

# DM11

DM11 DATA TST  
CZDMBD0

AH 8541D MC

COPYRIGHT 72 79  
FICHE 1 OF 1

NOV 1979

**digital**

MADE IN USA

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

.REM 2

IDENTIFICATION  
-----

PRODUCT CODE: AC-8540D-MC  
PRODUCT NAME: CZDMBD0 DM11 DATA TST  
PRODUCT DATE: JULY, 1979  
MAINTAINER: DIAGNOSTIC 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 DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1972,1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11 (ASYNCHRONOUS DATA MULTIPLEXER), CZDMA (DM11 LOGIC TESTS), AND CZDMB (DM11 MULTIPLE LINE DATA TESTS). THE LOGICC 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  
CZDMA\_\_ DM11 LOGIC TESTS

3. LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122

4. USE PROCEDURE  
4.1 STARTING PROCEDURE

BEFORE STARTING MAKE SURE THAT THE TTY IS IN REMOTE MODE.  
THREE STARTING ADDRESSES ARE PROVIDED.

0200 - THIS STARTING ADDRESS REQUESTS DM11 PARAMETERS, AND MUST  
BE USED TO INITIALLY START THE PROGRAM, AND WHENEVER ANY  
OF THE PARAMETERS LISTED BELOW IS CHANGED.

- A. VECTOR ADDRESS ?  
RESPONSE: TYPE IN THE VECTOR ADDRESS OF THE DM11 RECEIVER  
UNDER TEST. CARRIAGE RETURN SELECTS 0300
- B. UNIT #(8)?  
RESPONSE: THE DM11 UNIT NUMBER CORRESPONDS TO THE  
ADDRESS TO WHICH THE CLOCK STATUS REGISTER (CSR) RESPONDS.

CSR ADDRESS	DM11 UNIT #	CSR ADDRESS	DM11 UNIT #
175000	0	175100	10
175010	1	175110	11
175020	2	175120	12
175030	3	175130	13
175040	4	175140	14
175050	5	175150	15
175060	6	175160	16
175070	7	175170	17

CARRIAGE RETURN SELECTS UNIT # 0.

- C. PRG #  
RESPONSE: TYPE PROGRAM NUMBER OF PROGRAM YOU WISH TO  
RUN. CARRIAGE RETURN SELECTS PROGRAM # 0.

CARRIAGE RETURN TERMINATES ALL RESPONSES.  
ANY UNACCEPTABLE RESPONSE WILL RESULT IN A ? TYPEOUT AND  
THE PARAMETER WILL AGAIN BE REQUESTED.

0204 - THIS STARTING ADDRESS USES PREVIOUSLY DEFINED DM11  
PARAMETERS AND REQUESTS THE PROGRAM NUMBER OF THE  
PROGRAM YOU WISH TO RUN.

0210 - THIS STARTING ADDRESS STARTS THE PREVIOUSLY SELECTED  
PROGRAM USING PREVIOUSLY SELECTED PARAMETERS.

123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159

#### 4.2 SWITCH SETTINGS

THE FOLLOWING SWITCH SETTINGS APPLY TO PROGRAM #0.

SR 0-6	ROUTINE TO BE RUN (IF ENABLED BY SR-9)
SR 9	LOOP SELECTED ROUTINE
SR 11	INHIBIT ITERATION (DO EACH ROUTINE ONCE)
SR 13	INHIBIT PRINTOUT
SR 14	SCOPE (LOOP ROUTINE)
SR 15	HALT ON ERROR

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176 ) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR:XXXXXX NEW-

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. ^U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

160  
161  
162  
163  
164  
165  
166  
167  
168  
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 PRG0 - LOGIC TESTS

PRG0 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 PRG0.

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 INSURES 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.

6.0 CHANGE HISTORY

NOTE: CHANGE HISTORY STARTS WITH REV. D0

CZDMBDO - TABLE BASE ADDRESS (TBR) REGISTER WILL ONLY WORK IF SET TO 400 WORD BOUNDARY (1000,1400,2000,ETC.). THEREFOR LOC. 1100 CHANGED TO 1200, AND LOC. 1106 CHANGED TO 1400

203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258

z

.TITLE CZDMBDO DM11 DATA TSTS  
.MLIST MC,MD  
.LIST ME  
.ENABLE ABS,AMA

;CZDMBDO DM11 DATA TSTS  
;COPYRIGHT 1972,1979 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754  
;PRG0- DATA TESTS  
;PRG1- DATA TEST (ALL LINES SIMULTANEOUSLY)  
;PRG2- TRANSMIT TO TERMINALS  
;PRG3- ECHO RECEIVED DATA

;STANDARD SR SWITCH OPTIONS [SWITCH SET TO A 1 (UP)]  
;SR15- HALT ON ERROR  
;SR14- SCOPE.  
;SR13- INHIBIT PRINTOUT  
;SR12- INHIBIT TRACE  
;SR11- INHIBIT ITERATION.  
;SR9- LOOP ROUTINE.  
;SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE LOOPED.

;EQUATE STATEMENTS  
CC=177776  
PSW=177776  
ERRVEC=4  
NOP=240  
OPEN=0  
MANUAL=BIT15  
BIT15=100000  
BIT14=40000  
BIT13=20000  
BIT12=10000  
BIT11=4000  
BIT10=2000  
BIT9=1000  
BIT8=400  
BIT7=200  
BIT6=100  
BIT5=40  
BIT4=20  
BIT3=10  
BIT2=4  
BIT1=2  
BIT0=1  
POPSP=5726  
POPSP2=022626  
PRTY7=340  
PRTY6=300  
PRTY5=240  
PRTY4=200  
PRTY3=140  
PRTY2=100  
PRTY1=40

;POP THE STACK. SAME AS TST (6)\*  
;POP STACK TWICE. SAME AS CMP (6)\*, (6)\*  
;PRIORITY LEVEL DEFINITIONS

259	000000	PRTY0=0	
260	000000	R0=20	
261	000001	R1=21	
262	000002	R2=22	
263	000003	R3=23	
264	000004	R4=24	
265	000005	R5=25	
266	000006	SP=26	
267	000007	PC=27	
268		;EMT CALLS	
269	104000	TYPE=EMT+0	
270	104001	TYPES=EMT+1	
271	104002	STALL=EMT+2	
272	104003	ERROR=EMT+3	
273	104004	DATCHK=EMT+4	
274	104006	STRXV=EMT+6	
275	104007	STTXV=EMT+7	
276	104010	FHALT=EMT+10	
277	104012	SCOPE=EMT+12	
278	104013	SAVREG=EMT+13	
279	104014	RSTREG=EMT+14	
280	104015	ERROR1=EMT+15	
281	104016	SUSWR=EMT+16	
282	104017	KBDIN=EMT+17	
283	104020	CNTLU=EMT+20	
284			
285	000007	BELL=007	
286	177777	ATLAST=-1	
287	125252	ALTO=125252	;ALTERNATING 0'S PATTERN
288	052525	ALT1=052525	;ALTERNATING 1'S PATTERN
289	000000	Y=0	
290	177777	X=-1	
291	000000	A=0	
292	000000	. =0	
293	000000	HALT	
294	000002	HALT	
295	000004	.+2	;SP OVERFLOW, BUS ERROR TRAP
296	000006	HALT	
297	000010	.+2	;RESERVED INSTRUCTION TRAP
298	000012	HALT	
299	000014	.+2	;TRACE TRAP
300	000016	HALT	
301	000020	.+2	;TRAP TO CALL IOX
302	000022	HALT	
303	000024	.+2	;POWER FAIL TRAP
304	000026	HALT	
305	000030	EMTINT	;EMT TRAP
306	000032	PRTY7	
307	000034	.+2	
308	000036	HALT	
309	000040	.+2	
310	000042	HALT	;TRAPPED TO PREVIOUS ADDRESS.
311	000044	.+2	
312	000046	HALT	;TRAPPED TO PREVIOUS ADDRESS.
313	000050	.+2	
314	000052	HALT	;TRAPPED TO PREVIOUS ADDRESS.



315	000054	000056	.+2	
316	000056	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
317	000060	000062	.+2	
318	000062	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
319	000064	000066	.+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 003022      SENDAD
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 002404      JMP @#START      ;GO TO START OF DIAGNOSTIC.
434      000204 000137 002422      JMP @#RSTAT1     ;GO GET PROGRAM # & RESTART PROGRAM
435
436      000210 000137 002472      JMP @#RSTAT2     ;USING PREVIOUS DM11 PARAMETERS
437
438
439      001200 001200      .-1200
440
441      001200 000000      SPBOT: 0
442      001202 177570      SWR: 177570
443      001204 177570      DISPLAY:177570
444
445      001400 000000      .-1400
446      001400 001400      CAT: OPEN      ;STARTING ADDRESS OF
447      001440 000000      .-CAT+32.     ;CURRENT ADDRESS TABLE
448      001500 000000      WCT: OPEN     ;STARTING ADDRESS OF
449      001500 000000      .-WCT+32.    ;WORD COUNT TABLE
450      001540 000000      BAT: OPEN     ;STARTING ADDRESS OF
451      001540 000000      .-BAT+32.    ;BIT ASSEMBLY TABLE
452      001542 175000      VAC: OPEN     ;32. SPARE WORDS
453      001544 175002      CSR: 175000   ;ADDRESS OF CLOCK STATUS REGISTER
454      001546 175004      BAR: 175002   ;ADDRESS OF BUFFER ACTIVE REGISTER
455      001550 175006      BKCSR: 175004 ;ADDRESS OF BREAK STATUS REGISTER
456      001552 000000      BASREG: 175006 ;ADDRESS OF BASE REGISTER
457      001554 000240      CLKINT: OPEN  ;DM11 VECTOR ADDRESS (RECEIVER)
458      001556 000000      CLKLVL: PRTY5 ;PRIORITY LEVEL
459      001560 000240      XMTINT: OPEN  ;DM11 VECTOR ADDRESS (TRANSMITTER)
460      001562 000000      XMTLVL: PRTY5 ;TRANSMITTER PRIORITY LEVEL
461      001564 000000      BARIM: OPEN  ;PROGRAM BAR IMAGE
462      001566 000000      TTDAT: OPEN  ;TUMBLE TABLE DATA
463      001570 000000      LINBIT: OPEN ;LINE BIT (FOR BAR)
464      001572 000000      BARDAT: OPEN ;BAR DATA
465      001600 001600      TTPTR: OPEN  ;PROGRAM TUMBLE TABLE POINTER
466      001600 000000      .-VAC+32.
467      002000 002000      TUMTAB: OPEN ;STARTING ADDRESS OF
468      002004 000060      .-TUMTAB+128. ;TUMBLE TABLE
469      002002 000200      TKVTR: 60    ;LSR INTERRUPT VECTOR
470      002004 000064      TKLVL: PRTY4 ;LSR PRIORITY LEVEL
471      002006 000200      TPVTR: 64    ;LSP INTERRUPT VECTOR
472      002010 000000      TPLVL: PRTY4 ;LSP PRIORITY LEVEL
473      002012 000000      KSTART: OPEN ;CURRENT PROGRAM START ADDRESS.
474      002014 000000      CURTST: OPEN ;CONTAINS ADDR OF CURRENT TEST.
475      002016 000000      RTNNO: OPEN  ;CONTAINS CURRENT TEST #.
476      002020 000000      NXTST: OPEN  ;CONTAINS ADDR OF NEXT TEST.
477      002022 000000      ICTR: OPEN   ;CONTAINS CURRENT ITERATION COUNT
477      002022 000000      SCOPTR: OPEN ;CONTAINS CURRENT SCOPE POINTER.
    
```

(REV. 00)

(REV. 00)

478 002024 177774  
 479 002026 005660  
 480 002030 006640  
 481 002032 007024  
 482 002034 007034  
 483 002036 005702  
 484 002040 006656  
 485 002042 007030  
 486 002044 007052  
 487 002046 003220  
 488 002050 000000  
 489 002052 000000  
 490 002054 002224  
 491 002056 002156  
 492 002060 000000  
 493 002062 000000  
 494 002064 000000  
 495 002066 000000  
 496 002070 000000  
 497 002072 002650  
 498 002074 003116  
 499 002076 003156  
 500 002100 002242  
 501 002102 004412  
 502 002104 004256  
 503 002106 004332  
 504  
 505 002110 000000  
 506 002112 177560  
 507 002114 177562  
 508 002116 177564  
 509 002120 177566  
 510 002122 000000  
 511 002124 000000  
 512 002126 000000  
 513 002130 000000  
 514 002132 000000  
 515 002134 000000  
 516 002136 000000  
 517 002140 000000  
 518 002142 000000  
 519 002144 000000  
 520 002146 000000  
 521  
 522  
 523 002150 104000  
 524 002152 013300  
 525 002154 000207

PRGLIM: -4  
 PRGTAB: PRG0  
 PRG1  
 PRG2  
 PRG3  
 RSTART: PRGOR  
 PRG1R  
 PRG2R  
 PRG3R  
 EMTTAB: TYP  
 OPEN  
 OPEN  
 ERR  
 DTCHK  
 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  
 PASS: OPEN  
 COUNT: OPEN

INCRIN: TYPE  
 M1  
 RTS %7

: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

:TYPE INCORRECT ROUTINE SELECTED.  
 :EXIT.

```
526  
527  
528 ;DATA CHECK ROUTINE.  
529 002156 123737 002122 002124 DTCHK: CMPB RCVDAT,XMTDAT ;COMPARE EXPECTED AND RECEIVED  
530 002164 001416 BEQ 1$ ;CHARS. BRANCH IF SAME.  
531 002166 004737 002356 JSR 7,CNVDAT ;CONVERT RCVDAT & XMTDAT TO ASCII  
532 002172 032777 020000 177002 BIT #BIT13,@SWR ;ERROR TYPEOUT DESIRED?  
533 002200 001010 BNE !$ ;BRANCH IF NO TYPEOUT DESIRED  
534 002202 004537 005112 JSR 5,@#OACNV ;CONVERT LINE  
535 002206 002140 LINE ;NUMBER  
536 002210 013254 ALINE ;TO ASCII  
537 002212 000002 2  
538 002214 104015 ERROR1  
539 002216 104000 TYPE ;TYPE LINE # AS PART  
540 002220 013245 LINEM ;OF ERROR MESSAGE  
541 002222 000002 1$: RTI ;EXIT.  
542  
543 ;ERROR SERVICE ROUTINE CALLED BY TRAP (HLT)  
544 002224 012737 000402 002324 ERR: MOV #402,ERRB ;MOV BR .+6 TO ERRB  
545 002232 013737 002132 002134 MOV @#PCADD,@#APCADD ;GET PC WHERE ERROR OCCURRED  
546 002240 000410 BR ERRA  
547 002242 012737 000240 002324 ERR1: MOV #240,ERRB ;MOVE NOP TO ERRB  
548 002250 013737 002132 002134 MOV @#PCADD,@#APCADD ;GET PC WHERE ERROR OCCURRED  
549 002256 004737 002356 JSR 7,@#CNVDAT ;CONVERT RCVDAT & XMIT DAT TO ASCII  
550 002262 104017 ERRA: KBDIN ;CHECK FOR ^G  
551 002264 032777 020000 176710 BIT #BIT13,@SWR ;ERROR PRINTOUT DESIRED  
552 002272 001017 BNE ERRC ;BRANCH IF NO PRINTOUT  
553 002274 004537 005112 JSR 5,@#OACNV ;CONVERT  
554 002300 002134 APCADD ;DATA  
555 002302 013734 APC ;TO  
556 002304 000006 6 ;ASCII  
557 002306 004537 005112 JSR 5,@#OACNV ;FOR  
558 002312 002014 RTNNO ;PRINTOUT  
559 002314 013724 ATNUMB  
560 002316 000003 3  
561 002320 104000 TYPE ;TYPE ERROR  
562 002322 013721 EMO ;MESSAGE  
563 002324 000000 ERRB: OPEN ;NOP IF ERROR1, BR .+6 IF ERROR  
564 002326 104000 TYPE ;TYPE ANOTHER MESSAGE  
565 002330 013200 ERDAT ;IF ERROR 1  
566 002332 023737 000042 000046 ERRC: CMP @#42,@#46 ;ACT11?  
567 002340 001403 BEQ ERRHLT ;BR IF YES  
568 002342 005777 176634 TST @SWR ;HALT ON ERROR  
569 002346 100001 BPL ERREX ;GO TO EXIT IF NO HALT ON ERROR  
570 002350 000000 ERRHLT: HALT ;HALT  
571 002352 104017 ERREX: KBDIN ;CHECK FOR ^G  
572 002354 000002 RTI ;RETURN  
573  
574 ;SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE  
575 ;IN MESSAGE.  
576 002356 004537 005112 CNVDAT: JSR 5,OACNV  
577 002362 002124 XMTDAT  
578 002364 013220 AASB  
579 002366 000006 6  
580 002370 004537 005112 JSR 5,OACNV  
581 002374 002122 RCVDAT
```

```

582 002376 013235          AWAS
583 002400 000006          6
584 002402 000207          RTS      7          :EXIT
585
586
587
588 002404 012706 001200    START:  MOV      #SPBOT,%6      ;INITIALIZE STACK
589 002410 104016          SUSWR      ;CHECK FOR HARDWARE SWITCH REGISTER
590 002412 004737 003334    JSR      7,@#DMPAR      ;GET DM11 PARAMETERS
591 002416 004737 004042    JSR      7,@#OVLAY     ;PUT HALT,..+2 IN VECTOR AREA
592 002422 012706 001200    RSTAT1: MOV      #SPBOT,%6      ;INITIALIZE STACK
593 002426 023737 000042 000046  CMP      @#42,@#46      ;ACT11?
594 002434 001405          BEQ      PRGNUM+2      ;BR IF YES
595 002436 104000          TYPE
596 002440 013260          MO
597 002442 004537 004076    JSR      5,RECD        ;GET THE PRGNUM &
598 002446 000000          PRGNUM. 0            ;PUT IT HERE
599 002450 043737 002024 002446  BIC      PRGLIM,PRGNUM  ;MASK OFF UNUSED BITS
600 002456 006337 002446    ASL      PRGNUM        ;SHIFT PROGRAM #
601 002462 013700 002446    MOV      PRGNUM,%0     ;GET PROGRAM #
602 002466 000170 002026    JMP      @PRGTAB(0)    ;GO START PROGRAM
603 002472 012706 001200    RSTAT2: MOV      #SPBOT,%6      ;INITIALIZE STACK
604 002476 013700 002446    MOV      PRGNUM,%0     ;GET PROGRAM #
605 002502 000170 002036    JMP      @RSTART(0)   ;GO RESTART PROGRAM
606 002506 022737 000176 001202  SRSET:  CMP      #SWREG,SWR
607 002514 001410          BEQ      1$
608 002516 023737 000042 000046  CMP      @#42,@#46      ;ACT11?
609 002524 001405          BEQ      GETRDY       ;BR IF YES
610 002526 104000          TYPE                ;TYPE OPTIONS MESSAGE
611 002530 013330          M3
612 002532 000000          HALT                ;WAIT FOR USER TO SET OPTIONS
613 002534 000401          BR      GETRDY
614 002536 104020          1$:  CNTLU
615 002540 013737 002010 002016  GETRDY: MOV      KSTART,NXTST  ;ADDR OF 1ST ROUTINE TO NXTST
616 002546 012737 000006 000004  GTRDYX: MOV      #6,@#ERRVEC  ;RESET ERROR TRAP.
617 002554 005037 177776    CLR      PSW
618 002560 012706 001200    MOV      #SPBOT,%6
619 002564 000005          RESET                ;SET BOTTOM OF STACK.
620 002566 004737 003032          JSR      %7,FORWD     ;ISSUE RESET.
621 002572 032777 001000 176402  GTRDYA: BIT      #BIT9,@SWR    ;ROLL FORWARD TO "NEXT" ROUTINE.
622 002600 001003          BNE      GTRDYC       ;CHECK SELECT ROUTINE SWITCH
623 002602 000177 177204    JMP      @CURTST      ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
624 002606 000461          BR      SCOPED        ;GO RUN CURRENT ROUTINE.
625 002610 017700 176366    GTRDYC: MOV      @SWR,%0     ;NO GO. MANUAL RTN BYPASSED.
626 002614 042700 177600    BIC      #177600,%0   ;(SR) TO R0
627 002620 123700 002014    CMPB    RTNNO,%0     ;MASK UNDESIRED BITS
628 002624 001002          BNE      GTRDYD       ;COMPARE RTNNO TO (R0)
629 002626 000177 177160    JMP      @CURTST      ;BRANCH IF ROUTINE NOT FOUND YET.
630 002632 022737 177777 002016  GTRDYD: CMP      #-1,NXTST    ;GO RUN ROUTINE.
631 002640 001352          BNE      GTRDYA       ;NO. CHECK FOR LAST ROUTINE.
632 002642 004737 002150    JSR      %7,INCRTN   ;BRANCH IF NOT LAST ROUTINE.
633 002646 000734          BR      GETRDY       ;YES. INCORRECT ROUTINE SELECTED.
634
635
636 002650 000240          ;SCOPE ROUTINE (CALLED BY EMT INST.)
637 002652 005077 176664    ESCOPE: NOP
        CLR      @CSR      ;INITIALIZE
    
```

638	002656	005077	176664		CLR	@BKCSR		:THE
639	002662	005077	176656		CLR	@BAR		:DM11
640	002666	104017			KBDIN			
641	002670	012777	001400	176652	MOV	#CAT,@BASREG		
642	002676	032777	040000	176276	BIT	#BIT14,@SWR		:CHECK FOR SCOPE OPTION.
643	002704	001403			BEQ	SCOPEB		:BRANCH IF SCOPE SW NOT SET.
644	002706	013716	002022		SCOPEA: MOV	SCOPTR,@%6		:SET UP TO RETURN TO ROUTINE.
645	002712	000002			RTI			:RETURN TO ROUTINE.
646	002714	032777	004000	176260	SCOPEB: BIT	#BIT11,@SWR		:TEST INHIBIT ITERATION SWITCH
647	002722	001012			BNE	SCOPEC		:BRANCH IF INHIBIT ITERATION SW SET.
648	002724	023737	000042	000046	CMP	@#42,@#46		:ACT11?
649	002732	001003			BNE	1\$		:BR IF NO
650	002734	005737	002144		TST	@#PASS		:1ST PASS?
651	002740	001403			BEQ	SCOPEC		:BR IF YES
652	002742	005337	002020		1\$: DEC	ICTR		:DECREMENT ITERATION COUNT.
653	002746	001357			BNE	SCOPEA		:BRANCH IF COUNT NOT 0.
654	002750	022626			SCOPEC: POPSP2			:POP STACK TWICE
655	002752	032777	001000	176222	SCOPEB: BIT	#BIT9,@SWR		:CHECK SELECT ROUTINE SWITCH
656	002760	001267			BNE	GETRDY		:BRANCH IF SELECT RTN SW SET
657	002762	022737	177777	002016	CMP	#-1,NXTST		:LAST TEST?
658	002770	001266			BNE	GTRDYX		:BRANCH IF NOT LAST TEST.
659	002772	005237	002144		INC	@#PASS		:IND PASS
660	002776	104000			TYPE			:TYPE
661	003000	013303			M2			:END
662	003002	013702	000042		MOV	@#42,%2		:CHECK DDP/ACT11 MONITOR HOOK
663	003006	001654			BEQ	GETRDY		
664	003010	000005			RESET			
665	003012	000240			NOP			
666	003014	000240			NOP			
667	003016	000240			NOP			
668	003020	000240			NOP			
669	003022	004712			SENDAD: JSR	7,(2)		:RETURN TO DDP/ACT11 MONITOR
670	003024	000240			NOP			
671	003026	000240			NOP			
672	003030	000240			NOP			
673								
674	003032	013705	002016		FORWD: MOV	NXTST,%5		:ADDR OF NEXT ROUTINE TO R5.
675	003036	012537	002014		MOV	(5)+,RTNNO		:GET NEXT ROUTINE NUMBFR.
676	003042	012537	002016		MOV	(5)+,NXTST		:GET ADDR OF NEXT 'NEXT' ROUTINE.
677	003046	012537	002020		MOV	(5)+,ICTR		:GET ITERATION COUNT.
678	003052	012537	002022		MOV	(5)+,SCOPTR		:GET SCOPE LOOP ENTRY POINTER.
679	003056	010537	002012		MOV	%5,CURTST		:ADDR OF NOW CURRENT TEST TO CURTST.
680	003062	000207			RTS	%7		:EXIT FORWD SUBROUTINE.
681								
682					:EMT TRAP INTERPRETER			
683	003064	011646			EMTINT: MOV	(6),-(6)		:GET PC OF NEXT INSTRUCTION
684	003066	162716	000002		SUB	#2,(6)		:POINT SP TO PC OF EMT
685	003072	011637	002132		MOV	(6),PCADD		:GET PC OF EMT CALL
686	003076	017616	000000		MOV	@(6),(6)		:GET EMT CALL
687	003102	105066	000001		CLRB	1(6)		:STRIP EMT & SAVE IDENTIFIER
688	003106	006316			ASL	(6)		:SHIFT IDENTIFIER LEFT
689	003110	062716	002046		ADD	#EMTTAB,(6)		
690	003114	013607			MOV	@(6)+,%7		:GO TO PROPER EMT
691								
692					:SAVE REGS 0 TO 4 SUBROUTINE.			
693	003116	012637	003152		SAVRG: MOV	(6)+,%1\$		:SAVE PC AND PSW.

```

694 003122 012637 003154      MOV      (6)+,2$
695 003126 010446      MOV      %4,-(6)      ;SAVE REGS 0 - 4
696 003130 010346      MOV      %3,-(6)      ;IN STACK.
697 003132 010246      MOV      %2,-(6)
698 003134 010146      MOV      %1,-(6)
699 003136 010046      MOV      %0,-(6)
700 003140 013746 003154      MOV      2$,-(6)      ;RESTORE PC AND PSW.
701 003144 013746 003152      MOV      1$,-(6)
702 003150 000002      RTI
703 003152 000000      1$: OPEN      ;EXIT.
704 003154 000000      2$: OPEN      ;CONTAINS SAVED PC
                          ;CONTAINS SAVED PSW
705
706      ;RESTORE REGS 0 TO 4 SUBROUTINE.
707 003156 000240      RSTRG: NOP
708 003160 012637 003214      MOV      (6)+,1$      ;SAVE PC AND PSW.
709 003164 012637 003216      MOV      (6)+,2$
710 003170 012600      MOV      (6)+,%0      ;RESTORE REGS 0 - 4
711 003172 012601      MOV      (6)+,%1      ;FROM STACK.
712 003174 012602      MOV      (6)+,%2
713 003176 012603      MOV      (6)+,%3
714 003200 012604      MOV      (6)+,%4
715 003202 013746 003216      MOV      2$,-(6)      ;RESTORE PC AND PSW.
716 003206 013746 003214      MOV      1$,-(6)
717 003212 000002      RTI
718 003214 000000      1$: OPEN      ;EXIT
719 003216 000000      2$: OPEN      ;CONTAINS SAVED PC
                          ;CONTAINS SAVED PSW
720
721      ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
722 003220 000240      TYP:  NOP
723 003222 011600      MOV      (SP),%0      ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
724 003224 062716 000002      ADD      #2,(SP)      ;SET UP EXIT.
725 003230 011000      MOV      @%0,%0      ;ADDRESS OF MESSAGE TO R0.
726 003232 112037 003332      1$: MOVB   (0)+,%5      ;GET CHARACTER
727 003236 122737 000100 003332      CMPB    #100,%5      ;CHECK FOR '@' CHARACTER
728 003244 001001      BNE     2$           ;BRANCH IF NOT '@'
729 003246 000002      RTI
                          ;TERMINATOR CHAR. DONE. EXIT.
730 003250 122737 000045 003332      2$: CMPB    #45,%5      ;CHECK FOR '%'.
731 003256 001412      BEQ     4$           ;BRANCH IF '%'
732 003260 004737 003266      JSR     %7,3$      ;TYPE CHAR IN 5$
733 003264 000762      BR      1$
734 003266 113777 003332 176624      3$: MOVB   5$,@TPDBR   ;OUTPUT CHARACTER TO PRINTER
735 003274 105777 176616      TSTB   @TPCSR      ;WAIT FOR DONE FLAG.
736 003300 100375      BPL     -4
737 003302 000207      RTS     %7
                          ;EXIT
738 003304 112737 000015 003332      4$: MOVB   #15,%5      ;MOVE CARRIAGE RETURN CODE TO 5$
739 003312 004737 003266      JSR     %7,@#3$     ;GO TYPE CHAR.
740 003316 112737 000012 003332      MOVB   #12,%5      ;MOVE LF CODE TO 5$.
741 003324 004737 003266      JSR     %7,3$      ;GO TYPE CHAR.
742 003330 000740      BR      1$
743 003332 000000      5$: OPEN
744
745      ;SUBROUTINE TO GET DM11 PARAMETERS
746      ;VECTOR ADDRESS
747 003334 000240      DMPAR: NOP      ;BEGIN
748 003336 023737 000042 000046      CMP     @#42,@#46   ;ACT11?
749 003344 001060      BNE     6$         ;BR IF NO
    
```



```

750                                     ;SIZE FOR INTERRUPT VECTOR IN AUTO MODE
751 003346 012700 000302             MOV #302,R0 ;SET UP FLOATING VECT AREA
752 003352 010060 177776             4$: MOV R0,-2(R0)
753 003356 012720 000003             MOV #3,(R0)+
754 003362 005720                     TST (R0)+
755 003364 022700 000776             CMP #776,R0
756 003370 100370                     BPL 4$
757 003372 012737 003462 000014     MOV #58,@#14 ;SET BPT VECT
758 003400 012737 000340 000016     MOV #340,@#16 ;& PSW
759 003406 012737 177777 001440     3$: MOV #-1,WCT ;SET TO XMIT 1 CHAR
760 003414 012737 007362 001400     MOV #OUTBUF,CAT
761 003422 012777 000105 176112     MOV #BIT6+BIT2+BIT0,@CSR ;SET IE
762 003430 005037 177776             CLR @PSW ;LVL 0
763 003434 012777 000001 176102     MOV #BIT0,@BAR ;XMIT
764 003442 012737 177777 002146     MOV #-1,COUNT ;WAIT
765 003450 005337 002146             2$: DEC COUNT
766 003454 001375                     BNE 2$
767 003456 104003                     ERROR
768 003460 000752                     BR 3$ ;NO INT OCCURRED
769 003462 162716 000004             5$: SUB #4,(SP) ;REPEAT IT
770 003466 011637 003522             MOV (SP),@#VECTOR ;CALC INT VECT
771 003472 012737 000016 000014     MOV #16,@#14 ;STORE IT
772 003500 004737 004042             JSR 7,OVRLAY ;RESTORE BPT VECT
773 003504 000415                     BR VECOK ;.+2, HALT IN VECT AREA
774 003506 004737 004042             6$: JSR 7,OVRLAY ;PUT HALT,..+2 IN VECTOR AREA
775 003512 104000                     TYPE ;ASK USER FOR RECEIVER INT. VECTOR
776 003514 013050                     WHERE ;OF UNIT UNDER TEST
777 003516 004537 004076             JSR 5,RECD ;GET THE VECTOR &
778 003522 000000                     VECTOR: 0 ;PUT IT HERE
779 003524 005737 003522             TST VECTOR
780 003530 001003                     BNE VECOK
781 003532 012737 000300 003522     VECOK: MOV #300,VECTOR ;SET VECTOR - TO 0300
782 003540 023727 003522 000300     CMP VECTOR,#300 ;IS VECTOR HIGHER OR
783 003546 103003                     BHS VECOKB ;EQUAL TO 0300
784 003550 104000                     VECOKA: TYPE ;TYPE '?'
785 003552 013300                     M1
786 003554 000667                     BR DMPAR
787 003556 023727 003522 000770     VECOKB: CMP VECTOR,#770 ;ASK FOR ANOTHER VECTOR
788 003564 101371                     BHI VECOKA ;IS VECTOR = TO OR
789 003566 032737 000007 003522     BIT #7,VECTOR ;LESS THAN 770
790 003574 001365                     BNE VECOKA ;LSB OF VECTOR MUST BE ALL 0'S
791 003576 013737 003522 001552     MOV VECTOR,@#CLKINT
792 003604 062737 000004 003522     ADD #4,VECTOR
793 003612 013737 003522 001556     MOV VECTOR,@#XMTINT
794
795                                     ;UNIT NUMBER
796 003620 023737 000042 000046     DMPARB: CMP @#42,@#46 ;ACT11?
797 003626 001405                     BEQ UNIT+2 ;BR IF YES
798 003630 104000                     TYPE
799 003632 013143                     WHICH
800 003634 004537 004076             JSR 5,RECD ;GET THE UNIT &
801 003640 000000                     UNIT: 0 ;PUT IT HERE
802 003642 023727 003640 000017     CMP UNIT,#17
803 003650 101403                     BLOS 1$
804 003652 104000                     TYPE
805 003654 013300                     M1
    
```

806	003656	000760							
807	003660	006337	003640			1\$:	BR	DMPARB	
808	003664	006337	003640				ASL	UNIT	
809	003670	006337	003640				ASL	UNIT	
810	003674	012702	000004				ASL	UNIT	
811	003700	012701	001542				MOV	#4,%2	
812	003704	042711	000370			2\$:	MOV	#CSR,%1	
813	003710	063721	003640				BIC	#370,(1)	
814	003714	005302					ADD	UNIT,(1)+	
815	003716	001372					DEC	%2	
816							BNE	2\$	
817									
818	003720	012777	001400	175622					
819	003726	005077	175610						
820	003732	012737	177777	007362					
821	003740	012737	177777	001440					
822	003746	012737	177777	002126					
823	003754	012737	007362	001400					
824	003762	012777	004020	175562					
825	003770	012777	000340	175556					
826	003776	005037	001600						
827	004002	012777	000001	175534					
828	004010	012777	000105	175524					
829	004016	000001							
830	004020	005077	175516						
831	004024	143737	001600	002126		3\$:			
832	004032	005037	177776						
833	004036	022626							
834	004040	000207							
835									
836									
837									
838	004042	012701	000300						
839	004046	012702	000302						
840	004052	010221							
841	004054	005021							
842	004056	020227	000776						
843	004062	001403							
844	004064	062702	000004						
845	004070	000770							
846	004072	000240							
847	004074	000207							
848									
849									
850									
851									
852									
853									
854									
855									
856									
857	004076	010046							
858	004100	005015							
859	004102	012737	000007	004254					
860	004110	105777	175776						
861	004114	100375							

```

;CALCULATE CHARACTER LENGTH
MOV #CAT,@BASREG
CLR @CSR
MOV #-1,OUTBUF ;LOAD OUTBUF WITH CHAR TO BE TRANSMITTED
MOV #-1,WCT ;SET UP TO TRANSMIT 1 CHAR
MOV #-1,@#CARMSK ;PRE SET THE CHARACTER MASK
MOV #OUTBUF,CAT ;1 CHARACTER ON LINE 0
MOV #3$,@CLKINT ;LOAD RECEIVER INTERRUPT
MOV #340,@CLKLVL ;AND PRIORITY LEVEL
CLR TUMTAB
MOV #1,@BAR ;START TRANSMITTING
MOV #BIT6+BIT2+BIT0,@CSR ;SET IE,MAINT AND GO BITS
WAIT ;WAIT FOR RECEIVER INTERRUPT
3$: CLR @CSR
BICB TUMTAB,CARMSK ;LOAD CHARACTER LENGTH MASK
CLR PSW ;RESTORE PROCEESSER TO PRIORITY 0
POPSP2 ;RESTORE THE STACK POINTER
RTS 7 ;EXIT PARAMETERS ROUTINE

```

```

;ROUTINE TO LOAD TRAP/INTERRUPT VECTOR AREA WITH HALT,..+2. HALTS PROGRAM
;AT ADDRESS OF TRAP/INTERRUPT VECTOR +2.
OVRLAY: MOV #300,%1
MOV #302,%2
1$: MOV %2,(1)+
CLR (1)+
CMP %2,#776
BEQ 2$
ADD #4,%2
BR 1$
2$: NOP
RTS 7 ;EXIT

```

```

;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$

```

```
862 004116 117700 175772      MOVB   @TKDBR,RO
863 004122 142700 000200      BICB   #200,RO           ;STRIP OFF PARITY
864 004126 110077 175766      MOVB   RO,@TPDBR        ;ECHO CHARACTER
865 004132 122700 000025      CMPB   #25,RO           ;IS IT A ^U
866 004136 001443              BEQ    5$                ;BRANCH IF YES
867 004140 122700 000015      CMPB   #15,RO           ;IS IT A <CR>
868 004144 001415              BEQ    6$                ;BRANCH IF YESS
869 004146 142700 000060      BICB   #60,RO
870 004152 132700 000110      BITB   #110,RO          ;CHECK FOR 0-7 (8)
871 004156 001031              BNE    7$                ;BRANCH IF NOT
872 004160 006315              ASL    (5)
873 004162 006315              ASL    (5)
874 004164 006315              ASL    (5)                ;SHIFT DATA
875 004166 150015              BISB   RO,(5)            ;INSET NEW CHAR
876 004170 005337 004254      DEC    CNT
877 004174 001422              BEQ    7$                ;ONLY 6 CHAR'S PLEASE
878 004176 000744              BR     2$                ;NEXT CHARACTER
879 004200 105777 175712      6$:  TSTB   @TPCSR
880 004204 100375              BPL    6$                ;WAIT FOR READY
881 004206 012777 000012 175704  MOV    #12,@TPDBR        ;TYPE <LF>
882 004214 105777 175676      8$:  TSTB   @TPCSR
883 004220 100375              BPL    8$                ;WAIT FOR READY
884 004222 005077 175672      CLR    @TPDBR            ;NEXT CHARACTER
885 004226 105777 175664      9$:  TSTB   @TPCSR
886 004232 100375              BPL    9$                ;WAIT FOR READY
887 004234 005725              TST    (R5)+             ;ADJUST R5
888 004236 012600              MOV    (SP)+,RO         ;RESTORE RO
889 004240 000205              RTS    R5
890 004242 104000              7$:  TYPE
891 004244 013300              M1
892 004246 104000              5$:  TYPE
893 004250 013140              SCTLU
894 004252 000712              BR     1$                ;START OVER
895 004254 000000      CNT:  0
896
897
898      ;ROUTINE TO CHECK FOR ^G BEING TYPED
899
900 004256 022737 000176 001202  KBDINTT:  CMP    #SWREG,SWR
901 004264 001021              BNE    1$
902 004266 023737 000042 000046      CMP    @#42,@#46        ;ACT11?
903 004274 001415              BEQ    1$                ;BR IF YES
904 004276 005037 004370              CLR    TMP1              ;CLEAR TEMP AREA
905 004302 117737 175606 004370      MOVB   @TKDBR,TMP1      ;FETCH THE BUFFER
906 004310 142737 000200 004370      BICB   #20C,TMP1        ;STRIP OFF PARITY
907 004316 122737 000007 004370      CMPB   #7,TMP1          ;WAS IT ^G
908 004324 001001              BNE    1$                ;NOP
909 004326 104020              CNTLU
910 004330 000002      1$:  RTI                ;GO CHANGE IT
911
912
913      ;ROUTINE TO CHANGE CONTENTS OF SWREG(LOC 176)
914
915 004332 022737 000176 001202  CNTLUU:  CMP    #SWREG,SWR
916 004340 001023              BNE    FAJAG
917 004342 104000              TYPE
```

```

918 004344 013111          $SWREG
919 004346 004537 005112  JSR      R5,OACNV          :CONVERT TO ASCII
920 004352 000176          SWREG
921 004354 013120          $VALUE
922 004356 000006          6
923 004360 104000          TYPE
924 004362 013120          $VALUE
925 004364 004537 004076  JSR      5,RECD          :GET THE TMP1 &
926 004370 000000          TMP1: 0          :PUT IT HERE
927 004372 022737 000007 004254  CMP      #7,CNT
928 004400 001403          BEQ      FAJAG
929 004402 013777 004370 174572  MOV      TMP1,@SWR          :CHANGE CONTENTS OF SWREG
930 004410 000002          FAJAG: RTI
931
932
933 004412 013746 000006          SUSWRR: MOV      @#6,-(SP)          :SAVE VECTORS
934 004416 013746 000004          MOV      @#4,-(SP)
935 004422 012737 004442 000004  MOV      #15,@#4          :SET UP FOR TIMEOUT
936 004430 022777 177777 174544  CMP      #-1,@SWR          :REFERENCE HARDWARE SWITCH REGISTER
937 004436 001402          BEQ      2$
938 004440 000407          BR      3$
939 004442 022626          1$: CMP      (SP)+,(SP)+          :ADJUST STACK
940 004444 012737 000176 001202  2$: MOV      #SWREG,SWR          :POINT TO SOFTWARE SWITCH REG
941 004452 012737 000174 001204  MOV      #DISPREG,DISPLAY          :POINT TO SOFT DISPLAY REG
942 004460 012637 000004          3$: MOV      (SP)+,@#4          :RESTORE VECTORS
943 004464 012637 000006          MOV      (SP)+,@#6
944 004470 000002          RTI
945

```

```

946 ;SUBROUTINE TO TRANSMIT ON ALL LINES WITH A DELAY BETWEEN TRANSMITTING
947 ;ON SUCCESSIVE LINES. THE DELAY FOR THE TEST IS SUPPLIED BY THE
948 ;CALLING JSR INSTRUCTION. DATA IS CHECKED AFTER ALL
949 ;LINES HAVE FINISHED TRANSMITTING.
950
951 004472 000240 DLYXMT: NOP ;BEGIN TEST
952 004474 012777 001400 175046 MOV #CAT,@BASREG ;SET UP BASE REGISTER
953 004502 004737 005024 JSR 7,@#IDENT ;TRANSMIT LINE # ON EACH LINE
954 004506 000240 NOP ;NO:
955 004510 005077 175026 CLR @CSR
956 004514 012537 004524 MOV (5)+,10$ ;GET MESSAGE ADDRESS
957 004520 004537 005200 JSR 5,@#BMOVE ;LOAD OUTPUT BUFFER
958 004524 000000 10$: OPEN ;WITH DATA TO
959 004526 007362 OUTBUF ;BE TRANSMITTED
960 004530 000100 64.
961 004532 005037 001600 CLR @#TUMTAB ;CLEAR TUMBLE
962 004536 004537 005200 JSR 5,@#BMOVE ;TABLE (200
963 004542 001600 TUMTAB ;BYTES)
964 004544 001601 TUMTAB+1
965 004546 000177 177
966 004550 004537 005200 JSR 5,@#BMOVE ;CLEAR CHARACTER COUNT TABLE
967 004554 001600 TUMTAB
968 004556 012666 CNTTAB
969 004560 000020 16.
970 004562 005037 007526 CLR @#LNOBUF
971 004566 004537 005200 JSR 5,@#BMOVE ;CLEAR ALL
972 004572 007526 LNOBUF ;LINE'S INPUT
973 004574 007527 LNOBUF+1 ;BUFFERS
974 004576 003077 1599. ;(16. BUFFERS OF 100. CHARS. EACH)
975 004600 022737 000006 002446 CMP #6,PRGNUM
976 004606 001002 BNE .+6
977 004610 000137 007062 JMP PRG3A
978 004614 012504 MOV (5)+,%4 ;GET # OF CHARACTERS TO TRANSMIT BEFORE
979 ;TRANSMITTING ON NEXT LINE
980 004616 012737 001600 001572 MOV #TUMTAB,@#TTPTR ;INITIALIZE TUMBLE TABLE POINTER
981 004624 013701 001552 MOV @#CLKINT,%1 ;GET RECEIVER VECTOR ADDRESS
982 004630 012721 005426 MOV #RINT,(1)+ ;LOAD RECEIVER VECTOR
983 004634 013721 001554 MOV @#CLKLVL,(1)+ ;AND PRIORITY LEVEL
984 004640 012721 005620 MOV #TINT,(1)+ ;LOAD TRANSMITTER VECTOR
985 004644 013721 001560 MOV @#XMTLVL,(1)+ ;AND PRIORITY LEVEL
986 004650 005737 002446 TST PRGNUM ;RUNNING PROGRAM 0?
987 004654 001402 BEQ .+6
988 004656 000137 006666 JMP PRG1A ;RETURN TO PROGRAM 1 CODE
989 004662 012777 010101 174652 MOV #BIT12+BIT6+BIT0,@CSR ;SET IE & GO BITS
990 004670 012737 000001 001566 MOV #1,@#LINBIT
991 004676 005037 002140 CLR @#LINE
992 004702 013700 002140 1$: MOV LINE,%0 ;LINE # X2 TO R0
993 004706 000240 NOP ;NOP
994 004710 004537 005222 JSR 5,@#XMITD ;TRANSMIT 64 CHARACTERS
995 004714 177700 -64. ;ON LINE # AS SPECIFIED IN ADDRESS LINE
996 004716 020460 001440 2$: CMP %4,WCT(0) ;WAIT FOR THE WORD COUNT TO DEC TO THE
997 004722 001375 BNE 2$ ;CORRECT VALUE BEFORE STARTING NEXT LINE
998 004724 062737 000002 002140 ADD #2,LINE ;FORM NEXT LINE NUMBER
999 004732 006337 001566 ASL LINBIT ;SHIFT LINE BIT
1000 004736 103361 BCC 1$ ;START NEXT LINE
1001 004740 005760 001440 3$: TST WCT(0) ;WAIT FOR LAST LINE TO FINISH
    
```

```

1002 004744 001375      BNE      3$
1003 004746 042777 177400 174566      BIC      #177400,@CSR      ;CLEAR ODD BYTE OF CSR
1004 004754 062700 000001      ADD      #1,R0            ;WAIT FOR RECEIVER TO RECEIVE
1005 004760 001375      BNE      31$             ;ALL TRANSMITTED DATA
1006 004762 017737 174556 002122      MOV      @BAR,RCV DAT     ;GET AND TEST BAR CONTENTS
1007 004770 001410      BEQ      4$             ;BRANCH IF IS CLEAR
1008 004772 005037 002124      CLR      YMT DAT
1009 004776 005077 174540      CLR      @CSR
1010 005002 005077 174536      CLR      @BAR
1011 005006 104015      ERROR1
1012 005010 000403      BR       5$             ;ERROR. BAR DID NOT CLEAR IN SUFFICIENT TIME
1013 005012 000240      NOP
1014 005014 004737 005320      JSR      7,@CHK DAT      ;GO TEST DATA
1015 005020 022626      CMP      (6)+,(6)+      ;RESET THE STACK
1016 005022 104012      SCOPE
1017
1018
1019      ;SUBROUTINE TO TRANSMIT ON EACH LINE ITS LINE NUMBER (CRLF XX CRLF).
1020 005024 005037 002140      IDENT: CLR      @LINE      ;GET LINE NUMBER 0
1021 005030 012737 000001 001566      MOV      #1,@LIN BIT     ;GET LINE BIT
1022 005036 013702 002140      1$: MOV      LINE,%2
1023 005042 016262 012706 001400      MOV      ID(2),CAT(2)    ;LOAD CAT
1024 005050 012762 177772 001440      MOV      #-6,WCT(2)     ;LOAD WORD COUNT
1025 005056 053777 001566 174460      BIS      LIN BIT,@BAR    ;SET BAR BIT
1026 005064 062737 000002 002140      ADD      #2,LINE        ;FORM NEXT LINE NUMBER
1027 005072 006337 001566      ASL      LIN BIT        ;FORM NEXT LINE BIT
1028 005076 103357      BCC      1$             ;BRANCH IF NOT DONE
1029 005100 005777 174440      2$: TSI      @BAR        ;WAIT FOR BAR TO CLEAR
1030 005104 001375      BNE      2$
1031 005106 000240      NOP
1032 005110 000207      RTS      7             ;EXIT SUBROUTINE
1033
1034      ;OCTAL TO ASCII CONVERT ROUTINE
1035 005112 104013      OACNV: SAVREG          ;SAVE REGISTERS ON THE STACK
1036 005114 013537 005176      MOV      @ (5)+,%2      ;GET OCTAL VALUE.
1037 005120 012501      MOV      (5)+,%1        ;GET DESTINATION ADDR.
1038 005122 012502      MOV      (5)+,%2        ;GET CONVERT COUNT.
1039 005124 060201      ADD      %2,%1          ;DEVELOP ADDR TO STORE 1ST CHAR.
1040 005126 013703 005176      1$: MOV      %2,%3
1041 005132 042703 177770      BIC      #177770,%3     ;ISOLATE LEAST SIGNIFICANT DIGIT.
1042 005136 062703 000060      ADD      #60,%3         ;CONVERT DIGIT TO ASCII.
1043 005142 11034      MOV B    %3,-(1)        ;STORE ASCII CHARACTER.
1044 005144 042737 000007 005176      BIC      #7,%2
1045 005152 006037 005176      ROR      %2
1046 005156 006037 005176      ROR      %2
1047 005162 006037 005176      ROR      %2
1048 005166 005302      DEC      %2             ;DONE ALL DIGITS?
1049 005170 001356      BNE      1$             ;BRANCH IF NOT DONE.
1050 005172 104014      RSTREG
1051 005174 000205      RTS      %5             ;RESTORE THE REGISTERS
1052 005176 000000      2$: OPEN
1053
1054
1055
1056      ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1057 005200 104013      BMOVE: SAVREG          ;SAVE REGS.
    
```

1058	005202	012501		MOV	(5)+,Z1	:GET 'FROM' ADDRESS
1059	005204	012502		MOV	(5)+,Z2	:GET 'TO' ADDRESS
1060	005206	012503		MOV	(5)+,Z3	:GET COUNT
1061	005210	112122	18:	MOVB	(1)+,(2)+	:MOVE BYTE
1062	005212	005303		DEC	Z3	:DECREMENT COUNT
1063	005214	001375		BNE	18	:BRANCH IF NOT DONE.
1064	005216	104014		RSTREG		:RESTORE REGS.
1065	005220	000205		RTS	Z5	:DONE EXIT

```

1066
1067
1068
1069
1070 005222 000240
1071 005224 010046
1072 005226 013700 002140
1073 005232 012760 007362 001400
1074 005240 012560 001440
1075 005244 053737 001566 001562
1076 005252 053777 001566 174264
1077
1078 005260 012600
1079 005262 000240
1080 005264 000205
1081
1082
1083 005266 010046
1084 005270 005037 001566
1085 005274 013700 002140
1086 005300 000261
1087 005302 006137 001566
1088 005306 162700 000002
1089 005312 100373
1090 005314 012600
1091 005316 000207
1092
1093
1094
1095 005320 104013
1096 005322 000240
1097 005324 005001
1098 005326 012702 012626
1099 005332 005003
1100 005334 010337 002140
1101 005340 012237 002142
1102 005344 005337 002142
1103 005350 005237 002142
1104 005354 117737 174562 002122
1105 005362 116137 007362 002124
1106 005370 043737 002126 002124
1107 005376 104004
1108 005400 005201
1109 005402 020127 000100
1110 005406 001360
1111 005410 005001
1112 005412 005203
1113 005414 020327 000020
1114 005420 002745
1115 005422 104014
1116 005424 000207
1117
1118
1119
1120 005426 000240
1121 005430 104013

;SUBROUTINE TO TRANSMIT DATA. SUBROUTINE CALLED BY
;JSR 5,XMITD
XMITD:  NOP
      MOV    %0,-(SP)          ;SAVE R0 ON THE STACK
      MOV    @#LINE,%0        ;GET LINE
      MOV    #OUTBUF,CAT(0)    ;LOAD FIRST CHAR ADDRESS IN CAT
      MOV    (5)+,WCT(0)      ;LOAD WORD COUNT INTO LINE'S TABLE ADDRESS
      BIS    @#LINBIT,@#BARIM ;LOAD LINE POSITION INTO BAR IMAGE
      BIS    LINBIT,@BAR      ;START TRANSMITTING ON LINE SPECIFIED
      ;IN LINBIT
      MOV    (SP)+,%0         ;RESTORE R0
      NOP
      RTS    5                ;EXIT

;SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
GTLINB: MOV    %0,-(SP)          ;SAVE R0 ON THE STACK
      CLR    @#LINBIT
      MOV    @#LINE,%0        ;GET LINE
      SEC
      ;SET CARRY
1$:    ROL    LINBIT           ;SHIFT LINE BIT
      SUB    #2,%0            ;SUBTRACT 2 FROM LINE NUMBER
      BPL    1$
      MOV    (SP)+,%0         ;RESTORE R0
      RTS    7                ;EXIT

;SUBROUTINE TO CHECK TRANSMITTED DATA
CHKDAT: SAVREG                ;SAVE THE REGISTERS ON THE STACK
      NOP
      CLR    %1                ;CLEAR CHARACTER COUNT
      MOV    #INTAB,%2        ;GET ADDRESS OF LINE'S INPUT BUFFER
      CLR    %3                ;ADDRESS ;GET LINE COUNT
1$:    MOV    %3,@#LINE        ;MOVE LINE # TO LINE
      MOV    (2)+,@#LINBUF    ;GET LINE'S INPUT BUFFER ADDRESS
      DEC    LINBUF           ;SUBTRACT 1 FROM LINE'S INPUT BUFFER ADDRESS
2$:    INC    LINBUF          ;INCREMENT LINE'S INPUT BUFFER ADDRESS
      MOVB  @#LINBUF,@#RCV DAT ;GET RECEIVED CHARACTER
      MOVB  OUTBUF(1),XMTDAT ;GET TRANSMITTED CHARACTER
      BIC   @#CARMSK,XMTDAT  ;CLEAR UNTRANSMITTED BITS
      DATCHK
      INC    %1                ;INCREMENT CHARACTER COUNT
      CMP    %1,#64.         ;ALL CHARACTERS BEEN COMPARED
      BNE   2$                ;GO CHECK NEXT CHAR. IF NOT
      CLR    %1                ;CLEAR CHARACTER COUNT
      INC    %3                ;INCREMENT LINE COUNT
      CMP    %3,#10.         ;ALL LINES CHECKED?
      BLT   1$                ;BRANCH IF ALL LINES NOT CHECKED
      RSTREG
      RTS    7                ;EXIT SUBROUTINE

;RECEIVER INTERRUPT SERVICE ROUTINE
RINT:  NOP                    ;BEGIN
      SAVREG                    ;SAVE THE REGISTERS ON THE STACK

```



```

1122 005432 013701 001572      MOV    @#TTPTR,%1      ;GET TUMBLE TABLE POINTER
1123 005436 011137 001564      MOV    (1),TTDAT      ;GET TUMBLE TABLE ENTRY
1124 005442 100410                BMI    2$              ;BRANCH IF VALID DATA ENTRY
1125 005444 104003                ERROR  ;ERROR! FALSE INTERRUPT
1126 005446 000454                BR     6$              ;EXIT
1127 005450 011137 001564      1$:   MOV    (1),@#TTDAT  ;GET TUMBLE TABLE ENTRY
1128 005454 001451                BEQ    6$              ;GO TO EXIT IF NO DATA ENTRY
1129 005456 100402                BMI    2$              ;BRANCH IF VALID DATA ENTRY
1130 005460 104003                ERROR  ;ERROR! NO VALID DATA ENTRY INDICATOR
1131 005462 000425                BR     3$              ;EXIT
1132 005464 005011                CLR    (1)             ;CLEAR TUMBLE TABLE ENTRY
1133 005466 042737 160400 001564    2$:   BIC    #160400,@#TTDAT ;CLEAR ALL BUT CHAR. & LINE #
1134 005474 113702 001565                MOV    TTDAT+1,%2     ;PUT LINE # IN R2 (LINE WILL BE IN LSH)
1135 005500 010204                MOV    %2,%4
1136 005502 016237 012626 002142    MOV    INTAB(2),@#LINBUF ;GET LINE'S INPUT BUFFER ADDRESS
1137 005510 006202                ASR    %2              ;SHIFT LINE #
1138 005512 005003                CLR    %3
1139 005514 116203 012666                MOV    CNTTAB(2),%3   ;GET LINE'S RECEIVED CHAR. COUNT
1140 005520 105262 012666                INCB   CNTTAB(2)     ;INCREMENT CHARACTER COUNT
1141 005524 060337 002142                ADD    %3,LINBUF      ;FORM ADDRESS WHERE CHAR. IS TO BE STORED
1142 005530 113777 001564 174404    MOV    TTDAT,@#LINBUF ;STORE CHAR. IN LINE'S INPUT BUFFER
1143 005536 000240                NOP
1144 005540 016437 001440 002122    3$:   MOV    WCT(4),RCV DAT ;GET TRANSMITTERS WORD COUNT
1145 005546 003405                BLE    4$              ;BRANCH IF WORD COUNT IS 0 OR NEGATIVE
1146 005550 010437 002124                MOV    %4,XMTDAT      ;GET LINE # OF FAILING LINE
1147 005554 104015                ERROR  ;ERROR! INCORRECT WORD COUNT IN
1148                                ;TYPE OUT SHOWS FAILING LINE #, AND FAILING LINE'S WORD COUNT
1149 005556 000005                RESET ;STOP THE DM11
1150 005560 104012                SCOPE ;EXIT TEST
1151
1152 005562 022701 001776      4$:   CMP    #TUMTAB+176,%1 ;IS THE TUMBLE TABLE POINTER AT THE
1153 005566 001002                BNE    5$              ;THE END OF THE TABLE
1154 005570 012701 001576                MOV    #TUMTAB-2,%1   ;RESET POINTER
1155 005574 005721                5$:   TST    (1)+           ;INCREMENT POINTER
1156 005576 000724                BR     1$              ;GO CHECK NEXT ENTRY
1157 005600 042777 000200 173734    6$:   BIC    #BIT7,@CSR     ;CLEAR RECEIVER DONE FLAG
1158 005606 010137 001572                MOV    %1,TTPTR      ;SAVE POINTER
1159 005612 104014                RSTREG ;RESTORE THE REGISTERS
1160 005614 000240                NOP
1161 005616 000002                RTI
1162                                ;EXIT SERVICE ROUTINE
1163                                ;TRANSMITTER INTERRUPT SERVICE ROUTINE
1164 005620 000240      TINT:  NOP              ;BEGIN
1165 005622 032777 060000 173712    BIT    #BIT14+BIT13,@CSR ;TEST ERROR FLAGS
1166 005630 001404                BEQ    1$              ;BRANCH IF NO ERROR FLAGS
1167 005632 104003                ERROR  ;ERROR! ERROR FLAG IS SET
1168 005634 042777 060000 173700    BIC    #BIT14+BIT13,@CSR ;CLEAR ERROR FLAGS
1169 005642 005777 173674      1$:   TST    @CSR          ;TEST READY FLAG
1170 005646 100003                BPL    2$              ;BRANCH IF READY IS CLEAR
1171 005650 042777 100000 173664    BIC    #BIT15,@CSR    ;CLEAR READY FLAG
1172 005656 000002      2$:   RTI
1173
1174

```

```

1175
1176 005660 104000
1177 005662 013421
1178 005664 012737 005720 002010 PRGO: TYPE
1179 005672 005037 002014 PRGOM
1180 005676 000137 002506 PRGOA: MOV #RTO,KSTART ;GET ADDRESS OF FIRST TEST
1181 005702 012737 005720 002010 PRGOR: CLR RTNNO ;CLEAR ROUTINE #
1182 005710 005037 002014 PRGOR: JMP SRSET
1183 005714 000137 002540 PRGOR: MOV #RTO,KSTART ;GET ADDRESS OF FIRST TEST
1184 ;*****
1185 005720 000000 RT0: 0 ;ROUTINE # 0
1186 005722 005740 RT1 ;ADDR OF NEXT ROUTINE.
1187 005724 000002 2 ;ITERATION COUNT
1188 005726 005730 RTOA ;SCOPE ENTRY POINT.
1189 000000 X=X+1
1190 ;*****
1191 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1192 ;NEXT LINE.
1193 005730 004537 004472 RTOA: JSR 5,DLYXMT ;GO DO TEST.
1194 005734 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1195 005736 000000 0 ;DELAY THIS MUCH BETWEEN LINES
1196 ;*****
1197 005740 000001 RT1: 1 ;ROUTINE # 1
1198 005742 005760 RT2 ;ADDR OF NEXT ROUTINE.
1199 005744 000002 2 ;ITERATION COUNT
1200 005746 005750 RT1A ;SCOPE ENTRY POINT.
1201 000001 X=X+1
1202 ;*****
1203 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1204 ;NEXT LINE.
1205 005750 004537 004472 RT1A: JSR 5,DLYXMT ;GO DO TEST.
1206 005754 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1207 005756 177740 -32. ;DELAY THIS MUCH BETWEEN LINES
1208 ;*****
1209 005760 000002 RT2: 2 ;ROUTINE # 2
1210 005762 006000 RT3 ;ADDR OF NEXT ROUTINE.
1211 005764 000002 2 ;ITERATION COUNT
1212 005766 005770 RT2A ;SCOPE ENTRY POINT.
1213 000002 X=X+1
1214 ;*****
1215 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1216 ;NEXT LINE.
1217 005770 004537 004472 RT2A: JSR 5,DLYXMT ;GO DO TEST.
1218 005774 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1219 005776 177720 -48. ;DELAY THIS MUCH BETWEEN LINES
1220 ;*****
1221 006000 000003 RT3: 3 ;ROUTINE # 3
1222 006002 006020 RT4 ;ADDR OF NEXT ROUTINE.
1223 006004 000002 2 ;ITERATION COUNT
1224 006006 006010 RT3A ;SCOPE ENTRY POINT.
1225 000003 X=X+1
1226 ;*****
1227 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1228 ;NEXT LINE.
1229 006010 004537 004472 RT3A: JSR 5,DLYXMT ;GO DO TEST.
1230 006014 013743 MSG1 ;TRANSMIT THIS MESSAGE &
  
```

```
1231 006016 177710 -56. ;DELAY THIS MUCH BETWEEN LINES
1232 ;*****
1233 006020 000004 RT4: 4 ;ROUTINE # 4
1234 006022 006040 RT5 ;ADDR OF NEXT ROUTINE.
1235 006024 000002 2 ;ITERATION COUNT
1236 006026 006030 RT4A ;SCOPE ENTRY POINT.
1237 000004 X=X+1
1238 ;*****
1239 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1240 ;NEXT LINE.
1241 006030 004537 004472 RT4A: JSR 5,DLYXMT ;GO DO TEST.
1242 006034 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1243 006036 177704 -60. ;DELAY THIS MUCH BETWEEN LINES
1244 ;*****
1245 006040 000005 RT5: 5 ;ROUTINE # 5
1246 006042 006060 RT6 ;ADDR OF NEXT ROUTINE.
1247 006044 000002 2 ;ITERATION COUNT
1248 006046 006050 RT5A ;SCOPE ENTRY POINT.
1249 000005 X=X+1
1250 ;*****
1251 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1252 ;NEXT LINE.
1253 006050 004537 004472 RT5A: JSR 5,DLYXMT ;GO DO TEST.
1254 006054 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1255 006056 177702 -62. ;DELAY THIS MUCH BETWEEN LINES
1256 ;*****
1257 006060 000006 RT6: 6 ;ROUTINE # 6
1258 006062 006100 RT7 ;ADDR OF NEXT ROUTINE.
1259 006064 000002 2 ;ITERATION COUNT
1260 006066 006070 RT6A ;SCOPE ENTRY POINT.
1261 000006 X=X+1
1262 ;*****
1263 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1264 ;NEXT LINE.
1265 006070 004537 004472 RT6A: JSR 5,DLYXMT ;GO DO TEST.
1266 006074 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1267 006076 177701 -63. ;DELAY THIS MUCH BETWEEN LINES
1268 ;*****
1269 006100 000007 RT7: 7 ;ROUTINE # 7
1270 006102 006120 RT10 ;ADDR OF NEXT ROUTINE.
1271 006104 000002 2 ;ITERATION COUNT
1272 006106 006110 RT7A ;SCOPE ENTRY POINT.
1273 000007 X=X+1
1274 ;*****
1275 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1276 ;NEXT LINE.
1277 006110 004537 004472 RT7A: JSR 5,DLYXMT ;GO DO TEST.
1278 006114 013743 MSG1 ;TRANSMIT THIS MESSAGE &
1279 006116 177700 -64. ;DELAY THIS MUCH BETWEEN LINES
1280 ;*****
1281 006120 000010 RT10: 10 ;ROUTINE # 10
1282 006122 006140 RT11 ;ADDR OF NEXT ROUTINE.
1283 006124 000002 2 ;ITERATION COUNT
1284 006126 006130 RT10A ;SCOPE ENTRY POINT.
1285 000010 X=X+1
1286 ;*****
```

1287				;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1288				;NEXT LINE.
1289	006130	004537	004472	RT10A: JSR 5,DLYXMT ;GO DO TEST.
1290	006134	014044		MSG2 ;TRANSMIT THIS MESSAGE &
1291	006136	177740		-32. ;DELAY THIS MUCH BETWEEN LINES
1292				*****
1293	006140	000011		RT11: 11 ;ROUTINE # 11
1294	006142	006160		RT12 ;ADDR OF NEXT ROUTINE.
1295	006144	000002		2 ;ITERATION COUNT
1296	006146	006150		RT11A ;SCOPE ENTRY POINT.
1297		000011		X=X+1
1298				*****
1299				;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1300				;NEXT LINE.
1301	006150	004537	004472	RT11A: JSR 5,DLYXMT ;GO DO TEST.
1302	006154	014044		MSG2 ;TRANSMIT THIS MESSAGE &
1303	006156	177720		-48. ;DELAY THIS MUCH BETWEEN LINES
1304				*****
1305	006160	000012		RT12: 12 ;ROUTINE # 12
1306	006162	006200		RT13 ;ADDR OF NEXT ROUTINE.
1307	006164	000002		2 ;ITERATION COUNT
1308	006166	006170		RT12A ;SCOPE ENTRY POINT.
1309		000012		X=X+1
1310				*****
1311				;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1312				;NEXT LINE.
1313	006170	004537	004472	RT12A: JSR 5,DLYXMT ;GO DO TEST.
1314	006174	014044		MSG2 ;TRANSMIT THIS MESSAGE &
1315	006176	177710		-56. ;DELAY THIS MUCH BETWEEN LINES
1316				*****
1317	006200	000013		RT13: 13 ;ROUTINE # 13
1318	006202	006220		RT14 ;ADDR OF NEXT ROUTINE.
1319	006204	000002		2 ;ITERATION COUNT
1320	006206	006210		RT13A ;SCOPE ENTRY POINT.
1321		000013		X=X+1
1322				*****
1323				;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1324				;NEXT LINE.
1325	006210	004537	004472	RT13A: JSR 5,DLYXMT ;GO DO TEST.
1326	006214	014044		MSG2 ;TRANSMIT THIS MESSAGE &
1327	006216	177704		-60. ;DELAY THIS MUCH BETWEEN LINES
1328				*****
1329	006220	000014		RT14: 14 ;ROUTINE # 14
1330	006222	006240		RT15 ;ADDR OF NEXT ROUTINE.
1331	006224	000002		2 ;ITERATION COUNT
1332	006226	006230		RT14A ;SCOPE ENTRY POINT.
1333		006014		X=X+1
1334				*****
1335				;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1336				;NEXT LINE.
1337	006230	004537	004472	RT14A: JSR 5,DLYXMT ;GO DO TEST.
1338	006234	014044		MSG2 ;TRANSMIT THIS MESSAGE &
1339	006236	177702		-62. ;DELAY THIS MUCH BETWEEN LINES
1340				*****
1341	006240	000015		RT15: 15 ;ROUTINE # 15
1342	006242	006260		RT16 ;ADDR OF NEXT ROUTINE.

```
1343 006244 000002          2          : ITERATION COUNT          *
1344 006246 006250          RT15A         : SCOPE ENTRY POINT.      *
1345          000015          X=X+1
1346          :*****
1347          ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1348          ;NEXT LINE.
1349 006250 004537 004472    RT15A: JSR      5,DLYXMT      ;GO DO TEST.
1350 006254 014044          MSG2          ;TRANSMIT THIS MESSAGE &
1351 006256 177701          -63.         ;DELAY THIS MUCH BETWEEN LINES
1352          :*****
1353 006260 000016          RT16:  16          ;ROUTINE # 16
1354 006262 006300          RT17         ;ADDR OF NEXT ROUTINE.
1355 006264 000002          2           ;ITERATION COUNT
1356 006266 006270          RT16A        ;SCOPE ENTRY POINT.
1357          X=X+1
1358          :*****
1359          ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1360          ;NEXT LINE.
1361 006270 004537 004472    RT16A: JSR      5,DLYXMT      ;GO DO TEST.
1362 006274 014044          MSG2          ;TRANSMIT THIS MESSAGE &
1363 006276 177700          -64.         ;DELAY THIS MUCH BETWEEN LINES
1364          :*****
1365 006300 000017          RT17:  17          ;ROUTINE # 17
1366 006302 006320          RT20         ;ADDR OF NEXT ROUTINE.
1367 006304 000002          2           ;ITERATION COUNT
1368 006306 006310          RT17A        ;SCOPE ENTRY POINT.
1369          X=X+1
1370          :*****
1371          ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1372          ;NEXT LINE.
1373 006310 004537 004472    RT17A: JSR      5,DLYXMT      ;GO DO TEST.
1374 006314 014144          MSG3          ;TRANSMIT THIS MESSAGE &
1375 006316 177720          -48.         ;DELAY THIS MUCH BETWEEN LINES
1376          :*****
1377 006320 000020          RT20:  20          ;ROUTINE # 20
1378 006322 006340          RT21         ;ADDR OF NEXT ROUTINE.
1379 006324 000002          2           ;ITERATION COUNT
1380 006326 006330          RT20A        ;SCOPE ENTRY POINT.
1381          X=X+1
1382          :*****
1383          ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1384          ;NEXT LINE.
1385 006330 004537 004472    RT20A: JSR      5,DLYXMT      ;GO DO TEST.
1386 006334 014144          MSG3          ;TRANSMIT THIS MESSAGE &
1387 006336 177704          -60.         ;DELAY THIS MUCH BETWEEN LINES
1388          :*****
1389 006340 000021          RT21:  21          ;ROUTINE # 21
1390 006342 006360          RT22         ;ADDR OF NEXT ROUTINE.
1391 006344 000002          2           ;ITERATION COUNT
1392 006346 006350          RT21A        ;SCOPE ENTRY POINT.
1393          X=X+1
1394          :*****
1395          ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1396          ;NEXT LINE.
1397 006350 004537 004472    RT21A: JSR      5,DLYXMT      ;GO DO TEST.
1398 006354 014144          MSG3          ;TRANSMIT THIS MESSAGE &
```

```
1399 006356 177701 -63. ;DELAY THIS MUCH BETWEEN LINES
1400 ;*****
1401 006360 000022 RT22: 22 ;ROUTINE # 22
1402 006362 006400 RT23 ;ADDR OF NEXT ROUTINE.
1403 006364 000002 2 ;ITERATION COUNT
1404 006366 006370 RT22A ;SCOPE ENTRY POINT.
1405 000022 X-X+1
1406 ;*****
1407 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1408 ;NEXT LINE.
1409 006370 004537 004472 RT22A: JSR 5,DLYXMT ;GO DO TEST.
1410 006374 014144 MSG3 ;TRANSMIT THIS MESSAGE &
1411 006376 177700 -64. ;DELAY THIS MUCH BETWEEN LINES
1412 ;*****
1413 006400 000023 RT23: 23 ;ROUTINE # 23
1414 006402 006420 RT24 ;ADDR OF NEXT ROUTINE.
1415 006404 000002 2 ;ITERATION COUNT
1416 006406 006410 RT23A ;SCOPE ENTRY POINT.
1417 000023 X=X+1
1418 ;*****
1419 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1420 ;NEXT LINE.
1421 006410 004537 004472 RT23A: JSR 5,DLYXMT ;GO DO TEST.
1422 006414 014244 MSG4 ;TRANSMIT THIS MESSAGE &
1423 006416 177740 -32. ;DELAY THIS MUCH BETWEEN LINES
1424 ;*****
1425 006420 000024 RT24: 24 ;ROUTINE # 24
1426 006422 006440 RT25 ;ADDR OF NEXT ROUTINE.
1427 006424 000002 2 ;ITERATION COUNT
1428 006426 006430 RT24A ;SCOPE ENTRY POINT.
1429 000024 X=X+1
1430 ;*****
1431 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1432 ;NEXT LINE.
1433 006430 004537 004472 RT24A: JSR 5,DLYXMT ;GO DO TEST.
1434 006434 014244 MSG4 ;TRANSMIT THIS MESSAGE &
1435 006436 177710 -56. ;DELAY THIS MUCH BETWEEN LINES
1436 ;*****
1437 006440 000025 RT25: 25 ;ROUTINE # 25
1438 006442 006460 RT26 ;ADDR OF NEXT ROUTINE.
1439 006444 000002 2 ;ITERATION COUNT
1440 006446 006450 RT25A ;SCOPE ENTRY POINT.
1441 000025 X=X+1
1442 ;*****
1443 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1444 ;NEXT LINE.
1445 006450 004537 004472 RT25A: JSR 5,DLYXMT ;GO DO TEST.
1446 006454 014244 MSG4 ;TRANSMIT THIS MESSAGE &
1447 006456 177702 -62. ;DELAY THIS MUCH BETWEEN LINES
1448 ;*****
1449 006460 000026 RT26: 26 ;ROUTINE # 26
1450 006462 006500 RT27 ;ADDR OF NEXT ROUTINE.
1451 006464 000002 2 ;ITERATION COUNT
1452 006466 006470 RT26A ;SCOPE ENTRY POINT.
1453 000026 X-X+1
1454 ;*****
```

```
1455 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
1456 ;NEXT LINE.  
1457 006470 004537 004472 RT26A: JSR 5,DLYXMT ;GO DO TEST.  
1458 006474 014244 MSG4 ;TRANSMIT THIS MESSAGE &  
1459 006476 177700 -64. ;DELAY THIS MUCH BETWEEN LINES  
1460 ;*****  
1461 006500 000027 RT27: 27 ;ROUTINE # 27  
1462 006502 006520 RT30 ;ADDR OF NEXT ROUTINE.  
1463 006504 000002 2 ;ITERATION COUNT  
1464 006506 006510 RT27A ;SCOPE ENTRY POINT.  
1465 000027 X=X+1  
1466 ;*****  
1467 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
1468 ;NEXT LINE.  
1469 006510 004537 004472 RT27A: JSR 5,DLYXMT ;GO DO TEST.  
1470 006514 014344 MSG5 ;TRANSMIT THIS MESSAGE &  
1471 006516 177720 -48. ;DELAY THIS MUCH BETWEEN LINES  
1472 ;*****  
1473 006520 000030 RT30: 30 ;ROUTINE # 30  
1474 006522 006540 RT31 ;ADDR OF NEXT ROUTINE.  
1475 006524 000002 2 ;ITERATION COUNT  
1476 006526 006530 RT30A ;SCOPE ENTRY POINT.  
1477 000030 X=X+1  
1478 ;*****  
1479 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
1480 ;NEXT LINE.  
1481 006530 004537 004472 RT30A: JSR 5,DLYXMT ;GO DO TEST.  
1482 006534 014344 MSG5 ;TRANSMIT THIS MESSAGE &  
1483 006536 177710 -56. ;DELAY THIS MUCH BETWEEN LINES  
1484 ;*****  
1485 006540 000031 RT31: 31 ;ROUTINE # 31  
1486 006542 006560 RT32 ;ADDR OF NEXT ROUTINE.  
1487 006544 000002 2 ;ITERATION COUNT  
1488 006546 006550 RT31A ;SCOPE ENTRY POINT.  
1489 000031 X=X+1  
1490 ;*****  
1491 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
1492 ;NEXT LINE.  
1493 006550 004537 004472 RT31A: JSR 5,DLYXMT ;GO DO TEST.  
1494 006554 014344 MSG5 ;TRANSMIT THIS MESSAGE &  
1495 006556 177704 -60. ;DELAY THIS MUCH BETWEEN LINES  
1496 ;*****  
1497 006560 000032 RT32: 32 ;ROUTINE # 32  
1498 006562 006600 RT33 ;ADDR OF NEXT ROUTINE.  
1499 006564 000002 2 ;ITERATION COUNT  
1500 006566 006570 RT32A ;SCOPE ENTRY POINT.  
1501 000032 X=X+1  
1502 ;*****  
1503 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
1504 ;NEXT LINE.  
1505 006570 004537 004472 RT32A: JSR 5,DLYXMT ;GO DO TEST.  
1506 006574 014344 MSG5 ;TRANSMIT THIS MESSAGE &  
1507 006576 177702 -62. ;DELAY THIS MUCH BETWEEN LINES  
1508 ;*****  
1509 006600 000033 RT33: 33 ;ROUTINE # 33  
1510 006602 006620 RT34 ;ADDR OF NEXT ROUTINE.
```

```
1511 006604 000002          2          : ITERATION COUNT
1512 006606 006610          RT33A          : SCOPE ENTRY POINT.
1513          000033          X=X+1
1514          : *****
1515          : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1516          : NEXT LINE.
1517 006610 004537 004472 RT33A: JSR      5,DLYXMT      : GO DO TEST.
1518 006614 014344          MSG5          : TRANSMIT THIS MESSAGE &
1519 006616 177701          -63.          : DELAY THIS MUCH BETWEEN LINES
1520          : *****
1521 006620 000034          RT34:  34          : ROUTINE # 34
1522 006622 177777          RT35          : ADDR OF NEXT ROUTINE.
1523 006624 000002          2          : ITERATION COUNT
1524 006626 006630          RT34A          : SCOPE ENTRY POINT.
1525          000034          X=X+1
1526          : *****
1527          : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1528          : NEXT LINE.
1529 006630 004537 004472 RT34A: JSR      5,DLYXMT      : GO DO TEST.
1530 006634 014344          MSG5          : TRANSMIT THIS MESSAGE &
1531 006636 177700          -64.          : DELAY THIS MUCH BETWEEN LINES
1532          177777          RT35=-1
```



```
1533
1534
1535 ;PRG1- DATA TESTS ALL LINES SIMULTANEOUSLY. DATA TRANSMITTED IS 'THE
1536 006640 104000 ;QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
1537 006642 013447 PRG1: TYPE ;TYPE
1538 006644 022737 000176 001202 PRGX: PRG1M ;PROGRAM TITLE
1539 006652 001001 ;SEE IF SWITCH-LESS
1540 006654 104020 ;BRANCH IF NOT
1541 006656 004537 004472 PRG1R: CMTLU ;GET SWREG SETTINGS
1542 006662 013743 ;GO TO DLYXMT TO SET UP DM11
1543 006664 177700 ;MSG1 WILL BE THE DATA TRANSMITTED
1544 006666 012737 007362 001400 PRG1A: MOV #-64.,WCT ;DO NOT DELAY
1545 006674 004537 005200 ;LOAD CURRENT
1546 006680 001400 ;ADDRESS TABLE
1547 006682 001402 ;TO POINT TO
1548 006684 000040 ;OUTBUF
1549 006686 012737 177700 001440 MOV #-64.,WCT ;LOAD WORD COUNT
1550 006688 004537 005200 ;TO -64.
1551 006690 001440 WCT
1552 006692 001442 WCT+2
1553 006694 000040 32.
1554 006696 012777 010100 172606 MOV #BIT12+BIT6,@CSR ;SET TRANSMITTER & RECIEVER IE BITS
1555 006698 023727 002446 000004 CMP PRGNUM,#4 ;RUNNING PROGRAM #2?
1556 006700 001403 BEQ .+10
1557 006702 052777 000001 172570 BIS #BIT0,@CSR ;SET THE GO BIT
1558 006704 012777 177777 172564 MOV #-1,@BAR ;START TRANSMITTING ON ALL LINES
1559 006706 005777 172560 TST @BAR ;WAIT FOR ALL LINES TO COMPLETE
1560 006708 001375 BNE .-4
1561 006710 005205 INC %5
1562 006712 001376 BNE .-2
1563 006714 005077 172544 CLR @CSP
1564 006716 023727 002446 000004 PRG1C: CMP PRGNUM,#4 ;DO NOT CHECK DATA IF RUNNING
1565 006718 001402 BEQ PRG1D ;PROGRAM # 2
1566 006720 004737 005320 JSR 7,CHKDAT ;GO CHECK RECEIVED DATA
1567 006722 104000 PRG1D: TYPE ;TYPE
1568 006724 013303 M2 ;'PRGEND'
1569 006726 012706 001200 PRG1EX: MOV #SPBOT,SP ;RESET THE STACK POINTER
1570 006728 000715 BR PRG1R ;GO RESTART TEST
1571
1572
1573 ;PRG2-PROGRAM 2 RUNS PROGRAM 1 EXCEPT FOR THE DATA CHECKING
1574 ;WHEN ALL LINES ARE FINISHED TRANSMITTING. THIS ALLOWS THE DATA
1575 ;TRANSMITTED TO BE SENT TO TERMINALS. BEFORE STARTING THIS PROGRAM
1576 ;REMOVE THE JUMPERS CONNECTING THE TRANSMITTERS TO THE RECEIVERS.
1577 007024 104000 PRG2: TYPE ;TYPE PROGRAM TITLE
1578 007026 013527 PRG2M ;AND INSTRUCTIONS
1579 007030 000137 006644 PRG2R: JMP PRGX ;GO RUN PRG1
```

```

1580
1581
1582 ;PRG3-ECHO TEST THIS PROGRAM ECHOS BACK DATA RECEIVED FROM ANY DM11
1583 ;TERMINAL(S)
1584 ;NOTE: THIS TEST IS THE ONLY TEST THAT INSURES PROPER OPERATION
1585 ;OF THE DM11 DISTRIBUTION PANEL LOGIC.
1586 007034 104000 PRG3: TYPE ;TYPE PROGRAM
1587 007036 013556 PRG3M ;TITLE
1588 007040 022737 000176 001202 CMP #SWREG,SWR ;SEE IF SWITCH-LESS
1589 007046 001001 BNE PRG3R ;BRANCH IF NOT
1590 007050 104020 CNTLU ;GET SWREG SETTINGS
1591 007052 004537 004472 PRG3R: JSR 5,DLYXMT ;USE PART OF THE
1592 007056 013743 MSG1 ;DLYXMT ROUTINE TO
1593 007060 000240 NOP ;SET UP DM11
1594 007062 012737 001600 001572 PRG3A: MOV #TUMTAB,TTPTR ;INITIALIZE SOFTWARE POINTER
1595 007070 013701 001552 MOV CLKINT,%1 ;LOAD RECEIVER
1596 007074 012721 007132 MOV #RINT3,(1)+ ;AND TRANSMITTER
1597 007100 013721 001554 MOV CLKLVL,(1)+ ;VECTORS AND PRIORITY
1598 007104 012721 007336 MOV #TINT3,(1)+ ;LEVELS
1599 007110 013721 001560 MOV XMTLVL,(1)+
1600 007114 012777 010101 172420 MOV #BIT12+BIT6+BIT0,@CSR ;SET IE AND GO BITS
1601 007122 012700 000001 MOV #1,%0
1602 007126 005200 INC %0
1603 007130 000776 BR -2
1604
1605
1606 007132 000240 RINT3: NOP
1607 007134 000240 NOP
1608 007136 013701 001572 MOV TTPTR,%1 ;GET SOFTWARE POINTER
1609 007142 011137 001564 RINT3A: MOV (1),TTDAT ;GET TUMBLE TABLE ENTRY
1610 007146 001463 BEQ RINT3X ;EXIT IF NO ENTRY
1611 007150 005011 CLR (1) ;CLEAR ENTRY
1612 007152 032737 040000 001564 BIT #BIT14,TTDAT ;WAS BREAK RECEIVED
1613 007160 001047 BNE RINT3B ;DO NOTHING ABOUT IT
1614 007162 042737 160400 001564 BIC #160400,TTDAT ;CLEAR ALL BUT LINE # AND DATA
1615 007170 113702 001565 MOVB TTDAT+1,%2 ;GET LINE NUMBER
1616 007174 010237 002140 MOV %2,LINE ;FETCH LINE NUMBER
1617 007200 004737 005266 JSR 7,GTLINE ;FORM LINE BIT FOR BAR
1618 007204 033777 001566 172332 BIT LINBIT,@BAR ;IS THIS LINE ACTIVE
1619 007212 001414 BEQ NONACT ;LINE NOT ACTIVE
1620 007214 033777 001566 172322 BIT LINBIT,@BAR ;WAIT FOR LINE
1621 007222 001374 BNE -6
1622 007224 032777 060000 172310 BIT #BIT14+BIT13,@CSR
1623 007232 001401 BEQ -+4 ;BRANCH IF NO ERRORS
1624 007234 104003 ERROR
1625 007236 042777 100000 172276 BIC #BIT15,@CSR ;CLEAR TRANSMIT DONE
1626 007244 113762 001564 007362 NONACT: MOVB TTDAT,OUTBUF(2) ;STORE RECEIVED CHARACTER
1627 007252 012762 177777 001440 MOV #-1,WCT(2) ;LOAD LINE'S WORD COUNT
1628 007260 010203 MOV %2,%3
1629 007262 062703 007362 ADD #OUTBUF,%3
1630 007266 010362 001400 MOV %3,CAT(2) ;AND CURRENT ADDRESS
1631 007272 053777 001566 172244 BIS LINBIT,@BAR ;ECHO RECEIVED CHARACTER
1632 007300 022701 001776 RINT3B: CMP #TUMTAB+176,%1 ;CHECK TUMBLE
1633 007304 001002 BNE -+6 ;TABLE POINTER
1634 007306 012701 001576 MOV #TUMTAB-2,%1
1635 007312 005721 TST (1)+
    
```

```
1636 007314 000712 BR RINT3A
1637 007316 042777 000200 172216 RINT3X: BIC #BIT7,@CSR ;CLEAR CHARACTER DONE FLAG
1638 007324 010137 001572 MOV Z1,TIPTR ;RESTORE POINTER
1639 007330 000240 NOP
1640 007332 000240 NOP
1641 007334 000002 RTI ;EXIT
1642
1643 007336 000240 TINT3: NOP
1644 007340 032777 060000 172174 BIT #BIT14+BIT13,@CSR ;ANY ERROR FLAGS SET
1645 007346 001401 BEQ .+4
1646 007350 104003 ERROR
1647 007352 042777 160000 172162 BIC #BIT15+BIT14+BIT13,@CSR ;CLEAR ALL FLAGS
1648 007360 000002 RTI ;EXIT
```

1649	007362	000000	OUTBUF: 0
1650		007526	.=OUTBUF+100.
1651	007526	000000	LN0BUF: 0
1652		007672	.=LN0BUF+100.
1653	007672	000000	LN1BUF: 0
1654		010036	.=LN1BUF+100.
1655	010036	000000	LN2BUF: 0
1656		010202	.=LN2BUF+100.
1657	010202	000000	LN3BUF: 0
1658		010346	.=LN3BUF+100.
1659	010346	000000	LN4BUF: 0
1660		010512	.=LN4BUF+100.
1661	010512	000000	LN5BUF: 0
1662		010656	.=LN5BUF+100.
1663	010656	000000	LN6BUF: 0
1664		011022	.=LN6BUF+100.
1665	011022	000000	LN7BUF: 0
1666		011166	.=LN7BUF+100.
1667	011166	000000	LN10BF: 0
1668		011332	.=LN10BF+100.
1669	011332	000000	LN11BF: 0
1670		011476	.=LN11BF+100.
1671	011476	000000	LN12BF: 0
1672		011642	.=LN12BF+100.
1673	011642	000000	LN13BF: 0
1674		012006	.=LN13BF+100.
1675	012006	000000	LN14BF: 0
1676		012152	.=LN14BF+100.
1677	012152	000000	LN15BF: 0
1678		012316	.=LN15BF+100.
1679	012316	000000	LN16BF: 0
1680		012462	.=LN16BF+100.
1681	012462	000000	LN17BF: 0
1682		012626	.=LN17BF+100.
1683	012626	007526	INTAB: LN0BUF
1684	012630	007672	LN1BUF
1685	012632	010036	LN2BUF
1686	012634	010202	LN3BUF
1687	012636	010346	LN4BUF
1688	012640	010512	LN5BUF
1689	012642	010656	LN6BUF
1690	012644	011022	LN7BUF
1691	012646	011166	LN10BF
1692	012650	011332	LN11BF
1693	012652	011476	LN12BF
1694	012654	011642	LN13BF
1695	012656	012006	LN14BF
1696	012660	012152	LN15BF
1697	012662	012316	LN16BF
1698	012664	012462	LN17BF
1699	012666	000000	CNTTAB: 0
1700		012706	.=CNTTAB+16.
1701			
1702	012706	012746	ID: IDENT0
1703	012710	012752	IDENT1
1704	012712	012756	IDENT2

1705	012714	012762	
1706	012716	012766	IDENT3
1707	012720	012772	IDENT4
1708	012722	012776	IDENT5
1709	012724	013002	IDENT6
1710	012726	013006	IDENT7
1711	012730	013012	IDNT10
1712	012732	013016	IDNT11
1713	012734	013022	IDNT12
1714	012736	013026	IDNT13
1715	012740	013032	IDNT14
1716	012742	013036	IDNT15
1717	012744	013042	IDNT16
1718	012746	105215	IDNT17
1719	012750	030060	IDENT0: CRLF
1720	012752	105215	"00
1721	012754	030460	IDENT1: CRLF
1722	012756	105215	"01
1723	012760	031060	IDENT2: CRLF
1724	012762	105215	"02
1725	012764	031460	IDENT3: CRLF
1726	012766	105215	"03
1727	012770	032060	IDENT4: CRLF
1728	012772	105215	"04
1729	012774	032460	IDENT5: CRLF
1730	012776	105215	"05
1731	013000	033060	IDENT6: CRLF
1732	013002	105215	"06
1733	013004	033460	IDENT7: CRLF
1734	013006	105215	"07
1735	013010	030061	IDNT10: CRLF
1736	013012	105215	"10
1737	013014	030461	IDNT11: CRLF
1738	013016	105215	"11
1739	013020	031061	IDNT12: CRLF
1740	013022	105215	"12
1741	013024	031461	IDNT13: CRLF
1742	013026	105215	"13
1743	013030	032061	IDNT14: CRLF
1744	013032	105215	"14
1745	013034	032461	IDNT15: CRLF
1746	013036	105215	"15
1747	013040	033061	IDNT16: CRLF
1748	013042	105215	"16
1749	013044	033461	IDNT17: CRLF
1750	013046	105215	"17
1751		105215	CRLF
1752			CRLF-105215

1753						
1754						
1755	013050	042045	030515	020061		;MESSAGES
1756	013056	042522	042503	053111		WHERE: .ASCII '%DM11 RECEIVER VECTOR ADDRESS @'
1757	013064	051105	053040	041505		
1758	013072	047524	020122	042101		
1759	013100	051104	051505	020123		
1760	013106	020075	100			
1761	013111	045	053523	036522		\$SWREG: .ASCII '%SWR- @'
1762	013116	040040				
1763	013120	020040	020040	020040		\$VALUE: .ASCII ' NEW- @'
1764	013126	020040	020040	042516		
1765	013134	036527	040040			
1766	013140	036445	040			\$CTLU: .ASCII '%- '
1767	013143	045	044127	041511		WHICH: .ASCII '%WHICH DM11 ARE YOU TESTING @'
1768	013150	020110	046504	030461		
1769	013156	040440	042522	054440		
1770	013164	052517	052040	051505		
1771	013172	044524	043516	040040		
1772	013200	042045	052101	020101		ERDAT: .ASCII '%DATA ERR S/B: '
1773	013206	051105	020122	051440		
1774	013214	041057	020072			
1775	013220	020040	020040	020040		AASB: .ASCII ' WAS: '
1776	013226	020040	040527	035123		
1777	013234	040				
1778	013235	040	020040	020040		AWAS: .ASCII ' @'
1779	013242	020040	100			
1780	013245	114	047111	020105		LINEM: .ASCII 'LINE # '
1781	013252	020043				
1782	013254	020040	040040			ALINE: .ASCII ' @'
1783	013260	052045	050131	020105		MO: .ASCII '%TYPE PROGRAM #@'
1784	013266	051120	043517	040522		
1785	013274	020115	040043			
1786	013300	037445	100			M1: .ASCII '%?@'
1787	013303	045	042524	052123		M2: .ASCII '%TEST CZDMB COMPLETE@'
1788	013310	041440	042132	041115		
1789	013316	041440	046517	046120		
1790	013324	052105	040105			
1791	013330	051445	052105	051440		M3: .ASCII '%SET SR OPTIONS. NORMAL OPERATION'
1792	013336	020122	050117	044524		
1793	013344	047117	027123	047040		
1794	013352	051117	040515	020114		
1795	013360	050117	051105	052101		
1796	013366	047511	116			
1797	013371	123	020122	020075		.ASCII '%SR - 000000 PRESS CONT.@'
1798	013376	030060	030060	030060		
1799	013404	050040	042522	051523		
1800	013412	041440	047117	027124		
1801	013420	100				
1802	013421	045	040504	040524		PRGOM: .ASCII '%DATA TEST ALL LINES @'
1803	013426	052040	051505	020124		
1804	013434	046101	020114	044514		
1805	013442	042516	020123	100		
1806	013447	045	040504	040524		PRGIM: .ASCII '%DATA TEST TRANSMIT ON ALL LINES SIMULTANEOUSLY@'
1807	013454	052040	051505	020124		
1808	013462	051124	047101	046523		

1809	013470	052111	047440	020116			
1810	013476	046101	020114	044514			
1811	013504	042516	020123	044523			
1812	013512	052515	052114	047101			
1813	013520	047505	051525	054514			
1814	013526	100					
1815							
1816	013527	045	051124	047101	PRG2M:	.ASCII	'%TRANSMIT TO TERMINALS@'
1817	013534	046523	052111	052040			
1818	013542	020117	042524	046522			
1819	013550	047111	046101	040123			
1820	013556	042445	044103	020117	PRG3M:	.ASCII	'%ECHO TEST@'
1821	013564	042524	052123	100			
1822	013571	045	052520	020124	PRG1:	.ASCII	'%PUT CHAR IN SR(0-7),DELAY IN SR(8-15)@'
1823	013576	044103	051101	044440			
1824	013604	020116	051123	030050			
1825	013612	033455	026051	042504			
1826	013620	040514	020131	047111			
1827	013626	051440	024122	026470			
1828	013634	032461	040051				
1829	013640	052045	050131	020105	POPPAR:	.ASCII	'%TYPE PARITY OPTION (N NO' DESIRED O=ODD, E-EVEN@'
1830	013646	040520	044522	054524			
1831	013654	047440	052120	047511			
1832	013662	020116	047050	047075			
1833	013670	052117	042040	051505			
1834	013676	051111	042105	047440			
1835	013704	047475	042104	020054			
1836	013712	036505	053105	047105			
1837	013720	100					
1838	013721	045	020122		EMO:	.ASCII	'%R '
1839	013724	020040	020040	041520	ATNUMB:	.ASCII	' PC- '
1840	013732	020075					
1841	013734	020040	020040	020040	APC:	.ASCII	' @'
1842	013742	100					
1843	013743	015	012		MSG1:	.BYTE	15,12
1844	013745	040	044124	020105		.ASCII	' THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
1845	013752	052521	041511	020113			
1846	013760	051102	053517	020116			
1847	013766	047506	020130	052512			
1848	013774	050115	042105	047440			
1849	014002	042526	020122	044124			
1850	014010	020105	040514	054532			
1851	014016	042040	043517	020123			
1852	014024	040502	045503	030440			
1853	014032	031462	032464	033466			
1854	014040	034470	060				
1855		014044					
1856	014044	015	012		MSG2:	.EVEN	
1857	014046	177400				.BYTE	15,12
1858	014050	177400				177400	
1859	014052	177400				177400	
1860	014054	177400				177400	
1861	014056	177400				177400	
1862	014060	177400				177400	
1863	014062	177400				177400	
1864	014064	177400				177400	

1865	014066	177400			177400
1866	014070	177400			177400
1867	014072	177400			177400
1868	014074	177400			177400
1869	014076	177400			177400
1870	014100	177400			177400
1871	014102	177400			177400
1872	014104	177400			177400
1873	014106	177400			177400
1874	014110	177400			177400
1875	014112	177400			177400
1876	014114	177400			177400
1877	014116	177400			177400
1878	014120	177400			177400
1879	014122	177400			177400
1880	014124	177400			177400
1881	014126	177400			177400
1882	014130	177400			177400
1883	014132	177400			177400
1884	014134	177400			177400
1885	014136	177400			177400
1886	014140	177400			177400
1887	014142	177400			177400
1888	014144	015	012	MSG3:	.BYTE 15,12
1889	014146	125252			ALTO
1890	014150	125252			ALTO
1891	014152	125252			ALTO
1892	014154	125252			ALTO
1893	014156	125252			ALTO
1894	014160	125252			ALTO
1895	014162	125252			ALTO
1896	014164	125252			ALTO
1897	014166	125252			ALTO
1898	014170	125252			ALTO
1899	014172	125252			ALTO
1900	014174	125252			ALTO
1901	014176	125252			ALTO
1902	014200	125252			ALTO
1903	014202	125252			ALTO
1904	014204	125252			ALTO
1905	014206	125252			ALTO
1906	014210	125252			ALTO
1907	014212	125252			ALTO
1908	014214	125252			ALTO
1909	014216	125252			ALTO
1910	014220	125252			ALTO
1911	014222	125252			ALTO
1912	014224	125252			ALTO
1913	014226	125252			ALTO
1914	014230	125252			ALTO
1915	014232	125252			ALTO
1916	014234	125252			ALTO
1917	014236	125252			ALTO
1918	014240	125252			ALTO
1919	014242	125252			ALTO
1920	014244	015	012	MSG4:	.BYTE 15,12



1921	014246	052525			ALT1
1922	014250	052525			ALT1
1923	014252	052525			ALT1
1924	014254	052525			ALT1
1925	014256	052525			ALT1
1926	014260	052525			ALT1
1927	014262	052525			ALT1
1928	014264	052525			ALT1
1929	014266	052525			ALT1
1930	014270	052525			ALT1
1931	014272	052525			ALT1
1932	014274	052525			ALT1
1933	014276	052525			ALT1
1934	014300	052525			ALT1
1935	014302	052525			ALT1
1936	014304	052525			ALT1
1937	014306	052525			ALT1
1938	014310	052525			ALT1
1939	014312	052525			ALT1
1940	014314	052525			ALT1
1941	014316	052525			ALT1
1942	014320	052525			ALT1
1943	014322	052525			ALT1
1944	014324	052525			ALT1
1945	014326	052525			ALT1
1946	014330	052525			ALT1
1947	014332	052525			ALT1
1948	014334	052525			ALT1
1949	014336	052525			ALT1
1950	014340	052525			ALT1
1951	014342	052525			ALT1
1952	014344	015	012	MSG5:	.BYTE 15.12
1953	014346	000400			400
1954	014350	002002			2002
1955	014352	010010			10010
1956	014354	040040			40040
1957	014356	000200			200
1958	014360	177377			177377
1959	014362	175775			175775
1960	014364	167767			167767
1961	014366	137737			137737
1962	014370	177500			177500
1963	014372	000400			400
1964	014374	002002			2002
1965	014376	010010			10010
1966	014400	040040			40040
1967	014402	000200			200
1968	014404	177377			177377
1969	014406	175775			175775
1970	014410	167767			167767
1971	014412	137737			137737
1972	014414	177500			177500
1973	014416	000400			400
1974	014420	002002			2002
1975	014422	010010			10010
1976	014424	040040			40040

1977	014426	000200		200	
1978	014430	177377		177377	
1979	014432	175775		175775	
1980	014434	167767		167767	
1981	014436	137737		137737	
1982	014440	177500		177500	
1983	014442	015	012	.BYTE	15,12
1984	014444	015	012	.BYTE	15,12
1985		000001		.END	







PRG1A	006666	988	1544#					
PRG1C	006776	1564#						
PRG1D	007012	1565	1567#					
PRG1EX	007016	1569#						
PRG1M	013447	1537	1806#					
PRG1R	006656	484	1539	1541#		1570		
PRG2	007024	481	1577#					
PRG2M	013527	1578	1816#					
PRG2R	007030	485	1579#					
PRG3	007034	482	1586#					
PRG3A	007062	977	1594#					
PRG3M	013556	1587	1820#					
PRG3R	007052	486	1589	1591#				
PRTY0 =	000000	259#						
PRTY1 =	000040	258#						
PRTY2 =	000100	257#						
PRTY3 =	000140	256#						
PRTY4 =	000200	255#	469	471				
PRTY5 =	000240	254#	457	459				
PRTY6 =	000300	253#						
PRTY7 =	000340	252#	306					
PRVCNT	002136	516#						
PSW =	177776	229#	617*	762*	832*			
RCVDAT	002122	510#	529	581	1006*	1104*	1144*	
RECD	004076	597	777	800	857#	925		
RINT	005426	982	1120#					
RINT3	007132	1596	1606#					
RINT3A	007142	1609#	1636					
RINT3B	007300	1613	1632#					
RINT3X	007316	1610	1637#					
RSTART	002036	483#	605					
RSTAT1	002422	434	592#					
RSTAT2	002472	436	603#					
RSTREG=	104014	279#	1050	1064	1115	1159		
RSTRG	003156	499	707#					
RTNNO	002014	474#	558	627	675*	1179*	1'82*	
RT0	005720	1178	1181	1185#				
RT0A	005730	1188	1193#					
RT1	005740	1186	1197#					
RT1A	005750	1200	1205#					
RT10	006120	1270	1281#					
RT10A	006130	1284	1289#					
RT11	006140	1282	1293#					
RT11A	006150	1296	1301#					
RT12	006160	1294	1305#					
RT12A	006170	1308	1313#					
RT13	006200	1306	1317#					
RT13A	006210	1320	1325#					
RT14	006220	1318	1329#					
RT14A	006230	1332	1337#					
RT15	006240	1330	1341#					
RT15A	006250	1344	1349#					
RT16	006260	1342	1353#					
RT16A	006270	1356	1361#					
RT17	006300	1354	1365#					
RT17A	006310	1368	1373#					

RT2	005760	1198	1209#				
RT2A	005770	1212	1217#				
RT20	006320	1366	1377#				
RT20A	006330	1380	1385#				
RT21	006340	1378	1389#				
RT21A	006350	1392	1397#				
RT22	006360	1390	1401#				
RT22A	006370	1404	1409#				
RT23	006400	1402	1413#				
RT23A	006410	1416	1421#				
RT24	006420	1414	1425#				
RT24A	006430	1428	1433#				
RT25	006440	1426	1437#				
RT25A	006450	1440	1445#				
RT26	006460	1438	1449#				
RT26A	006470	1452	1457#				
RT27	006500	1450	1461#				
RT27A	006510	1464	1469#				
RT3	006000	1210	1221#				
RT3A	006010	1224	1229#				
RT30	006520	1462	1473#				
RT30A	006530	1476	1481#				
RT31	006540	1474	1485#				
RT31A	006550	1488	1493#				
RT32	006560	1486	1497#				
RT32A	006570	1500	1505#				
RT33	006600	1498	1509#				
RT33A	006610	1512	1517#				
RT34	006620	1510	1521#				
RT34A	006630	1524	1529#				
RT35 =	177777	1522	1532#				
RT4	006020	1222	1233#				
RT4A	006030	1236	1241#				
RT5	006040	1234	1245#				
RT5A	006050	1248	1253#				
RT6	006060	1246	1257#				
RT6A	006070	1260	1265#				
RT7	006100	1258	1269#				
RT7A	006110	1272	1277#				
SAVREG=	104013	278#	1035	1057	1095	1121	
SAVRG	003116	498	693#				
SCOPE =	104012	277#	1016	1150			
SCOPEA	002706	644#	653				
SCOPEB	002714	643	646#				
SCOPEC	002750	647	651	654#			
SCOPEd	002752	624	655#				
SCOPIR	002022	477#	644	678*			
SPBOT	001200	441#	588	592	603	618	1569
SRSET	002506	606#	1180				
SRT	002110	505#					
STALL =	104002	271#					
START	002404	433	588#				
STRXV =	104006	274#					
STTXV =	104007	275#					
SUSWR -	104016	281#	589				
SUSWRR	004412	501	933#				

SWR	001202	442#	532	551	568	606	621	625	642	646	655	900	915	929*
SWREG	000176	936	940*	1538	1588									
TEMP	002130	430#	606	900	915	920	940	1538	1588					
TINT	005620	513#												
TINT3	007336	984	1164#											
TKCSR	002112	1598	1643#											
TKDBR	002114	506#	860											
TKLVL	002002	507#	862	905										
TKVTR	002000	469#												
TMP1	004370	468#												
TPCSR	002116	904*	905*	906*	907	926#	929							
TPDBR	002120	508#	735	879	882	885								
TPLVL	002006	509#	734*	864*	881*	884*								
TPVTR	002004	471#												
TIDAT	001564	470#												
TIPTR	001572	461#	1123*	1127*	1133*	1134	1142	1609*	1612	1614*	1615	1626		
TUMTAB	001600	464#	980*	1122	1158*	1594*	1608	1638*						
		466#	467	826*	831	961*	963	964	967	980	1152	1154	1594	1632
		1634												
TYP	003220	487	722#											
TYPE =	104000	269#	523	539	561	564	595	610	660	775	784	798	804	890
		892	917	923	1176	1536	1567	1577	1586					
TYPES =	104001	270#												
UNIT	003640	797	801#	802	807*	808*	809*	813						
VAC	001540	451#	465											
VECOK	003540	773	780	782#										
VECOKA	003550	784#	788	790										
VECOKB	003556	783	787#											
VECTOR	003522	770*	778#	779	781*	782	787	789	791	792*	793			
WCT	001440	447#	448	759*	821*	996	1001	1024*	1074*	1144	1549*	1551	1552	1627*
WHERE	013050	776	1755#											
WHICH	013143	799	1767#											
X =	000034	290#	1184	1189#	1196	1201#	1208	1213#	1220	1225#	1232	1237#	1244	1249#
		1256	1261#	1268	1273#	1280	1285#	1292	1297#	1304	1309#	1316	1321#	1328
		1333#	1340	1345#	1352	1357#	1364	1369#	1376	1381#	1388	1393#	1400	1405#
		1412	1417#	1424	1429#	1436	1441#	1448	1453#	1460	1465#	1472	1477#	1484
		1489#	1496	1501#	1508	1513#	1520	1525#						
XMITD	005222	994	1070#											
XMTDAT	002124	511#	529	577	1008*	1105*	1106*	1146*						
XMTINT	001556	458#	793*											
XMTLVL	001560	459#	985	1599										
Y =	000000	289#												
\$CTLU	013140	893	1766#											
\$ENDAD	003022	423	669#											
\$SWREG	013111	918	1761#											
\$VALUE	013120	921	924	1763#										
.	= 014446	292#	295	297	299	301	303	307	309	311	313	315	317	319
		321	323	325	327	329	331	333	335	337	339	341	343	345
		347	349	351	353	355	357	359	361	363	365	367	369	371
		373	375	377	379	381	383	385	387	389	391	393	395	397
		399	401	403	405	407	409	411	413	415	417	419	422#	424#
		428#	432#	439#	444#	446#	448#	450#	465#	467#	736	976	987	1556
		1560	1562	1603	1621	1623	1633	1645	1650#	1652#	1654#	1656#	1658#	1660#
		1662#	1664#	1666#	1668#	1670#	1672#	1674#	1676#	1678#	1680#	1682#	1700#	1855#



GET	211#	597	777	800	925										
HEADER	211#	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	1508	1520
XMTDLY	211#	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	1508	1520

. ABS. 014446 000

ERRORS DETECTED: 0

CZDMBD.BIN,CZDMBD.LST/CRF/SOL/NL:TOC=CZDMBD.P11  
RUN-TIME: 6 12 2 SECONDS  
RUN-TIME RATIO: 121/21=5.5  
CORE USED: 8K (15 PAGES)