

# DL11-E

ONLINE TEST

MD-11-DZDLB-B

EP-DZDLB-B-DL-A

OCT 1976

COPYRIGHT ©1976

**digital**

FICHE 1 OF 1

Made in U.S.A.

This microfiche card contains a grid of frames. The first column on the left contains frames with graphical data, possibly waveforms or plots. The remaining columns contain frames with text-based data, likely test results or system logs. The text is too small to be legible in this image.



.NEXT 3

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDLB-B-C

PRODUCT NAME: DL11-E ON LINE TESTS

DATE: 21-DECEMBER-1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: ROBERT WHITTON

REVISED BY: AL LOSCHAK JUN-75  
M. McNALLY DEC-75 : SUPPORT SOFTWARE SWITCH REGISTE

THIS MAINDEC OBSOLETE MAINDEC-11-DZDLBA

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION  
1972, 1975

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION  
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.  
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY  
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT  
SUPPLIED BY IT.  
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY  
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

11-11-75 15:21 17-SEP-75 15:21 PAGE 1

## 2. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR THE DL11-E (ASYNCHRONOUS LINE INTERFACE), MAINDEC-11-DZDLA (DL11-E OFF LINE TESTS) AND MAINDEC-11-DZDLB (DL11-E ON LINE TESTS). THE OFF LINE TESTS TEST ALL DL11-E LOGIC AND MAY BE USED TO INDIVIDUALLY TEST UP TO 31 DL11-E'S. THE OFF LINE TESTS DO NOT REQUIRE THE USE OF A MODEM, HOWEVER A SPECIAL JUMPER CONNECTOR IS REQUIRED. THE ON LINE TESTS ARE ESSENTIALLY DATA RELIABILITY TESTS REQUIRING THE USE OF MODEMS AND A SUITABLE TERMINAL DEVICE.

THREE STARTING ADDRESSES ARE PROVIDED. THEY ARE:

200 - NORMAL START  
 210 - REMAP DEVICES PRESENT AND RESTART  
 220 - MODIFY DEVICE ADDRESSES IF NON STANDARD INSTRUCTIONS TO DO THIS ARE TYPED OUT.

THIS DOCUMENT DESCRIBES THE ON LINE TESTS.

THE AVAILABLE TESTS ARE:

PRG0 SINGLE CHARACTER LINE MODE DATA TEST  
 PRG1 BINARY COUNT LINE MODE DATA TEST  
 PRG2 MESSAGE TRANSMIT ONLY W/W 0 PARITY  
 PRG3 RECEIVE DATA TEST  
 PRG4 MESSAGE TRANSMIT (SPIRAL) ONLY W/W 0 PARITY.

## 3. REQUIREMENTS

## 3.1 EQUIPMENT

- A. POP 11 SYSTEM
- B. DL11-E(S)
- C. SUITABLE TERMINAL DEVICE (ASR 33, 37, DATA POINT, ETC)
- D. MODEM TYPE 103 OR 202 OR EQUIVALENT

## 3.2 STORAGE

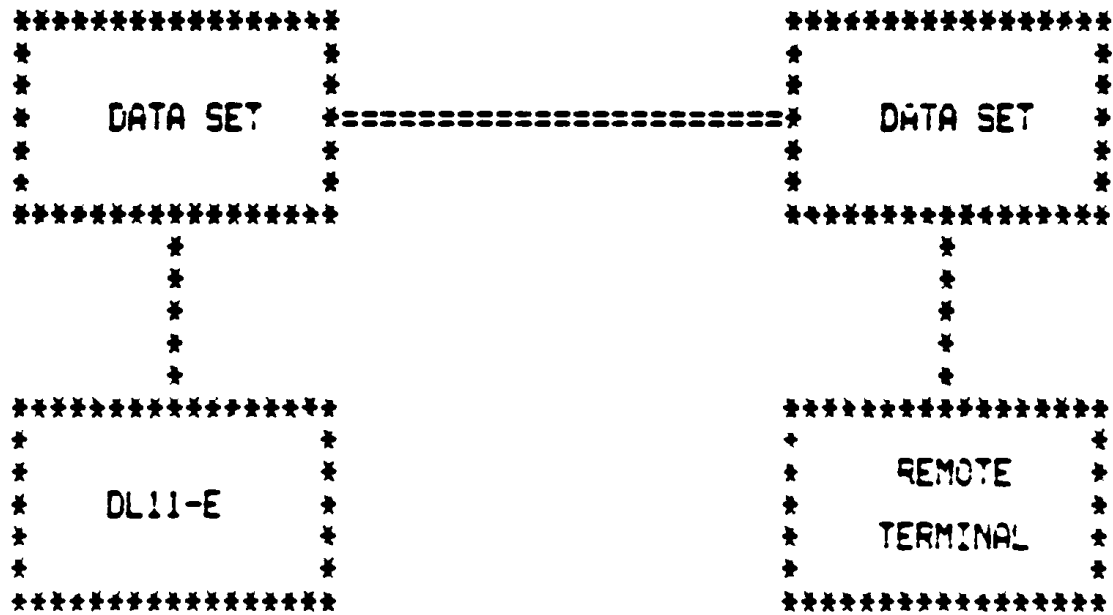
THIS PROGRAM USES 8K OF MEMORY



3.4 GENERAL

THERE ARE THREE CONFIGURATIONS USING DL11-E/MODEM PAIRS WHICH MAY BE SELECTED BY PRG0 AND PRG1. THESE CONFIGURATIONS ARE SELECTED BY THE USER WHEN REQUESTED BY THE PROGRAM DURING THE LINE CONNECTION ROUTINE. THE FOLLOWING PROCEDURES SHOULD BE FOLLOWED TO SELECT ANY OF THE THREE SELECTABLE CONFIGURATIONS:

CONFIGURATION 0: THIS CONFIGURATION TRANSMITS DATA FROM THE DL11-E CONNECTED TO THE LINE THAT WAS CALLED TO THE MODEM THAT CALLED (THE CALLER). THIS CONFIGURATION MAY BE USED TO TRANSMIT DATA TO A TERMINAL DEVICE. NOTE NO DATA CHECKING IS PERFORMED BY THE PROGRAM, HOWEVER DATA MAY BE VISUALLY CHECKED AT THE TERMINAL DEVICE. TO INITIATE PROGRAM ACTION CALL THE MODEM CONNECTED TO A DL11-E FROM A MODEM CONNECTED TO THE TERMINAL DEVICE. WHEN THE PHONE RINGS AT THE PDPI! THE PROGRAM WILL REQUEST THE CONFIGURATION. SET BIT0-1=00 BY TYPING IN OCTAL FOLLOWED BY A CR. WHEN THE 'HANDSHAKING' IS COMPLETED THE PROGRAM WILL REQUEST DL11-E PARAMETERS. TYPE THE PARAMETERS AS REQUESTED. THE PROGRAM WILL TYPE 'LINE CONNECTION MADE' AND BEGIN DATA TRANSMISSION.



CONFIGURATION 0

CONFIGURATION 1: THIS CONFIGURATION TRANSMITS DATA FROM THE DL11-E CONNECTED TO THE LINE THAT WAS CALLED TO THE DL11-E CONNECTED TO THE LINE THAT CALLED (THE CALLER). TO INITIATE PROGRAM ACTION CALL THE DL11-E YOU WISH TO TRANSMIT ON FROM THE LINE CONNECTED TO THE DL11-E YOU WISH TO RECEIVE THE DATA ON. WHEN THE PHONE RINGS AT THE PDPI! THE PROGRAM WILL REQUEST THE CONFIGURATION AND MODEM TYPE. TYPE BIT0-1=01 & BIT2=0 IF A 103 (OR EQUIV.) AND BIT2=1 IF A 202 (OR EQUIV.). THE PROGRAM WILL REQUEST THE LINE NUMBER THAT YOU CALLED FROM. TYPE THIS IN OCTAL FOLLOWED BY A CARRIAGE RETURN. WHEN THE CARRIER

170  
171  
172  
173  
174  
175  
176



15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
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  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

CONFIGURATION 2: THIS CONFIGURATION TRANSMITS DATA FROM BOTH THE CALLED TRANSMITTER AND THE TRANSMITTER CONNECTED TO THE LINE THAT WAS CALLING, I.E. IN ADDITION TO THE DATA TRANSMITTED AS IN CONFIGURATION 1 DATA IS ALSO TRANSMITTED IN THE REVERSE DIRECTION. TO INITIATE PROGRAM ACTION CALL THE DL11-E YOU WISH TO TRANSMIT ON FROM THE DL11-E YOU WISH TO RECEIVE/TRANSMIT ON. WHEN THE PHONE RINGS AT THE PDP11 THE PROGRAM WILL REQUEST THE CONFIGURATION AND MODEM TYPE. TYPE BIT0=1 AND BIT2=0. NOTE: \*\*\*\*\*DO NOT USE MODEM TYPE 202 (OR EQUIV) USING CONFIG #2\*\*\*\* THE PROGRAM WILL REQUEST THE LINE YOU CALLED FROM. TYPE THE NUMBER IN OCTAL FOLLOWED BY A CR. WHEN THE CARRIER IS HEARD IN THE HEADSET PRESS THE DATA BUTTON ON THE DATA SET. NOTE YOU HAVE APPROXIMATELY 10 SECONDS IN WHICH TO DO THIS. WHEN THE 'HANDSHAKING IS COMPLETED THE PROGRAM WILL REQUEST TWO SETS OF DL11-E PARAMETERS. THE CHARACTER LENGTH OF BOTH SETS MUST BE THE SAME AND THE SPEED OF THE SECOND SET MUST BE GREATER THAN THE SPEED OF THE FIRST. WHEN THE PARAMETERS HAVE BEEN LOADED THE PROGRAM WILL TYPE 'LINE CONNECTION MADE' AND BEGIN TO WAY DATA TRANSMISSION. WHEN 100. CHARACTERS HAVE BEEN RECEIVED AND CHECKED THE BELL WILL RING AT THE TTY, AND ANOTHER BLOCK OF 100. CHARACTERS WILL BE PROCESSED. NOTE, DL11-E#X REFERS TO THE 'CALLED' DL11-E, AND DL11-E#Y REFERS TO THE 'CALLING' DL11-E.

3.5 LINE NUMBERS

LINE NUMBER REFERS TO THE ADDRESSES TO WHICH THE DL11-E RESPONDS.

LINE 00 77561X	LINE 10 77571X	LINE 20 77601X	LINE 30 77611X
LINE 01 77562X	LINE 11 77572X	LINE 21 77602X	LINE 31 77612X
LINE 02 77563X	LINE 12 77573X	LINE 22 77603X	LINE 32 77613X
LINE 03 77564X	LINE 13 77574X	LINE 23 77604X	LINE 33 77614X
LINE 04 77565X	LINE 14 77575X	LINE 24 77605X	LINE 34 77615X
LINE 05 77566X	LINE 15 77576X	LINE 25 77606X	LINE 35 77616X
LINE 06 77567X	LINE 16 77577X	LINE 26 77607X	LINE 36 77617X
LINE 07 77570X	LINE 17 77600X	LINE 27 77610X	





308  
309

'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK  
'0123456799' WILL BE TRANSMITTED. TO TERMINATE, HANG UP.

310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357

4.4 PRG3 - RECEIVE TRANSMIT MESSAGE TEST

- A. LOAD ADDRESS = 000200
- B. START - PROGRAM WILL REQUEST PROGRAM NUMBER
- C. THE PROGRAM WILL IDENTIFY ITSELF AND TYPE INSTRUCTIONS TO SELECT DESIRED OPTIONS.
- D. SET IN OPTIONS AND PRESS CONTINUE.
- E. WHEN 'MAKE LINE CONNECTION' IS TYPED CALL THE DL11-E YOU WISH TO TRANSMIT ON. WHEN THE 'HANDSHAKING' IS COMPLETED THE DL11-E WILL TRANSMIT A CRLF TO THE TERMINAL DEVICE. AT THIS TIME YOU MAY BEGIN TO SEND DATA FROM THE DEVICE TO THE DL11-E WHERE IT WILL BE ECHOED BACK TO THE TERMINAL. TYPE ANY CHARACTER TO SIGNAL START OF MESSAGE. THEN TYPE MESSAGE AND THE SAME CHARACTER TO SIGNAL END OF MESSAGE. CONTROL C WILL CAUSE THE BUFFERS CONTENTS TO BE TRANSMITTED WHEN TYPED.
- F. IF NO ECHO IS DESIRED (ON A CHARACTER BASIS FOR EXAMPLE WHEN USING A TERMINAL THAT PRODUCES ITS OWN LOCAL COPY) SET BIT7 OF SWITCH REGISTER.

4.5 PRG4 - SPECIAL MESSAGE XMIT ONLY

- A. LOAD ADDRESS = 000200
- B. OPTIONS
  - 1. BITS 0-2 = 4
  - 2. BITS 3-6 = LINE NUMBER (SEE SECT 3.5)
- C. DEPRESS START - THE PROGRAM WILL IDENTIFY ITSELF AND TYPE INSTRUCTIONS TO SELECT DESIRED DL11-E PARAMETERS (SEE SECT 3.2)
- D. SET IN PARAMETERS IF IT IS DESIRED TO TRANSMIT DATA WITH PARITY RAISE SR6. ALSO RAISE SR5 TO TRANSMIT ODD PARITY AND LOWER TO TRANSMIT EVEN PARITY.

BIT6	1/0	ENABLE/DISABLE PARITY
BIT5	1/0	TRANSMIT ODD/EVEN PARITY

- E. WHEN 'MAKE LINE CONNECTION' IS TYPED CALL THE DL11-E YOU WISH TO TRANSMIT ON FROM THE TERMINAL MODEM. WHEN THE 'HANDSHAKING' IS COMPLETED A SPIRAL PATTERN WILL BE TRANSMITTED. TO TERMINATE, HANG UP.

358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390

5. PROGRAM DESCRIPTIONS
- 5.1 PRG0 - SINGLE CHARACTER LINE MODE DATA TEST  
PRG0 TRANSMITS USER SPECIFIED DATA AND A CARRIAGE RETURN/LINE FEED EVERY 72ND CHARACTER.
- 5.2 PRG1 - BINARY COUNT PATTERN LINE MODE DATA TEST  
PRG1 TRANSMITS A BINARY COUNT PATTERN. THIS PROGRAM IS THE SAME AS PRG0 EXCEPT FOR THE DATA TRANSMITTED.
- 5.3 PRG2 - SPECIAL MESSAGE TRANSMIT ONLY  
PRG2 TRANSMITS THE MESSAGE  
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789.  
NO DATA ERROR CHECKING IS PERFORMED BY THE PROGRAM.
- 5.4 PRG3 - RECEIVE/TRANSMIT MESSAGE TEST  
PRG3 - RECEIVES DATA FROM A TERMINAL AND READS THE RECEIVED MESSAGE BACK, AND TYPES THE MESSAGE ON THE PDP-11 TTY WHEN THE MESSAGE IS TERMINATED. CHARACTERS MAY BE ECHOED BACK (IF REQUIRED) ON A CHARACTER BASIS THEREBY CREATING LOCAL COPY AS THE MESSAGE IS TYPED.  
TRANSMISSION MAY BEGIN AT THE TERMINAL WHEN A CR/LF IS RECEIVED AT THE TERMINAL. THIS PROGRAM IS RESTRICTED TO USE BY ONLY FULL DUPLEX MODEMS.
- 5.5 PRG4 - SPECIAL MESSAGE TRANSMIT ONLY  
PRG4 TRANSMITS A SPIRAL PATTERN.  
NO DATA CHECKING IS PERFORMED BY THE PROGRAM.



391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439

6.0

ERRORS

THERE ARE TWO TYPES OF ERRORS WHICH ARE DETECTED BY THESE TESTS LINE FAILURE AND DATA ERRORS.

LINE FAILURES ARE DETECTED AND REPORTED BY ALL TESTS, AND DATA ERRORS ARE DETECTED ONLY IN PRG 0 & 1 WHEN USING CONFIGURATIONS 1 OR 2. DATA ERRORS IN THE OTHER TESTS MAY BE DETECTED BY VISUAL INSPECTION OF THE DATA AT THE TERMINAL.

LINE FAILURES ARE REPORTED BY TYPING THE PC, THE RECEIVER CONTROL STATUS REGISTER ADDRESS, AND ITS CONTENTS. SEE THE PROGRAM LISTING FOR A DETAILED DESCRIPTION OF THE ERROR.

THE MOST FREQUENTLY ENCOUNTERED ERROR WILL PROBABLY BE THE LOSS OF CARRIER. THIS ERROR WILL BE REPORTED IF AFTER A LINE CONNECTION IS MADE THE CARRIER IS LOST, EITHER BY 'HANGING UP' OR A 'GLITCH' ON THE LINE CAUSING THE CARRIER TO MOMENTARILY DROP. IN EITHER INSTANCE THE PROGRAM DISCONNECTS THE DL11-E FROM THE MODEM (BY CLEARING DATA TERMINAL READY) AND THE LINE WILL HAVE TO BE RECONNECTED TO RESUME TESTING.

IF IT IS PHYSICALLY IMPOSSIBLE TO GET TO THE DATA BUTTON WITHIN THE TIME ALLOTTED (APPROX. 10 SECONDS) TO MAKE THE LINE CONNECTION, THIS TIME MAY BE INCREASED BY PUTTING A LARGER NUMBER INTO THE DELAY. PATCH THE LARGER NUMBER INTO THE ADDRESS FOLLOWING THE DELAY EMT (BETWEEN RINTBG AND RINTBH). FOR EXAMPLE PATCHING IN 72460 WILL ALLOW APPROXIMATELY 30 SECONDS IN WHICH TO RESPOND.

DATA ERRORS ARE REPORTED BY TYPING THE PC, THE RECEIVER CONTROL REGISTER ADDRESS OF THE LINE THAT FAILED, WHAT THE DATA SHOULD HAVE BEEN, WHAT THE DATA WAS, AND THE CHARACTER NUMBER.

PC=XXXXXX 174010 DATA S/B 301 WAS 321 CHAR NO 23

THIS TYPEOUT INDICATES A DATA ERROR ON LINE 1 IF CONFIGURATION 2 IS SELECTED TWO ERROR T/POUTS MAY OCCUR FOR A SINGLE ERROR DEPENDING ON WHERE THE ERROR OCCURED. CONFIGURATION 2 COMPARES THE DATA RECEIVED AT THE CALLED DL11-E WITH THE DATA TRANSMITTED BY THE CALLED DL11-E, AND ALSO THE DATA RECEIVED AT THE CALLER, DL11-E (CALLER) WITH THE DATA TRANSMITTED BY THE CALLED DL11-E.

IF FOR EXAMPLE A DATA ERROR OCCURRED AT THE RECEIVER OF THE CALLING DL11-E CAUSING IT TO TRANSMIT INCORRECT DATA TO THE CALLED DL11-E TWO TYPEOUTS WILL OCCUR AS SHOWN BELOW:

PC=XXXXXX 174010 DATA S/B 301 WAS 321 CHAR NO 23  
PC=XXXXXX 174000 DATA S/B 301 WAS 321 CHAR NO 23

THESE TYPEOUTS SHOW THAT THE RECEIVER ON LINE 0 WAS THE CAUSE OF THE ERROR AND THE RECEIVER ON LINE 1 RECEIVED THE CORRECT INCORRECT DATA.

.ENDR

MO1

441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466 000000 000000  
467 000002 000000  
468 000004 004520  
469 000006 000040  
470 000010 004520  
471 000012 000100  
472 000014 004520  
473 000016 000140  
474 000020 004626  
475 000022 000340  
476 000024 002646  
477 000026 000340  
478 000030 002222  
479 000032 000340  
480 000034 004520  
481 000036 000340  
482 000040 000042  
483 000042 000000  
484  
485  
486  
487  
488  
489  
490 177776  
491 001200  
492 000000  
493 100000  
494 100000  
495 040000  
496 020000

```
.TITLE DZDLBB
.ENABLE ABS,AMA
: THIS TEST CHECKS THE DL11-E USING MODEMS
: REFER ALSO TO TEST DZDLA (DL11-E OFF LINE TESTS)
: STARTING PROCEEDURE
: LOAD ADDRESS 200

: STACK POINTER IS AT 1200
: PRESS START

: AVAILABLE PROGRAMS
: PRG0- SINGLE CHARACTER LINE MODE DATA TEST.
: PRG1- SPECIAL BINARY COUNT LINE MODE DATA TEST.
: PRG2- SPECIAL MESSAGE XMIT ONLY W/W/O PARITY
: PRG3- RECEIVE DATA TEST
: PRG4- SPIRAL PATTERN MESSAGE XMIT ONLY W/W/O PARITY
: PRG5- DATA ECHO TEST (USES FACILITY AT MAYNARD)

: STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
: SR15- HALT ON ERROR.
: SR14- SCOPE (NOT USED)
: SR13- INHIBIT PRINTOUT
: SR12- INHIBIT TRACE (NOT USED)
: SR11- INHIBIT ITERATION (NOT USED)

.=0
ERTP :UNASSIGNED TRAP
0
MACHER: ERTP :SP OVERFLOW, BUS ERROR TRAP
40
ERTP :RESERVED INSTRUCTION TRAP
100
ERTP :TRACE TRAP
140
MAPVEC :TRAP TO MAP VECTOR
PRTY7
PFAIL :POWER FAIL TRAP
PRTY7
EMTINT :EMT TRAP
PRTY7
ERTP
340
.+2
HALT
.REPT 119.
.+2
4 :TRAP TO MAP MAKER
.ENDR

:EQUATE STATEMENTS
PSW=177776
STKPTR=1200
OPEN=0
MANUAL=BIT15
BIT15=100000
BIT14=40000
BIT13=20000
```









665	001402	000000		CALLER: OPEN	
666	001404	000000		CALLED: OPEN	
667	001406	000000		OTBUFP: OPEN	: CONTAINS ADDRESS FROM WHERE NEXT TRANSMITTED CHAR. (IN OUTBUF) IS TO COME
668	001410	000000		TBUFP: OPEN	
669	001412	000000		MODEM: OPEN	: CONTAINS MODEM TYPE 0=103, 4=202
670	001414	000000		OPEN	: CONTAINS ADDRESS FROM WHERE NEXT TRANSMITTED CHAR. (CALLER'S LINE) IS TO COME
671	001416	005037	001344	REMAP: CLR	FTITLE
672	001422	012706	001200	START: MOV	#STKPTR,%6 ;SET BOTTOM OF SP STACK.
673	001426	000005		RESET	
674	001430	005037	177776	CLR	PSW
675	001434	012737	004520	MOV	#ERTP,MACHER
676	001438	012737	000040	MOV	#40,MACHER+2
677	001442	005737	001344	TST	FTITLE ;TITLE PRINTED
678	001450	001145		BNE	START1 ;YES, SKIP THIS
679	001454	104000		TYPE	
680	001456	012012		MTITLE	
681	001460	005237	001344	INC	FTITLE ;SET TITLE PRINTED FLAG
682	001462	005037	001346	CLR	FNONE ;CLEAR NO DEVICE FLAG
683	001466	012737	177777	MOV	#-1,LINENO
684	001470	012737	001550	MOV	#MAPNE,MACHER ;SET UP FOR NO DEVICE ANSWER
685	001500	012737	000340	MOV	#PTY7,MACHER+2
686	001506	012737	000006	MOV	#RCSR,%4 ;SET UP DEVICE POINTER
687	001514	012704	010432	MOV	#RCSR,%4
688	001520	005237	001354	MAPA: INC	LINENO
689	001524	020427	010530	CMP	%4,#RBUF ;LAST DEVICE
690	001530	001477		BEQ	MAPEND ;YES
691	001532	005037	177776	CLR	PSW
692	001536	005774	000000	TST	0(4) ;TEST DEVICE
693	001542	000240		NOP	
694	001546	000240		NOP	
695	001548	000404		BR	MAPOK
696	001550	062704	000002	MAPNE: ADD	#2,%4
697	001554	022626		POPSP2	
698	001556	000760		BR	MAPA
699	001560	011437	001352	MAPOK: MOV	(4),TEMP1 ;SAME DEVICE FOR TYPING
700	001564	004537	003530	JSR	%5,0ACNV
701	001570	001352		TEMP1	
702	001572	012154		MADR	
703	001574	000006		6	
704	001576	004537	003530	JSR	%5,0ACNV
705	001602	001354		LINENO	
706	001604	012146		MLINE	
707	001606	000002		2	
708	001610	011401		MOV	(4),%1 ;GET RXCSR DEVICE ADDRESS
709	001612	004737	004704	JSR	%7,FORMAD
710	001616	052737	000001	BIS	#BIT0,FMAP
711	001624	042777	000100	BIC	#BIT6,RTXCSR
712	001632	052777	000100	BIS	#BIT6,RTXCSR
713	001640	000240		NOP	
714	001642	012737	000340	MOV	#PTY7,PSW
715	001650	005737	001210	TST	RXVTR
716	001654	001406		BEQ	MAPOKA
717	001656	013701	001354	MOV	LINENO,%1
718	001662	006301		ASL	%1



721	001664	013761	001210	010334		MOV	RXVTR,VECTAB(1)	;STORE VECTOR
722	001672	042777	000100	177304	MAPOKA:	BIC	#BIT6,DTXCSR	
723	001700	004537	003530			JSR	%5,0ACNV	
724	001704	001210				RXVTR		
725	001706	012166				MTRAP		
726	001710	000004				4		
727	001712	104000				TYPE		
728	001714	012146				MLINE		
729	001716	005937	001346			INC	FNONE	
730	001722	062704	000002			ADD	#2,%4	
731	001726	000674				BR	MAPA	
732	001730	012737	004520	000004	MAPEND:	MOV	#ERTP,MACHER	
733	001736	012737	000040	000006		MOV	#40,MACHER+2	
734	001744	005737	001346			TST	FNONE	
735	001750	001007				BNE	START1	
736	001752	104000			MAPERR:	TYPE		
737	001754	012176				MNONE		
738	001756	005037	001344			CLR	FTITLE	
739	001762	000000				HALT		
740	001764	000137	001422			JMP	START	
741	001770	005037	177776		START1:	CLR	PSW	
742	001774	104000				TYPE		
743	001776	012213				MSWSEL		
744	002000	004737	003314			JSR	PC,RDOCT	;GET INPUT
745	002004	012600				MOV	(SP)+,%0	;((SR) TO RD
746	002006	042700	177770			BIC	#177770,%0	;LIMIT (SR) TO BITS 2-0
747	002012	010037	001240			MOV	%0,PRGNUM	;SAVE PROGRAM #
748	002016	006300				ASL	%0	;ROX2
749	002020	000170	001244			JMP	@PRGTAB(0)	;GO TO SELECTED PROGRAM.
750								
751	002024	012706	001200		MODEV:	MOV	#STKPTR,%6	
752	002030	000005				RESET		
753	002032	005037	177776			CLR	PSW	
754	002036	104000			MODEV1:	TYPE		
755	002040	014435				MMOD1		
756	002042	004737	003314			JSR	PC,RDOCT	;GET INPUT
757	002046	011600				MOV	(SP),%0	
758	002050	042700	177740			BIC	#177740,%0	
759	002054	006300				ASL	%0	
760	002056	022627	000036			CMP	(SP)+,%35	
761	002062	101403				BLOS	MODEV2	;BRANCH IF > 36
762	002064	104000				TYPE		
763	002066	014363				MMODX		
764	002070	000762			MCDEV2:	BR	MODEV1	
765	002072	104000				TYPE		
766	002074	014553				MMOD2		
767	002076	004737	003314		JSR	PC,RDOCT		;GET INPUT
768	002102	032716	000001			BIT	#BIT0,(SP)	
769	002106	001403				BEQ	MODEV3	
770	002110	104000				TYPE		
771	002112	014435				MMOD0		
772	002114	000762			MODEV3:	BR	MODEV2	
773	002116	012601				MOV	(SP)+,%1	;SAVE DEV ADR
774	002120	019003				MOV	%0,%3	
775	002122	062703	010432			ADD	#ROSF,%3	
776	002126	010113				MOV	%1,%3	

```

777 002130 062701 000002      ADD      #2,%1      ;UPDATE DEV ADR
778 002134 010003      MOV      %0,%3
779 002136 062703 010530      ADD      #RBUF,%3
780 002142 010113      MOV      %1,(3)
781 002144 062701 000002      ADD      #2,%1      ;UPDATE DEV ADR
782 002150 010003      MOV      %0,%3
783 002152 062703 010626      ADD      #TCSR,%3
784 002156 010113      MOV      %1,(3)
785 002160 062701 000002      ADD      #2,%1      ;UPDATE DEV ADR
786 002164 010003      MOV      %0,%3
787 002166 062703 010724      ADD      #TBUF,%3
788 002172 010113      MOV      %1,(3)
789 002174 104000      TYPE
790 002176 014621      MMOD3
791 002200 004737 003314      JSR      PC,RDOCT   ;GET INPUT
792 002204 022627 177777      CMP      (SP)+,#177777
793 002210 001712      BEQ
794 002212 005037 001344      CLR
795 002216 000137 001422      JMP      START
796
797      ;EMT TRAP INTERPRETER
798 002222 011646      EMTINT: MOV      @%6,-(6)   ;GET SAVED PC.
799 002224 162716 000002      SUB      #2,@%6       ;DECREMENT PC BY 2.
800 002230 017616 000000      MOV
801 002234 006116      EMTA:  ROL      @%6       ;EMT ARG X 2.
802 002236 042716 177001      BIC      #177001,@%6   ;REMOVE 7 MSB.
803 002242 062716 001264      ADD      #EMTTAB,@%6   ;FORM EMT RTN ADDR.
804 002246 017616 000000      MOV
805 002252 000136      JMP      @%6+         ;GO TO EMT ROUTINE.
806
807      ;SAVE REGS 0 TO 4 SUBROUTINE.
808 002254 012637 002310      SAVRG: MOV      (6)+,SVRPC   ;SAVE PC AND PSW.
809 002260 012637 002312      MOV      (6)+,SVRPSW
810 002264 010446      MOV      %4,-(6)       ;SAVE REGS 0 - 4
811 002266 010346      MOV      %3,-(6)       ;IN STACK.
812 002270 010246      MOV      %2,-(6)
813 002272 010146      MOV      %1,-(6)
814 002274 010046      MOV      %0,-(6)
815 002276 013746 002312      MOV      SVRPSW,-(6)   ;RESTORE PC AND PSW.
816 002302 013746 002310      MOV      SVRPC,-(6)
817 002306 000002      RTI
818 002310 000000      SVRPC: OPEN
819 002312 000000      SVRPSW: OPEN
820
821      ;RESTORE REGS 0 TO 4 SUBROUTINE.
822 002314 012637 002350      RSTRG: MOV      (6)+,RSTPC   ;SAVE PC AND PSW.
823 002320 012637 002352      MOV      (6)+,RSTPSW
824 002324 012600      MOV      (6)+,%0       ;RESTORE REGS 0 - 4
825 002326 012601      MOV      (6)+,%1       ;FROM STACK.
826 002330 012602      MOV      (6)+,%2
827 002332 012603      MOV      (6)+,%3
828 002334 012604      MOV      (6)+,%4
829 002336 013746 002352      MOV      RSTPSW,-(6)   ;RESTORE PC AND PSW.
830 002342 013746 002350      MOV      RSTPC,-(6)
831 002346 000002      RTI
832 002350 000000      RSTPC: OPEN

```

833	002352	000000			RSTPSW: OPEN	
834	002354	104000			INCPRG: TYPE	:TYPE INCORRECT PROGRAM SELECTED.
835	002356	013352			AINPRG	
836	002360	000000			HALT	
837	002362	000137	001422		JMP START	
838					:COMMON HALT ROUTINE	
839	002366	011600			CHLT: MOV @%6,%0	:DEVELOP ADDRESS OF CALLER.
840	002370	162700	000002		SUB #2,%0	
841	002374	000000			HALT	:HALT ADDRESS OF CALL INSTRUCTION
842	002376	000002			RTI	:IN DATA LIGHTS.
843						
844					:CONDITIONAL ERROR HALT ROUTINE.	
845	002400	005777	175570		EHLT: TST @SRPTR	:CHECK FOR HALT ON ERROR.
846	002404	000001			BPL EHLTA	:BRANCH IF NO HALT DESIRED.
847	002406	000000			HALT	:HALT
848	002410	000002			EHLTA: RTI	:IN DATA LIGHTS.
849						
850					:DATA CHECK ROUTINE.	
851	002412	043737	001362	001360	DTCHK: BIC CARMSK,XMTDAT	:CLEAR UNTRANSMITTED BITS
852	002420	123737	001356	001360	CMPB RECDAT,XMTDAT	:COMPARE TRANSMITTED AND RECEIVED
853	002426	001430			BEQ DTCHKA	:CHARS. BRANCH IF SAME.
854	002430				CNVOA RECDAT,CWAS,3	
855	002442				CNVOA XMTDAT,CSB,3	
856	002454				CNVOA RXCSR,CSRADD,6	
857	002466	104013			ERRC:	
858	002470	012357			CSRAUD	
859	002472	004537	004216		JSR 5,BDCNV	:CONVERT
860	002476	001364			CTRD	:CHAR #
861	002500	013343			CRNUM	:TO DECIMAL
862	002502	000004			4	:4 BITS
863	002504	104013			ERROR1	
864	002506	013300			CERDAT	
865	002510	000002			DTCHKA: RTI	:EXIT.
866						
867	002512	012737	177777	002624	ERR: MOV #-1,ERRB	:SET UP ONE MESSAGE CALL.
868	002520	012737	000240	002626	MOV #240,ERRB+2	
869	002526	005037	002644		CLR ERRE	
870	002532	000413			BR ERRA	
871	002534	011637	002624		ERR1: MOV @%6,ERRB	:DEVELOP ADDT'L MESSAGE ADDR.
872	002540	017737	000060	002624	MOV @ERRB,ERRB	:STORE AT ERRB.
873	002546	012737	177777	002626	MOV #-1,ERRB+2	
874	002554	012737	000002	002644	MOV #2,ERRE	
875	002562	032777	020000	175404	ERRA: BIT #BIT13,@SRPTR	:INHIBIT ERROR PRINT?
876	002570	001020			BNE ERRC	:BRANCH TO INHIBIT PRINT.
877	002572	011637	002642		MOV @%6,ERRD	:DEVELOP CALLING ADDR.
878	002576	162737	000002	002642	SUB #2,ERRD	
879	002604				CNVOA ERRD,APC,6	:CONVERT CALL ADDR TO ASCII.
880	002616	104011			SAVREG	
881	002620	104001			TYPES	:TYPE:
882	002622	012427			EMD	:ERROR HEADER,
883	002624	000000			ERRB: OPEN	:ADDT'L ERROR MESSAGE IF ANY.
884	002626	177777			-1	
885	002630	104012			RSTREG	
886	002632	104010			ERRC: EHALT	:GO ERR HALT IF DESIRED.
887	002634	063716	002644		ADD ERRE,@%6	
888	002640	000002			RTI	:EXIT.



```

899 002642 000000      ERRD:  OPEN
899 002644 000000      ERRE:  OPEN
899                ;
899                ;POWER FAIL SERVICE
893 002646 012737 002656 000024 PFAIL:  MOV      #PWRUP,0#24
894 002654 000000      HALT
895 002656 012737 002646 000024 PWRUP:  MOV      #PFAIL,0#24
896 002654 000005      RESET
897 002666 012706 001200      MOV      #STKPTR,%6
898 002572 104000      TYPE
899 002674 014325      MPWRF
900 002676 013700 001240      RESTRT: MOV      PRGNUM,%0
901 002702 006300      ASL      %0
902 002704 000170 001244      JMP      @PRGTAB(0)
903
904 002710                TXERR:  CNVOA   TXCSR,ATXWAS,6 ;CONVERT CONTENTS OF TXCSR TO ASCII.
905 002722 012737 012445 003026      MOV      #ATXCSR,CRXTXB
906 002730 000410      BR       CRXTX
907 002732                RXERR:  CNVOA   RXCSR,ARXWAS,6 ;CONVERT CONTENTS OF RXCSR TO ASCII.
908 002744 012737 012466 003026      MOV      #ARXCSR,CRXTXB
909 002752 011637 003024      CRXTX:  MOV      @%6,CRXTXA ;DEVELOP ADDR OF ADDTT'L ERROR MESSAGE.
910 002756 017737 000042 003024      MOV      @CRXTXA,CRXTXA
911 002764 032777 020000 175202      BIT      #BIT13,@SRPTR ;INHIBIT PRINT?
912 002772 001017      BNE     CRXTXC ;BRANCH TO INHIBIT PRINT.
913 002774 011637 002642      MOV      @%6,ERRD ;DEVELOP CALLING ADDR.
914 003000 162737 000002 002642      SUB      #2,ERRD
915 003006      CNVOA   ERRD,APC,6 ;CONVERT CALLING ADDR TO ASCII.
916 003020 104001      TYPES ;TYPE ERROR MESSAGE.
917 003022 012427      EMO     ;ERR HEADER
918 003024 000000      CRXTXA: OPEN ;ADDT'L ERR MESSAGE
919 003026 000000      CRXTXB: OPEN ;TXCSR OR RXCSR CONTENTS.
920 003030 177777      -1
921 003032 104010      CRXTXC: EHALT ;GO HALT IF DESIRED.
922 003034 062716 000002      ADD      #2,@%6
923 003040 000002      RTI     ;EXIT.
924
925                ;ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
926 003042 017637 000000 003062 STRVRV: MOV      @6,STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2
927 003050 062716 000002      ADD      #2,@%6 ;SET UP EXIT
928 003054 013701 001210      MOV      RXVTR,%1
929 003060 012721 000000      STPRA:  MOV      #OPEN,(1)+ ;SET VECTOR ADDRESS
930 003064 013721 001212      MOV      RXLVL,(1)+ ;SET PRIORITY
931 003070 000002      RTI     ;EXIT
932
933                ;ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
934 003072 017637 000000 003112 STXMTV: MOV      @6,STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
935 003100 062716 000002      ADD      #2,@%6 ;SET UP EXIT
936 003104 013701 001214      MOV      TXVTR,%1
937 003110 012721 000000      STPPA:  MOV      #OPEN,(1)+ ;SET VECTOR ADDRESS.
938 003114 013721 001216      MOV      TXLVL,(1)+ ;SET PRIORITY
939 003120 000002      RTI     ;EXIT.
940
941                ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
942 003122 010037 003312      TYP:   MOV      %0,SAVR0 ;SAVE R0
943 003126 011600      MOV      @%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
944 003130 062716 000002      ADD      #2,@%6 ;SET UP EXIT.
  
```

945	003134	011000				MOV	%0,%0	; ADDRESS OF MESSAGE TO RO.
946	003136	112037	003242			MOVE	(0)+,TYPDAT	; GET CHARACTER
947	003142	122737	000100	003242	TYPA:	CMPB	#10C,TYPDAT	; CHECK FOR "a" CHARACTER
948	003150	001003				BNE	TYPC	; BRANCH IF NOT "a"
949	003152	013700	003312			MOV	SAVRO,%0	; RESTORE RO
950	003156	000002				RTI		; TERMINATOR CHAR. DONE. EXIT.
951	003160	122737	000045	003242	TYPC:	CMPB	#45,TYPDAT	; CHECK FOR "%"
952	003166	001412				BEQ	TYPF	; BRANCH IF "%"
953	003170	004737	003176			JSR	%7,TYPD	; TYPE CHAR IN TYPDAT
954	003174	000760				BR	TYPA	
955	003176	113777	003242	176022	TYPD:	MOVSB	TYPDAT,%TPB	; OUTPUT CHARACTER TO PRINTER
956	003204	105777	176014			TSTB	@TPS	; WAIT FOR DONE FLAG.
957	003210	100375				BPL	-4	
958	003212	000207				RTS	%7	; EXIT
959	003214	112737	000015	003242	TYPF:	MOVSB	#15,TYPDAT	; MOVE CARRIAGE RETURN CODE TO TYPDAT
960	003222	004737	003176			JSR	%7,TYPD	; GO TYPE CHAR.
961	003226	112737	000012	003242		MOVSB	#12,TYPDAT	; MOVE LF CODE TO TYPDAT.
962	003234	004737	003176			JSR	%7,TYPD	; GO TYPE CHAR.
963	003240	000736				BR	TYPA	
964	003242	000000				TYPDAT: OPEN		
965						; SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER		
966	003244	010037	003312			TYPB:	MOV	%0,SAVRO
967	003250	011600				TYPBAA:	MOV	%6,%0
968	003252	062716	000002			ADD	#2,%6	; GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
969	003256	011037	003306			MOV	%0,TYPSB	; UPDATE TO NEXT MESSAGE ADDRESS
970	003262	022737	177777	003306		CMP	#-1,TYPSB	; ADDRESS OF MESSAGE TO TYPSB
971	003270	001003				BNE	TYPBA	; CHECK FOR TERMINATOR
972	003272	013700	003312			MOV	SAVRO,%0	; BRANCH IF NOT TERMINATOR.
973	003276	000002				RTI		; RESTORE RO
974	003300	013700	003312		TYPBA:	MOV	SAVRO,%0	; TERMINATOR, EXIT
975	003304	104000				TYPE		; CALL ON TYP SUB TO TYPE MESSAGE
976	003306	000000			TYPBB:	OPEN		; ADDRESS OF MESSAGE GOES HERE
977	003310	000757				BR	TYPBAA	; GO PROCESS NEXT MESSAGE
978	003312	000000			SAVRO:	OPEN		
979						; SUBROUTINE TO READ OCTAL DATA FROM THE TELETYPE PRINTER		
980					RDOCT:	MOV	(SP),-(SP)	; MAKE ROOM FOR DATA WORD
981	003314	011646				MOV	%0,-(SP)	; SAVE RO
982	003316	010046				MOV	%1,-(SP)	; SAVE RI
983	003320	010146			INDAT:	CLR	%1	; CLEAR DATA WORD
984	003322	005001				CLR	COUNT	; SET NO. OF DIGITS = 0
985	003324	005037	001324		RDDAT:	TSTB	@TKS	; TEST TTY READ STATUS
986	003330	105777	175664			BPL	RDDAT	; WAIT
987	003334	100375				MOVSB	@TKB,-(SP)	; PUSH DIGIT ON STACK
988	003336	117746	175660			BIC	#BIT7,(SP)	
989	003342	042716	000200		ECDAT:	TSTB	@TPS	; TEST TTY PRINT STATUS
990	003346	105777	175652			BPL	ECDAT	; WAIT
991	003352	100375				MOVSB	(SP),@TPB	; ECHO CHARACTER
992	003354	111677	175646			CMPB	#15,(SP)	; IS IT A TERMINATOR?
993	003360	122716	000015			BEQ	RETRN	; BR IF YES
994	003364	001432				CMPB	#177,(SP)	; IS IT A RUBOUT?
995	003366	122716	000177			BEQ	RREAD	; BR IF YES
996	003372	001423				CMPB	#60,(SP)	; IS IT AN OCTAL DIGIT?
997	003374	122716	000060			BGT	RREAD	; BR IF NO
998	003400	003020				CMPB	#67,(SP)	; TEST AGAIN
999	003402	122716	000067			BLT	RREAD	; BR IF NO
1000	003406	002415						

```

1001 003410 005237 001324          INC      COUNT          ;INC NO. OF DIGITS
1002 003414 022737 000067 001324      CMP      #67,COUNT      ;MORE THAN SIX DIGITS?
1003 003422 003407                    BLE      RREAD          ;BR IF YES
1004 003424 006301                    ASL      %1              ;CLEAR LOWEST THREE BITS
1005 003426 006301                    ASL      %1              ;OF DATA WORD
1006 003430 006301                    ASL      %1
1007 003432 162716 000060          SUB      #60,(SP)       ;CONVERT TO BINARY
1008 003436 062601                    ADD      (SP)+,%1       ;ADD DIGIT TO DATA WORD
1009 003440 000733                    BR       RRDAT          ;GET NEXT DIGIT
1010 003442 104000          RREAD:  TYPE          ;TELL USER ABOUT ILLEGAL CHARACTER
1011 003444 013232                    DTERR
1012 003446 005726                    TST      (SP)+          ;GET RID OF ILLEGAL CHARACTER
1013 003450 000724                    BR       INDAT          ;START SUBROUTINE AGAIN
1014 003452 010166 000010      RETRN:  MOV      %1,10(SP) ;STORE DATA WORD ON STACK
1015 003455 005726                    TST      (SP)+          ;INC STACK POINTER
1016 003460 012601                    MOV      (SP)+,%1       ;RESTORE R1
1017 003462 012600                    MOV      (SP)+,%0       ;RESTORE R0
1018 003464 000207                    RTS       PC             ;RETURN
1019
1020          ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1021 003466 011637 003526      DLY:    MOV      @%6,DLCNT ;GET DELAY COUNT ADDRESS.
1022 003472 062716 000002          ADD      #2,@%6         ;SET UP EXIT ADDRESS
1023 003476 017746 000024          MOV      @DLCNT,-(6)    ;DELAY COUNT TO STACK
1024 003502 001407                    BEQ      DLYC
1025 003504 012746 000226      DLYA:  MOV      #226,-(6) ;1 MSEC COUNT TO STACK
1026 003510 005316      DLYB:  DEC      @%6         ;DECREMENT 1 MSEC COUNT
1027 003512 001376                    BNE      DLYB           ;BRANCH IF NOT 0.
1028 003514 005726                    POPSP
1029 003516 005316                    DEC      @%6           ;ZERO. UNCOVER MSECS. COUNT.
1030 003520 001371                    BNE      DLYA           ;DECREMENT IT
1031 003522 005726      DLYC:  POPSP           ;BR IF NOT DONE DELAYING
1032 003524 000002                    RTI                    ;DONE
1033 003526 000000      DLCNT: OPEN          ;EXIT.
1034          ;CONTAINS MILLISECONDS COUNT ADDRESS.
1035
1036          ;OCTAL TO ASCII CONVERT ROUTINE
1037 003530 104011      OACNV: SAVREG          ;SAVE REGS.
1038 003532 013500                    MOV      @(%5)+,%0      ;GET OCTAL VALLE.
1039 003534 012501                    MOV      (%5)+,%1       ;GET DESTINATION ADDR.
1040 003536 012502                    MOV      (%5)+,%2       ;GET CONVERT COUNT.
1041 003540 060201                    ADD      %2,%1          ;DEVELOP ADDR TO STORE 1ST CHAR.
1042 003542 010003      OACNVA: MOV      %0,%3
1043 003544 042703 177770          BIC      #177770,%3     ;ISOLATE LEAST SIGNIFICANT DIGIT.
1044 003550 062703 000060          ADD      #60,%3         ;CONVERT DIGIT TO ASCII.
1045 003554 110341                    MOV      %3,-(1)       ;STORE ASCII CHARACTER.
1046 003556 042700 000507          BIC      #7,%0
1047 003562 006000                    ROR      %0
1048 003564 006000                    ROR      %0
1049 003566 006000                    ROR      %0
1050 003570 005302                    DEC      %2              ;DONE ALL DIGITS?
1051 003572 001363                    BNE      OACNVA        ;BRANCH IF NOT DONE.
1052 003574 104012                    RSTREG
1053 003576 000205                    RTS       %5            ;RESTORE REGS.
1054          ;DONE. EXIT.
1055          ;SUBROUTINE TO GENERATE PARITY ON DATA FOR 5,6,7,8 LEVEL CODE.
1056          ;PARITY BIT IS THE MSB OF THE CHARACTER PARITY CAN BE EITHER
          ;EVEN OR ODD
          ;GENERATES ODD/EVEN PARITY.

```

```

1057
1058 003600 032737 000200 001362 GENPAR: BIT #BIT7,CARMSK ;TEST LSB CHAR LENGTH
1059 003606 001411 BEQ EIGHT ;CHAR IS 8
1060 003610 032737 000100 001362 BIT #BIT6,CARMSK ;TEST MSB CHAR LENGTH
1061 003616 001427 BEQ SEVEN ;CHAR LENGTH IS 7
1062 003620 032737 000040 001362 BIT #BITS,CARMSK
1063 003626 001412 BEQ SIX
1064 003630 000433 BR FIVE
1065 003632 012737 000200 001322 EIGHT: MOV #BIT7,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1066 003640 012737 000007 001324 MOV #7,COUNT ;SET UP ROTATE COUNTER=7
1067 003646 042701 177600 BIC #177600,%1 ;MASK OFF UNUSED BITS
1068 003652 000433 BR DOIT ;GO AND GENERATE PARITY FOR 8
1069 003654 012737 000040 001322 SIX: MOV #BITS,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1070 003662 012737 000005 001324 MOV #5,COUNT ;SET UP ROTATE COUNTER=5
1071 003670 042701 177740 BIC #177740,%1 ;MASK OFF UNUSED BITS
1072 003674 000422 BR DOIT ;GO AND GENERATE PARITY FOR
1073 003676 012737 000100 001322 SEVEN: MOV #BIT6,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1074 003704 012737 000006 001324 MOV #6,COUNT ;SET UP ROTATE COUNTER=6
1075 003712 042701 177700 BIC #177700,%1 ;MASK OFF UNUSED BITS
1076 003716 000411 BR DOIT ;GO AND GENERATE PARITY FOR 7
1077 003720 012737 000020 001322 FIVE: MOV #BIT4,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1078 003726 012737 000004 001324 MOV #4,COUNT ;SET UP ROTATE COUNTER=4
1079 003734 042701 177760 BIC #177760,%1 ;MASK OFF UNUSED BITS
1080 003740 000400 BR DOIT ;GO AND GENERATE PARITY FOR
1081 003742 010137 001326 DOIT: MOV %1,SAVE ;SAVE DATA
1082 003746 006001 AGAIN: ROR %1 ;ROTATE DATA
1083 003750 103415 BCS ADD1 ;IF CARRY SET ADD IN PARBIT
1084 003752 005337 001324 RTN: DEC COUNT ;DECREMENT COUNTER
1085 003756 001373 BNE AGAIN ;NOT DONE DO IT AGAIN
1086 003760 032737 000040 001374 BIT #BITS,SRT ;DONE EVEN OR ODD PARITY?
1087 003766 001403 BEQ DONE ;IF EVEN FINISHED
1088 003770 063737 001322 001326 ADD PARBIT,SAVE ;IF ODD ADD IN ANOTHER 1
1089 003776 013701 001326 DONE: MOV SAVE,%1 ;PLACE DATA + PARITY BACK IN R1
1090 004002 000207 RTS 7 ;AND EXIT
1091 004004 063737 001322 001326 ADD1: ADD PARBIT,SAVE ;ADD PARBIT TO DATA
1092 004012 000757 BR RTN ;RETURN TO COUNTER
1093
1094 ;SUBROUTINE TO SELECT LINE AND LOAD VECTOR ASSIGNMENT
1095 004014 104000 LINSSEL: TYPE
1096 004016 014256 LLINE
1097 004020 004737 003314 JSR PC,ROOCT ;GET INPUT
1098 004024 012601 MOV (SP)+,%1 ;LOAD R1
1099 004026 042701 177407 BIC #177407,%1 ;MASK OFF ALL BUT LINE BITS
1100 004032 006201 ASR %1
1101 004034 006201 ASR %1
1102 004036 010137 001372 MOV %1,TEMP ;SAVE LINE #
1103 004042 012703 001200 MOV #RXCSR,%3 ;LOAD ADDRESS OF REGISTERS
1104 004046 012704 000004 MOV #4,%4 ;SET UP COUNTER
1105 004052 016102 010432 MOV RCSR(1),%2
1106 004056 010223 LINSR: MOV %2,(3)+
1107 004060 062702 000002 ADD #2,%2
1108 004064 005304 DEC %4
1109 004066 001373 BNE LINSR
1110 004070 016101 010334 MOV VECTAB(1),%1 ;GET LINE VECTOR ADDRESS
1111 004074 010123 MOV %1,(3)+ ;LOAD INTO PROG. RXVTR
1112 004076 022121 CMP (1)+,(1)+ ;ADD +4 TO RXVTR TO = TXVTR
    
```



1113	004100	005723		TST	(3)+		;POINT TO PROG TXVTR
1114	004102	010113		MOV	%1,(3)		;LOAD INTO PROG TXVTR
1115	004104	022737	000005 001240	CMP	#5,PRGNUM		;RUNNING PROGRAM # 5
1116	004112	001001		BNE	+4		
1117	004114	000205		RTS	5		;RETURN TO PROG 5
1118	004116	006237	001372	ASR	TEMP		;POSITION
1119	004122			CNVOA	TEMP,TLINEX,2		
1120	004134	1040J0		TYPE			;TYPE LINE # THAT
1121	004136	013665		ALINEX			;WAS CALLED
1122	004140	000205		RTS	5		
1123							
1124							;SUBROUTINE TO LOAD BINARY COUNT PATTERN INTO OUTPUT BUFFER
1125	004142	105037	001334	INFIL:	CLRB	NUMBER	;INITIALIZE BINARY COUNT
1126	004146	012500		FILL:	MOV	(5)+,%0	;GET ADDRESS
1127	004150	012537	001364		MOV	(5)+,CTRD	;GET COUNT
1128	004154	113720	001334	FILLA:	MOV	NUMBER,(0)+	;LOAD ADDRESS WITH BINARY COUNT
1129	004160	105237	001334		INCB	NUMBER	;INC. BINARY COUNT
1130	004164	005337	001364		DEC	CTRD	;DEC. COUNT
1131	004170	001871			BNE	FILLA	
1132	004172	000205			RTS	5	;EXIT
1133							
1134							;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1135	004174	104011		BMOVE:	SAVREG		;SAVE REGS.
1136	004176	012501			MOV	(5)+,%1	;GET"FROM"ADDRESS
1137	004200	012502			MOV	(5)+,%2	;GET"TO"ADDRESS
1138	004202	012503			MOV	(5)+,%3	;GET COUNT
1139	004204	112122		BMOVA:	MOV	(1)+,(2)+	;MOVE BYTE
1140	004206	005303			DEC	%3	;DECREMENT COUNT
1141	004210	001375			BNE	BMOVA	;BRANCH IF NOT DONE.
1142	004212	104012			RSTREG		;RESTORE REGS.
1143	004214	000205			RTS	%5	;DONE EXIT
1144							
1145							;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
1146	004216	104011		BDCNV:	SAVREG		;SAVE REGS.
1147	004220	012700	004374		MOV	#DECVAL,%0	;SET UP ADDR TO STORE DECIMAL ASCII IN R0
1148	004224	013501			MOV	2(5)+,%1	;BINARY VALUE TO R1.
1149	004226	012537	004304		MOV	(5)+,BDCNVC	;DESTINATION ADDR TO BDCNVC.
1150	004232	012537	004306		MOV	(5)+,BDCNVD	;COUNT TO BDCNVD.
1151	004236	012702	004362		MOV	#ADTEMP,%2	;ADDR OF TEN POWER STRING TO R2.
1152	004242	012737	000005 004354		MOV	#5,CNVCTR	;SET UP FOR 5 POWER CONVERSIONS.
1153	004250	012237	004360	BDCNVA:	MOV	(2)+,TENPWR	;MOVE POWER OF TEN VALUE TO TENPWR.
1154	004254	004737	004314		JSR	%7,SUBTEN	;PERFORM CONVERSION
1155	004260	005337	004354		DEC	CNVCTR	;DONE 5 CONVERSIONS?
1156	004264	001371			BNE	BDCNVA	;BRANCH IF NOT YET 5.
1157	004266	163700	004306		SUB	BDCNVD,%0	;SET UP ADDR TO MOVE DECIMAL
1158	004272	010037	004302		MOV	%0,BDCNVB	;DATA FROM.
1159	004276	004537	004174		JSR	%5,BMOVE	;MOVE DECIMAL DATA TO DESTINATION.
1160	004302	000000		BDCNVB:	OPEN		;SRC ADDR.
1161	004304	000000		BDCNVC:	OPEN		;DEST ADDR.
1162	004306	000000		BDCNVD:	OPEN		;COUNT.
1163	004310	104012			RSTREG		;RESTORE REGS.
1164	004312	000205			RTS	%5	;YES, EXIT.
1165	004314	005037	004356	SUBTEN:	CLR	DIGIT	;CLEAR DIGIT
1166	004320	163701	004360	SUBTNA:	SUB	TENPWR,%1	;SUBTRACT TEN POWER FROM BINARY VALUE.
1167	004324	103403			BCS	SUBTNB	;BRANCH IF UNSUCCESSFUL SUBTRACTION.
1168	004326	005237	004356		INC	DIGIT	

```

1159 004332 000772
1170 004334 063701 004360
1171 004340 062737 000060 004356
1172 004346 113720 004356
1173 004352 000207
1174 004354 000000
1175 004356 000000
1176 004360 000000
1177 004362 023420
1178 004364 001750
1179 004366 000144
1180 004370 000012
1181 004372 000001
1182 004374 040 040 040
1183 004377 040 040 040
1184
1185 004402 104000
1186 004404 013055
1187 004406 004737 003314
1188 004412 012637 001374
1189 004416
1190 004430 104000
1191 004432 013355
1192 004434 012737 177400 001362
1193 004442 032737 000002 001374
1194 004450 001413
1195 004452 012737 177700 001362
1196 004460 032737 000001 001374
1197 004466 001403
1198 004470 012737 177740 001362
1199 004476 000207
1200 004500 032737 000001 001374
1201 004506 001773
1202 004510 012737 177600 001362
1203 004516 000767
1204
1205
1206 004520 013737 177776 001336
1207 004526 012737 000340 177776
1208 004534 006237 001336
1209 004540 006237 001336
1210 004544 006237 001336
1211 004550 042737 177740 001336
1212 004556 013737 001336 001340
1213 004564 011637 001342
1214 004570 004537 003530
1215 004574 001340
1216 004576 012314
1217 004600 000006
1218 004602 004537 003530
1219 004606 001342
1220 004610 012346
1221 004612 000006
1222 004614 104000
1223 004616 012247
1224 004620 000000

SUBTNB: BR SUBTNA
ADD TENPWR,%1 ;RESTORE SUBTRACTED VALUE.
ADD #60,DIGIT ;CONVERT (DIGIT) TO ASCII
MOV8 DIGIT,(0)+ ;MOVE ASCII CHAR TO DECVAL FIELD.
RTS %7 ;EXIT.

CNVCTR: OPEN
DIGIT: OPEN
TENPWR: OPEN
ADTENP: 10000.
1000.
100.
10.
1

DECVAL: .BYTE 040,040,040,040,040,040

;SUBROUTINE TO SET CHARACTER LENGTH PARAMETER
SETPAR: TYPE ;TYPE: SELECT PARAMETERS.
SELPAR
JSR PC,RDOCT ;GET INPUT
MOV (SP)+,SRT
CNVOA SRT,APARM,3
TYPE
PARMTS

TBIT1: MOV #177400,CARMSK ;SET CHARACTER MASK TO 8 BITS.
BIT #BIT1,SRT ;SEE IF SR BIT 1 IS SET.
BEQ STPARA ;BRANCH IF NOT SET.
MOV #177700,CARMSK ;CHANGE CHAR MASK TO 6 BITS.
BIT #BIT0,SRT ;SEE IF SR BIT0 IS SET.
BEQ PAREX ;BRANCH IF NOT SET.
MOV #177740,CARMSK ;CHANGE CHAR MASK TO 5 BITS.
RTS %7 ;EXIT.

STPARA: BIT #BIT0,SRT ;SEE IF SR BIT0 IS SET.
BEQ STPARA-2 ;BRANCH IF NOT SET.
MOV #177600,CARMSK ;CHANGE CHAR MASK TO 7 BITS.
BR PAREX

;ERROR TRAP HANDLER - TYPE TO AND FROM WHERE ERROR TRAP OCCURS
ERTP: MOV PSW,OLDPS ;SAVE OLDPS
MOV #PRTY7,PSW
ASR OLDPS
ASR OLDPS
ASR OLDPS
BIC #177740,OLDPS
MOV OLDPS,TOPC
MOV @%6,FROMPC
ERTPA: JSR %5,OACNV
TOPC
MTO
6
JSR %5,OACNV
FROMPC
MFROM
6
TYPE
MERR
HALT

```

```

1225 004622 000137 001422          JMP      START
1226
1227          ;MAPVEC - MAP VECTOR VECTOR OR REPORT ERROR DEPENDING ON FMAP FLAG
1228          MAPVEC: MOV      @%6, TOPC
1229 004626 011637 001340          POPSP2
1230 004634 011637 001342          MOV      @%6, FROMPC
1231 004640 162737 000004 001340          SUB      #4, TOPC
1232 004646 005737 001350          TST      FMAP
1233 004652 001746          BEQ      ERTPA          ;NOT MAPPING, REPORT ERROR
1234 004654 013737 001340 001214          MOV      TOPC, TXVTR    ;STORE VECTOR
1235 004662 152737 000004 001340          SUB      #4, TOPC
1236 004670 013737 001340 001210          MOV      TOPC, RXVTR
1237 004676 005037 001350          CLR      FMAP
1238 004702 000002          RTI
1239
1240          ;FORMAD - FORM DEVICE AT ADDRESS
1241 004704 010137 001200          FORMAD: MOV      %1, RXCSR
1242 004710 062701 000002          ADD      #2, %1
1243 004714 010137 001202          MOV      %1, RXBUF
1244 004720 062701 000002          ADD      #2, %1
1245 004724 010137 001204          MOV      %1, TXCSR
1246 004730 062701 000002          ADD      #2, %1
1247 004734 010137 001206          MOV      %1, TXBUF
1248 004740 000207          RTS      %7
1249
1250          ;SUBROUTINE TO MAKE LINE CONNECTION.
1251 004742 017737 174232 001370          LINCON: MOV      @RXCSR, RXCSRT
1252 004750 032737 020000 001370          BIT      #BIT13, RXCSRT          ;YES. IS CLEAR TO SEND UP
1253 004756 001046          BNE      LINEUP          ;YES CONNECTION IS MADE.
1254 004760 042777 000146 174212          LINCA: BIC      #146, @RXCSR          ;CLEAR IE BIT AND DTR, RQ TO SND
1255 004766 005777 174210          TST      @RXBUF          ;CLEAR DONE FLAG
1256 004772 104000          TYPE          ;TYPE
1257 004774 013251          MAKCON          ;'MAKE LINE CONNECTION'
1258 004776 017737 174176 001370          LINCB: MOV      @RXCSR, RXCSRT
1259 005004 032737 040000 001370          BIT      #BIT14, RXCSRT          ;DID YOU RING
1260 005012 001771          BEQ      LINCB          ;GO WAIT FOR RING
1261 005014 052777 000006 174156          BIS      #6, @RXCSR          ;SET DTR, RQ TO SND
1262 005022 104016          DELAY          ;WAIT 10 SECONDS FOR
1263 005024 023420 10000.          ;CLEAR TO SEND
1264 005026 017737 174146 001370          MOV      @RXCSR, RXCSRT
1265 005034 005777 174142          TST      @RXBUF          ;CLEAR DONE
1266 005040 032737 020000 001370          BIT      #BIT13, RXCSRT          ;IS CLEAR TO SEND UP?
1267 005046 001003          BNE      LINCFC          ;YES. GO TO LINCFC
1268 005050 104015          ERRRX          ;NO. PRINT ERROR MESSAGE
1269 005052 013377          LINCHM          ;'CLEAR TO SEND NOT SET'
1270 005054 000741          BR          ;START OVER AGAIN
1271 005056 017737 174116 001370          LINCFC: MOV      @RXCSR, RXCSRT          ;CLEAR ALL FLAGS
1272 005064 005777 174112          TST      @RXBUF          ;AND DONE
1273 005070 104000          TYPE          ;TYPE MESSAGE
1274 005072 013427          LINMAD          ;CONNECTION IS MADE'
1275 005074 000205          LINEUP: RTS      5          ;EXIT LINE CONNECTION ROUTINE WITH-
1276          ;SUBROUTINE TO OVERLAY <CRLF> IN DATA PATTERN (EVERY 72.ND CHAR)
1277 005076 012701 014710          OVRLAY: MOV      #OUTBUF, %1          ;GET OUTBUF ADDRESS
1278 005102 012702 000016          MOV      #14, %2          ;GET COUNTER
1279 005106 012711 105215          OVRLYA: MOV      #105215, (1)          ;INSERT CR&LF
1280 005112 062701 000110          ADD      #72, %1          ;ADD OFFSET

```

```

DEC 2 :DONE
BNE 2 :EXIT
RTS 2 :EXIT

:RECEIVER INTERRUPTS COMMON HANDLER
RISR:  NOP
      MOV  %D,LINE
      BSL  %D
      MOV  RCSR(0),RXCSPR :GET ADDRESS OF INTERRUPTING DL11-E'S RCSR
      MOV  %RXCSPR,%RXCSPR :GET CSR CONTENTS
      BMY  DCERR :CHECK INT
      TSTB  RXCSR :TEST DONE
      RISRA :FALSE INTERRUPT
      BRROR :EXIT
      RISRA:  CMP  %D,CALLED :DID CALLED LINE INTERRUPT?
      BNE  RISRB :BRANCH IF CALLER INTERRUPTED
      TSTB  MODEM :CHECK MODEM TYPE
      BEQ  RISRAA :BRANCH IF 103
      RISRAA:  JRBUFF(0) :READ CALLED LINES DATA
      MOVB  JRBUFF(0),JINBUFP :STORE CHARACTER IN INPUT BUFFER
      INCB  JINBUFP :INCREMENT POINTER
      CMP  #JINBUFP+100.,JINBUFP :HAVE 100. CHARACTERS BEEN RECEIVED?
      BEQ  RISRC :GO CHECK DATA IF YES
      RTI :EXIT IF NO
      RISRB:  MOVB  JRBUFF(0),JBUFFP :STORE CHARACTER IN INTERMEDIATE DATA BUFFER
      INCB  JBUFFP :INCREMENT POINTER
      CMP  #JBUFFP+10.,JBUFFP :HAVE 10 CHARACTERS BEEN RECEIVED
      BLT  .+4
      RTI :EXIT
      CMP  #2,CONFIG :RUNNING CONFIGURATION 2?
      BEQ  RISRBB
      CMP  #JBUFFP+100.,JBUFFP :HAVE 100. CHARACTERS BEEN RECEIVED?
      BEQ  RISRC :GO CHECK DATA IF YES, OTHERWISE
      RTI :EXIT
      RISRBB:  BIS  #BIT6,%RCSR.0 :START CALLERS TRANSMITTER
      RTI :EXIT
:CHECK DATA CONFIGURATION #1
RISRC:  NOP
      MOV  #1,CTRD :INITIALIZE CHARACTER COUNT
      MOV  #BUFF,%2 :POINT R2 TO CALLERS RECEIVED DATA BUFFER
      MOV  #OUTBUF,%3 :R3 = FIRST ADDRESS OF OUTPUT DATA BUFFER
      MOV  #2,BUFFP :RESTORE CALLERS RCVD DATA BUFFER PTR
      CMP  #1,CONFIG :CHECK CONFIGURATION
      BNE  RISRC
      RISRC:  MOVB  (3)+,%XMTDAT :GET TRANSMITTED CHARACTER
      MOVB  (2)+,%RECDAT :GET RECEIVED CHARACTER
      DATCK :CHECK DATA
      INCB  CTRD :INCREMENT CHARACTER COUNT
      CMP  #101,CTRD :HAS ALL DATA BEEN CHECKED
      BNE  RISRC
      JMP  FINISH

:CHECK DATA CONFIGURATION #2
RISPD:  NOP

```

000000	000000	012704	016660			MOV	#INBUF, 4	:POINT R4 TO CALLED LINES RECEIVER
000000	000000	012704	013776			MOV	4, INBUFP	:DATA BUFFER & INIT. POINTER
000000	000000	012704	015054	001410		MOV	#BUFF, TBUFP	
000000	000000	012704	013701			RISRDA:	MOV	CALLER, %1
000000	000000	016137	010432	001200		MOV	RCSR(1), RXCSR	
000000	000000	112237	001356			MOV	(3)+, XMTDAT	
000000	000000	112237	001356			MOV	(2)+, RECDAT	:COMPARE TRANSMITTED DATA WITH DATA
000000	000000	104004				DATCHK		:RECEIVED BY CALLED LINE
000000	000000	013701	001402			MOV	CALLER, %1	
000000	000000	016137	010432	001200		MOV	RCSR(1), RXCSR	
000000	000000	112237	001356			MOV	(4)+, RECDAT	:COMPARE TRANSMITTED DATA WITH DATA
000000	000000	104004				DATCHK		:RECEIVED BY CALLER
000000	000000	005237	001364			INC	CTRD	
000000	000000	002737	000101	001364		CMP	#101, CTRD	
000000	000000	001356				BNE	RISRDA	
000000	000000	000240				FINISH:	NOP	
000000	000000	013701	001404			MOV	CALLER, %1	
000000	000000	004537	004146			JSR	S.FILL	
000000	000000	014710				OUTBUF		
000000	000000	000144				IOO.		
000000	000000	104000				TYPE		
000000	000000	012356				ENDPAS		
000000	000000	052737	000100	010626		BIS	#BIT6, @TCSR(1)	
000000	000000	000240				NOP		
000000	000000	000002				RTI		
000000	000000	032770	100000	010530		:ERROR SERVICE ROUTINE		
000000	000000	001402				OCERR:	BIT	#BIT15, @RBUF(0) :TEST ERROR
000000	000000	040115				BEQ	RISRF	
000000	000000	012357				ERRRX		
000000	000000	012737	014710	001406		RISRF:	MOV	#OUTBUF, @TBUFP :SET OUTPUT BUFFER POINTER
000000	000000	012737	016660	001376		MOV	#INBUF, INBUFP	:SET INPUT BUFFER POINTER
000000	000000	012737	015054	001400		MOV	#BUFF, @BUFP	:SET INTERMEDIATE BUFFER POINTER
000000	000000	012737	015054	001410		MOV	#BUFF, TBUFP	:SET POINTER FOR CONFIG #2 TRANSMITTER
000000	000000	032737	040000	001370		BIT	#BIT14, RXCSR	:CHECK RING INDICATOR
000000	000000	001005				BNE	RISREX	:BRANCH IF RING
000000	000000	004737	006234			JSR	7, DISCON	:ERROR SET - NO RING
000000	000000	104015				ERRRX		
000000	000000	012357				CSRADD		
000000	000000	000002				RTI		
000000	000000	104000				RISREX:	CNVDA	LINE, TLINE, 2
000000	000000	013715				TYPE		
000000	000000	010037	001404			MOV	%0, CALLED	
000000	000000	004737	003314			JSR	PC, RDOCT	:GET INPUT
000000	000000	01637	001412			MOV	(SP), MODEM	:GET MODEM TYPE
000000	000000	042737	177773	001412		BIC	#177773, MODEM	:0=103, 4=202
000000	000000	012637	001332			MOV	(SP)+, CONFIG	
000000	000000	042737	177774	001332		BIC	#177774, CONFIG	
000000	000000	001042				BNE	RISRFC	:GO TO SERVICE FOR CONFIG 1 OR 2
000000	000000	004737	006250			JSR	7, CONN	:CONNECT LINE IF CONFIGURATION 0
000000	000000	104000				TYPE		:TYPE MESSAGE TO PRESS DATA
000000	000000	014215				BUTTON		:BUTTON ON DATA PHONE
000000	000000	104016				DELAY		:WAIT FOR CARRIER
000000	000000	023420				10000.		:10 SECONDS
000000	000000	005770	010530			TS	@RBUF(0)	:READ BUFFER TO CLEAR DONE



1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448

```

005714 032770 020000 010432 BIT #BIT13,DRCSR(0) :TEST FOR CLEAR TO SEND
005722 001004 BNE RISRFB
005724 104003 ERROR :ERROR! DID NOT RECEIVE CLEAR TO SEND
:WITHIN TIME ALLOTTED 10 SEC.
005726 004737 006234 JSR 7,DISCON :DISCONNECT LINE
005732 000002 RTI :AND EXIT
005734 016037 010626 001204 RISRFB: MOV TCSR(0),TXCSR :GET CALLED LINES TXCSR ADDRESS
005742 004737 004432 JSR 7,SETPAR :LOAD USER PARAMETERS
005746 104000 TYPE :TYPE 'LINE CONNECTION
005750 013427 LINMAD :MADE
005752 000000 CNVOR CONFIG,TCONFIG.2
005756 104000 TYPE
005760 013456 ACCNFIC
005770 052771 000100 010626 BIS #BIT6,DTCSR(0)
005776 000002 RTI :AND EXIT

:HERE IF CONFIGURATION 1 OR 2
RISRFG: TYPE :ASK USER WHICH LINE HE IS
006000 104000 WRU :DIALING ON
006002 014055 JSR PC,ROCT :GET INPUT
006004 004737 003314 MOV (SP)+,%1 :GET LINE #
006010 012601 BIC #177740,%1 :MASK UNUSED BITS
006016 042701 177740 MOV %1,LINE :REPORT LINE # ON TTY
006022 010137 CNVOR LINE,URA.2
006024 104000 TYPE
006026 014162 UR
006030 006301 ASL #1
006034 010137 001402 MOV #1,CALLER :SAVE CALLERS LINE #
006036 004737 006250 JSR 7,CONN :CONNECT CALLED LINE
006038 052771 000002 010432 BIS #BIT1,DRCSR.1 :SET DTR ON CALLERS LINE
006040 104000 TYPE :TYPE MESSAGE TO PRESS DATA
006042 014215 BUTTON :ON DATA PHONE
006044 104016 DELAY :WAIT 10 SECONDS FOR CLEAR TO SEND
006046 023420 10000. :SET AT CALLED LINE
006050 027071 010530 010530 CMP DRBUF(0),DRBF.1) :READ BUFFERS
006052 032770 020000 010432 BIT #BIT13,DRCSR(0) :TEST FOR CLEAR TO SEND AT CALLED LINE
006054 001007 BNE RISRFF
006056 104003 ERROR :ERROR! CLEAR TO SEND NOT SET AT CALLED LINE
006060 004737 006234 RISRFB: JSR 7,DISCON :DISCONNECT
006062 042771 000006 010432 BIC #6,DRCSR.1) :LINE
006064 000002 RTI :AND EXIT
006066 022737 000002 001332 RISRFF: CMP #2,CONFIG
006068 001414 BEQ RISRFG
006070 022771 010000 010432 CMP #BIT12,DRCSR.1) :CHECK CARRIER AT CALLERS LINE
006072 001033 BNE RISRFE
006074 104003 ERROR :ERROR! NO CARRIER AT CALLERS LINE
006076 000137 006110 JMP RISRFB :GO DISCONNECT LINES
006078 016137 010432 001200 RISRFE: MOV RCSR(1),RXCSR :GO GET PARAMETERS AND ENABLE
006080 000137 005734 JMP RISRFB :CALLED TRANSMITTER AND EXIT

:HERE IF CONFIGURATION 2
006164 032771 020000 010432 RISRFG: BIT #BIT13,DRCSR.1) :TEST CALLERS CLEAR TO SEND
006172 001003 BNE RISRFB
006174 104003 ERROR :ERROR! NO CTS AT CALLERS LINE
006176 000137 006110 JMP RISRFB :GO DISCONNECT LINE AND EXIT

```

E03

```

1449 006203 016037 010432 001200 R:SRFH: MOV RCSR(0),RXCSR
1450 006210 016137 010626 001204 MOV TCSR(1),TXCSR
1451 006216 004737 004402 JSR 7,SETPAP ;GO GET PARAMETERS FOR CALLERS
;TRANSMITTER AND CALLED RECEIVER
1452 006222 016137 010432 001200 MOV RCSR(1),RXCSR
1453 006230 000137 005734 JMP RISRFB
;SUBROUTINE TO DISCONNECT LINE RO HAS LINE #
1454 006234 042770 000006 010432 DISCON: BIC #6,RCSR(0)
1455 006242 005770 010432 TST RCSR(0)
1456 006246 000207 RTS 7
;SUBROUTINE TO CONNECT LINE RO HAS LINE #
1457 006250 052770 000006 010432 CONN: BIS #6,RCSR(0) ;SET DTR. RO TO SMC
1458 006256 000207 RTS 7
;TRANSMITTER INTERRUPT COMMON HANDLER
1459 006260 000240 TISR: NOP
1460 006262 006300 ASL %D ;RD HAS LINE #
1461 006264 105770 010626 TSTB @TCSR(0) ;CHECK FOR DONE
1462 006270 100402 BMI TISRA ;BRANCH IF DONE
1463 006272 104003 ERROR ;ERROR! FALSE INTERRUPT
1464 006274 000002 TISRAA: RTI ;EXIT
1465 006276 005737 001332 TISRA: TST CONFIG ;THIS CONFIGURATIO 0?
1466 006302 001420 SEQ TISRC ;BRANCH IF YES
1467 006304 020077 001402 CMP %D,CALLER ;DID CALLER INTERRUPT?
1468 006310 001015 ENE TISRC
1469 006312 117770 173072 010724 MOVVB @TBUFP,@TBUF(0) ;TRANSMIT
1470 006320 005237 001410 INC TBUFP ;STEP POINTER
1471 006324 022737 015220 001410 CMP #BUFP+100.,TBUFP
1472 006332 001003 BNE .+10
1473 006334 042770 000100 010626 BIC #BIT6,@TCSR(0)
1474 006342 000002 RTI
1475 006344 117770 173036 010724 TISRC: MOVVB @OTBUFP,@TBUF(0) ;TRANSMIT THE NEXT CHARACTER
1476 006352 005237 001406 INC OTBUFP ;STEP POINTER TO NEXT CHAR.
1477 006356 005737 001332 TST CONFIG ;WAS CONFIGURATION 0 SELECTED
1478 006362 001010 BNE TISRB ;BRANCH IF CONFIG #1 OR #2
1479 006364 022737 016660 001406 CMP #OUTBUF+1000.,OTBUFP ;HAVE 1000. CHARS. BEEN SENT
1480 006372 001340 BNE TISRAA ;EXIT IF NOT
1481 006374 012737 014710 001406 TISRBS: MOV #OUTBUF,OTBUFP ;RESET POINTER
1482 006402 000002 RTI ;AND EXIT
1483 006404 022737 015054 001406 TISRB: CMP #OUTBUF+100.,OTBUFP ;HAVE 100. CHARS. BEEN SENT?
1484 006412 001330 BNE TISRAA ;EXIT IF NOT
1485 006414 042770 000100 010626 BIC #BIT6,@TCSR(0) ;DISABLE TRANSMITTER INTERRUPT
1486 006422 000764 BR TISRBS ;RESET POINTER AND EXIT
;*****
;PRGD - SINGLE CHARACTER LINE MODE TEST.
;*****
1487 006424 000240 PRGD: NOP ;BEGIN PRGD
1488 006426 104000 TYPE ;TYPE
1489 006430 012507 POTIT ;PROGRAM TITLE
1490 006432 104000 TYPE
1491 006434 013143 SELCAR
1492 006436 004737 003314 JSR PC,RDOCT ;GET INPUT
1493 006442 112601 MOVVB (SP)+,%I ;GET USER SPECIFIED DATA

```

1505	006444	010137	014710		MOV	%1,OUTBUF		;AND
1506	006450	004537	004174		JSR	5,9MOVE		;LOAD
1507	006454	014710			OUTBUF			;INTO
1508	006456	014711			OUTBUF+1			;OUTPUT
1509	006460	001747			999.			;BUFFER
1510	006462	004737	005076		JSR	7,OVRLAY		;OVER LAY CR,LF'S IN DATA
1511	006466	004737	010274		JSR	7,LDPRI		;LOAD PRIORITY LEVEL IN VECTOR+2
1512	006471	004737	010226		JSR	7,LDTVEC		;LOAD TRANSMITTER VECTORS
1513	006476	004737	010162		JSR	7,LDVECS		;LOAD RECEIVER VECTORS
1514	006502	012737	000340	177776	MOV	#PRTY7,PSW		;SET PROCESSOR PRIORITY=7
1515	006510	012702	000140		MOV	#140,%2		;SET IE
1516	006514	012701	010432		MOV	#RCSR,%1		;BIT IN
1517	006520	004537	010116		JSR	5,MOVIT		;ALL RECEIVERS
1518	006524	104000			TYPE			;TYPE
1519	006526	013251			MAKCON			; 'MAKE LINE CONNECTION'
1520	006530	005037	177776		CLR	PSW		;SET PROCESSOR PRIORITY=0
1521	006534	000001			WAIT			;WAIT
1522	006536	000776			BR	PRG0A		;HERE
1523					*****			
1524					;PRG1 - SPECIAL BINARY COUNT PATTERN LINE MODE TEST.			
1525					*****			
1526	006540	104000			PRG1:	TYPE		;TYPE PROGRAM TITLE.
1527	006542	012553			PITIT			
1528	006544	012737	105215	014710	MOV	#105215,OUTBUF		;LOAD CRLF
1529	006552	004537	004142		JSR	5,INFIL		;LOAD OUTPUT
1530	006556	014712			OUTBUF+2			;WITH BINARY
1531	006560	001750			1000.			;COUNT PATTERN
1532	006562	012737	000100	001334	MOV	#100,NUMBER		
1533	006570	004737	010274		JSR	7,LDPRI		;LOAD PRIORITY LEVEL IN VECTOR +2
1534	006574	004737	010226		JSR	7,LDTVEC		;LOAD TRANSMITTER VECTORS
1535	006600	004737	010162		JSR	7,LDVECS		;LOAD RECEIVER VECTORS
1536	006604	012737	000340	177776	MOV	#PRTY7,PSW		;SET PROCESSOR PRIORITY=7
1537	006612	012702	000140		MOV	#140,%2		;GET IE BIT
1538	006616	012701	010432		MOV	#RCSR,%1		;GET FIRST CSR ADDRESS
1539	006622	004537	010116		JSR	5,MOVIT		;AND MOVE IT
1540	006626	104000			TYPE			;TYPE
1541	006630	013251			MAKCON			; 'MAKE LINE CONNECTION'
1542	006632	005037	177776		CLR	PSW		;SET PROCESSOR PRIORITY=0
1543	006636	000001			PRG1C:	WAIT		;WAIT
1544	006640	000776			BR	PRG1C		;HERE
1545					*****			
1546					;PRG2-SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS			
1547					*****			
1548					;THE MESSAGE 'A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS			
1549					;BACK 1234567890.'			
1550					.			
1551	006642	104000			PRG2:	TYPE		;TYPE PROGRAM
1552	006644	012622			P2TIT			;TITLE
1553	006646	004537	004014		JSR	5,LINSEL		
1554	006652	004737	004402		JSR	7,SETPAR		;GO SET PARAMETERS
1555	006656	052777	000004	172314	BIS	#BIT2,3RXC5R		;SET REQUEST TO SEND
1556	006664	004537	004742		PRG2A:	JSR	5,LINCON	;GO MAKE LINE CONNECTION
1557	006670	012702	013504		PRG2B:	MOV	#PRG2M,%2	;GET ADDRESS OF MESSAGE
1558	006674	112201			PRG2C:	MOVB	(2)+,%1	;GET FIRST CHARACTER
1559	006676	020127	000045		CMP	%1,%1		;TERMINATOR CHARACTER
1560	006702	001772			BEQ	PRG2B		;RESEND MESSAGE

```

1561 006704 032737 000100 001374 BIT #BIT6,SRT ;PARITY ENABLED
1562 006712 001402 BEQ .+6
1563 006714 004737 003600 JSR 7,GENPAR ;GENERATE PARITY
1564 006720 004537 004742 JSR 5,LINCON ;CHECK LINE CONNECTION
1565 006724 010177 172256 MOV #1,@TXBUF ;LOAD BUFFER
1566 006730 105777 172250 TSTB @TXCSR ;AND WAIT FOR CHARACTER
1567 006734 100375 SPL -4 ;TO BE TRANSMITTED
1568 006736 000756 BR PRG2C ;GET NEXT CHARACTER.
1569
1570 ;*****
1571 ;PRG3-PROGRAM TO RECEIVE A MESSAGE.
1572 ;*****
1573 PRG3: TYPE ;TYPE PROGRAM
1574 P3TIT ;TITLE
1575 JSR 5,LINSEL
1576 JSR 7,SETPAR ;GET PARAMETERS
1577 PRG3A: MOV #STKPTR-2,%6 ;REPOSITION STACK POINTER
1578 BIS #BIT2,@RXCSR ;SET REQUEST TO SEND
1579 JSR 5,LINCON ;MAKE LINE CONNECTION
1580 STRXV ;SET RECEIVER INTERRUPT
1581 RINT3 ;TO THIS ADDRESS
1582 STTXV ;SET TRANSMITTER INTERRUPT
1583 TINT3 ;TO THIS ADDRESS
1584 CLR WORDS
1585 MOV TPVTR,%0
1586 MOV #TPINT,(0)+ ;LOAD TELEPRINTER VECTOR
1587 MOV TPLVL,(0) ;AND PRIORITY
1588 MOV #OUTBUF,%1 ;GET BUF ADD
1589 BIS #BIT15,TFLAG ;SET BIT 15
1590 JSR 7,TCRLF ;SEND CRLF
1591 BIS #140,@RXCSR ;ENABLE RECEIVER INTERRUPTS
1592 CLR PSW
1593 WAIT ;DO
1594 BR -2 ;NOTHING
1595 RINT3: MOV @RXCSR,RXCST ;GET RXCSR DATA
1596 BMI ERR3A ;BRANCH IF ERROR
1597 TSTB RXCSR ;TEST
1598 BPL ERR3B
1599 INC WORDS
1600 MOV @RXBUF,RXBUFT ;GET DATA
1601 MOVB RXBUFT,(1)
1602 TST RXBUFT
1603 BMI ERR3C
1604 TSTB @SRPTR ;ECHO OPTION SELECTED
1605 BMI RINT3A
1606 TSTB @TXCSR
1607 BPL -4
1608 MOVB (1),@TXBUF ;ECHO CHARACTER
1609 RINT3A: CMP WORDS,#900. ;END OF BUFFER ALLOWED
1610 BEQ RINT3B ;YES EXIT
1611 TST TFLAG ;IS THIS THE FIRST
1612 BMI RINT3E ;CHARACTER BRANCH IF YES.
1613 CMPB (1),%3 ;LAST CHARACTER RECEIVED
1614 BEQ RINT3B
1615 CMPB (1)+,#203 ;CONTROL C
1616 BEQ RINT3E ;EXIT
RTI

```

```

1617 007172 005037 007470 RINT3B: CLR WORDS
1618 007176 042777 000140 171774 BIC #140, @RXCSR ;DISABLE RECEIVER
1619 007204 012701 014711 MOV #OUTBUF+1, %1 ;INITIALIZE BUFFER POINTER
1620 007210 010102 MOV %1, %2
1621 007212 052777 000100 171764 BIS #BIT6, @TXCSR ;ENABLE TRANSMITTER
1622 007220 052777 000100 171776 BIS #BIT6, @TPS ;ENABLE TELEPRINTER
1623 007226 000002 RTI ;EXIT
1624 007230 104015 ERR3A: ERRRX ;TYPE ERROR MESSAGE
1625 007232 013605 LFAIL
1626 007234 042777 000140 171736 BIC #140, @RXCSR ;DISABLE RECEIVER
1627 007242 000644 BR PRG3A
1628 007244 104015 ERR3B: ERRRX ;TYPE
1629 007246 012366 RINTM ;ERROR MESSAGE
1630 007250 000002 RTI ;EXIT
1631 007252 104015 ERR3C: ERPRX
1632 007254 013623 ROVER
1633 007256 000002 RTI
1634
1635 007260 005J37 007466 RINT3E: CLR TFLAG
1636 007264 112103 MOV (1)+, %3
1637 007266 000002 RTI
1638
1639 007270 017737 171710 001366 TINT3: MOV @TXCSR, TXCSR ;GET TXCSR DATA
1640 007276 105737 001366 TSTB TXCSR ;TEST
1641 007302 100016 BPL TINT3B
1642 007304 112177 171676 MOV (1)+, @TXBUF ;TRANSMIT CHARACTER
1643 007310 005237 007470 INC WORDS
1644 007314 121103 CMPB (1), %3 ;ALL CHARACTERS TRANSMITTED
1645 007316 001431 BEQ TINT3C
1646 007320 023727 007470 001604 CMP WORDS, #900.
1647 007326 001425 BEQ TINT3C
1648 007330 121127 000203 CMPB (1), #203 ;= CONTROL C
1649 007334 00 02 BEQ TINT3C
1650 007336 000002 RTI ;RETURN TO MAIN PROGRAM
1651 007340 017737 171660 001372 TINT3B: MOV @TPS, TEMP ;SAVE TELEPRINTER STATUS
1652 007346 005077 171652 CLR @TPS ;DISABLE INTERRUPT
1653 007352 105777 171646 TSTB @TPS ;WAIT FOR
1654 007356 100375 BPL -4 ;TELEPRINTER TO FINISH
1655 007360 104014 ERRTX ;TYPE
1656 007362 012407 TINTM ;ERROR MESSAGE
1657 007364 105777 171634 TSTB @TPS ;WAIT FOR TELEPRINTER
1658 007370 100375 BPL -4 ;TO FINISH
1659 007372 013777 001372 171624 MOV TEMP, @TPS ;RESTORE TELEPRINTER STATUS
1660 007400 000002 RTI ;EXIT
1661
1662 007402 042777 000100 171574 TINT3C: BIC #BIT6, @TXCSR ;DISABLE INTERRUPT
1663 007410 032777 000100 171606 BIT #BIT6, @TPS ;IS TTY ACTIVE
1664 007416 001421 BEQ PRG3EX
1665 007420 000002 RTI
1666
1667
1668 007422 112277 171600 TPINT: MOV (2)+, @TPB ;TYPE CHARACTER
1669 007426 121203 CMPB (2), %3 ;WAS THIS THE LAST CHAR.
1670 007430 001404 BEQ TPINTA
1671 007432 121227 000203 CMPB (2), #203 ;= CONTROL C
1672 007436 001401 BEQ TPINTA
  
```

```

1673 007440 000J02          RFI
1674 007442 042777 000100 171554 TPINTA: BIC      #BIT6,@TPS      ;DISABLE INTERRUPT
1675 007450 032777 000100 171526          BIT      #BIT6,@TXCSR ;IS TRANSMITTER ACTIVE
1676 007456 001401          BEQ      .+4
1677 007460 000002          RTI
1678 007462 000137 006754          PRG3EX: JMP      PRG3A      ;EXIT
1679 007466 000000          TFLAG: 0
1680 007470 000000          WORDS: 0
1681 007472 000000          RXBUFT: 0
1682
1683          ;*****
1684          ;PRG4-SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS
1685          ;*****
1686          ;MESSAGE SPIRAL PATTERN
1687
1687 007474 104000          PRG4:  TYPE
1688 007476 012730          P4TIT          ;TYPE PROGRAM
1689 007500 004537 004014          JSR      5,LINSEL          ;TITLE
1690 007504 004737 004402          JSR      7,SETPAR          ;GO SET PARAMETERS
1691 007510 052777 000004 171462          BIS      #BIT2,@RXCSR          ;SET REQUEST TO SEND
1692 007516 012737 000110 007652          MOV      #72.,COLMN          ;INIT PAGE WIDTH
1693 007524 012703 000040          MOV      #40,%3          ;SET LINE START CHAR
1694 007530 004537 004742          PRG4A: JSR      5,LINCON          ;GO MAKE LINE CONNECTION
1695 007534 004737 007654          JSR      7,TCRLF
1696 007540 010302          PRG4B: MOV      %3,%2          ;GET FIRST CHARACTER
1697 007542 110201          PRG4C: MOV      %2,%1          ;GET CHARACTER
1698 007544 120127 000136          CMP      %1,#136          ;TERMINATOR CHARACTER
1699 007550 001003          BNE      PRG4D          ;RESEND MESSAGE
1700 007552 012702 000040          MOV      #40,%2
1701 007556 000771          BR       PRG4C
1702 007560 032737 000100 001374          PRG4D: BIT      #BIT6,SRT          ;PARITY ENABLED
1703 007566 001402          BEQ      .+6
1704 007570 004737 003600          JSR      7,GENPAR          ;GENERATE PARITY
1705 007574 004537 004742          JSR      5,LINCON          ;CHECK LINE CONNECTION
1706 007600 010177 171402          MOV      %1,@TXBUF          ;LOAD BUFFER
1707 007604 105777 171374          TST      @TXCSR          ;AND WAIT FOR CHARACTER
1708 007610 100375          BPL      .-4          ;TO BE TRANSMITTED
1709 007612 005202          INC      %2          ;SET FOR NEXT CHAR
1710 007614 005337 007652          DEC      COLMN          ;ALL COLUMNS PRINTED?
1711 007620 001350          BNE      PRG4C          ;NO, GET NEXT CHAR
1712 007622 012737 000110 007652          MOV      #72.,COLMN          ;RESET COLUMN COUNTER
1713 007630 004737 007654          JSR      7,TCRLF
1714 007634 005203          INC      %3          ;UPDATE LINE START CHAR
1715 007636 120327 000136          CMP      %3,#136          ;LAST IN SET
1716 007642 001336          BNE      PRG4B          ;NO
1717 007644 012703 000040          MOV      #40,%3          ;YES, RESET
1718 007650 006733          BR       PRG4B          ;GET NEXT CHARACTER.
1719 007652 000000          COLMN: 0
1720
1721          ;SEND CR LF
1722
1723 007654 112777 000015 171324          TCRLF: MOV      #15,@TXBUF          ;SEND CR,LF
1724 007662 105777 171316          TST      @TXCSR
1725 007666 100375          BPL      .-4
1726 007670 112777 000012 171310          MOV      #12,@TXBUF
1727 007676 105777 171302          TST      @TXCSR
1728 007702 100375          BPL      .-4
  
```



1729	007704	000207			RIS	%7	
1730					*****		
1731					PROGRAM 5		
1732					*****		
1733	007706	104000			PRGS:	TYPE	
1734	007710	012777				PSTIT	
1735	007712	004537	004014			JSR	5,LINSEL
1736	007716	000005				RESET	
1737	007720	004737	004402			JSR	7,SETPAR
1738	007724	052777	000006	171246		BIS	#6,DRXCSR
1739	007732	104000			PRGSA:	TYPE	;SET DTR RQ TO SND
1740	007734	013251				MAKCON	;TYPE MESSAGE TO MAKE
1741	007736	000000				HALT	;LINE CONNECTION
1742	007740	005777	171236			TST	;WAIT FOR USER TO MAKE LINE CONNECTION
1743	007744	032777	020000	171226		BIT	;READ BUFFER
1744	007752	001003				BNE	;TEST FOR CLEAR TO SEND
1745	007754	104000			PRGSA:	TYPE	;TYPE ERROR MESSAGE
1746	007756	013377				LINCHM	
1747	007760	000764				BR	PRGSA
1748	007762	104000			PRGSB:	TYPE	;AND TRY AGAIN
1749	007764	013427				LINMAD	
1750	007766	005037	010114			CLR	ERRCNT
1751	007772	012702	013504		PRGSBB:	MOV	#PRG2M,%2
1752	007776	112201			PRGSC:	MOVB	(2)+,%1
1753	010000	020127	000045			CMP	%1,%1
1754	010004	001440				BEQ	PRGSE
1755	010006	032737	000100	001374		BIT	#BIT6,SRT
1756	010014	001402				BEQ	+6
1757	010016	004737	003600			JSR	7,GENPAR
1758	010022	032777	020000	171150		BIT	#BIT13,DRXCSR
1759	010030	001751				BEQ	PRGSA
1760	010032	010177	171150			MOV	%1,DTXBUF
1761	010036	005777	171140			TST	DRXBUF
1762	010042	100001				BPL	+4
1763	010044	104003				ERROR	
1764	010046	105777	171126			TSTB	DRXCSR
1765	010052	100375				BPL	-4
1766	010054	117703	171122			MOVB	DRXBUF,%3
1767	010060	043701	001352			BIC	CARMSK,%1
1768	010064	120103				CMPB	%1,%3
1769	010066	001403				BEQ	PRGSD
1770	010070	104003				ERROR	;ERROR! DATA ERROR
1771	010072	005237	010114			INC	ERRCNT
1772	010076	105777	171102		PRGSD:	TSTB	DTXCSR
1773	010102	100375				BPL	-4
1774	010104	000734				BR	PRGSC
1775	010106	104000			PRGSE:	TYPE	
1776	010110	013350				ENDPAS	
1777	010112	000727				BR	PRGSBB
1778	010114	000000			ERRCNT:	OPEN	
1779							;THIS ROUTINE MOVES THE CONTENTS OF R2 TO THE ADDRESS SPECIFIED
1780							;BY R1
1781	010116	012737	000006	000004	MOVIT:	MOV	#6,4
1782	010124	012737	000002	000006		MOV	#2,6
1783	010132	012700	000037			MOV	#31,%0
1784	010136	010231			MOVITA:	MOV	%2,%(1)+

```

1795 010140 005300          DEC      %0          ;ALL DATA MOVED?
1796 010142 001375          BNE     MOVITA       ;NO. RETURN
1797 010144 012737 004520 000004  MOV     #ERTP,MACHER
1798 010152 012737 000040 000006  MOV     #40,MACHER+2
1799 010160 000205          RTS      5          ;RETURN

;SUBROUTINE TO LOAD ALL VECTORS
1791
1792 010162 012701 011022  LDVECS: MOV     #RISRO,%1
1793 010166 012702 010334          MOV     #VECTAB,%2
1794 010172 012703 000010          MOV     #10,%3
1795 010176 012704 000037          MOV     #31,%4
1796 010202 032712 000001  LDVECB: BIT     #BIT0,(2)          ;DOES THIS VECTOR EXIST
1797 010206 001002          BNE     LDVEC1       ;NO, SKIP LOADING
1798 010210 010172 000000  LDVEC1: MOV     %1,%(2)          ;LOAD VECTOR
1799 010214 060301          ADD     %3,%1
1800 010216 005722          TST    (2)+
1801 010220 005304          DEC     %4
1802 010222 001367          BNE     LDVECB
1803 010224 000207          RTS

;LDTVEC
1804
1805 010226 012701 011412  LDTVEC: MOV     #TISRO,%1
1806 010232 012702 010334          MOV     #VECTAB,%2
1807 010236 012703 000010          MOV     #10,%3
1808 010242 012704 000037          MOV     #31,%4
1809 010246 032712 000001  LDTVED: BIT     #BIT0,(2)          ;DOES THIS VECTOR EXIST
1810 010252 001003          BNE     LDVEC2       ;NO, SKIP LOADING
1811 010254 011200          MOV     (2),%0
1812 010256 010160 000004  LDVEC2: MOV     %1,%(3)
1813 010262 060301          ADD     %3,%1
1814 010264 005722          TST    (2)+
1815 010266 005304          DEC     %4
1816 010270 001356          BNE     LDTVED
1817 010272 000207          RTS      7

;ROUTINE TO LOAD PRIORITY LEVEL 7 IN VECTOR +2
1818
1819
1820 010274 012701 010334  LDPRI: MOV     #VECTAB,%1          ;GET BASE VECTOR
1821 010300 012702 000340          MOV     #340,%2          ;GET LEVEL 7
1822 010304 012703 000037          MOV     #31,%3          ;LOAD COUNTER
1823 010310 032711 000001  LDPRIA: BIT     #BIT0,(1)          ;DOES THIS VECTOR EXIST
1824 010314 001003          BNE     LDPRIX       ;NO SKIP LOADING
1825 010316 011104          MOV     (1),%4          ;LOAD VECTOR +2
1826 010320 010264 000002  LDPRIX: MOV     %2,%(4)
1827 010324 005721          TST    (1)+          ;POINT TO NEXT VECTOR
1828 010326 005303          DEC     %3          ;DECREMENT COUNTER
1829 010330 001367          BNE     LDPRIA
1830 010332 000207          RTS      7

;VECTOR ASSIGNMENT TABLE
1831
1832 010334 000301  VECTAB: 301          ;LINE 0 VECTOR
1833 010336 000311          311          ;LINE 1 VECTOR
1834 010340 000321          321          ;LINE 2 VECTOR
1835 010342 000331          331          ;LINE 3 VECTOR
1836 010344 000341          341          ;LINE 4 VECTOR
1837 010346 000351          351          ;LINE 5 VECTOR
1838 010350 000361          361          ;LINE 6 VECTOR
1839 010352 000371          371          ;LINE 7 VECTOR
1840 010354 000401          401          ;LINE 10 VECTOR

```

1841	010356	000411	411	;LINE	11	VECTOR
1842	010360	000421	421	;LINE	12	VECTOR
1843	010362	000431	431	;LINE	13	VECTOR
1844	010364	000441	441	;LINE	14	VECTOR
1845	010366	000451	451	;LINE	15	VECTOR
1846	010370	000461	461	;LINE	16	VECTOR
1847	010372	000471	471	;LINE	17	VECTOR
1848	010374	000501	501	;LINE	20	VECTOR
1849	010376	000511	511	;LINE	21	VECTOR
1850	010400	000521	521	;LINE	22	VECTOR
1851	010402	000531	531	;LINE	23	VECTOR
1852	010404	000541	541	;LINE	24	VECTOR
1853	010406	000551	551	;LINE	25	VECTOR
1854	010410	000561	561	;LINE	26	VECTOR
1855	010412	000571	571	;LINE	27	VECTOR
1856	010414	000601	601	;LINE	30	VECTOR
1857	010416	000611	611	;LINE	31	VECTOR
1858	010420	000621	621	;LINE	32	VECTOR
1859	010422	000631	631	;LINE	33	VECTOR
1860	010424	000641	641	;LINE	34	VECTOR
1861	010426	000651	651	;LINE	35	VECTOR
1862	010430	000661	661	;LINE	36	VECTOR
1863						
1864		000000				
1865		000000				
1866	010432					
1867						
1868						
1869						
1870						
1871		000000				
1872		000000				
1873	010530					
1874						
1875						
1876						
1877						
1878		000000				
1879		000000				
1880	010626					
1881						
1882						
1883						
1884						
1885		000000				
1886		000000				
1887	010724					
1888						
1889						
1890						
1891						
1892		000000				
1893						
1894						
1895						
1896						

;DL11-E REGISTER ADDRESSES

```

N=0
A=0
RCSR: .REPT 31.
      RRCV  \N,\A
      N=N+10
      A=A+1
      .ENDR
N=0
A=0
RBUF: .REPT 31.
      RBUFF \N,\A
      N=N+10
      A=A+1
      .ENDR
N=0
A=0
TCSR: .REPT 31.
      TXMT  \N,\A
      N=N+10
      A=A+1
      .ENDR
N=0
A=0
TBUF: .REPT 31.
      TBUFF \N,\A
      N=N+10
      A=A+1
      .ENDR
N=0
ISR:  .REPT 31.
      ISR  \N
      N=N+1
      .ENDR

```

M03

1997		000000			N=0	
1998					.REPT	32.
1999					ISRT	\N
1900					N=N+1	
1901					.ENDR	
1902					:MESSAGES	
1903	012012	042045	030514	026461	MTITLE: .ASCII	'%DL11-E ON LINE TEST - MAINDEC-11-DZDLB-B:'
1904	012020	020105	047117	046040		
1905	012026	047111	020105	042524		
1906	012034	052123	026440	046440		
1907	012042	044501	042116	041505		
1908	012050	030455	026461	055104		
1909	012056	046104	026502	022502		
1910	012064	046445	050101	047440	.ASCII	'%MAP OF DEVICES PRESENT%'
1911	012072	020106	042504	044526		
1912	012100	042503	020123	051120		
1913	012106	051505	047105	022524		
1914	012114	046045	047111	020105	.ASCII	'%LINE D-ADR TRAP AT%a'
1915	012122	042040	040455	051104		
1916	012130	020040	020040	052040		
1917	012136	040522	020120	052101		
1918	012144	040045				
1919	012146	020040	020040	020040	MLINE: .ASCII	' , , '
1920	012154	020040	020040	020040	MDADR: .ASCII	' , , '
1921	012162	020040	020040			
1922	012166	020040	020040	020040	MTRAP: .ASCII	' %a '
1923	012174	040045				
1924	012176	047045	047117	020105	MNONE: .ASCII	'%NONE FOUND%a'
1925	012204	047506	047125	022504		
1926	012212	100				
1927	012213	045	054524	042520	MSWSEL: .ASCII	'%TYPE IN PROGRAM NUMBER a'
1928	012220	044440	020116	051120		
1929	012226	043517	040522	020115		
1930	012234	052516	041115	051105		
1931	012242	020040	020040	100		
1932	012247	045	051105	047522	MTERR: .ASCII	'%ERROR - UNEXPECTED TRAP'
1933	012254	020122	020055	047125		
1934	012262	054105	042520	052103		
1935	012270	042105	052040	040522		
1936	012276	120				
1937	012277	045	051124	050101	.ASCII	'%TRAPPED TO '
1938	012304	042520	020104	047524		
1939	012312	020040				
1940	012314	020040	020040	020040	MTO: .ASCII	' , , '
1941	012322	020040				
1942	012324	052045	040522	050120	.ASCII	'%TRAPPED FROM PC '
1943	012332	042105	043040	047522		
1944	012340	020115	041520	020040		
1945	012346	020040	020040	020040	MFROM: .ASCII	' , a '
1946	012354	020040	100			
1947	012357	040	020040	020040	CSRADD: .ASCII	' , a '
1948	012364	040040				
1949	012366	043045	046101	042523	RINTM: .ASCII	'%FALSE INT. RCVRa'
1950	012374	044440	052116	020056		
1951	012402	041522	051126	100		
1952	012407	045	040506	051514	TINTM: .ASCII	'%FALSE INT XMITa'

1953	012414	020105	047111	020124			
1954	012422	046530	052111	100			
1955	012427	045	041520	020075	EMO: .ASCII	'%PC= '	
1956	012434	020040	020040	020040	APC: .ASCII	' @'	
1957	012442	020040	100				
1958	012445	040	052040	041530	ATXCSR: .ASCII	' TXCSR = '	
1959	012452	051123	036440	040			
1960	012457	040	020040	020040	ATXWAS: .ASCII	' @'	
1961	012464	040040					
1962	012466	020040	054122	051503	ARXCSR: .ASCII	' RXCSR = '	
1963	012474	020122	020075				
1964	012500	020040	020040	020040	ARXWAS: .ASCII	' @'	
1965	012506	100					
1966	012507	045	050045	043522	POTIT: .ASCII	'%PRGO - SINGLE CHAR LINE MODE TEST@'	
1967	012514	020060	020055	044523			
1968	012522	043516	042514	041440			
1969	012530	040510	020122	044514			
1970	012536	042516	046440	042117			
1971	012544	020105	042524	052123			
1972	012552	100					
1973	012553	045	050045	043522	PITIT: .ASCII	'%PRG1 - SPEC BIN COUNT LINE MODE TEST@'	
1974	012560	020061	020055	050123			
1975	012566	041505	041040	047111			
1976	012574	041440	052517	052116			
1977	012602	046040	047111	020105			
1978	012610	047515	042504	052040			
1979	012616	051505	040124				
1980	012622	022445	051120	031107	P2TIT: .ASCII	'%PRG2 - SPECIAL MESSAGE LINE MODE TEST@'	
1981	012630	026440	051440	042520			
1982	012636	044503	046101	046440			
1983	012644	051505	040523	042507			
1984	012652	046040	047111	020105			
1985	012660	047515	042504	052040			
1986	012666	051505	040124				
1987	012672	022445	051120	031507	P3TIT: .ASCII	'%PRG3 - RECEIVE MESSAGE TEST@'	
1988	012700	026440	051040	041505			
1989	012706	044505	042526	046440			
1990	012714	051505	040523	042507			
1991	012722	052040	051505	040124			
1992	012730	022445	051120	032107	P4TIT: .ASCII	'%PRG4 - SPECIAL MESSAGE TEST (SPIRAL)@'	
1993	012736	026440	051440	042520			
1994	012744	044503	046101	046440			
1995	012752	051505	040523	042507			
1996	012760	052040	051505	020124			
1997	012766	051450	044520	040522			
1998	012774	024514	100				
1999	012777	045	051120	032507	P5TIT: .ASCII	'%PRG5 - DATA ECHO TEST USING MAYNARD FACILITY@'	
2000	013004	026440	042040	052101			
2001	013012	020101	041505	047510			
2002	013020	052040	051505	020124			
2003	013026	051525	047111	020107			
2004	013034	040515	047131	051101			
2005	013042	020104	040506	044503			
2006	013050	044514	054524	100			
2007	013055	045	054524	042520	SELPAR: .ASCII	'%TYPE IN PARAMETERS AS FOLLOWS:'	
2008	013062	044440	020116	040520			









000000	014634	033467	033467	020040
000000	014642	047524	041440	040510
000000	014650	043516	020105	047101
000000	014656	052117	042510	122
000000	014663	040	042504	044526
000000	014670	042503	040440	042104
000000	014676	042522	051523	020040
000000	014704	020040	100	
000000	014710	014710		
000000	014710	000000		
000000	016550	016660		
000000		000000		
000000		020030		
000000		015054		
000000	020630	000001		

.ASCII ' DEVICE ADDRESS 3'

```

      .EVEN
OUTBUF: OPEN
        =OUTBUF+1000.
INBUF:  OPEN
        =INBUF+1000.
BUFF=OUTBUF+100.
DEVD:  .END

```



CRXTXC	003032	912	921#							
CSB	013313	856	2036#							
CSRADD	012357	857	858	1366	1375	1947#				
CTRD	001364	558#	860	1127*	1130*	1321*	1330*	1331	1349*	1350
CWAS	013225	855	2038#							
DATCH# =	104004	524#	1329	1344	1348					
DERR	005532	1291	1363#							
DECVL	004374	1147	1182#							
DELAY =	104016	534#	1262	1390	1425					
DEND	020530	2191#								
DIGIT	004356	1165*	1168*	1171*	1172	1175#				
DISCON	006234	1373	1397	1431	1457#					
DLONT	003526	1021*	1023	1033#						
DLY	003466	639	1021#							
DLYA	003504	1025#	1000							
DLYB	003510	1026#	1027							
DLYC	003522	1024	1031#							
DCIT	003742	1068	1072	1076	1080	1081#				
DONE	003776	1087	1089#							
DTCHK	002412	629	851#							
DTCHKA	002510	853	865#							
DTERR	013232	1011	2027#							
ECDAT	003346	990#	991							
EHALT =	104010	528#	886	921						
EHLT	002400	633	845#							
EHLTA	002410	846	848#							
EIGHT	003632	1059	1065#							
EMTA	002234	801#								
EMTINT	002222	478	798#							
EMTTAB	001264	625#	803							
EMO	012427	882	917	1955#						
EXCPAS	013350	1358	1776	2042#						
ERR	002512	628	867#							
ERRA	002562	870	875#							
ERRB	002624	867*	868*	871*	872*	873*	883#			
ERRC	002632	876	886#							
ERRCNT	010114	1750*	1771*	778#						
ERRD	002642	877*	878*	880	889#	913*	914*	916		
ERRE	002644	863*	874*	887	890#					
ERROR =	104003	523#	1294	1395	1430	1438	1447	1469	1763	1770
ERROR1 =	104013	531#	857	863						
ERRRX =	104015	533#	1268	1365	1374	1624	1628	1631		
ERRTX =	104014	532#	1655							
ERR1	002534	636	871#							
ERR3A	007230	1595	1624#							
ERR3B	007244	1597	1628#							
ERR3C	007252	1602	1631#							
ERTP	004520	466	468	470	472	480	678	732	1206#	1787
ERTPA	004570	1214#	1233							
FILL	004146	1126#	1354							
FILLA	004154	1128#	1131							
FINISH	005476	1333	1352#							
FIVE	003720	1064	1077#							
FMAP	001350	652#	712*	1232	1237*					
FMONE	001346	651#	685*	729*	734					
FORMAC	004704	711	1241#							













TISR3	011442	1902*																		
TISR30	011712	1902*																		
TISR31	011722	1902*																		
TISR32	011732	1902*																		
TISR33	011742	1902*																		
TISR34	011752	1902*																		
TISR35	011762	1902*																		
TISR36	011772	1902*																		
TISR37	012002	1902*																		
TISR4	011452	1902*																		
TISR5	011462	1902*																		
TISR6	011472	1902*																		
TISR7	011502	1902*																		
TKB	001222	608*	998																	
TKLVL	001232	612*																		
TKS	001220	607*	986																	
TKVTR	001230	611*																		
TLINE	013725	1378	2092*																	
TLINEX	013675	1120	2087*																	
TCPC	001347	648*	1212*	1215	1228*	1231*	1234	1235*	1236											
TPB	001226	610*	955*	992*	1668*															
TPINT	007422	1585	1668*																	
TPINTA	007442	1670	1672	1674*																
TPLVL	001236	614*	1586																	
TPS	001224	609*	956	990	1622*	1651	1652*	1653	1657	1659*	1663	1674*								
TPVTR	001234	613*	1584																	
TXBUF	001206	602*	1247*	1565*	1607*	1642*	1706*	1723*	1726*	1760*										
TXCSR	001204	601*	713*	714*	722*	1245*	1399*	1450*	1566	1605	1621*	1639	1662*	1675						
		1707	1724	1727	1772															
TXCSRT	001366	659*	905	1639*	1640															
TXEPR	002710	637	904*																	
TXLVL	001216	606*	938																	
TXVTR	001214	605*	936	1234*																
TYP	003122	625	942*																	
TYPA	003136	946*	954	963																
TYPC	003160	948	951*																	
TYPD	003176	952	955*	960	962															
TYPDAT	003242	946*	947	951	955	959*	961*	964*												
TYPE =	104000	520*	682	727	736	742	754	762	765	770	789	834	898	975						
		1010	1095	1120	1185	1190	1222	1256	1273	1357	1378	1388	1401	1404						
		1410	1417	1423	1499	1501	1518	1526	1540	1551	1572	1687	1733	1739						
		1745	1748	1775																
TYPES =	104001	521*	881	916																
TYPF	003214	952	959*																	
TYPB	003244	626	966*																	
TYPSA	003300	971	974*																	
TYPSAA	003250	967*	977																	
TYPSB	003306	969*	970	976*																
UR	014162	1418	2121*																	
URA	014212	1417	2125*																	
VECTAB	010334	721*	1110	1793	1806	1920	1832*													
WORDS	007470	1583*	1598*	1608	1617*	1643*	1646	1680*												
WRU	014065	1411	2110*																	
XMTDAT	001360	656*	851*	852	856	1327*	1342*													
.	= 020630	465*	482	488	574*	577*	598*	957	1116	1310	1478	1562	1567	1593						
		1606	1654	1658	1676	1703	1708	1725	1728	1756	1762	1765	1773	2062*						

N04

DZDL68 MACY11 27(732) 17-SEP-76 15:21 PAGE 53  
DZDL68.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

2185# 2187# 2189#



TABLE 15:21 PAGE 55  
MACRO NAMES

TABLE	MACRO	NAME	955	956	879	904	907	915	1119	1189	1377	1403	1416
TABLE 15:21	MACRO	NAME	955	956	879	904	907	915	1119	1189	1377	1403	1416

1065	1066	1069	1070	1073	1074	1077	1079	1081	1089	1099	1098	1102	1103	1107	1109	1111	1114	1115	1116	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

1105	1110	1111	1114	1126	1127	1136	1137	1138	1147	1148	1149	1150	1151	1152
1106	1158	1188	1192	1195	1198	1202	1206	1207	1212	1213	1228	1230	1234	1236
1107	1243	1245	1247	1251	1258	1264	1271	1277	1278	1279	1287	1289	1290	1321
1108	1323	1324	1327	1338	1339	1340	1341	1345	1346	1353	1367	1368	1369	1370
1109	1388	1389	1399	1413	1415	1420	1440	1449	1450	1453	1487	1505	1514	1515
1110	1508	1532	1536	1537	1538	1557	1565	1576	1584	1585	1596	1587	1594	1599
1111	1600	1639	1651	1659	1692	1693	1696	1700	1706	1712	1717	1751	1760	1791
1112	1703	1787	1787	1788	1792	1793	1794	1795	1798	1805	1906	1907	1808	1811
1113	1800	1821	1822	1825	1826	1897	1902	1902	1902	1902	1906	1907	1808	1811
1114	1900	1909	1911	1919	1922	1922	1924	1927	1937	1940	1942	1945	1947	1949
1115	1914	1956	1958	1960	1962	1964	1964	1966	1972	1980	1992	1992	2007	2013
1116	2020	2030	2034	2036	2038	2041	2041	2043	2044	2048	2049	2054	2059	2061
1117	2076	2076	2079	2085	2087	2090	2090	2092	2095	2110	2121	2125	2126	2132
1118	2149	2149	2154	2165	2168	2175	2181	2181	2181	2181	2181	2181	2181	2181
1119	2062	2062	2154	2165	2168	2175	2181	2181	2181	2181	2181	2181	2181	2181
1120	2185	2185	2185	2185	2185	2185	2185	2185	2185	2185	2185	2185	2185	2185
1121	484	551	557	561	565	569	569	569	569	569	569	569	569	569
1122	1866	1873	1880	1887	1893	1898	1898	1898	1898	1898	1898	1898	1898	1898
1123	1347	1475	1481	1504	1558	1600	1607	1636	1642	1668	1697	1723	1725	1732
1124	596	715	1286	1320	1336	1352	1360	1465	1498					
1125	1047	1048	1082	865	888	923	931	939	950	973	1032	1238	1295	1301
1126	1311	1316	1318	1361	1376	1399	1407	1433	1470	1490	1488	1616	1623	1630
1127	1637	1650	1660	1665	1673	1677	1677	1677	1677	1677	1677	1677	1677	1677
1128	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1129	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1130	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1131	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1132	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1133	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1134	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1135	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1136	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1137	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1138	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1139	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1140	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1141	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1142	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1143	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1144	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1145	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1146	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1147	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1148	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1149	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1150	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1151	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703
1152	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703	1703

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\* DZDLBB SEQ/SOL/CRF/PAGE: 1=DZDLBB  
RUN-TIME: 9 15 3 SECONDS  
RUN-TIME RATIO: 93/30=3.0  
CORE USED: 11K (21 PAGES)

