1652 012122 011556  
1653 012124 011722  
1654 012126 000000  
1655 012146 012146  
1656  
1657 012146 012206  
1658 012150 012212  
1659 012152 012216

OUTBUF: 0  
          .=OUTBUF+100.  
LNOBUF: 0  
          .=LNOBUF+100.  
LN1BUF: 0  
          .=LN1BUF+100.  
LN2BUF: 0  
          .=LN2BUF+100.  
LN3BUF: 0  
          .=LN3BUF+100.  
LN4BUF: 0  
          .=LN4BUF+100.  
LN5BUF: 0  
          .=LN5BUF+100.  
LN6BUF: 0  
          .=LN6BUF+100.  
LN7BUF: 0  
          .=LN7BUF+100.  
LN10BF: 0  
          .=LN10BF+100.  
LN11BF: 0  
          .=LN11BF+100.  
LN12BF: 0  
          .=LN12BF+100.  
LN13BF: 0  
          .=LN13BF+100.  
LN14BF: 0  
          .=LN14BF+100.  
LN15BF: 0  
          .=LN15BF+100.  
LN16BF: 0  
          .=LN16BF+100.  
LN17BF: 0  
          .=LN17BF+100.  
INTAB: LNOBUF  
         LN1BUF  
         LN2BUF  
         LN3BUF  
         LN4BUF  
         LN5BUF  
         LN6BUF  
         LN7BUF  
         LN10BF  
         LN11BF  
         LN12BF  
         LN13BF  
         LN14BF  
         LN15BF  
         LN16BF  
         LN17BF  
CNTTAB: 0  
          .=CNTTAB+16.  
ID: IDENT0  
     IDENT1  
     IDENT2



1660 012154 012222  
1661 012156 012226  
1662 012160 012232  
1663 012162 012236  
1664 012164 012242  
1665 012166 012246  
1666 012170 012252  
1667 012172 012256  
1668 012174 012262  
1669 012176 012266  
1670 012200 012272  
1671 012202 012276  
1672 012204 012302  
1673 012206 105215  
1674 012210 030060  
1675 012212 105215  
1676 012214 030460  
1677 012216 105215  
1678 012220 031060  
1679 012222 105215  
1680 012224 031460  
1681 012226 105215  
1682 012230 032060  
1683 012232 105215  
1684 012234 032460  
1685 012236 105215  
1686 012240 033060  
1687 012242 105215  
1688 012244 033460  
1689 012246 105215  
1690 012250 030061  
1691 012252 105215  
1692 012254 030461  
1693 012256 105215  
1694 012260 031061  
1695 012262 105215  
1696 012264 031461  
1697 012266 105215  
1698 012270 032061  
1699 012272 105215  
1700 012274 032461  
1701 012276 105215  
1702 012300 033061  
1703 012302 105215  
1704 012304 033461  
1705 012306 105215  
1706 105215  
1707

IDENT3  
IDENT4  
IDENT5  
IDENT6  
IDENT7  
IDNT10  
IDNT11  
IDNT12  
IDNT13  
IDNT14  
IDNT15  
IDNT16  
IDNT17  
IDENT0: CRLF  
"00  
IDENT1: CRLF  
"01  
IDENT2: CRLF  
"02  
IDENT3: CRLF  
"03  
IDENT4: CRLF  
"04  
IDENT5: CRLF  
"05  
IDENT6: CRLF  
"06  
IDENT7: CRLF  
"07  
IDNT10: CRLF  
"10  
IDNT11: CRLF  
"11  
IDNT12: CRLF  
"12  
IDNT13: CRLF  
"13  
IDNT14: CRLF  
"14  
IDNT15: CRLF  
"15  
IDNT16: CRLF  
"16  
IDNT17: CRLF  
"17  
CRLF  
CRLF=105215

1708						
1709						:MESSAGES
1710	012310	042045	0305'5	020061		WHERE: .ASCII '%DM11 RECEIVER VECTOR ADDRESS = @'
1711	012316	042522	042503	053111		
1712	012324	051105	053040	041505		
1713	012332	047524	020122	042101		
1714	012340	051104	051505	020123		
1715	012346	020075	100			
1716	012351	045	053523	036522		\$SWREG: .ASCII '%SWR= @'
1717	012356	040040				
1718	012360	020040	020040	020040		\$VALUE: .ASCII ' NEW= @'
1719	012366	020040	020040	042516		
1720	012374	036527	040040			
1721	012400	036445	040			\$CTLU: .ASCII '%= '
1722	012403	045	044127	041511		WHICH: .ASCII '%WHICH DM11 ARE YOU TESTING @'
1723	012410	020110	046504	030461		
1724	012416	040440	042522	054440		
1725	012424	052517	052040	051505		
1726	012432	044524	043516	040040		
1727	012440	042045	052101	020101		ERDAT: .ASCII '%DATA ERR S/B: '
1728	012446	051105	020122	051440		
1729	012454	041057	020072			
1730	012460	020040	020040	020040		AASB: .ASCII ' WAS: '
1731	012466	020040	040527	035123		
1732	012474	040				
1733	012475	040	020040	020040		AWAS: .ASCII ' @'
1734	012502	020040	100			
1735	012505	114	047111	020105		LINEM: .ASCII 'LINE # '
1736	012512	020043				
1737	012514	020040	040040			ALINE: .ASCII ' @'
1738	012520	052045	050131	020105		MO: .ASCII '%TYPE PROGRAM #@'
1739	012526	051120	043517	040522		
1740	012534	020115	040043			
1741	012540	037445	100			M1: .ASCII '%?@'
1742	012543	045	042524	052123		M2: .ASCII '%TEST DZDMB COMPLETE@'
1743	012550	042040	042132	041115		
1744	012556	041440	046517	046120		
1745	012564	052105	040105			
1746	012570	051445	052105	051440		M3: .ASCII '%SET SR OPTIONS. NORMAL OPERATION'
1747	012576	020122	050117	044524		
1748	012604	047117	027123	047040		
1749	012612	051117	040515	020114		
1750	012620	050117	051105	052101		
1751	012626	047511	116			
1752	012631	123	020122	020075		.ASCII 'SR = 000000 PRESS CONT.@'
1753	012636	030060	030060	030060		
1754	012644	050040	042522	051523		
1755	012652	041440	047117	027124		
1756	012660	100				
1757	012661	045	040504	040524		PRGOM: .ASCII '%DATA TEST ALL LINES @'
1758	012666	052040	051505	020124		
1759	012674	046101	020114	044514		
1760	012702	042516	020123	100		
1761	012707	045	040504	040524		PRGIM: .ASCII '%DATA TEST TRANSMIT ON ALL LINES SIMULTANEOUSLY@'
1762	012714	052040	051505	020124		
1763	012722	051124	047101	046523		



1764	012730	052111	047440	020116		
1765	012736	046101	020114	044514		
1766	012744	042516	020123	044523		
1767	012752	052515	052114	047101		
1768	012760	047505	051525	054514		
1769	012766	100				
1770						
1771	012767	045	051124	047101	PRG2M:	.ASCII '%TRANSMIT TO TERMINALS@'
1772	012774	046523	052111	052040		
1773	013002	020117	042524	046522		
1774	013010	047111	046101	040123		
1775	013016	042445	044103	020117	PRG3M:	.ASCII '%ECHO TEST@'
1776	013024	042524	052123	100		
1777	013031	045	052520	020124	PRGI:	.ASCII '%PUT CHAR IN SR(0-7),DELAY IN SR(8-15)@'
1778	013036	044103	051101	044440		
1779	013044	020116	051123	030050		
1780	013052	033455	026051	042504		
1781	013060	040514	020131	047111		
1782	013066	051440	024122	026470		
1783	013074	032461	040051			
1784	013100	052045	050131	020105	POPPAR:	.ASCII '%TYPE PARITY OPTION (N=NOT DESIRED O=ODD, E=EVEN)@'
1785	013106	040520	044522	054524		
1786	013114	047440	052120	047511		
1787	013122	020116	047050	047075		
1788	013130	052117	042040	051505		
1789	013136	051111	042105	047440		
1790	013144	047475	042104	020054		
1791	013152	036505	053105	047105		
1792	013160	100				
1793	013161	045	020122		EMO:	.ASCII '%R '
1794	013164	020040	020040	041520	ATNUMB:	.ASCII ' PC= '
1795	013172	020075				
1796	013174	020040	020040	020040	APC:	.ASCII ' @'
1797	013202	100				
1798	013203	015	012		MSG1:	.BYTE 15,12
1799	013205	040	044124	020105		.ASCII ' THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
1800	013212	052521	041511	020113		
1801	013220	051102	053517	020116		
1802	013226	047506	020130	052512		
1803	013234	050115	042105	047440		
1804	013242	042526	020122	044124		
1805	013250	020105	040514	054532		
1806	013256	042040	043517	020123		
1807	013264	040502	045503	030440		
1808	013272	031462	032464	033466		
1809	013300	034470	060			
1810		013304				
1811	013304	015	012		MSG2:	.EVEN
1812	013306	177400				.BYTE 15,12
1813	013310	177400				177400
1814	013312	177400				177400
1815	013314	177400				177400
1816	013316	177400				177400
1817	013320	177400				177400
1818	013322	177400				177400
1819	013324	177400				177400

1820	013326	177400				177400
1821	013330	177400				177400
1822	013332	177400				177400
1823	013334	177400				177400
1824	013336	177400				177400
1825	013340	177400				177400
1826	013342	177400				177400
1827	013344	177400				177400
1828	013346	177400				177400
1829	013350	177400				177400
1830	013352	177400				177400
1831	013354	177400				177400
1832	013356	177400				177400
1833	013360	177400				177400
1834	013362	177400				177400
1835	013364	177400				177400
1836	013366	177400				177400
1837	013370	177400				177400
1838	013372	177400				177400
1839	013374	177400				177400
1840	013376	177400				177400
1841	013400	177400				177400
1842	013402	177400				177400
1843	013404	015	012	MSG3:	.BYTE	15,12
1844	013406	125252				ALTO
1845	013410	125252				ALTO
1846	013412	125252				ALTO
1847	013414	125252				ALTO
1848	013416	125252				ALTO
1849	013420	125252				ALTO
1850	013422	125252				ALTO
1851	013424	125252				ALTO
1852	013426	125252				ALTO
1853	013430	125252				ALTO
1854	013432	125252				ALTO
1855	013434	125252				ALTO
1856	013436	125252				ALTO
1857	013440	125252				ALTO
1858	013442	125252				ALTO
1859	013444	125252				ALTO
1860	013446	125252				ALTO
1861	013450	125252				ALTO
1862	013452	125252				ALTO
1863	013454	125252				ALTO
1864	013456	125252				ALTO
1865	013460	125252				ALTO
1866	013462	125252				ALTO
1867	013464	125252				ALTO
1868	013466	125252				ALTO
1869	013470	125252				ALTO
1870	013472	125252				ALTO
1871	013474	125252				ALTO
1872	013476	125252				ALTO
1873	013500	125252				ALTO
1874	013502	125252				ALTO
1875	013504	015	012	MSG4:	.BYTE	15,12





013666	000200	200	
013670	177377	177377	
013672	175775	175775	
013674	167767	167767	
013676	137737	137737	
013700	177500	177500	
013702	015	.BYTE	15.12
013704	015	.BYTE	15.12
000001		.END	











PRG1M	012707	1492	1761#					
PRG1R	006116	483	1494	1496#	1525			
PRG2	006264	480	1532#					
PRG2H	012767	1533	1771#					
PRG2R	006270	484	1534#					
PRG2W	006274	481	1541#					
PRG2D	006322	921	1543#					
PRG2H	013016	1542	1775#					
PRG2R	006312	485	1544	1546#				
PRTY0	000000	259#						
PRTY1	000040	258#						
PRTY2	000100	257#						
PRTY3	000140	256#						
PRTY4	000200	255#	468	470				
PRTY5	000240	254#	456	458				
PRTY6	000300	253#						
PRTY7	000340	252#	306					
PRVONT	001644	251#						
PSW	177776	229#	606*	788*				
RCV0AT	001630	509#	524	574	960*	1059*	1099*	
RFCD	003346	588#	735	756	913#	879		
RINT	004666	936	1075#					
RINT3	006372	1551	1561#					
RINT3A	006402	1564#	1591					
RINT3B	006540	1568#	1587#					
RINT3X	006556	1565#	1592#					
RSTAT1	001544	482#	596					
RSTAT2	002114	434	585#					
RSTAT3	002154	436	594#					
RSTRG0	104014	279#	1005	1019	1070	1114		
RSTRG	002606	498	691#					
RTRNO	001522	473#	553	616	659*	1134*	1137*	
RT0	005160	1133	1136	1140#				
RT0R	005170	1143	1148#					
RT1	005200	1141	1152#					
RT1A	005210	1155	1160#					
RT10R	005360	1225	1236#					
RT11R	005370	1239	1244#					
RT111R	005400	1237	1248#					
RT1111R	005410	1251	1256#					
RT11111R	005420	1243	1260#					
RT111111R	005430	1263	1268#					
RT1111111R	005440	1261	1272#					
RT11111111R	005450	1275	1280#					
RT111111111R	005460	1273	1284#					
RT1111111111R	005470	1287	1292#					
RT11111111111R	005500	1285	1296#					
RT111111111111R	005510	1299	1304#					
RT1111111111111R	005520	1297	1308#					
RT11111111111111R	005540	1311	1316#					
RT111111111111111R	005540	1309	1320#					
RT1111111111111111R	005550	1323	1328#					
RT11111111111111111R	005550	1153	1164#					
RT111111111111111111R	005550	1167	1172#					
RT1111111111111111111R	005550	1321	1322#					
RT11111111111111111111R	005550	1325	1340#					









GET	211#	598	735	756	879										
HEADER	211#	1139	1151	1163	1175	1187	1199	1211	1223	1235	1247	1259	1271	1283	1295
	1307	1319	1331	1343	1355	1367	1379	1391	1403	1415	1427	1439	1451	1463	1475
XMTDLY	211#	1139	1151	1163	1175	1187	1199	1211	1223	1235	1247	1259	1271	1283	1295
	1307	1319	1331	1343	1355	1367	1379	1391	1403	1415	1427	1439	1451	1463	1475

ADD	673	708	750	769	800	952	958	981	994	997	1096	1584			
ASL	591	672	763	764	765	828	929	830	953	982					
ASR	1092														
BCC	954	983													
BEQ	525	599	632	647	715	799	822	924	833	882	891	941	961	1083	1121
	1511	1520	1565	1574	1578	1600									
BHI	746														
BHIS	741														
BIC	590	615	768	957	996	999	1061	1088	1112	1123	1126	1569	1580	1592	1602
BICB	787	819	825	860											
BIS	980	1030	1031	1512	1586										
BISB	831														
BIT	527	546	610	631	635	640	747	1120	1567	1573	1575	1577	1599		
BITB	826														
BLE	1100														
BLOS	759														
BLT	1069														
BMI	1079	1084													
BNE	528	547	611	617	620	636	638	641	643	712	738	748	771	827	857
	862	870	930	951	956	959	985	1004	1018	1065	1108	1494	1515	1517	1544
	1568	1576	1588												
BPL	562	720	817	836	839	842	1044	1125							
BR	541	602	613	622	717	726	744	762	801	834	850	892	967	1081	1086
	1111	1525	1558	1591											
CLR	506	626	627	628	775	782	786	788	797	814	840	858	909	915	924
	945	963	964	965	975	1039	1052	1054	1066	1087	1093	1134	1137	1518	1566
CLRB	671														
CMP	597	619	642	740	745	758	798	856	869	881	890	893	929	950	970
	1064	1068	1107	1493	1510	1519	1543	1587							
CMPB	524	616	711	714	821	823	861								
DEC	637	770	832	1003	1017	1057									
EMT	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283
HALT	293	294	296	298	300	302	304	308	310	312	314	316	318	320	322
	324	326	328	330	332	334	336	338	340	342	344	346	348	350	352
	354	356	358	360	362	364	366	368	370	372	374	376	378	380	382
	384	386	388	390	392	394	396	398	400	402	404	406	408	410	412
	414	416	418	420	563	601									
INC	1058	1063	1067	1516	1557										
INCB	1095														
JMP	433	434	436	593	596	612	618	931	942	1135	1138	1534			
JSR	526	529	544	548	552	569	573	583	584	588	609	621	653	716	723
	725	732	735	756	873	879	907	911	916	920	925	948	969	1149	1160
	1172	1184	1196	1208	1220	1232	1244	1256	1268	1280	1292	1304	1316	1328	1340
	1352	1364	1376	1388	1400	1412	1424	1436	1448	1460	1472	1484	1496	1500	1505
	1521	1546	1572												
MOV	539	540	542	543	581	585	592	594	595	604	605	607	614	630	633
	646	658	659	660	661	662	663	667	669	670	674	677	678	679	680
	681	682	683	684	685	692	693	694	695	696	697	698	699	700	707
	709	739	749	751	766	767	774	776	777	778	779	780	781	783	784
	794	795	796	813	815	837	844	883	887	888	889	894	895	896	897
	906	910	932	934	935	936	937	938	939	943	944	946	960	976	977
	978	979	991	992	993	995	1013	1014	1015	1026	1027	1028	1029	1033	1038
	1040	1045	1053	1055	1056	1077	1078	1082	1090	1091	1099	1101	1109	1113	1133
	1136	1499	1504	1509	1513	1524	1549	1550	1551	1552	1553	1554	1555	1556	1563
	1564	1571	1582	1583	1585	1589	1593								
MOV8	710	718	722	724	818	820	859	998	1016	1059	1060	1089	1094	1097	1570



NOP	1591 625	649	650	651	652	654	655	656	691	706	731	802	905	908	947
RESET	968	986	1025	1034	1051	1075	1098	1115	1119	1548	1561	1562	1594	1595	1598
ROL	608	649	1104												
ROR	1042														
RTI	1000	1001	1002												
RTS	536	565	634	686	701	713	864	884	898	1116	1127	1596	1603		
SEC	521	577	664	721	790	803	845	987	1006	1020	1035	1046	1071		
SUB	1041														
TST	668	1043													
TSTB	561	737	843	940	955	984	1110	1124	1514	1590					
WAIT	719	816	835	838	841										
.ASCII	785														
.BYTE	1710	1716	1718	1721	1722	1727	1730	1733	1735	1737	1738	1741	1742	1746	1752
.ENABL	1757	1761	1771	1775	1777	1784	1793	1794	1796	1799					
.END	1798	1811	1843	1875	1907	1938	1939								
.EVEN	209														
.LIST	1940														
.MACR	1810														
.NLIST	208	211													
.REM	211														
.REPT	207	211													
.TITLE	3														
	1812	1844	1876	1908											
	206														

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DZDMBB.SEQ/SOL/CRF=DZDMBB.SRC  
 RUN-TIME: 5 12 2 SECONDS  
 RUN-TIME RATIO: 120/20=5.7  
 CORE USED: 8K (15 PAGES)

