

DL11-E

DL11-E ON LINE TEST
CZDLBC0

AH-8521C-MC

COPYRIGHT 72-80
FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-8520C-MC
PRODUCT NAME: CZDLBCO DL11-E ON LINE TEST
PRODUCT DATE: JULY 1979
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: ROBERT WHITTON

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1972,1979 BY DIGITAL EQUIPMENT CORPORATION

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

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR THE DL11-E (ASYNCHRONOUS LINE INTERFACE), CZDLA (DL11-E OFF LINE TESTS) AND CZDLB (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/O PARITY
- PRG3 RECEIVE DATA TEST
- PRG4 MESSAGE TRANSMIT (SPIRAL) ONLY W/W/O PARITY.

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

THIS PROGRAM USES 8K OF MEMORY

94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122

3. OPERATING PROCEDURE:

3.1 LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

3.2 DL11-E PARAMETER SELECTION

THE SELECTABLE DL11-E PARAMETERS ARE:

BIT1-0 CHARACTER LENGTH

| BIT1 | BIT0 | CHAR. LENGTH |
|------|------|--------------|
| 0 | 0 | 8 |
| 0 | 1 | 7 |
| 1 | 0 | 6 |
| 1 | 1 | 5 |

WHEN A TERMINAL IS INVOLVED DL11-E PARAMETERS SHOULD BE SET ACCORDING TO TERMINAL SPECIFICATIONS.

3.3 PDP-11 STANDARD OPERATING PARAMETERS

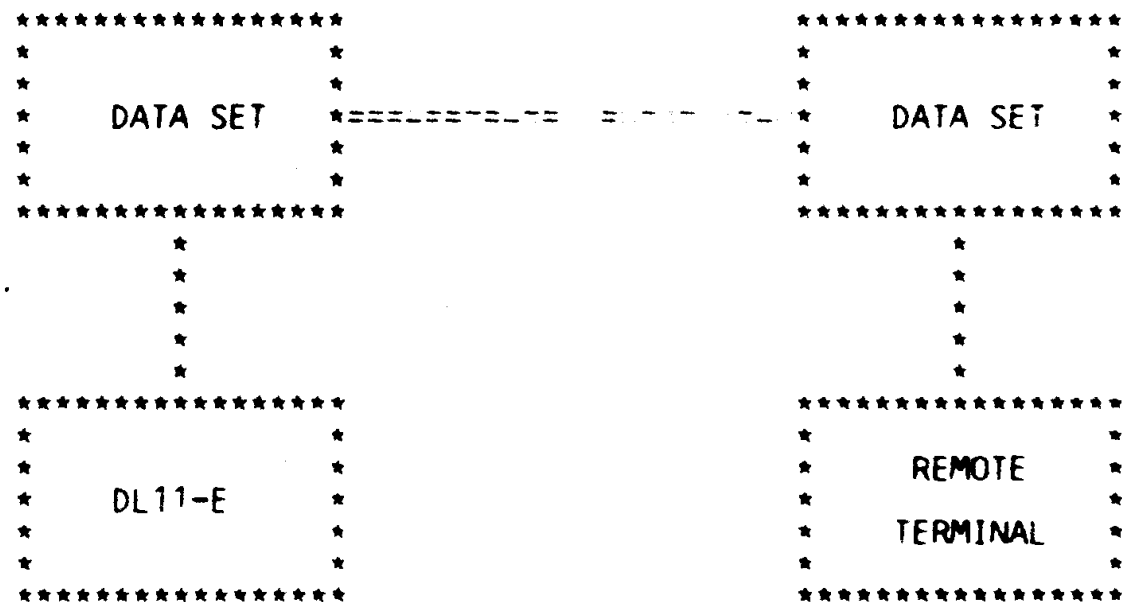
| | | |
|------|---|-------------------------------|
| SW15 | 1 | HALT ON ERROR |
| SW14 | 1 | SCOPE LOOP (NOT USED) |
| SW13 | 1 | INHIBIT ERROR PRINTOUT |
| SW12 | 1 | INHIBIT TRACE TRAP (NOT USED) |
| SW11 | 1 | INHIBIT ITERATION (NOT USED) |

123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178

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 PDP11 THE PROGRAM WILL REQUEST THE CONFIGURATION. SET BITS0-1=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.

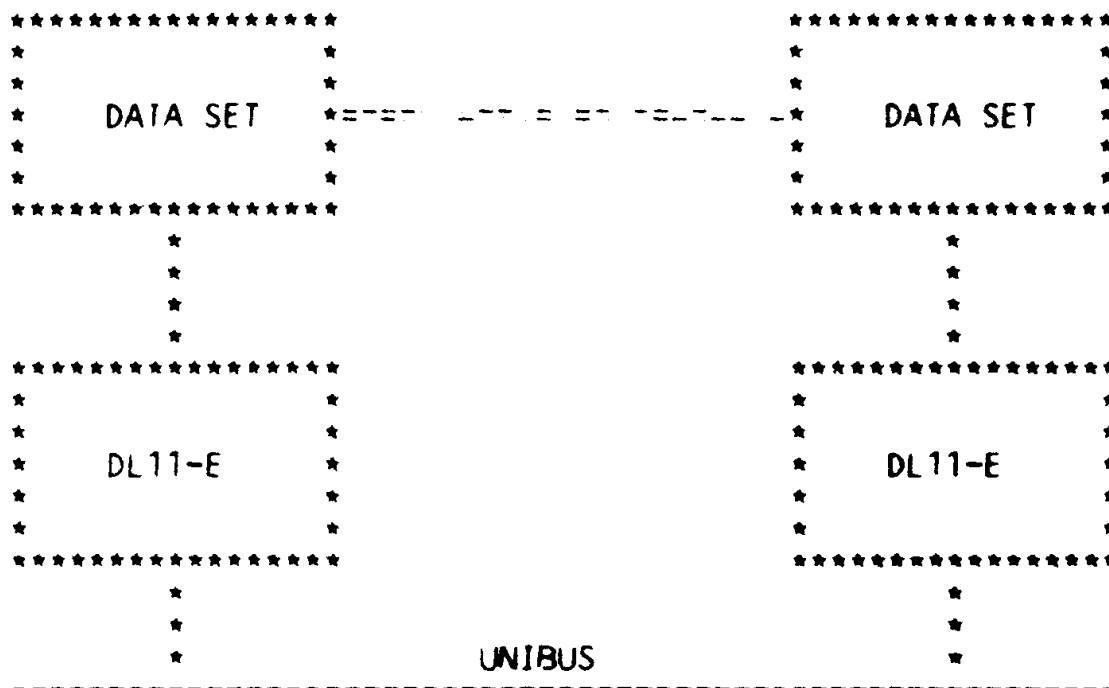


CONFIGURATION 0

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 PDP11 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

179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216

IS HEARD IN THE HEADSET PRESS THE DATA BUTTON ON THE DATA SET.
YOU HAVE APPROXIMATELY 10 SECONDS IN WHICH TO DO THIS. 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. WHEN 100. CHARACTERS HAVE BEEN
PROCESSED (TRANSMITTED/RECEIVED AND CHECKED) THE BELL WILL
RING AT THE TTY, AND ANOTHER 100. CHARACTER BLOCK WILL BE
PROCESSED.
NOTE, DL11-E#X REFERS TO THE 'CALLED' DL11-E, AND DL11-E#Y RE-
FERS TO THE CALLING DL11-E.



CONFIGURATION 1 103 OR 202
CONFIGURATION 2 103 ONLY

217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253

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 -10 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 | |

254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309

4. USE PROCEDURE

THIS PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT
A CONSOLE PROCESSOR.

IF A CONSOLE MACHINE IS USED; THEN THE PROGRAM
LOOKS AT THE HARDWARE SWITCH REGISTER.
IF A CONSOLE-LESS MACHINE IS USED; THEN THE PROGRAM
AUTOMATICALLY LOOKS AT THE CONTENTS OF LOCATION
SOFTSR (176) AS A SWITCH REGISTER.

IF A HARDWARE SWITCH REGISTER DOES NOT EXIST, THE PROGRAM WILL
USE THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES.
THE PROGRAM WILL PRINT OUT THE PRESENT CONTENTS OF THE SOFTWARE
SWITCH REGISTER WHEN THE PROGRAM IS STARTED. IT WILL THEN ASK
FOR THE NEW CONTENTS TO BE INPUT TO THE SOFTWARE SWITCH REGISTER.
TYPE CARRIAGE RETURN TO FINISH INPUT.

4.1 PRG0 SINGLE CHARACTER LINE MODE DATA TEST

- A. LOAD ADDRESS = 000200 (RESTART LOAD ADDR. = 000200)
- B. START - PROGRAM WILL REQUEST PROGRAM NUMBER
- C. THE PROGRAM WILL NOW REQUEST THE DATA. TYPE THE DATA IN OCTAL
FOLLOWED BY A CARRIAGE RETURN.
- D. MAKE LINE CONNECTION. SEE SECT 3.4

4.2 PRG1 - BINARY COUNT LINE MODE DATA TEST

- A. LOAD ADDRESS = 000200
- B. START - PROGRAM WILL REQUEST PROGRAM NUMBER
- C. MAKE LINE CONNECTION SEE SECT 3.4

4.3 PRG2 - SPECIAL MESSAGE XMIT ONLY

- A. LOAD ADDRESS = 000200
- B. START - PROGRAM WILL REQUEST PROGRAM NUMBER
- C. DEPRESS START - THE PROGRAM WILL IDENTIFY ITSELF AND
TYPE INSTRUCTIONS TO SELECT DESIRED DL11-E PARAMETERS

(SEE SECT 3.2)
- D. TYPE IN PARAMETERS. IF IT IS DESIRED TO TRANSMIT
DATA WITH PARITY SET BIT6. ALSO SET BITS TO TRANSMIT ODD
PARITY AND CLEAR TO TRANSMIT EVEN PARITY.

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

310
311
312
313
314
315

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 THE MESSAGE
'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK
0123456789' WILL BE TRANSMITTED. TO TERMINATE, HANG UP.

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
358
359
360
361
362
363

- 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.

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
391
392
393
394
395
396

- 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.

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
440
441
442
443
444
445
446
447
448
449
450
451
452

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 TYPEOUTS 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 CALLING DL11-E (CALLER) WITH THE DATA TRANSMITTED BY THE CALLED DL11-E. IF FOR EXAMPLE A DATA ERROR OCCURED 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.

- ***** ECO HISTORY *****
CHGC1 - NEW STORAGE LOCATIONS FOR REFERENCE IN ^G MODE
CHGC2 - CHANGE START UP FOR ENTERING SOFTWARE SWITCH REGISTER INFO.
CHGC3 - SOFTWARE STORAGE LOCATIONS
CHGC4 - ROUTINE TO CHECK PRINTOUT, UPDATE SOFTWARE SWITCH REG.

453
454
455
456

CHGC5 - NEW MESSAGES FOR SOFTWARE SWITCH REG ROUTINES.
CHGC6 - PART OF CHGC2. NECESSARY TO DIVIDE INTO TWO PARTS

.ENDR

| | | |
|-----|--------|---------------|
| 513 | 010000 | BIT12=10000 |
| 514 | 004000 | BIT11=4000 |
| 515 | 002000 | BIT10=2000 |
| 516 | 001000 | BIT9=1000 |
| 517 | 000400 | BIT8=400 |
| 518 | 000200 | BIT7=200 |
| 519 | 000100 | BIT6=100 |
| 520 | 000040 | BIT5=40 |
| 521 | 000020 | BIT4=20 |
| 522 | 000010 | BIT3=10 |
| 523 | 000004 | BIT2=4 |
| 524 | 000002 | BIT1=2 |
| 525 | 000001 | BIT0=1 |
| 526 | 005726 | POPSP=5726 |
| 527 | 022626 | POPSP2=022626 |
| 528 | 000340 | PRTY7=340 |
| 529 | 000300 | PRTY6=300 |
| 530 | 000240 | PRTY5=240 |
| 531 | 000200 | PRTY4=200 |
| 532 | 000140 | PRTY3=140 |
| 533 | 000100 | PRTY2=100 |
| 534 | 000040 | PRTY1=40 |
| 535 | 000000 | PRTY0=0 |
| 536 | 104000 | TYPE=EMT+0 |
| 537 | 104001 | TYPES=EMT+1 |
| 538 | 104002 | STALL=EMT+2 |
| 539 | 104003 | ERROR=EMT+3 |
| 540 | 104004 | DATCHK=EMT+4 |
| 541 | 104005 | CHALT=EMT+5 |
| 542 | 104006 | STRXV=EMT+6 |
| 543 | 104007 | STTXV=EMT+7 |
| 544 | 104010 | EHALT=EMT+10 |
| 545 | 104011 | SAVREG=EMT+11 |
| 546 | 104012 | RSTREG=EMT+12 |
| 547 | 104013 | ERROR1=EMT+13 |
| 548 | 104014 | ERRTX=EMT+14 |
| 549 | 104015 | ERRRX=EMT+15 |
| 550 | 104016 | DELAY=EMT+16 |
| 551 | 000000 | N=0 |
| 552 | 000000 | A=0 |

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

| | | | | | |
|-----|--|-------|----------|---------------|--------------------------------|
| 554 | | .MACR | CNVGA | SRC,DST,COUNT | |
| 555 | | JSR | %5,OACNV | | ;GO TO OCTAL TO ASCII CONVERT. |
| 556 | | SRC | | | ;SOURCE ADDR. |
| 557 | | DST | | | ;DESTINATION ADDR. |
| 558 | | COUNT | | | ;#OF DIGITS TO CONVERT. |
| 559 | | .ENDM | | | |

| | | | | | |
|-----|----------|-------|------|---------|---|
| 561 | | .MACR | ISR | N | |
| 562 | RISR'N': | MOV | | #'N',%0 | |
| 563 | | JMP | RISR | | ;GO TO COMMON INTERRUPT SERVICE ROUTINE |
| 564 | | | | | |
| 565 | | .ENDM | | | |

| | | | | | |
|-----|----------|-------|------|---------|-------------------|
| 567 | | .MACR | ISRT | N | |
| 568 | TISR'N': | MOV | | #'N',%0 | ;PUT LINE # IN RO |

```
569          JMP      TISR          ;GO TO COMMON INTERRUPT SERVICE
570
571          .ENDM
572
573          .MACR   RRCV      N,A
574          175610+N          ;ADDRESS OF RECEIVER LINE # 'A'
575          .ENDM
576
577          .MACR   RBUFF     N,A
578          175612+N          ;ADDRESS OF RECEIVER BUFFER LINE # 'A'
579          .ENDM
580
581          .MACR   TXMT      N,A
582          175614+N          ;ADDRESS OF TRANSMITTER CSR LINE # 'A'
583          .ENDM
584
585          .MACR   TBUF      N,A
586          175616+N          ;ADDRESS OF TRANSMIT BUFFER LINE # 'A'
587          .ENDM
588
589
590
591          ;***** CHGC3 *****
592
593
594          . =170
595          000170 000170          SRPTR:      177570          ;SOFT SW REG POINTER
596          000172 000172          DISPREG:    177570          ;DISPLAY POINTER
597          000174 000000          DISPLAY:    OPEN           ;SOFTWARE DISPLAY REGISTER
598          000176 000000          SWREG:      OPEN           ;SOFTWARE SWITCH REG
599
600
601          ;*****
602
603
604
605          . =200
606          000200 012737 001436 000312          MOV      #START,STAD          ;SET UP FOR NORMAL START
607          000206 000407                                BR      STCONT                ;CONTINUE
608          000210 012737 001432 000312          MOV      #REMAP,STAD          ;SET FOR A REMAP START
609          000216 000403                                BR      STCONT
610          000220 012737 002102 000312          MOV      #MODEV,STAD          ;SET TO MODIFY ON START
611          000226 012706 001200          STCONT:  MOV      #STKPTR,%6          ;SET BOTTOM OF STAC"
612          ;*****
613          000232 013746 000006          CHGC2:  MOV      6,-(SP)          ;SAVE CURRENT VECTOR
614          000236 013746 000004          MOV      4,-(SP)
615          000242 012737 000262 000004          MOV      #64$,@#4          ;SET UP LOC. 4 FOR NON-EXISTANT
616          ;MEMORY TRAP
617          000250 022777 177777 177712          CMP      #-1,@SRPTR          ;REFERENCE HARDWARE SWITCH REGISTER
618          000256 001402                                BEQ     65$                    ;IF IT = -1, USE SOFT SW REG
619          000260 000407                                BR      66$                    ;THEN USE HARDWARE SW REG
620          000262 022626          64$:    CMP      (SP)+,(SP)+          ;CORRECT THE STACK NO HDWE SW REG
621          000264 012737 000176 000170          65$:    MOV      #SWREG,SRPTR          ;POINT TO SOFT SW REG
622          000272 012737 000174 000172          MOV      #DISPLAY,DISPREG      ;POINT TO SOFT DISP REG
623          000300 012637 000004          66$:    MOV      (SP)+,@#4          ;RESTORE VECTORS
624          000304 012637 000006          MOV      (SP)+,@#6
```



```
625 :*****
626 000310 000137 :JMP @ (PC)+ ;JUMP TO SELECTED START
627 000312 000000 STAD: 0
628 001200 001200 :=1200
629 001200 000000 RXCSR: 0 ;RECEIVER CSR UNDER TEST
630 001202 000000 RXBUF: 0 ;RECEIVER BUFFER UNDER TEST
631 001204 000000 TXCSR: 0 ;TRANSMITTER CSR UNDER TEST
632 001206 000000 TXBUF: 0 ;TRANSMITTER BUFFER UNDER TEST
633 001210 000000 RXVTR: OPEN ;RECEIVER VECTOR
634 001212 000200 RXLVL: PRTY4 ;RECEIVER PRIORITY LEVEL
635 001214 000000 TXVTR: OPEN ;TRANSMITTER VECTOR
636 001216 000200 TXLVL: PRTY4 ;TRANSMITTER PRIORITY LEVEL
637 001220 177560 TKS: 177560 ;LSR CSR
638 001222 177562 TKB: 177562 ;LSR BUFFER
639 001224 177564 TPS: 177564 ;LSP CSR
640 001226 177566 TPB: 177566 ;LSP BUFFER
641 001230 000060 TKVTR: 60 ;LSR INTERRUPT VECTOR
642 001232 000200 TKLVL: PRTY4 ;LSR PRIORITY LEVEL
643 001234 000064 TPVTR: 64 ;LSP INTERRUPT VECTOR
644 001236 000200 TPLVL: PRTY4 ;LSP PRIORITY LEVEL
645 001240 000000 PRGNUM: OPEN ;CONTAINS CURRENT PROGRAM#
646 001242 000000 PRGID: OPEN ;CONTAINS PROGRAM INDICATORS
647 001244 006502 PRGTAB: PRG0 ;PRG0 START ADDRESS
648 001246 006616 PRG1 ;PRG1 START ADDRESS
649 001250 006720 PRG2 ;PRG2 START ADDRESS
650 001252 007016 PRG3 ;PRG3 START ADDRESS
651 001254 007552 PRG4 ;PRG4 START ADDRESS
652 001256 007764 PRG5 ;PRG5 START ADDRESS
653 001260 002432 INCPRG
654 001262 002432 INCPRG
655 001264 003200 EMTTAB: TYP ;POINTER TO TYPEOUT ROUTINE
656 001266 003322 TYP5 ;POINTER TO CHAINED MESSAGES ROUTINE
657 001270 000000 OPEN ;POINTER TO RANDOM STALL ROUTINE
658 001272 002570 ERR ;POINTER TO ERROR ROUTINE
659 001274 002470 DTCHK ;POINTER TO DATA CHECK ROUTINE
660 001276 002444 CHLT ;COMMON HALT
661 001300 003120 STRVRV ;POINTER TO ROUTINE TO SET RCVR VECTOR AND PRIORITY
662 001302 003150 STXMTV ;POINTER TO ROUTINE TO SET XMIT VECTOR AND PRIORITY
663 001304 002456 EHLT ;POINTER TO ERROR HALT ROUTINE.
664 001306 002332 SAVRG ;POINTER TO SAVE REGISTERS ROUTINE
665 001310 002372 RSTRG ;POINTER TO RESTORE REGISTERS ROUTINE
666 001312 002612 ERR1 ;POINTER TO ERROR ROUTINE
667 001314 002766 TXERR ;POINTER TO XMIT ERROR ROUTINE
668 001316 003010 RXERR ;POINTER TO RCVR ERROR ROUTINE
669 001320 003544 DLY ;POINTER TO DELAY ROUTINE
670 001322 000000 PARBIT: OPEN
671 001324 000000 COUNT: OPEN
672 001326 000000 SAVE: OPEN
673 001330 000000 LINE: OPEN
674 001332 000000 CONFIG: OPEN
675 001334 000000 NUMBER: OPEN
676
677 001336 000000 OLDPS: 0
678 001340 000000 TOPC: 0
679 001342 000000 FROMPC: 0
680 001344 000000 FTITLE: 0
```

681 001346 000000
682 001350 000000
683 001352 000000
684 001354 000000
685 001356 000000
686 001360 000000
687 001362 000000
688 001364 000000
689 001366 000000
690 001370 000000
691 001372 000000
692 001374 000000
693 001376 000000
694 001400 000000
695 001402 000000
696 001404 000000
697 001406 000000
698
699 001410 000000
700 001412 000000
701 001414 000000
702
703
704 001416 000000
705 001420 000000
706 001422 000000
707 001424 000000
708 001426 000000
709 001430 000000
710
711
712
713
714 001432 005037 001344
715 001436 012706 001200
716 001442 000005
717 001444 005037 177776
718 001450 012737 004576 000004
719 001456 012737 000040 000006
720 001464 005737 001344
721 001470 001145
722 001472 104000
723 001474 012576
724 001476 005237 001344
725 001502 005037 001346
726 001506 012737 177777 001354
727 001514 012737 001564 000004
728 001522 012737 000340 000006
729 001530 012704 011216
730 001534 005237 001354
731 001540 020427 011314
732 001544 001477
733 001546 005037 177776
734 001552 005774 000000
735 001556 000240
736 001560 000240

FNONE: 0
FMAP: 0
TEMP1: 0
LINENO: 0
RECDAT: OPEN
XMTDAT: OPEN
CARMSK: OPEN
CTRD: OPEN
TXCSRT: OPEN
RXCSRT: OPEN
TEMP: OPEN
SRT: OPEN
INBUFP: OPEN
BUFFP: OPEN
CALLER: OPEN
CALLED: OPEN
OTBUFP: OPEN

TBUFFP: OPEN
MODEM: OPEN
OPEN

***** CHGC1 *****
COUNT1: OPEN
CNT: OPEN
FILL1: OPEN
MODE: OPEN
TEMPST: OPEN
TIB: OPEN

;CONTAINS ADDRESS FROM WHERE NEXT TRAN-
SMITTED CHAR. (IN OUTBUF) IS TO COME

;CONTAINS MODEM TYPE 0=103,4-202
;CONTAINS ADDRESS FROM WHERE NEXT TRANS-
MITTED CHAR. (CALLER'S LINE) IS TO COME

;LOOP COUNT FOR TEST
;ITERATION COUNT
;ZERO FILL SWITCH
;NUMBER OF DIGITS TO TYPE
;TEMP WORK LOCATION
;TEMPORARY KYBD BUFFER STORAGE

REMAP: CLR FTITLE
START: MOV #STKPTR,%6
RESET
CLR PSW
MOV #ERTP,MACHER
MOV #40,MACHER+2
TST FTITLE
BNE START1
TYPE
MTITLE
INC FTITLE
CLR FNONE
MOV #-1,LINENO
MOV #MAPNE,MACHER
MOV #PRTY7,MACHER+2
MOV #RCSR,%4
MAPA: INC LINENO
CMP %4,#RBUF
BEQ MAPEND
CLR PSW
TST @(%4)
NOP
NOP

;SET BOTTOM OF SP STACK.

;TITLE PRINTED
;YES, SKIP THIS

;SET TITLE PRINTED FLAG
;CLEAR NO DEVICE FLAG

;SET UP FOR NO DEVICE ANSWER

;SET UP DEVICE POINTER

;LAST DEVICE
;YES

;TEST DEVICE

```
737 001562 000404          BR      MAPOK
738 001564 062704 000002    MAPNE:  ADD      #2,%4
739 001570 022626          POPSP2
740 001572 000760          BR      MAPA
741 001574 011437 001352    MAPOK:  MOV      (4),TEMP1      ;SAVE DEVICE FOR TYPING
742 001600 004537 003606    JSR      %5,OACNV
743 001604 001352          TEMP1
744 001606 012723          MDADR
745 001610 000006          6
746 001612 004537 003606    JSR      %5,OACNV
747 001616 001354          LINENO
748 001620 012715          MLINE
749 001622 000002          2
750 001624 011401          MOV      (4),%1      ;GET RXCSR DEVICE ADDRESS
751 001626 004737 004762    JSR      %7,FORMAD
752 001632 052737 000001 001350    BIS      #BIT0,FMAP
753 001640 042777 000100 177336    BIC      #BIT6,@TXCSR
754 001646 052777 000100 177330    BIS      #BIT6,@TXCSR
755 001654 000240          NOP
756 001656 012737 000340 177776    MOV      #PRTY7,PSW
757 001664 005737 001210    TST      RXVTR
758 001670 001406          BEQ      MAPOKA
759 001672 013701 001354    MOV      LINENO,%1
760 001676 006301          ASL      %1
761 001700 013761 001210 011120    MOV      RXVTR,VECTAB(1) ;STORE VECTOR
762 001706 042777 000100 177270    MAPOKA: BIC      #BIT6,@TXCSR
763 001714 004537 003606    JSR      %5,OACNV
764 001720 001210          RXVTR
765 001722 012735          MTRAP
766 001724 000004          4
767 001726 104000          TYPE
768 001730 012715          MLINE
769 001732 005237 001346    INC      FNONE
770 001736 062704 000002    ADD      #2,%4
771 001742 000674          BR      MAPA
772 001744 012737 004576 000004    MAPEND: MOV      #ERTP,MACHER
773 001752 012737 000040 000006    MOV      #40,MACHER+2
774 001760 005737 001346    TST      FNONE
775 001764 001007          BNE      START1
776 001766 104000          MAPERR: TYPE
777 001770 012745          MNONE
778 001772 005037 001344    CLR      FTITLE
779 001776 000000          HALT
780 002000 000137 001436    JMP      START
781                                     ;*****
782 002004          CHGC6:
783 002004 022737 000176 000170    START1: CMP      #SWREG,SRPTR      ;IS SOFT SWREG SELECTED
784 002012 001015          BNE      60$              ;IF NOT, BR OVER NEXT OPER
785 002014 004737 010464          JSR      PC,CNTLU        ;SOFT SW REG INPUT ROUTINE
786 002020 013701 001230          MOV      TKVTR,R1
787 002024 012721 010712          MOV      #TTINTS,(R1)+   ;VECTOR ADDRESS TO LOC 60
788 002030 013721 001232          MOV      TKLVL,(R1)+    ;PRIORITY TO LOC 62
789 002034 005777 177162          TST      @TKB           ;CLEAR DONE FLAG
790 002040 012777 000100 177152    MOV      #100,@TKS      ;SET TTY INTERRUPT ON
791                                     ;*****
792 002046 005037 177776    60$:   CLR      PSW
```

```

793 002052 104000          TYPE
794 002054 012762          MSWSEL
795 002056 004737 003372  JSR      PC,RDOCT      ;GET INPUT
796 002062 012600          MOV      (SP)+,%0      ;(SR) TO R0
797 002064 042700 177770  BIC      #177770,%0    ;LIMIT (SR) TO BITS 2-0
798 002070 010037 001240  MOV      %0,PRGNUM     ;SAVE PROGRAM #
799 002074 006300          ASL      %0            ;R0X2
800 002076 000170 001244  JMP      @PRGTAB(0)    ;GO TO SELECTED PROGRAM.
801
802 002102 012706 001200  ;MODEV: MOV      #STKPTR,%6
803 002106 000005          RESET
804 002110 005037 177776  CLR      PSW
805 002114 104000          MODEV1: TYPE
806 002116 015203          MMOD1
807 002120 004737 003372  JSR      PC,RDOCT      ;GET INPUT
808 002124 011600          MOV      (SP),%0
809 002126 042700 177740  BIC      #177740,%0
810 002132 006300          ASL      %0
811 002134 022627 000036  CMP      (SP)+,#36
812 002140 101403          BLOS     MODEV2        ;BRANCH IF > 36
813 002142 104000          TYPE
814 002144 015131          MMODX
815 002146 000762          BR       MODEV1
816 002150 104000          MODEV2: TYPE
817 002152 015321          MMOD2
818 002154 004737 003372  JSR      PC,RDOCT      ;GET INPUT
819 002160 032716 000001  BIT      #BIT0,(SP)
820 002164 001403          BEQ      MODEV3
821 002166 104000          TYPE
822 002170 015153          MMODD
823 002172 000766          BR       MODEV2
824 002174 012601          MODEV3: MOV      (SP)+,%1      ;SAVE DEV ADR
825 002176 010003          MOV      %0,%3
826 002200 062703 011216  ADD      #RCSR,%3
827 002204 010113          MOV      %1,(3)
828 002206 062701 000002  ADD      #2,%1        ;UPDATE DEV ADR
829 002212 010003          MOV      %0,%3
830 002214 062703 011314  ADD      #RBUF,%3
831 002220 010113          MOV      %1,(3)
832 002222 062701 000002  ADD      #2,%1        ;UPDATE DEV ADR
833 002226 010003          MOV      %0,%3
834 002230 062703 011412  ADD      #TCSR,%3
835 002234 010113          MOV      %1,(3)
836 002236 062701 000002  ADD      #2,%1        ;UPDATE DEV ADR
837 002242 010003          MOV      %0,%3
838 002244 062703 011510  ADD      #TBUF,%3
839 002250 010113          MOV      %1,(3)
840 002252 104000          TYPE
841 002254 015367          MMOD3
842 002256 004737 003372  JSR      PC,RDOCT      ;GET INPUT
843 002262 022627 177777  CMP      (SP)+,#177777
844 002266 001712          BEQ      MODEV1
845 002270 005037 001344  CLR      FTITLE
846 002274 000137 001436  JMP      START
847
848 ;EMT TRAP INTERPRETER

```

| | | | | | | |
|-----|--------|--------|--------|--------|----------------------------------|------------------------------------|
| 849 | 002300 | 011646 | | | EMTINT: MOV @%6,-(6) | :GET SAVED PC. |
| 850 | 002302 | 162716 | 000002 | | SUB #2,@%6 | :DECREMENT PC BY 2. |
| 851 | 002306 | 017616 | 000000 | | MOV @%6,@%6 | |
| 852 | 002312 | 006116 | | | EMTA: ROL @%6 | :EMT ARG X 2. |
| 853 | 002314 | 042716 | 177001 | | BIC #177001,@%6 | :REMOVE 7 MSB. |
| 854 | 002320 | 062716 | 001264 | | ADD #EMTTAB,@%6 | :FORM EMT RTN ADDR. |
| 855 | 002324 | 017616 | 000000 | | MOV @%6,@%6 | |
| 856 | 002330 | 000136 | | | JMP @%6+ | :GO TO EMT ROUTINE. |
| 857 | | | | | | |
| 858 | | | | | :SAVE REGS 0 TO 4 SUBROUTINE. | |
| 859 | 002332 | 012637 | 002366 | | SAVRG: MOV (6)+,SVRPC | :SAVE PC AND PSW. |
| 860 | 002336 | 012637 | 002370 | | MOV (6)+,SVRPSW | |
| 861 | 002342 | 010446 | | | MOV %4,-(6) | :SAVE REGS 0 - 4 |
| 862 | 002344 | 010346 | | | MOV %3,-(6) | :IN STACK. |
| 863 | 002346 | 010246 | | | MOV %2,-(6) | |
| 864 | 002350 | 010146 | | | MOV %1,-(6) | |
| 865 | 002352 | 010046 | | | MOV %0,-(6) | |
| 866 | 002354 | 013746 | 002370 | | MOV SVRPSW,-(6) | :RESTORE PC AND PSW. |
| 867 | 002360 | 013746 | 002366 | | MOV SVRPC,-(6) | |
| 868 | 002364 | 000002 | | | RTI | :EXIT. |
| 869 | 002366 | 000000 | | | SVRPC: OPEN | |
| 870 | 002370 | 000000 | | | SVRPSW: OPEN | |
| 871 | | | | | | |
| 872 | | | | | :RESTORE REGS 0 TO 4 SUBROUTINE. | |
| 873 | 002372 | 012637 | 002426 | | RSTRG: MOV (6)+,RSTPC | :SAVE PC AND PSW. |
| 874 | 002376 | 012637 | 002430 | | MOV (6)+,RSTPSW | |
| 875 | 002402 | 012600 | | | MOV (6)+,%0 | :RESTORE REGS 0 - 4 |
| 876 | 002404 | 012601 | | | MOV (6)+,%1 | :FROM STACK. |
| 877 | 002406 | 012602 | | | MOV (6)+,%2 | |
| 878 | 002410 | 012603 | | | MOV (6)+,%3 | |
| 879 | 002412 | 012604 | | | MOV (6)+,%4 | |
| 880 | 002414 | 013746 | 002430 | | MOV RSTPSW,-(6) | :RESTORE PC AND PSW. |
| 881 | 002420 | 013746 | 002426 | | MOV RSTPC,-(6) | |
| 882 | 002424 | 000002 | | | RTI | :EXIT |
| 883 | 002426 | 000000 | | | RSTPC: OPEN | |
| 884 | 002430 | 000000 | | | RSTPSW: OPEN | |
| 885 | 002432 | 104000 | | | INCPRG: TYPE | :TYPE INCORRECT PROGRAM SELECTED. |
| 886 | 002434 | 014121 | | | AINPRG | |
| 887 | 002436 | 000000 | | | HALT | |
| 888 | 002440 | 000137 | 001436 | | JMP START | |
| 889 | | | | | :COMMON HALT ROUTINE | |
| 890 | 002444 | 011600 | | | CHLT: MO' @%6,%0 | :DEVELOP ADDRESS OF CALLER. |
| 891 | 002446 | 162700 | 000002 | | SUB #2,%0 | |
| 892 | 002452 | 000000 | | | HALT | :HALT. ADDRESS OF CALL INSTRUCTION |
| 893 | 002454 | 000002 | | | RTI | :IN DATA LIGHTS. |
| 894 | | | | | | |
| 895 | | | | | :CONDITIONAL ERROR HALT ROUTINE. | |
| 896 | 002456 | 005777 | 175506 | | EHLT: TST @SRPTR | :CHECK FOR HALT ON ERROR. |
| 897 | 002462 | 100001 | | | BPL EHLTA | :BRANCH IF NO HALT DESIRED. |
| 898 | 002464 | 000000 | | | HALT | :HALT. |
| 899 | 002466 | 000002 | | | EHLTA: RTI | :IN DATA LIGHTS. |
| 900 | | | | | | |
| 901 | | | | | :DATA CHECK ROUTINE. | |
| 902 | 002470 | 043737 | 001362 | 001360 | DTCHK: BIC CARMASK,XMTDAT | :CLEAR UNTRANSMITTED BITS |
| 903 | 002476 | 123737 | 001356 | 001360 | CMPB RECDAT,XMTDAT | :COMPARE TRANSMITTED AND RECEIVED |
| 904 | 002504 | 001430 | | | BEQ DTCHKA | :CHARS. BRANCH IF SAME. |

```
905 002506 CNVOA RECDAT,CWAS,3
906 002520 CNVOA XMTDAT,CSB,3
907 002532 CNVOA RXCSR,CSRADD,6
908 002544 104013 ERROR1
909 002546 013126 CSRADD
910 002550 004537 004274 JSR 5,BDCNV ;CONVERT
911 002554 001364 CTRD ;CHAR #
912 002556 014112 CRNUM ;TO DECIMAL
913 002560 000004 4 ;4 BITS
914 002562 104013 ERROR1
915 002564 014047 CERDAT
916 002566 000002 DTCHKA: RTI ;EXIT.
917
918 002570 012737 177777 002702 ERR: MOV #-1,ERRB ;SET UP ONE MESSAGE CALL.
919 002576 012737 000240 002704 MOV #240,ERRB+2
920 002604 005037 002722 CLR ERRE
921 002610 000413 BR ERRA
922 002612 011637 002702 ERR1: MOV @%6,ERRB ;DEVELOP ADDT'L MESSAGE ADDR.
923 002616 017737 000060 002702 MOV @ERRB,ERRB ;STORE AT ERRB.
924 002624 012737 177777 002704 MOV #-1,ERRB+2
925 002632 012737 000002 002722 MOV #2,ERRE
926 002640 032777 020000 175322 ERRA: BIT #BIT13,@SRPTR ;INHIBIT ERROR PRINT?
927 002646 001020 BNE ERRC ;BRANCH TO INHIBIT PRINT.
928 002650 011637 002720 MOV @%6,ERRD ;DEVELOP CALLING ADDR.
929 002654 162737 000002 002720 SUB #2,ERRD
930 002662 CNVOA ERRD,APC,6 ;CONVERT CALL ADDR TO ASCII.
931 002674 104011 SAVREG
932 002676 104001 TYPES
933 002700 013176 EMO
934 002702 000000 ERRB: OPEN ;ERROR HEADER,
935 002704 177777 -1 ;ADDT'L ERROR MESSAGE IF ANY.
936 002706 104012 RSTREG
937 002710 104010 ERR(: EHALT ;GO ERR HALT IF DESIRED.
938 002712 063716 002722 ADD ERRE,@%6
939 002716 000002 RTI ;EXIT.
940 002720 000000 ERRD: OPEN
941 002722 000000 ERRE: OPEN
942
943 ;POWER FAIL SERVICE
944 002724 012737 002734 000024 PFAIL: MOV #PWRUP,@#24
945 002732 000000 HALT
946 002734 012737 002724 000024 PWRUP: MOV #PFAIL,@#24
947 002742 000005 RESET
948 002744 012706 001200 MOV #STKPTR,%6
949 002750 104000 TYPE
950 002752 015073 MPWRF
951 002754 013700 001240 RESTRT: MOV PRGNUM,%0
952 002760 006300 ASL %0
953 002762 000170 001244 JMP @PRGTAB(0)
954
955 002766 TXERR: CNVOA TXCSRT,ATXWAS,6 ;CONVERT CONTENTS OF TXCSR TO ASCII.
956 003000 012737 013214 003104 MOV #ATXCSR,CRXTXB
957 003006 000410 BR CRXTX
958 003010 RXERR: CNVOA RXCSRT,ARXWAS,6 ;CONVERT CONTENTS OF RXCSR TO ASCII.
959 003022 012737 013235 003104 MOV #ARXCSR,CRXTXB
960 003030 011637 003102 CRXTX: MOV @%6,CRXTXA ;DEVELOP ADDR OF ADDTT'L ERROR MESSAGE.
```

```
961 003034 017737 000042 003102 MOV @CRXTXA,CRXTXA
962 003042 032777 020000 175120 BIT #BIT13,@SRPTR ;INHIBIT PRINT?
963 003050 001017 BNE CRXTXC ;BRANCH TO INHIBIT PRINT.
964 003052 011637 002720 MOV @%6,ERRD ;DEVELOP CALLING ADDR.
965 003056 162737 000002 002720 SUB #2,ERRD
966 003064 CNVOA ERRD,APC,6 ;CONVERT CALLING ADDR TO ASCII.
967 003076 104001 TYPES ;TYPE ERROR MESSAGE.
968 003100 013176 EMO ;ERR HEADER
969 003102 000000 CRXTXA: OPEN ;ADDT'L ERR MESSAGE
970 003104 000000 CRXTXB: OPEN ;TXCSR OR RXCSR CONTENTS.
971 003106 177777 -1
972 003110 104010 CRXTXC: EHALT ;GO HALT IF DESIRED.
973 003112 062716 000002 ADD #2,@%6
974 003116 000002 RTI ;EXIT.
975
976 ;ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
977 003120 017637 000000 003140 STRVRV: MOV @6),STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2
978 003126 062716 000002 ADD #2,@%6 ;SET UP EXIT
979 003132 013701 001210 MOV RXVTR,%1
980 003136 012721 000000 STPRA: MOV #OPEN,(1)+ ;SET VECTOR ADDRESS
981 003142 013721 001212 MOV RXLVL,(1)+ ;SET PRIORITY
982 003146 000002 RTI ;EXIT
983
984 ;ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
985 003150 017637 000000 003170 STXMTV: MOV @6),STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
986 003156 062716 000002 ADD #2,@%6 ;SET UP EXIT
987 003162 013701 001214 MOV TXVTR,%1
988 003166 012721 000000 STPPA: MOV #OPEN,(1)+ ;SET VECTOR ADDRESS.
989 003172 013721 001216 MOV TXLVL,(1)+ ;SET PRIORITY
990 003176 000002 RTI ;EXIT.
991
992 ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
993 003200 010037 003370 TYP: MOV %0,SAVRO ;SAVE RO
994 003204 011600 MOV @%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
995 003206 062716 000002 ADD #2,@%6 ;SET UP EXIT.
996 003212 011000 MOV @%0,%0 ;ADDRESS OF MESSAGE TO RO.
997 003214 112037 003320 TYFA: MOV (0)+,TYPDAT ;GET CHARACTER
998 003220 122737 000100 003320 CMPB #100,TYPDAT ;CHECK FOR '@' CHARACTER
999 003226 001003 BNE TYPC ;BRANCH IF NOT '@'.
1000 003230 013700 003370 MOV SAVRO,%0 ;RESTORE RO
1001 003234 000002 RTI ;TERMINATOR CHAR. DONE. EXIT.
1002 003236 122737 000045 003320 TYPC: CMPB #45,TYPDAT ;CHECK FOR '%'.
1003 003244 001412 BEQ TYPF ;BRANCH IF '%'.
1004 003246 004737 003254 JSR %7,TYPD ;TYPE CHAR IN TYPDAT
1005 003252 000760 BR TYPA
1006 003254 113777 003320 175744 TYPD: MOVB TYPDAT,@TPB ;OUTPUT CHARACTER TO PRINTER
1007 003262 105777 175736 TSTB @TPS ;WAIT FOR DONE FLAG.
1008 003266 100375 BPL -4
1009 003270 000207 RTS %7 ;EXIT
1010 003272 112737 000015 003320 TYPF: MOVB #15,TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1011 003300 004737 003254 JSR %7,TYPD ;GO TYPE CHAR.
1012 003304 112737 000012 003320 MOVB #12,TYPDAT ;MOVE LF CODE TO TYPDAT.
1013 003312 004737 003254 JSR %7,TYPD ;GO TYPE CHAR.
1014 003316 000736 BR TYPA
1015 003320 000000 TYPDAT: OPEN
1016 ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
```

| | | | | | | | |
|------|--------|--------|--------|---------|-------|------------|--|
| 1017 | 003322 | 010037 | 003370 | TYP5: | MOV | %0, SAVRO | |
| 1018 | 003326 | 011600 | | TYP5AA: | MOV | @%6,%0 | ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS |
| 1019 | 003330 | 062716 | 000002 | | ADD | #2,@%6 | ;UPDATE TO NEXT MESSAGE ADDRESS |
| 1020 | 003334 | 011037 | 003364 | | MOV | @%0,TYP5B | ;ADDRESS OF MESSAGE TO TYP5B |
| 1021 | 003340 | 022737 | 177777 | 003364 | CMF | #-1,TYP5B | ;CHECK FOR TERMINATOR |
| 1022 | 003346 | 001003 | | | BNE | TYP5A | ;BRANCH IF NOT TERMINATOR. |
| 1023 | 003350 | 013700 | 003370 | | MOV | SAVRO,%0 | ;RESTORE R0 |
| 1024 | 003354 | 000002 | | | RTI | | ;TERMINATOR, EXIT |
| 1025 | 003356 | 013700 | 003370 | TYP5A: | MOV | SAVRO,%0 | |
| 1026 | 003362 | 104000 | | | TYPE | | ;CALL ON TYP SUB TO TYPF MESSAGE |
| 1027 | 003364 | 000000 | | TYP5B: | OPEN | | ;ADDRESS OF MESSAGE GOES HERE |
| 1028 | 003366 | 000757 | | | BR | TYP5AA | ;GO PROCESS NEXT MESSAGE |
| 1029 | 003370 | 000000 | | SAVRO: | OPEN | | |
| 1030 | | | | | | | |
| 1031 | | | | | | | ;SUBROUTINE TO READ OCTAL DATA FROM THE TELETYPE PRINTER |
| 1032 | 003372 | 011646 | | RDOCT: | MOV | (SP),-(SP) | ;MAKE ROOM FOR DATA WORD |
| 1033 | 003374 | 010046 | | | MOV | %0,-(SP) | ;SAVE R0 |
| 1034 | 003376 | 010146 | | | MOV | %1,-(SP) | ;SAVE R1 |
| 1035 | 003400 | 005001 | | INDAT: | CLR | %1 | ;CLEAR DATA WORD |
| 1036 | 003402 | 005037 | 001324 | | CLR | COUNT | ;SET NO. OF DIGITS - 0 |
| 1037 | 003406 | 105777 | 175606 | RDDAT: | TSTR | @TKS | ;TEST TTY READ STATUS |
| 1038 | 003412 | 100375 | | | BPL | RDDAT | ;WAIT |
| 1039 | 003414 | 117746 | 175602 | | MOVB | @TKB,-(SP) | ;PUSH DIGIT ON STACK |
| 1040 | 003420 | 042716 | 000200 | | BIC | #BIT7,(SP) | |
| 1041 | 003424 | 105777 | 175574 | ECDAT: | TSTB | @TPS | ;TEST TTY PRINT STATUS |
| 1042 | 003430 | 100375 | | | BPL | ECDAT | ;WAIT |
| 1043 | 003432 | 111677 | 175570 | | MOVB | (SP),@TPB | ;ECHO CHARACTER |
| 1044 | 003436 | 122716 | 000015 | | CMPB | #15,(SP) | ;IS IT A TERMINATOR? |
| 1045 | 003442 | 001432 | | | BEQ | RETRN | ;BR IF YES |
| 1046 | 003444 | 122716 | 000177 | | CMPB | #177,(SP) | ;IS IT A RUBOUT? |
| 1047 | 003450 | 001423 | | | BEQ | RREAD | ;BR IF YES |
| 1048 | 003452 | 122716 | 000060 | | CMPB | #60,(SP) | ;IS IT AN OCTAL DIGIT? |
| 1049 | 003456 | 003020 | | | BGT | RREAD | ;BR IF NO |
| 1050 | 003460 | 122716 | 000067 | | CMPB | #67,(SP) | ;TEST AGAIN |
| 1051 | 003464 | 002415 | | | BLT | RREAD | ;BR IF NO |
| 1052 | 003466 | 005237 | 001324 | | INC | COUNT | ;INC NO. OF DIGITS |
| 1053 | 003472 | 022737 | 000067 | 001324 | CMF | #67,COUNT | ;MORE THAN SIX DIGITS? |
| 1054 | 003500 | 003407 | | | BLE | RREAD | ;BR IF YES |
| 1055 | 003502 | 006301 | | | ASL | %1 | ;CLEAR LOWEST THREE BITS |
| 1056 | 003504 | 006301 | | | ASL | %1 | ;OF DATA WORD |
| 1057 | 003506 | 006301 | | | ASL | %1 | |
| 1058 | 003510 | 162716 | 000060 | | SUB | #60,(SP) | ;CONVERT TO BINARY |
| 1059 | 003514 | 062601 | | | ADD | (SP)+,%1 | ;ADD DIGIT TO DATA WORD |
| 1060 | 003516 | 000733 | | | BR | RDDAT | ;GET NEXT DIGIT |
| 1061 | 003520 | 104000 | | RREAD: | TYPE | | ;TELL USER ABOUT ILLEGAL CHARACTER |
| 1062 | 003522 | 014001 | | | DTERR | | |
| 1063 | 003524 | 005726 | | | TST | (SP)+ | ;GET RID OF ILLEGAL CHARACTER |
| 1064 | 003526 | 000724 | | | BR | INDAT | ;START SUBROUTINE AGAIN |
| 1065 | 003530 | 010166 | 000010 | RETRN: | MOV | %1,10(SP) | ;STORE DATA WORD ON STACK |
| 1066 | 003534 | 005726 | | | TST | (SP)+ | ;INC STACK POINTER |
| 1067 | 003536 | 012601 | | | MOV | (SP)+,%1 | ;RESTORE R1 |
| 1068 | 003540 | 012600 | | | MOV | (SP)+,%0 | ;RESTORE R0 |
| 1069 | 003542 | 000207 | | | RTS | PC | ;RETURN |
| 1070 | | | | | | | |
| 1071 | | | | | | | ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS |
| 1072 | 003544 | 011637 | 003604 | DLY: | MOV | @%6,DLCNT | ;GET DELAY COUNT ADDRESS. |

| | | | | | | | |
|------|--------|--------|--------|--------|------|-------------|---------------------------------------|
| 1073 | 003550 | 062716 | 000002 | | ADD | #2,@%6 | :SET UP EXIT ADDRESS |
| 1074 | 003554 | 017746 | 000024 | | MOV | @DLCNT,-(6) | :DELAY COUNT TO STACK |
| 1075 | 003560 | 001407 | | | BEQ | DLYC | |
| 1076 | 003562 | 012746 | 000226 | DLYA: | MOV | #226,-(6) | :1 MSEC COUNT TO STACK |
| 1077 | 003566 | 005316 | | DLYB: | DEC | @%6 | :DECREMENT 1 MSEC COUNT |
| 1078 | 003570 | 001376 | | | BNE | DLYB | :BRANCH IF NOT 0. |
| 1079 | 003572 | 005726 | | | POPS | | :ZERO. UNCOVER MSECS. COUNT. |
| 1080 | 003574 | 005316 | | | DEC | @%6 | :DECREMENT IT |
| 1081 | 003576 | 001371 | | | BNE | DLYA | :BR IF NOT DONE DELAYING |
| 1082 | 003600 | 005726 | | DLYC: | POPS | | :DONE |
| 1083 | 003602 | 000002 | | | RTI | | :EXIT. |
| 1084 | 003604 | 000000 | | DLCNT: | OPEN | | :CONTAINS MILLISECONDS COUNT ADDRESS. |
| 1085 | | | | | | | |
| 1086 | | | | | | | |
| 1087 | 003606 | 104011 | | | | | |
| 1088 | 003610 | 013500 | | | | | |
| 1089 | 003612 | 012501 | | | | | |
| 1090 | 003614 | 012502 | | | | | |
| 1091 | 003616 | 060201 | | | | | |
| 1092 | 003620 | 010003 | | | | | |
| 1093 | 003622 | 042703 | 177770 | | | | |
| 1094 | 003626 | 062703 | 000060 | | | | |
| 1095 | 003632 | 110341 | | | | | |
| 1096 | 003634 | 042700 | 000007 | | | | |
| 1097 | 003640 | 006000 | | | | | |
| 1098 | 003642 | 006000 | | | | | |
| 1099 | 003644 | 006000 | | | | | |
| 1100 | 003646 | 005302 | | | | | |
| 1101 | 003650 | 001363 | | | | | |
| 1102 | 003652 | 104012 | | | | | |
| 1103 | 003654 | 000205 | | | | | |
| 1104 | | | | | | | |
| 1105 | | | | | | | |
| 1106 | | | | | | | |
| 1107 | | | | | | | |

OCTAL TO ASCII CONVERT ROUTINE

OACNV: SAVREG ;SAVE REGS.
MOV @ (5)+,%0 ;GET OCTAL VALUE.
MOV (5)+,%1 ;GET DESTINATION ADDR.
MOV (5)+,%2 ;GET CONVERT COUNT.
ADD %2,%1 ;DEVELOP ADDR TO STORE 1ST CHAR.
OACNVA: MOV %0,%3
BIC #177770,%3 ;ISOLATE LEAST SIGNIFICANT DIGIT.
ADD #60,%3 ;CONVERT DIGIT TO ASCII.
MOVB %3,-(1) ;STORE ASCII CHARACTER.
BIC #7,%0
ROR %0
ROR %0
ROR %0
DEC %2 ;DONE ALL DIGITS?
BNE OACNVA ;BRANCH IF NOT DONE.
RSTREG ;RESTORE REGS.
RTS %5 ;DONE. EXIT.

:SUBROUTINE TO GENERATE PARITY ON DATA FOR 5,6,7,8 LEVEL CODE.
:PARITY BIT IS THE MSB OF THE CHARACTER PARITY CAN BE EITHER
:EVEN OR ODD
:GENERATES ODD/EVEN PARITY.

```

1108
1109 003656 032737 000200 001362 GENPAR: BIT #BIT7,CARMSK ;TEST LSB CHAR LENGTH
1110 003664 001411 BEQ EIGHT ;CHAR IS 8
1111 003666 032737 000100 001362 BIT #BIT6,CARMSK ;TEST MSB CHAR LENGTH
1112 003674 001427 BEQ SEVEN ;CHAR LENGTH IS 7
1113 003676 032737 000040 001362 BIT #BIT5,CARMSK
1114 003704 001412 BEQ SIX
1115 003706 000433 BR FIVE
1116 003710 012737 000200 001322 EIGHT: MOV #BIT7,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1117 003716 012737 000007 001324 MOV #7,COUNT ;SET UP ROTATE COUNTER=7
1118 003724 042701 177600 BIC #177600,%1 ;MASK OFF UNUSED BITS
1119 003730 000433 BR DOIT ;GO AND GENERATE PARITY FOR 8
1120 003732 012737 000040 001322 SIX: MOV #BIT5,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1121 003740 012737 000005 001324 MOV #5,COUNT ;SET UP ROTATE COUNTER=5
1122 003746 042701 177740 BIC #177740,%1 ;MASK OFF UNUSED BITS
1123 003752 000422 BR DOIT ;GO AND GENERATE PARITY FOR
1124 003754 012737 000100 001322 SEVEN: MOV #BIT6,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1125 003762 012737 000006 001324 MOV #6,COUNT ;SET UP ROTATE COUNTER=6
1126 003770 042701 177700 BIC #177700,%1 ;MASK OFF UNUSED BITS
1127 003774 000411 BR DOIT ;GO AND GENERATE PARITY FOR 7
1128 003776 012737 000020 001322 FIVE: MOV #BIT4,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1129 004004 012737 000004 001324 MOV #4,COUNT ;SET UP ROTATE COUNTER=4
1130 004012 042701 177760 BIC #177760,%1 ;MASK OFF UNUSED BITS
1131 004016 000400 BR DOIT ;GO AND GENERATE PARITY FOR
1132 004020 010137 001326 DOIT: MOV %1,SAVE ;SAVE DATA
1133 004024 006001 AGAIN: ROR %1 ;ROTATE DATA
1134 004026 103415 BCS ADD1 ;IF CARRY SET ADD IN PARBIT
1135 004030 005337 001324 RTN: DEC COUNT ;DECREMENT COUNTER
1136 004034 001373 BNE AGAIN ;NOT DONE DO IT AGAIN
1137 004036 032737 000040 001374 BIT #BIT5,SRT ;DONE EVEN OR ODD PARITY?
1138 004044 001403 BEQ DONE ;IF EVEN FINISHED
1139 004046 063737 001322 001326 ADD PARBIT,SAVE ;IF ODD ADD IN ANOTHER 1
1140 004054 013701 001326 DONE: MOV SAVE,%1 ;PLACE DATA + PARITY BACK IN R1
1141 004060 000207 RTS 7 ;AND EXIT
1142 004062 063737 001322 001326 ADD1: ADD PARBIT,SAVE ;ADD PARBIT TO DATA
1143 004070 000757 BR RTN ;RETURN TO COUNTER
1144
1145 ;SUBROUTINE TO SELECT LINE AND LOAD VECTOR ASSIGNMENT
1146 004072 104000 LINSSEL: TYPE
1147 004074 015024 LDLINE
1148 004076 004737 003372 JSR PC,RDOCT ;GET INPUT
1149 004102 012601 MOV (SP)+,%1 ;LOAD R1
1150 004104 042701 177407 BIC #177407,%1 ;MASK OFF ALL BUT LINE BITS
1151 004110 006201 ASR %1
1152 004112 006201 ASR %1
1153 004114 010137 001372 MOV %1,TEMP ;SAVE LINE #
1154 004120 012703 001200 MOV #RXCSR,%3 ;LOAD ADDRESS OF REGISTERS
1155 004124 012704 000004 MOV #4,%4 ;SET UP COUNTER
1156 004130 016102 011216 MOV RCSR(1),%2
1157 004134 010223 LINSA: MOV %2,(3)+
1158 004136 062702 000002 ADD #2,%2
1159 004142 005304 DEC %4
1160 004144 001373 BNE LINSA
1161 004146 016101 011120 MOV VECTAB(1),%1 ;GET LINE VECTOR ADDRESS
1162 004152 010123 MOV %1,(3)+ ;LOAD INTO PROG. RXVTR
1163 004154 022121 CMP (1)+,(1)+ ;ADD +4 TO RXVTR TO = TXVTR
  
```

```

1164 004156 005723          TST      (3)+          ;POINT TO PROG TXVTR
1165 004160 010113          MOV      %1,(3)        ;LOAD INTO PROG TXVTR
1166 004162 022737 000005 001240  CMP      #5,PRGNUM     ;RUNNING PROGRAM # 5
1167 004170 001001          BNE     .+4            ;
1168 004172 000205          RTS      5             ;RETURN TO PROG 5
1169 004174 006237 001372  ASR      TEMP          ;POSITION
1170 004200          CNVOA   TEMP,TLINEX,2 ;
1171 004212 104000          TYPE     ;TYPE LINE # THAT
1172 004214 014433          ALINEX  ;WAS CALLED
1173 004216 000205          RTS      5
1174
1175          ;SUBROUTINE TO LOAD BINARY COUNT PATTERN INTO OUTPUT BUFFER
1176 004220 105037 001334  INFIL:  CLRB   NUMBER   ;INITIALIZE BINARY COUNT
1177 004224 012500          FILL:   MOV    (5)+,%0  ;GET ADDRESS
1178 004226 012537 001364  FILL:   MOV    (5)+,CTRD ;GET COUNT
1179 004232 113720 001334  FILLA:  MOVB   NUMBER,(0)+ ;LOAD ADDRESS WITH BINARY COUNT
1180 004236 105237 001334  FILLA:  INCB   NUMBER    ;INC. BINARY COUNT
1181 004242 005337 001364  FILLA:  DEC    CT RD     ;DEC. COUNT
1182 004246 001371          FILLA:  BNE    FILLA    ;
1183 004250 000205          FILLA:  RTS      5      ;EXIT
1184
1185          ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1186 004252 104011  BMOVE:  SAVREG          ;SAVE REGS.
1187 004254 012501          BMOVE:  MOV    (5)+,%1   ;GET FROM ADDRESS
1188 004256 012502          BMOVE:  MOV    (5)+,%2   ;GET TO ADDRESS
1189 004260 012503          BMOVE:  MOV    (5)+,%3   ;GET COUNT
1190 004262 112122  BMOVA:  MOVB   (1)+,(2)+ ;MOVE BYTE
1191 004264 005303          BMOVA:  DEC    %3        ;DECREMENT COUNT
1192 004266 001375          BMOVA:  BNE    BMOVA    ;BRANCH IF NOT DONE.
1193 004270 104012          BMOVA:  RSTREG          ;RESTORE REGS.
1194 004272 000205          BMOVA:  RTS      %5     ;DONE EXIT
1195
1196          ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
1197 004274 104011  BDCNV:  SAVREG          ;SAVE REGS.
1198 004276 012700 004452  BDCNV:  MOV    #DECVAL,%0 ;SET UP ADDR TO STORE DECIMAL ASCII IN RO
1199 004302 013501          BDCNV:  MOV    @(%5)+,%1  ;BINARY VALUE TO R1.
1200 004304 012537 004362  BDCNV:  MOV    (5)+,BDCNVC ;DESTINATION ADDR TO BDCNVC.
1201 004310 012537 004364  BDCNV:  MOV    (5)+,BDCNVD ;COUNT TO BDCNVD.
1202 004314 012702 004440  BDCNV:  MOV    #ADTENP,%2  ;ADDR OF TEN POWER STRING TO R2.
1203 004320 012737 000005 004432  BDCNV:  MOV    #5,CNVCTR   ;SET UP FOR 5 POWER CONVERSIONS.
1204 004326 012237 004436  BDCNVA:  MOV    (2)+,TENPWR ;MOVE POWER OF TFN VALUE TO TENPWR.
1205 004332 004737 004372  BDCNVA:  JSR    %7,SUBTEN  ;PERFORM CONVERSION
1206 004336 005337 004432  BDCNVA:  DEC    CNVCTR     ;DONE 5 CONVERSIONS?
1207 004342 001371          BDCNVA:  BNE    BDCNVA   ;BRANCH IF NOT YET 5.
1208 004344 163700 004364  BDCNVA:  SUB    BDCNVD,%0  ;SET UP ADDR TO MOVE DECIMAL
1209 004350 010037 004360  BDCNVA:  MOV    %0,BDCNVB  ;DATA FROM.
1210 004354 004537 004252  BDCNVA:  JSR    %5,BMOVE   ;MOVE DECIMAL DATA TO DESTINATION.
1211 004360 000000          BDCNVB:  OPEN          ;SRC ADDR.
1212 004362 000000          BDCNVC:  OPEN          ;DEST ADDR.
1213 004364 000000          BDCNVD:  OPEN          ;COUNT.
1214 004366 104012          BDCNVD:  RSTREG          ;RESTORE REGS.
1215 004370 000205          BDCNVD:  RTS      %5     ;YES, EXIT.
1216 004372 005037 004434  SUBTEN:  CLR    DIGIT    ;CLEAR DIGIT
1217 004376 163701 004436  SUBTNA:  SUB    TENPWR,%1 ;SUBTRACT TEN POWER FROM BINARY VALUE.
1218 004402 103403          SUBTNA:  BCS    SUBTNB   ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
1219 004404 005237 004434  SUBTNA:  INC    DIGIT

```

```
1220 004410 000772
1221 004412 063701 004436
1222 004416 062737 000060 004434
1223 004424 113720 004434
1224 004430 000207
1225 004432 000000
1226 004434 000000
1227 004436 000000
1228 004440 023420
1229 004442 001750
1230 004444 000144
1231 004446 000012
1232 004450 000001
1233 004452 040 040 040
1234 004455 040 040
1235
1236 004460 104000
1237 004462 013624
1238 004464 004737 003372
1239 004470 012637 001374
1240 004474
1241 004506 104000
1242 004510 014124
1243 004512 012737 177400 001362
1244 004520 032737 000002 001374
1245 004526 001413
1246 004530 012737 177700 001362
1247 004536 032737 000001 001374
1248 004544 001403
1249 004546 012737 177740 001362
1250 004554 000207
1251 004556 032737 000001 001374
1252 004564 001773
1253 004566 012737 177600 001362
1254 004574 000767
1255
1256
1257 004576 013737 177776 001336
1258 004604 012737 000340 177776
1259 004612 006237 001336
1260 004616 006237 001336
1261 004622 006237 001336
1262 004626 042737 177740 001336
1263 004634 013737 001336 001340
1264 004642 011637 001342
1265 004646 004537 003606
1266 004652 001340
1267 004654 013063
1268 004656 000006
1269 004660 004537 003606
1270 004664 001342
1271 004666 013115
1272 004670 000006
1273 004672 104000
1274 004674 013016
1275 004676 000000

BR SUBTNA
SUBTNB: ADD TENPWR,%1 ;RESTORE SUBTRACTED VALUE.
ADD #60,DIGIT ;CONVERT (DIGIT) TO ASCII
MOVB 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
MTERR
HALT
```

```
1276 004700 000137 001436 JMP START
1277
1278 ;MAPVEC - MAP VECTOR VECTOR OR REPORT FROM DEPENDING ON FMAP FLAG
1279 004704 011637 001340 MAPVEC: MOV @%6, TOPC
1280 004710 022626 POPSP2
1281 004712 011637 001342 MOV @%6, FROMPC
1282 004716 162737 000004 001340 SUB #4, TOPC
1283 004724 005737 001350 TST FMAP
1284 004730 001746 BEQ ERTPA ;NO MAPPING, REPORT ERROR
1285 004732 013737 001340 001214 MOV TOPC, TXVTR ;STORE VECTOR
1286 004740 162737 000004 001340 SUB #4, TOPC
1287 004746 013737 001340 001210 MOV TOPC, RXVTR
1288 004754 005037 001350 CLR FMAP
1289 004760 000002 RTI
1290
1291 ;FORMAD - FORM DEVICE AT ADDRESS
1292 004762 010137 001200 FORMAD: MOV %1, RXCSR
1293 004766 062701 000002 ADD #2, %1
1294 004772 010137 001202 MOV %1, RXBUF
1295 004776 062701 000002 ADD #2, %1
1296 005002 010137 001204 MOV %1, TXCSR
1297 005006 062701 000002 ADD #2, %1
1298 005012 010137 001206 MOV %1, TXBUF
1299 005016 000207 RTS %7
1300
1301 ;SUBROUTINE TO MAKE LINE CONNECTION.
1302 005020 017737 174154 001370 LINCON: MOV @RXCSR, RXCSRT
1303 005026 032737 020000 001370 BIT #BIT13, RXCSRT ;YES, IS CLEAR TO SEND UP
1304 005034 001046 BNE LINEUP ;YES CONNECTION IS MADE.
1305 005036 042777 000146 174134 LINCA: BIC #146, @RXCSR ;CLEAR IE BIT AND DTR, RQ TO SND
1306 005044 005777 174132 TST @RXBUF ;CLEAR DONE FLAG
1307 005050 104000 TYPE ;TYPE
1308 005052 014020 MAKCON ;'MAKE LINE CONNECTION'
1309 005054 017737 174120 001370 LINC B: MOV @RXCSR, RXCSRT
1310 005062 032737 040000 001370 BIT #BIT14, RXCSRT ;DID YOU RING
1311 005070 001771 BEQ LINC B ;GO WAIT FOR RING
1312 005072 052777 000006 174100 BIS #6, @RXCSR ;SET DTR, RQ TO SND
1313 005100 104016 DELAY ;WAIT 10 SECONDS FOR
1314 005102 023420 10000. ;CLEAR TO SEND
1315 005104 017737 174070 001370 MOV @RXCSR, RXCSRT
1316 005112 005777 174064 TST @RXBUF ;CLEAR DONE
1317 005116 032737 020000 001370 BIT #BIT13, RXCSRT ;IS CLEAR TO SEND UP?
1318 005124 001003 BNE LINC F ;YES. GO TO LINC F
1319 005126 104015 ERRRX ;NO. PRINT ERROR MESSAGE
1320 005130 014146 LINCHM ;'CLEAR TO SEND NOT SET'
1321 005132 000741 BR LINCA ;START OVER AGAIN
1322 005134 017737 174040 001370 LINC F: MOV @RXCSR, RXCSRT ;CLEAR ALL FLAGS
1323 005142 005777 174034 TST @RXBUF ;AND DONE
1324 005146 104000 TYPE ;TYPE MESSAGE
1325 005150 014176 LINMAD ;CONNECTION IS MADE'
1326 005152 000205 LINEUP: RTS 5 ;EXIT LINE CONNECTION ROUTINE WITH
1327 ;SUBROUTINE TO OVERLAY <CRLF> IN DATA PATTERN (EVERY 72. ND CHAR)
1328 005154 012701 015512 OVRLAY: MOV #OUTBUF, %1 ;GET OUTBUF ADDRESS
1329 005160 012702 000016 MOV #14, %2 ;GET COUNTER
1330 005164 012711 105215 OVRLYA: MOV #105215, (1) ;INSERT CR&LF
1331 005170 062701 000110 ADD #72, %1 ;ADD OFFSET
```

```
1332 005174 005302          DEC      %2          ;DONE?
1333 005176 001372          BNE     OVRLYA
1334 005200 000207          RTS     7            ;EXIT
1335
1336          ;RECEIVER INTERRUPTS COMMON HANDLER
1337 005202 000240          RISR:  NOP
1338 005204 010037 001330      MOV     %0,LINE
1339 005210 006300          ASL     %0
1340 005212 016037 011216 001200  MOV     RCSR(0),RXCSR ;GET ADDRESS OF INTERRUPTING DL11-E'S RCSR
1341 005220 017737 173754 001370  MOV     @RXCSR,RXCST ;GET CSR CONTENTS
1342 005226 100570          BMI     DCERR       ;CHECK INT
1343 005230 105737 001370  TSTB   RXCST        ;TEST DONE
1344 005234 001002          BNE     RISRA
1345 005236 104003          ERROR
1346 005240 000002          RTI
1347 005242 020037 001404  RISRA:  CMP     %0,CALLED   ;DID CALLED LINE INTERRUPT?
1348 005246 001020          BNE     RISRB       ;BRANCH IF CALLER INTERRUPTED
1349 005250 005737 001412  TST    MODEM        ;CHECK MODEM TYPE
1350 005254 001403          BEQ     RISRAA      ;BRANCH IF 103
1351 005256 005770 011314  TST    @RBUF(0)     ;READ CALLED LINES DATA
1352 005262 000002          RTI
1353 005264 117077 011314 174104  RISRAA: MOVB   @RBUF(0),@INBUFP ;STORE CHARACTER IN INPUT BUFFER
1354 005272 005237 001376      INC    INBUFP        ;INCREMENT POINTER
1355 005276 022737 017626 001376  CMP    #INBUFP+100.,INBUFP ;HAVE 100. CHARACTERS BEEN RECEIVED?
1356 005304 001430          BEQ     RISRC        ;GO CHECK DATA IF YES
1357 005306 000002          RTI
1358 005310 117077 011314 174062  RISRB:  MOVB   @RBUF(0),@BUFP   ;STORE CHARACTER IN INTERMEDIATE DATA BUFFER
1359 005316 005237 001400      INC    BUFP          ;INCREMENT POINTER
1360 005322 022737 015670 001400  CMP    #BUFP+10.,BUFP ;HAVE 10 CHARACTERS BEEN RECEIVED
1361 005330 002401          BLT    .+4
1362 005332 000002          RTI
1363 005334 022737 000002 001332  CMP    #2,CONFIG    ;RUNNING CONFIGURATION 2?
1364 005342 001405          BEQ     RISRBB
1365 005344 022737 016022 001400  CMP    #BUFP+100.,BUFP ;HAVE 100. CHARACTERS BEEN RECEIVED?
1366 005352 001405          BEQ     RISRC        ;GO CHECK DATA IF YES,OTHERWISE
1367 005354 000002          RTI
1368 005356 052770 000100 011412  RISRBB: BIS    #BIT6,@TCSR(0) ;START CALLERS TRANSMITTER
1369 005364 000002          RTI
1370          ;CHECK DATA CONFIGURATION #1
1371 005366 000240          RISRC:  NOP
1372 005370 012737 000001 001364  MOV    #1,CTRD       ;INITIALIZE CHARACTER COUNT
1373 005376 012702 015656      MOV    #BUFP,%2      ;POINT R2 TO CALLERS RECEIVED DATA BUFFER
1374 005402 012703 015512      MOV    #OUTBUF,%3    ;R3 = FIRST ADDRESS OF OUTPUT DATA BUFFER
1375 005406 010237 001400      MOV    %2,BUFP      ;RESTORE CALLERS RCVD DATA BUFFER PTR
1376 005412 022737 000001 001332  CMP    #1,CONFIG    ;CHECK CONFIGURATION
1377 005420 001015          BNE     RISRD
1378 005422 112337 001360  RISRCA: MOVB   (3)+,XMTDAT ;GET TRANSMITTED CHARACTER
1379 005426 112237 001356      MOVB   (2)+,RECDAT ;GET RECEIVED CHARACTER
1380 005432 104004          DATCHK ;CHECK DATA
1381 005434 005237 001364      INC    CTRD         ;INCREMENT CHARACTER COUNT
1382 005440 022737 000101 001364  CMP    #101,CTRD    ;HAS ALL DATA BEEN CHECKED
1383 005446 001365          BNE     RISRCA
1384 005450 000137 005554      JMP    FINISH
1385
1386          ;CHECK DATA CONFIGURATION #2
1387 005454 000240          RISRD:  NOP
```

```

1388 005456 012704 017462      MOV      #INBUF,%4      ;POINT R4 TO CALLED LINES RECEIVER
1389 005462 010437 001376      MOV      %4,INBUFP     ;DATA BUFFER & INIT. POINTER
1390 005466 012737 015656 001410  RISRDA: MOV      #BUFF,TBUFFP
1391 005474 013701 001404      MOV      CALLED,%1
1392 005500 016137 011216 001200  MOV      RCSR(1),RXCSR
1393 005506 112337 001360      MOV      (3)+,XMTDAT
1394 005512 112237 001356      MOV      (2)+,RECDAT   ;COMPARE TRANSMITTED DATA WITH DATA
1395 005516 104004      DATCHK                ;RECEIVED BY CALLED LINE
1396 005520 013701 001402      MOV      CALLER,%1
1397 005524 016137 011216 001200  MOV      RCSR(1),RXCSR
1398 005532 112437 001356      MOV      (4)+,RECDAT   ;COMPARE TRANSMITTED DATA WITH DATA
1399 005536 104004      DATCHK                ;RECEIVED BY CALLER
1400 005540 005237 001364      INC      CTRD
1401 005544 022737 000101 001364  CMP      #101,CTRD
1402 005552 001350      BNE     RISRDA
1403 005554 000240      FINISH: NOP
1404 005556 013701 001404      MOV      CALLED,%1
1405 005562 004537 004224      JSR     5,FILL
1406 005566 015512      OUTBUF
1407 005570 000144      100.
1408 005572 104000      TYPE
1409 005574 014117      ENDPAS
1410 005576 052771 000100 011412  BIS     #BIT6,@TCSR(1)
1411 005604 000240      NOP
1412 005606 000002      RTI
1413      ;ERROR SERVICE ROUTINE
1414 005610 032770 100000 011314  DCERR: BIT     #BIT15,@RBUF(0) ;TEST ERROR
1415 005616 001402      BEQ     RISRF
1416 005620 104015      ERRRX
1417 005622 013126      CSRADD
1418 005624 012737 015512 001406  RISRF: MOV     #OUTBUF,OTBUFP ;SET OUTPUT BUFFER POINTER
1419 005632 012737 017462 001376  MOV     #INBUF,INBUFP ;SET INPUT BUFFER POINTER
1420 005640 012737 015656 001400  MOV     #BUFF,BUFFP ;SET INTERMEDIATE BUFFER POINTER
1421 005646 012737 015656 001410  MOV     #BUFF,TBUFFP ;SET POINTER FOR CONFIG #2 TRANSMITTER
1422 005654 032737 040000 001370  BIT     #BIT14,RXCST ;CHECK RING INDICATOR
1423 005662 001005      BNE     RISREX ;BRANCH IF RING
1424 005664 004737 006312  JSR     7,DISCON ;ERROR SET - NO RING
1425 005670 104015      ERRRX
1426 005672 013126      CSRADD
1427 005674 000002      RTI
1428 005676      RISREX: CNVOA  LINE,TLINE,2
1429 005710 104000      TYPE
1430 005712 014463      ALINE
1431 005714 010037 001404      MOV     %0,CALLED
1432 005720 004737 003372      JSR     PC,RDOCT ;GET INPUT
1433 005724 011637 001412      MOV     (SP),MODEM ;GET MODEM TYPE
1434 005730 042737 177773 001412  BIC     #177773,MODEM ;0=103,4=202
1435 005736 012637 001332      MOV     (SP)+,CONFIG
1436 005742 042737 177774 001332  BIC     #177774,CONFIG
1437 005750 001042      BNE     RISRFC ;GO TO SERVICE FOR CONFIG 1 OR 2
1438 005752 004737 006326  JSR     7,CONN ;CONNECT LINE IF CONFIGURATION 0
1439 005756 104000      TYPE ;TYPE MESSAGE TO PRESS DATA
1440 005760 014763      BUTTON ;BUTTON ON DATA PHONE
1441 005762 104016      DELAY ;WAIT FOR CARRIER
1442 005764 023420      10000. ;10 SECONDS
1443 005766 005770 011314      ST     @RBUF(0) ;READ BUFFER TO CLEAR DONE
  
```

```

1444 005772 032770 020000 011216 BIT #BIT13,@RCSR(0) ;TEST FOR CLEAR TO SEND
1445 006000 001004 BNE RISRFB
1446 006002 104003 ERROR ;ERROR! DID NOT RECEIVE CLEAR TO SEND
1447 ;WITHIN TIME ALLOTTED (10 SEC.)
1448 006004 004737 006312 JSR 7,DISCON ;DISCONNECT LINE
1449 006010 000002 RTI ;AND EXIT
1450 006012 016037 011412 001204 RISRFB: MOV TCSR(0),TXCSR ;GET CALLED LINES TXCSR ADDRESS
1451 006020 004737 004460 JSR 7,SETPAR ;LOAD USER PARAMETERS
1452 006024 104000 TYPE ;TYPE 'LINE CONNECTION'
1453 006026 014176 LINMAD ;MADE'
1454 006030 CNVOA CONFIG,TCONFIG,2
1455 006042 104000 TYPE
1456 006044 014225 ACONFIG
1457 006046 052770 000100 011412 BIS #BIT6,@TCSR(0)
1458 006054 000002 RTI ;AND EXIT
1459
1460 ;HERE IF CONFIGURATION 1 OR 2
1461 006056 104000 RISRFC: TYPE ;ASK USER WHICH LINE HE IS
1462 006060 014633 WRU ;DIALING ON
1463 006062 004737 003372 JSR PC,RDOCT ;GET INPUT
1464 006066 012601 MOV (SP)+,%1 ;GET LINE #
1465 006070 042701 177740 BIC #177740,%1 ;MASK UNUSED BITS
1466 006074 010137 001330 MOV %1,LINE
1467 006100 CNVOA LINE,URA,2
1468 006112 104000 TYPE ;REPORT LINE # ON TTY
1469 006114 014730 UR
1470 006116 006301 ASL %1
1471 006120 010137 001402 MOV %1,CALLER ;SAVE CALLERS LINE #
1472 006124 004737 006326 JSR 7,CONN ;CONNECT CALLED LINE
1473 006130 052771 000002 011216 BIS #BIT1,@RCSR(1) ;SET DTR ON CALLERS LINE
1474 006136 104000 TYPE ;TYPE MESSAGE TO PRESS DATA
1475 006140 014763 BUTTON ;ON DATA PHONE
1476 006142 104016 DELAY ;WAIT 10 SECONDS FOR CLEAR TO SEND
1477 006144 023420 10000. ;SET AT CALLED LINE
1478 006146 027071 011314 011314 CMP @RBUF(0),@RBUF(1) ;READ BUFFERS
1479 006154 032770 020000 011216 BIT #BIT13,@RCSR(0) ;TEST FOR CLEAR TO SEND AT CALLED LINE
1480 006162 001007 BNE RISRFF
1481 006164 104003 ERROR ;ERROR! CLEAR TO SEND NOT SET AT CALLED LINE
1482 006166 004737 006312 RISRFD: JSR 7,DISCON ;DISCONNECT
1483 006172 042771 000006 011216 BIC #6,@RCSR(1) ;LINE
1484 006200 000002 RTI ;AND EXIT
1485 006202 022737 000002 001332 RISRFF: CMP #2,CONFIG
1486 006210 001414 BEQ RISRFG
1487 006212 022771 010000 011216 CMP #BIT12,@RCSR(1) ;CHECK CARRIER AT CALLERS LINE
1488 006220 001003 BNE RISRFE
1489 006222 104003 ERROR ;ERROR! NO CARRIER AT CALLERS LINE
1490 006224 000137 006166 JMP RISRFD ;GO DISCONNECT LINES
1491 006230 016137 011216 001200 RISRFE: MOV RCSR(1),RXCSR
1492 006236 000137 006012 JMP RISRFB ;GO GET PAREMETERS AND ENABLE
1493 ;CALLED TRANSMITTER AND EXIT
1494
1495 ;HERE IF CONFIGURATION 2
1496 006242 032771 020000 011216 RISRFG: BIT #BIT13,@RCSR(1) ;TEST CALLERS CLEAR TO SEND
1497 006250 001003 BNE RISRFB
1498 006252 104003 ERROR ;ERROR! NO CTS AT CALLERS LINE
1499 006254 000137 006166 JMP RISRFD ;GO DISCONNECT LINE AND EXIT
    
```



```
1500 006260 016037 011216 001200 RISRFH: MOV RCSR(0),RXCSR
1501 006266 016137 011412 001204 MOV TCSR(1),TXCSR
1502 006274 004737 004460 JSR 7,SETPAR ;GO GET PARAMETERS FOR CALLERS
1503 ;TRANSMITTER AND CALLED RECEIVER
1504 006300 016137 011216 001200 MOV RCSR(1),RXCSR
1505 006306 000137 006012 JMP RISRFB
1506 ;
1507 ;SUBROUTINE TO DISCONNECT LINE R0 HAS LINE #
1508 006312 042770 000006 011216 DISCON: BIC #6,@RCSR(0)
1509 006320 005770 011216 TST @RCSR(0)
1510 006324 000207 RTS 7
1511 ;
1512 ;SUBROUTINE TO CONNECT LINE R0 HAS LINE #
1513 006326 052770 000006 011216 CONN: BIS #6,@RCSR(0) ;SET DTR, R0 TO SND
1514 006334 000207 RTS 7
1515 ;TRANSMITTER INTERRUPT COMMON HANDLER
1516 006336 000240 TISR: NOP
1517 006340 006300 ASL %0 ;R0 HAS LINE #
1518 006342 105770 011412 TSTB @TCSR(0) ;CHECK FOR DONE
1519 006346 100402 BMI TISRA ;BRANCH IF DONE
1520 006350 104003 ERROR ;ERROR! FALSE INTERRUPT
1521 006352 000002 TISRAA: RTI ;EXIT
1522 006354 005737 001332 TISRA: TST CONFIG ;THIS CONFIGURATIO 0?
1523 006360 001420 BEQ TISRC ;BRANCH IF YES
1524 006362 020037 001402 CMP %0,CALLER ;DID CALLER INTERRUPT
1525 006366 001015 BNE TISRC
1526 006370 117770 173014 011510 MOVB @TBUFP,@TBUF(0) ;TRANSMIT
1527 006376 005237 001410 INC TBUFP ;STEP POINTER
1528 006402 022737 016022 001410 CMP #BUFP+100.,TBUFP
1529 006410 001003 BNE .+10
1530 006412 042770 000100 011412 BIC #BIT6,@TCSR(0)
1531 006420 000002 RTI
1532 006422 117770 172760 011510 TISRC: MOVB @OTBUFP,@TBUF(0) ;TRANSMIT THE NEXT CHARACTER
1533 006430 005237 001406 INC OTBUFP ;STEP POINTER TO NEXT CHAR.
1534 006434 005737 001332 TST CONFIG ;WAS CONFIGURATION 0 SELECTED
1535 006440 001010 BNE TISRBB ;BRANCH IF CONFIG #1 OR #2
1536 006442 022737 017462 001406 CMP #OUTBUF+1000.,OTBUFP;HAVE 1000. CHARS. BEEN SENT
1537 006450 001340 BNE TISRAA ;EXIT IF NOT
1538 006452 012737 015512 001406 TISRBB: MOV #OUTBUF,OTBUFP ;RESET POINTER
1539 006460 000002 RTI ;AND EXIT
1540 ;
1541 006462 022737 015656 001406 TISRB: CMP #OUTBUF+100.,OTBUFP;HAVE 100. CHARS. BEEN SENT?
1542 006470 001330 BNE TISRAA ;EXIT IF NOT
1543 006472 042770 000100 011412 BIC #BIT6,@TCSR(0) ;DISABLE TRANSMITTER INTERRUPT
1544 006500 000764 BR TISRBB ;RESET POINTER AND EXIT
1545 ;
1546 ;*****
1547 ;PRGO - SINGLE CHARACTER LINE MODE TEST.
1548 ;*****
1549 006502 000240 PRGO: NOP ;BEGIN PRGO
1550 006504 104000 TYPE ;TYPE
1551 006506 013256 POTIT ;PROGRAM TITLE
1552 006510 104000 TYPE
1553 006512 013712 SELCAR
1554 006514 004737 003372 JSR PC,RDOCT ;GET INPUT
1555 006520 112601 MOVB (SP)+,%1 ;GET USER SPECIFIED DATA
```

```
1556 006522 010137 015512      MOV      %1,OUTBUF      ;AND
1557 006526 004537 004252      JSR      5,BMOVE       ;LOAD
                                OUTBUF      ;INTO
1558 006532 015512      OUTBUF      ;OUTPUT
1559 006534 015513      OUTBUF+1    ;BUFFER
1560 006536 001747      999.        ;OVER LAY CR,LF'S IN DATA
1561 006540 004737 005154      JSR      7,OVRLAY      ;LOAD PRIORITY LEVEL IN VECTOR+2
1562 006544 004737 010352      JSR      7,LDPRI       ;LOAD TRANSMITTER VECTORS
1563 006550 004737 010304      JSR      7,LDTVEC      ;LOAD RECEIVER VECTORS
1564 006554 004737 010240      JSR      7,LDVECS      ;SET PROCESSOR PRIORITY=7
1565 006560 012737 000340 177776  MOV      #PRTY7,PSW     ;SET IE
1566 006566 012702 000140      MOV      #140,%2       ;BIT IN
1567 006572 012701 011216      MOV      #RCSR,%1      ;ALL RECEIVERS
1568 006576 004537 010174      JSR      5,MOVIT       ;TYPE
1569 006602 104000      TYPE        ;'MAKE LINE CONNECTION'
1570 006604 014020      MAKCON      ;SET PROCESSOR PRIORITY=0
1571 006606 005037 177776      CLR      PSW          ;WAIT
1572 006612 000001      PRG0A: WAIT          ;HERE
1573 006614 000776      BR      PRG0A
1574
1575      ;*****
1576      ;PRG1 - SPECIAL BINARY COUNT PATTERN LINE MODE TEST.
1577      ;*****
1577 006616 104000      PRG1: TYPE          ;TYPE PROGRAM TITLE.
1578 006620 013322      P1TIT
1579 006622 012737 105215 015512  MOV      #105215,OUTBUF ;LOAD CRLF
1580 006630 004537 004220      JSR      5,INFIL      ;LOAD OUTPUT
1581 006634 015514      OUTBUF+2    ;WITH BINARY
1582 006636 001750      1000.      ;COUNT PATTERN
1583 006640 012737 000100 001334  MOV      #100,NUMBER
1584 006646 004737 010352      JSR      7,LDPRI       ;LOAD PRIORITY LEVEL IN VECTOR +2
1585 006652 004737 010304      JSR      7,LDTVEC      ;LOAD TRANSMITTER VECTORS
1586 006656 004737 010240      JSR      7,LDVECS      ;LOAD RECEIVER VECTORS
1587 006662 012737 000340 177776  MOV      #PRTY7,PSW     ;SET PROCESSOR PRIORITY=7
1588 006670 012702 000140      MOV      #140,%2       ;GET IE BIT
1589 006674 012701 011216      MOV      #RCSR,%1      ;GET FIRST CSR ADDRESS
1590 006700 004537 010174      JSR      5,MOVIT       ;AND MOVE IT
1591 006704 104000      TYPE        ;TYPE
1592 006706 014020      MAKCON      ;'MAKE LINE CONNECTION'
1593 006710 005037 177776      CLR      PSW          ;SET PROCESSOR PRIORITY=0
1594 006714 000001      PRG1C: WAIT          ;WAIT
1595 006716 000776      BR      PRG1C        ;HERE
1596
1597      ;*****
1598      ;PRG2-SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS
1599      ;*****
1600      ;THE MESSAGE 'A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS
1601      ;BACK 1234567890.'
1602 006720 104000      PRG2: TYPE          ;TYPE PROGRAM
1603 006722 013371      P2TIT        ;TITLE
1604 006724 004537 004072      JSR      5,LINSEL
1605 006730 004737 004460      JSR      7,SETPAR     ;GO SET PARAMETERS
1606 006734 052777 000004 172236  BIS      #BIT2,@RXCSCR ;SET REQUEST TO SEND
1607 006742 004537 005020      PRG2A: JSR      5,LINCON ;GO MAKE LINE CONNECTION
1608 006746 012702 014252      PRG2B: MOV      #PRG2M,%2 ;GET ADDRESS OF MESSAGE
1609 006752 112201      PRG2C: MOVB     (2)+,%1 ;GET FIRST CHARACTER
1610 006754 020127 000045      CMP      %1,#1%       ;TERMINATOR CHARACTER
1611 006760 001772      BEQ     PRG2B        ;RESEND MESSAGE
```

```
1612 006762 032737 000100 001374 BIT #BIT6,SRT ;PARITY ENABLED
1613 006770 001402 BEQ .+6
1614 006772 004737 003656 JSR 7,GENPAR ;GENERATE PARITY
1615 006776 004537 005020 JSR 5,LINCON ;CHECK LINE CONNECTION
1616 007002 010177 172200 MCV %1,@TXBUF ;LOAD BUFFER
1617 007006 105777 172172 TSTB @TXCSR ;AND WAIT FOR CHARACTER
1618 007012 100375 BPL .-4 ;TO BE TRANSMITTED
1619 007014 000756 BR PRG2C ;GET NEXT CHARACTER.
1620
1621 ;*****
1622 ;PRG3-PROGRAM TO RECEIVE A MESSAGE.
1623 ;*****
1623 007016 104000 PRG3: TYPE ;TYPE PROGRAM
1624 007020 013441 P3TIT ;TITLE
1625 007022 004537 004072 JSR 5,LINSEL
1626 007026 004737 004460 JSR 7,SETPAR ;GET PARAMETERS
1627 007032 012706 001176 PRG3A: MOV #STKPTR-2,%6 ;REPOSITION STACK POINTER
1628 007036 052777 000004 172134 BIS #BIT2,@RXCSR ;SET REQUEST TO SEND
1629 007044 004537 005020 JSR 5,LINCON ;MAKE LINE CONNECTION
1630 007050 104006 STRXV ;SFT RECEIVER INTERRUPT
1631 007052 007134 RINT3 ;TO THIS ADDRESS
1632 007054 104007 STTXV ;SET TRANSMITTER INTERRUPT
1633 007056 007346 TINT3 ;TO THIS ADDRESS
1634 007060 005037 007546 CLR WORDS
1635 007064 013700 001234 MOV TPVTR,%0
1636 007070 012720 007500 MOV #TPINT,(0)+ ;LOAD TELEPRINTER VECTOR
1637 007074 013710 001236 MOV TPLVL,(0) ;AND PRIORITY
1638 007100 012701 015512 MOV #OUTBUF,%1 ;GET BUF ADD
1639 007104 052737 100000 007544 BIS #BIT15,TFLAG ;SET BIT 15
1640 007112 004737 007732 JSR 7,TCRLF ;SEND CRLF
1641 007116 052777 000140 172054 BIS #140,@RXCSR ;ENABLE RECEIVER INTERRUPTS
1642 007124 005037 177776 CLR PSW
1643 007130 000001 WAIT ;DO
1644 007132 000776 BR .-2 ;NOTHING
1645 007134 017737 172040 001370 RINT3: MOV @RXCSR,RXCST ;GET RXCSR DATA
1646 007142 100461 BMI ERR3A ;BRANCH IF ERROR
1647 007144 105737 001370 TSTB RXCSR ;TEST
1648 007150 100064 BPL ERR3B
1649 007152 005237 007546 INC WORDS
1650 007156 017737 172020 007550 MOV @RXBUF,RXBUFT ;GET DATA
1651 007164 113711 007550 MOVB RXBUFT,(1)
1652 007170 005737 007550 TST RXBUFT
1653 007174 100455 BMI ERR3C
1654 007176 105777 170766 TSTB @SRPTR ;ECHO OPTION SELECTED
1655 007202 100405 BMI RINT3A
1656 007204 105777 171774 TSTB @TXCSR
1657 007210 100375 BPL .-4
1658 007212 111177 171770 MOVB (1),@TXBUF ;ECHO CHARACTER
1659 007216 023727 007546 001604 RINT3A: CMP WORDS,#900. ;END OF BUFFER ALLOWED
1660 007224 001411 BEQ RINT3B ;YES EXIT
1661 007226 005737 007544 TST TFLAG ;IS THIS THE FIRST
1662 007232 100441 BMI RINT3E ;CHARACTER BRANCH IF YES.
1663 007234 121103 CMPB (1),%3 ;LAST CHARACTER RECEIVED
1664 007236 001404 BEQ RINT3B
1665 007240 122127 000203 CMPB (1)+,#203 ;CONTROL C
1666 007244 001401 BEQ RINT3B
1667 007246 000002 RTI ;EXIT
```

```
1668 007250 005037 007546 RINT3B: CLR WORDS
1669 007254 042777 000140 171716 BIC #140,@RXCSR ;DISABLE RECEIVER
1670 007262 012701 015513 MOV #OUTBUF+1,%1 ;INITIALIZE BUFFER POINTER
1671 007266 010102 MOV %1,%2
1672 007270 052777 000100 171706 BIS #BIT6,@TXCSR ;ENABLE TRANSMITTER
1673 007276 052777 000100 171720 BIS #BIT6,@TPS ;ENABLE TELEPRINTER
1674 007304 000002 RTI ;EXIT
1675 007306 104015 ERR3A: ERRRX ;TYPE ERROR MESSAGE
1676 007310 014353 LFAIL ;
1677 007312 042777 000140 171660 BIC #140,@RXCSR ;DISABLE RECEIVER
1678 007320 000644 BR PRG3A
1679 007322 104015 ERR3B: ERRRX ;TYPE
1680 007324 013135 RINTM ;ERROR MESSAGE
1681 007326 000002 RTI ;EXIT
1682 007330 104015 ERR3C: ERRRX
1683 007332 014371 ROVER
1684 007334 000002 RTI
1685
1686 007336 005037 007544 RINT3E: CLR TFLAG
1687 007342 112103 MOV (1)+,%3
1688 007344 000002 RTI
1689
1690 007346 017737 171632 001366 TINT3: MOV @TXCSR,TXCSTR ;GET TXCSR DATA
1691 007354 105737 001366 TSTB TXCSR ;TEST
1692 007360 100016 BPL TINT3B
1693 007362 112177 171620 MOV (1)+,@TXBUF ;TRANSMIT CHARACTER
1694 007366 005237 007546 INC WORDS
1695 007372 121103 CMPB (1),%3 ;ALL CHARACTERS TRANSMITTED
1696 007374 001431 BEQ TINT3C
1697 007376 023727 007546 001604 CMP WORDS,#900.
1698 007404 001425 BEQ TINT3C
1699 007406 121127 000203 CMPB (1),#203 ;= CONTROL C
1700 007412 001422 BEQ TINT3C
1701 007414 000002 RTI ;RETURN TO MAIN PROGRAM
1702 007416 017737 171602 001372 TINT3B: MOV @TPS,TEMP ;SAVE TELEPRINTER STATUS
1703 007424 005077 171574 CLR @TPS ;DISABLE INTERRUPT
1704 007430 105777 171570 TSTB @TPS ;WAIT FOR
1705 007434 100375 BPL .-4 ;TELEPRINTER TO FINISH
1706 007436 104014 ERRTX ;TYPE
1707 007440 013156 TINTM ;ERROR MESSAGE
1708 007442 105777 171556 TSTB @TPS ;WAIT FOR TELEPRINTER
1709 007446 100375 BPL .-4 ;TO FINISH
1710 007450 013777 001372 171546 MOV TEMP,@TPS ;RESTORE TELEPRINTER STATUS
1711 007456 000002 RTI ;EXIT
1712
1713 007460 042777 000100 171516 TINT3C: BIC #BIT6,@TXCSR ;DISABLE INTERRUPT
1714 007466 032777 000100 171530 BIT #BIT6,@TPS ;IS TTY ACTIVE
1715 007474 001421 BEQ PRG3EX
1716 007476 000002 RTI
1717
1718
1719 007500 112277 171522 TPINT: MOV (2)+,@TPB ;TYPE CHARACTER
1720 007504 121203 CMPB (2),%3 ;WAS THIS THE LAST CHAR.
1721 007506 001404 BEQ TPINTA
1722 007510 121227 000203 CMPB (2),#203 ;= CONTROL C
1723 007514 001401 BEQ TPINTA
```

```

1724 007516 000002 RTI
1725 007520 042777 000100 171476 TPINTA: BIC #BIT6,@TPS ;DISABLE INTERRUPT
1726 007526 032777 000100 171450 BIT #BIT6,@TXCSR ;IS TRANSMITTER ACTIVE
1727 007534 001401 BEQ .+4
1728 007536 000002 RTI ;EXIT
1729 007540 000137 007032 PRG3EX: JMP PRG3A
1730 007544 000000 TFLAG: 0
1731 007546 000000 WORDS: 0
1732 007550 000000 RXBUF: 0
1733 ;*****
1734 ;FRG4-SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS
1735 ;*****
1736 ;MESSAGE SPIRAL PATTERN
1737 ;
1738 007552 104000 PRG4: TYPE ;TYPE PROGRAM
1739 007554 013477 P4TIT ;TITLE
1740 007556 004537 004072 JSR 5,LINSEL
1741 007562 004737 004460 JSR 7,SETPAR ;GO SET PARAMETERS
1742 007566 052777 000004 171404 BIS #BIT2,@RXCSR ;SET REQUEST TO SEND
1743 007574 012737 000110 007730 MOV #72,COLMN ;INIT PAGE WIDTH
1744 007602 012703 000040 MOV #40,%3 ;SET LINE START CHAR
1745 007606 004537 005020 PRG4A: JSR 5,LINCON ;GO MAKE LINE CONNECTION
1746 007612 004737 007732 JSR 7,TCRLF
1747 007616 010302 PRG4B: MOV %3,%2 ;GET FIRST CHARACTER
1748 007620 110201 PRG4C: MOVB %2,%1 ;GET CHARACTER
1749 007622 120127 000136 CMPB %1,#136 ;TERMINATOR CHARACTER
1750 007626 001003 BNE PRG4D ;RESEND MESSAGE
1751 007630 012702 000040 MOV #40,%2
1752 007634 000771 BR PRG4C
1753 007636 032737 000100 001374 PRG4D: BIT #BIT6,SRT ;PARITY ENABLED
1754 007644 001402 BEQ .+6
1755 007646 004737 003656 JSR 7,GENPAR ;GENERATE PARITY
1756 007652 004537 005020 JSR 5,LINCON ;CHECK LINE CONNECTION
1757 007656 010177 171324 MOV %1,@TXBUF ;LOAD BUFFER
1758 007662 105777 171316 TSTB @TXCSR ;AND WAIT FOR CHARACTER
1759 007666 100375 BPL .-4 ;TO BE TRANSMITTED
1760 007670 005202 INC %2 ;SET FOR NEXT CHAR
1761 007672 005337 007730 DEC COLMN ;ALL COLUMNS PRINTED?
1762 007676 001350 BNE PRG4C ;NO, GET NEXT CHAR
1763 007700 012737 000110 007730 MOV #72,COLMN ;RESET COLUMN COUNTER
1764 007706 004737 007732 JSR 7,TCRLF
1765 007712 005203 INC %3 ;UPDATE LINE START CHAR
1766 007714 120327 000136 CMPB %3,#136 ;LAST IN SET
1767 007720 001336 BNE PRG4B ;NO
1768 007722 012703 000040 MOV #40,%3 ;YES, RESET
1769 007726 000733 BR PRG4B ;GET NEXT CHARACTER.
1770 007730 000000 COLMN: 0
1771 ;
1772 ;SEND CR LF
1773 ;
1774 007732 112777 000015 171246 TCRLF: MOVB #15,@TXBUF ;SEND CR,LF
1775 007740 105777 171240 TSTB @TXCSR
1776 007744 100375 BPL .-4
1777 007746 112777 000012 171232 MOVB #12,@TXBUF
1778 007754 105777 171224 TSTB @TXCSR
1779 007760 100375 BPL .-4
  
```

| Line | Address | Code | Label | Comment | |
|------|---------|--------|-----------|---|--|
| 1780 | 007762 | 000207 | RTS | %7 | |
| 1781 | | | ***** | | |
| 1782 | | | PROGRAM 5 | | |
| 1783 | | | ***** | | |
| 1784 | 007764 | 104000 | PRG5: | TYPE | |
| 1785 | 007766 | 013546 | | PSTIT | |
| 1786 | 007770 | 004537 | | JSR 5,LINSEL | |
| 1787 | 007774 | 000005 | | RESET | |
| 1788 | 007776 | 004737 | | JSR 7,SETPAR | |
| 1789 | 010002 | 052777 | 000006 | 171170 | BIS #6,@RXCSR ;SET DTR RO TO SND |
| 1790 | 010010 | 104000 | PRG5A: | TYPE ;TYPE MESSAGE TO MAKE | |
| 1791 | 010012 | 014020 | | MAKCON ;LINE CONNECTION | |
| 1792 | 010014 | 000000 | | HALT ;WAIT FOR USER TO MAKE LINE CONNECTION | |
| 1793 | 010016 | 005777 | 171160 | TST @RXBUF ;READ BUFFER | |
| 1794 | 010022 | 032777 | 020000 | 171150 | BIT #BIT13,@RXCSR ;TEST FOR CLEAR TO SEND |
| 1795 | 010030 | 001003 | | BNE PRG5B | |
| 1796 | 010032 | 104000 | PRG5AA: | TYPE ;TYPE ERROR MESSAGE | |
| 1797 | 010034 | 014146 | | LINCHM | |
| 1798 | 010036 | 000764 | | BR PRG5A ;AND TRY AGAIN | |
| 1799 | 010040 | 104000 | PRG5B: | TYPE | |
| 1800 | 010042 | 014176 | | LINMAD | |
| 1801 | 010044 | 005037 | 010172 | CLR ERRCNT | |
| 1802 | 010050 | 012702 | 014252 | PRG5BB: MOV #PRG2M,%2 ;GET BASE ADDRESS OF DATA TO BE TRANSMITTED | |
| 1803 | 010054 | 112201 | | PRG5C: MOV (2)+,%1 ;GET A CHARACTER | |
| 1804 | 010056 | 020127 | 000045 | CMP %1,%1 ;WAS IT THE TERMINATOR? | |
| 1805 | 010062 | 001440 | | BEQ PRG5E | |
| 1806 | 010064 | 032737 | 000100 | 001374 | BIT #BIT6,SRT ;WAS PARITY OPTION SELECTED? |
| 1807 | 010072 | 001402 | | BEQ .+6 ;BRANCH IF NO PARITY DESIRED | |
| 1808 | 010074 | 004737 | 003656 | JSR 7,GENPAR ;GENERATE PARITY ON CHAR. IN R1 | |
| 1809 | 010100 | 032777 | 020000 | 171072 | BIT #BIT13,@RXCSR ;CHECK CLEAR TO SEND |
| 1810 | 010106 | 001751 | | BEQ PRG5AA ;TYPE ERROR MSG. IF NOT SET | |
| 1811 | 010110 | 010177 | 171072 | MOV %1,@TXBUF ;TRANSMIT THE CHARACTER | |
| 1812 | 010114 | 005777 | 171062 | TST @RXBUF ;ANY ERROR FLAGS? | |
| 1813 | 010120 | 100001 | | BPL .+4 ;BRANCH IF NO ERROR FLAGS | |
| 1814 | 010122 | 104003 | | ERROR ;ERROR! SOME ERROR FLAG IS SET | |
| 1815 | 010124 | 105777 | 171050 | TSTB @RXCSR ;WAIT FOR THE RECEIVER TO RECEIVE | |
| 1816 | 010130 | 100375 | | BPL .-4 ;THE TRANSMITTED CHARACTER | |
| 1817 | 010132 | 117703 | 171044 | MOV @RXBUF,%3 ;SAVE IT IN R3 | |
| 1818 | 010136 | 043701 | 001362 | BIC CARMSK,%1 ;CLEAR NON- TRANSMITTED BITS | |
| 1819 | 010142 | 120103 | | CMFB %1,%3 ;WAS RECEIVED & TRANSMITTED DATA THE SAME | |
| 1820 | 010144 | 001403 | | BEQ PRG5D | |
| 1821 | 010146 | 104003 | | ERROR ;ERROR! DATA ERROR | |
| 1822 | 010150 | 005237 | 010172 | INC ERRCNT | |
| 1823 | 010154 | 105777 | 171024 | PRG5D: TSTB @TXCSR ;WAIT FOR TRANSMITTER TO FINISH | |
| 1824 | 010160 | 100375 | | BPL .-4 | |
| 1825 | 010162 | 000734 | | BR PRG5C | |
| 1826 | 010164 | 104000 | | PRG5E: TYPE | |
| 1827 | 010166 | 014117 | | ENDPAS | |
| 1828 | 010170 | 000727 | | BR PRG5BB | |
| 1829 | 010172 | 000000 | | ERRCNT: OPEN | |
| 1830 | | | | ;THIS ROUTINE MOVES THE CONTENTS OF R2 TO THE ADDRESS SPECIFIED | |
| 1831 | | | | ;BY R1 | |
| 1832 | 010174 | 012737 | 000006 | 000004 | MOVIT: MOV #6,4 ;SET UP FOR RETURN |
| 1833 | 010202 | 012737 | 000002 | 000006 | MOV #2,6 |
| 1834 | 010210 | 012700 | 000037 | | MOV #31,%0 ;GET COUNTER |
| 1835 | 010214 | 010231 | | | MOVITA: MOV %2,@(1)+ ;MOVE THE DATA |

```
1836 010216 005300          DEC      %0          ;ALL DATA MOVED?
1837 010220 001375          BNE      MOVITA      ;NO. RETURN
1838 010222 012737 004576 000004  MOV      #ERTP,MACHER
1839 010230 012737 000040 000006  MOV      #40,MACHER+2
1840 010236 000205          RTS      5          ;RETURN
1841
1842          ;SUBROUTINE TO LOAD ALL VECTORS
1843 010240 012701 011606  LDVECS: MOV      #RISRO,%1
1844 010244 012702 011120          MOV      #VECTAB,%2
1845 010250 012703 000010          MOV      #10,%3
1846 010254 012704 000037          MOV      #31,%4
1847 010260 032712 000001  LDVECB: BIT      #BIT0,(2)          ;DOES THIS VECTOR EXIST
1848 010264 001002          BNE      LDVEC1      ;NO, SKIP LOADING
1849 010266 010172 000000          MOV      %1,%(2)    ;LOAD VECTOR
1850 010272 060301  LDVEC1: ADD     %3,%1
1851 010274 005722          TST     (2)+
1852 010276 005304          DEC     %4
1853 010300 001367          BNE     LDVECB
1854 010302 000207          RTS     7
1855
1856 010304 012701 012176  LDTVEC: MOV      #TISRO,%1
1857 010310 012702 011120          MOV      #VECTAB,%2
1858 010314 012703 000010          MOV      #10,%3
1859 010320 012704 000037          MOV      #31,%4
1860 010324 032712 000001  LDTVED: BIT      #BIT0,(2)          ;DOES THIS VECTOR EXIST
1861 010330 001003          BNE     LDVEC2      ;NO, SKIP LOADING
1862 010332 011200          MOV     (2),%0
1863 010334 010160 000004  LDVEC2: MOV     %1,%(0)
1864 010340 060301          ADD     %3,%1
1865 010342 005722          TST     (2)+
1866 010344 005304          DEC     %4
1867 010346 001366          BNE     LDTVED
1868 010350 000207          RTS     7
1869
1870          ;ROUTINE TO LOAD PRIORITY LEVEL 7 IN VECTOR +2
1871 010352 012701 011120  LDPRI: MOV      #VECTAB,%1          ;GET BASE VECTOR
1872 010356 012702 000340          MOV      #340,%2          ;GET LEVEL 7
1873 010362 012703 000037          MOV      #31,%3          ;LOAD COUNTER
1874 010366 032711 000001  LDPRIA: BIT      #BIT0,(1)          ;DOES THIS VECTOR EXIST
1875 010372 001003          BNE     LDPRIX      ;NO SKIP LOADING
1876 010374 011104          MOV     (1),%4          ;LOAD VECTOR +2
1877 010376 010264 000002  LDPRIX: MOV     %2,%(4)
1878 010402 005721          TST     (1)+          ;POINT TO NEXT VECTOR
1879 010404 005303          DEC     %3          ;DECREMENT COUNTER
1880 010406 001367          BNE     LDPRIA
1881 010410 000207          RTS     7
1882
1883
1884
1885
1886          ;*****
1887
1888          ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
1889          ;OF LOC 176. ROUTINE IS ENTERED AT CNTLU FOR START UP PURPOSES.
1890          ;WHEN A ^G IS GIVEN, THE PROGRAM ENTERS AT CKSWR. THE PROGRAM
1891          ;GETS CONTENTS OF SOFT. SWITCH REG., TYPES IT OUT, AND THEN SEEKS
```

```

1892      :NEW DATA FROM OPERATOR.  ONCE DATA IS SUPPLIED, IT INSERTS THIS
1893      :DATA INTO THE SOFT. SWITCH REG. AND RESUMES OPERATION IN THE
1894      :MAIN PROGRAM.  INCORRECT ENTRIES (SUCH AS 8,9,LETTERS) ARE DELETED
1895      :AND THE PROCESS RESTARTED.  TYPING ^U ALLOWS THE PRESENT ENTRY
1896      :TO BE DELETED AND THE PROCESS RESTARTED.
1897      :
1898      :
1899      :*****
1900
1901
1902
1903      010412      CHGC4:
1904      010412      022737      000176      000170      CKSWR:  CMP      #SWREG,SRPTR      ;SOFTWARE SW REG PRES?
1905      010420      001133      BNE      OUT              ;NO, GET OUT
1906      010422      105777      170572      TSTB     @TKS             ;YES, IS CHARACTER READY?
1907      010426      100130      BPL      OUT              ;IF NOT, GET OUT
1908      010430      017737      170566      001430      MOV      @TKB,TIB        ;STORE BUFFER
1909      010436      042737      177600      001430      BIC      #177600,TIB     ;STRIP OFF GARBAGE
1910      010444      022737      000007      001430      CMP      #7,TIB          ;IS IT A ^G
1911      010452      001116      BNE      OUT              ;IF NOT GET OUT
1912      010454      104000      TYPE
1913      010456      015455      CNTG
1914      010460      005077      170542      CLR      @TPB
1915      010464      104000      CNTLU:  TYPE              ;ALLOW SWR TO BE TYPED
1916      010466      015463      SWR
1917      010470      017746      167474      MOV      @SRPTR,-(SP)    ;MOV CONTENTS OF SWR
1918      010474      004737      010730      JSR      PC,TYPOC       ;OCTAL TYPE OUT ROUTINE
1919      010500      022600      CMP      (SP)+,R0       ;CORRECT STACK POINTER
1920      010502      104000      TYPE
1921      010504      015473      NEW
1922      010506      005037      001426      CLR      TEMPST         ;CLEAR TEMP STORAGE LOC
1923      010512      012737      000007      001416      MOV      #7,COUNT1      ;SET UP TO ACCEPT 7 CHAR
1924      010520      105777      170474      1$:      TSTB     @TKS             ;IS CHARACTER THERE?
1925      010524      100375      BPL      1$              ;IF NOT, TRY AGAIN
1926      010526      117737      170470      001430      MOVB    @TKB,TIB        ;PICKUP CHARACTER
1927      010534      105777      170464      8$:      TSTB     @TPS             ;CHECK PRINTER STATUS
1928      010540      100375      BPL      8$              ;NOT READY, TRY AGAIN
1929      010542      113777      001430      170456      MOVB    TIB,@TPB        ;PRINT IT
1930      010550      042737      177600      001430      BIC      #177600,TIB     ;STRIP OFF GARBAGE
1931      010556      122737      000025      001430      CMPB    #25,TIB         ;IS IT A ^U
1932      010564      001001      BNE      2$              ;BRANCH IF NOT
1933      010566      000736      3$:      BR       CNTLU           ;START OVER
1934      010570      122737      000015      001430      2$:      CMPB    #15,TIB         ;IS IT A <CR>
1935      010576      001006      BNE      4$              ;BRANCH IF NOT
1936      010600      104000      TYPE
1937      010602      015510      SCRLF
1938      010604      022737      000007      001416      CMP      #7,COUNT1      ;WAS <CR> FIRST CHAR
1939      010612      001033      BNE      7$              ;CHANGE SWREG IF NOT FIRST <CR>
1940      010614      122737      000060      001430      4$:      CMPB    #60,TIB         ;IS IT LESS THAN 0
1941      010622      003004      BGT      5$              ;GO TO ? ROUTINE IF SO
1942      010624      122737      000067      001430      CMPB    #67,TIB         ;IS IT GREATER THAN 7
1943      010632      002003      BGE      6$              ;GO TO ? ROUTINE IS SO
1944      010634      104000      5$:      TYPE
1945      010636      015503      QUEST
1946      010640      000752      BR       3$              ;START INPUT STRING OVER
1947      010642      006337      001426      6$:      ASL     TEMPST          ;MULTIPLY BY TEN

```


| | | | | | | | | | | |
|------|--------|--------|--------|--------|----------|------|----------------|--|--|--------------------------------------|
| 1948 | 010646 | 006337 | 001426 | | | ASL | TEMPST | | | |
| 1949 | 010652 | 006337 | 001426 | | | ASL | TEMPST | | | |
| 1950 | 010656 | 142737 | 000060 | 001430 | | BICB | #60,TIB | | | :CLEAR OF ASCII |
| 1951 | 010664 | 153737 | 001430 | 001426 | | BISB | TIB,TEMPST | | | :MOV CHAR TO TEMPST |
| 1952 | 010672 | 005337 | 001416 | | | DEC | COUNT1 | | | :ONLY WANT 6 NUMBERS AND <CR> |
| 1953 | 010676 | 001756 | | | | BEQ | 5\$ | | | :IF = 7 TOO MANY NUMBERS |
| 1954 | 010700 | 000707 | | | | BR | 1\$ | | | :GET NEXT CHAR |
| 1955 | 010702 | 013777 | 001426 | 167260 | 7\$: | MOV | TEMPST, @SRPTR | | | :CHANGE SWR CONTENTS |
| 1956 | 010710 | 000207 | | | OUT: | RTS | PC | | | :RETURN TO PROGRAM |
| 1957 | | | | | | | | | | |
| 1958 | | | | | | | | | | |
| 1959 | | | | | | | | | | |
| 1960 | 010712 | 010046 | | | TTINT'S: | MOV | R0,-(SP) | | | :INTERRUPT SERVICE ROUTINE |
| 1961 | 010714 | 010146 | | | | MOV | R1,-(SP) | | | :SAVE R0 AND R1 |
| 1962 | 010716 | 004737 | 010412 | | | JSR | PC,CKSWR | | | :GO TO SUBR TO SERVICE ITY INTERRUPT |
| 1963 | 010722 | 012601 | | | | MOV | (SP)+,R1 | | | :RESTORE R1 AND R0 |
| 1964 | 010724 | 012600 | | | | MOV | (SP)+,R0 | | | |
| 1965 | 010726 | 000002 | | | | RTI | | | | :RETURN FROM INTERRUPT |
| 1966 | | | | | | | | | | |
| 1967 | | | | | | | | | | |
| 1968 | | | | | | | | | | |
| 1969 | 010730 | 112737 | 000001 | 001422 | TYPOC: | MOVB | #1,FILL1 | | | :SET THE ZERO FILL SWITCH |
| 1970 | 010736 | 112737 | 000006 | 001425 | | MOVB | #6,MODE+1 | | | :SET FOR SIX (6) DIGITS |
| 1971 | 010744 | 112737 | 000005 | 001420 | TYPON: | MOVB | #5,CNT | | | :SET THE ITERATION COUNT |
| 1972 | 010752 | 010346 | | | | MOV | R3,-(SP) | | | :SAVE R3 |
| 1973 | 010754 | 010446 | | | | MOV | R4,-(SP) | | | :SAVE R4 |
| 1974 | 010756 | 010546 | | | | MOV | R5,-(SP) | | | :SAVE R5 |
| 1975 | 010760 | 113704 | 001425 | | | MOVB | MODE+1,R4 | | | :GET THE NUMBER OF DIGITS TO TYPE |
| 1976 | 010764 | 005404 | | | | NEG | R4 | | | |
| 1977 | 010766 | 062704 | 000006 | | | ADD | #6,R4 | | | :SUBTRACT IT FOR MAX. ALLOWED |
| 1978 | 010772 | 110437 | 001424 | | | MOVB | R4,MODE | | | :SAVE IT FOR USE |
| 1979 | 010776 | 113704 | 001422 | | | MOVB | FILL1,R4 | | | :GET THE ZERO FILL SWITCH |
| 1980 | 011002 | 016605 | 000010 | | | MOV | 10(SP),R5 | | | :PICKUP THE INPUT NUMBER |
| 1981 | 011006 | 005003 | | | | CLR | R3 | | | :CLEAR THE OUTPUT WORD |
| 1982 | 011010 | 006105 | | | 1\$: | ROL | R5 | | | :ROTATE MSB INTO 'C' |
| 1983 | 011012 | 000404 | | | | BR | 3\$ | | | :GO DO MSB |
| 1984 | 011014 | 006105 | | | 2\$: | ROL | R5 | | | :FORM THIS DIGIT |
| 1985 | 011016 | 006105 | | | | ROL | R5 | | | |
| 1986 | 011020 | 006105 | | | | ROL | R5 | | | |
| 1987 | 011022 | 010503 | | | | MOV | R5,R3 | | | |
| 1988 | 011024 | 006103 | | | 3\$: | ROL | R3 | | | :GET LSB OF THIS DIGIT |
| 1989 | 011026 | 105337 | 001424 | | | DECB | MODE | | | :TYPE THIS DIGIT? |
| 1990 | 011032 | 100020 | | | | BPL | 7\$ | | | :BR IF NO |
| 1991 | 011034 | 042703 | 177770 | | | BIC | #177770,R3 | | | :GET RID OF JUNK |
| 1992 | 011040 | 001002 | | | | BNE | 4\$ | | | :TEST FOR 0 |
| 1993 | 011042 | 005704 | | | | TST | R4 | | | :SUPPRESS THIS 0 |
| 1994 | 011044 | 001403 | | | | BEQ | 5\$ | | | :BR IF YES |
| 1995 | 011046 | 005204 | | | 4\$: | INC | R4 | | | :DON'T SUPPRESS ANYMORE 0'S |
| 1996 | 011050 | 052703 | 000060 | | | BIS | #60,R3 | | | :MAKE THIS DIGIT ASCII |
| 1997 | 011054 | 105777 | 170144 | | 5\$: | TSTB | @TPS | | | :IS PRINTER READY FOR CHARACTER? |
| 1998 | 011060 | 100375 | | | | BPL | 5\$ | | | :IF NOT, TRY AGAIN |
| 1999 | 011062 | 110377 | 170140 | | | MOVB | R3,@TPB | | | :TYPE OUT NUMBER |
| 2000 | 011066 | 105777 | 170132 | | 8\$: | TSTB | @TPS | | | :MAKE SURE LAST DIGIT TYPES |
| 2001 | 011072 | 100375 | | | | BPL | 8\$ | | | |
| 2002 | 011074 | 105337 | 001420 | | 7\$: | DECB | CNT | | | :COUNT BY 1 |
| 2003 | 011100 | 003345 | | | | BGT | 2\$ | | | :BR IF MORE TO DO |

```
2004 011102 002402          BLT      6$          ;BR IF DONE
2005 011104 005204          INC      R4          ;INSURE LAST DIGIT ISN'T A BLANK
2006 011106 000742          BR       2$          ;GO DO THE LAST DIGIT
2007 011110 012605      6$: MOV    (SP)+,R5    ;RESTORE R5
2008 011112 012604          MOV    (SP)+,R4    ;RESTORE R4
2009 011114 012603          MOV    (SP)+,R3    ;RESTORE R3
2010 011116 000207          RTS     PC          ;RETURN FROM INTERRUPT PC
2011                          ;*****
```

```
2012
2013
2014          ;VECTOR ASSIGNMENT TABLE
2015          VECTAB: 301          ;LINE 0 VECTOR
2016          311          ;LINE 1 VECTOR
2017          321          ;LINE 2 VECTOR
2018          331          ;LINE 3 VECTOR
2019          341          ;LINE 4 VECTOR
2020          351          ;LINE 5 VECTOR
2021          361          ;LINE 6 VECTOR
2022          371          ;LINE 7 VECTOR
2023          401          ;LINE 10 VECTOR
2024          411          ;LINE 11 VECTOR
2025          421          ;LINE 12 VECTOR
2026          431          ;LINE 13 VECTOR
2027          441          ;LINE 14 VECTOR
2028          451          ;LINE 15 VECTOR
2029          461          ;LINE 16 VECTOR
2030          471          ;LINE 17 VECTOR
2031          501          ;LINE 20 VECTOR
2032          511          ;LINE 21 VECTOR
2033          521          ;LINE 22 VECTOR
2034          531          ;LINE 23 VECTOR
2035          541          ;LINE 24 VECTOR
2036          551          ;LINE 25 VECTOR
2037          561          ;LINE 26 VECTOR
2038          571          ;LINE 27 VECTOR
2039          601          ;LINE 30 VECTOR
2040          611          ;LINE 31 VECTOR
2041          621          ;LINE 32 VECTOR
2042          631          ;LINE 33 VECTOR
2043          641          ;LINE 34 VECTOR
2044          651          ;LINE 35 VECTOR
2045          661          ;LINE 36 VECTOR
```

```
2046          ;DL11-E REGISTER ADDRESSES
2047          000000
2048          000000
2049 011216 000037      RCSR:  .REPT  31.
2050          RRCV    \N,\A
2051          N=N+10
2052          A-A+1
2053          .ENDR
2054          N=0
2055          A=0
2056 011314 000037      RBUF:  .REPT  31.
2057          RBUF    \N,\A
2058          N=N+10
2059          A-A+1
```

```

2060                                  .ENDR
2061                                N=0
2062                                A=0
2063 011412 000037                TCSR:  .REPT   31.
2064                                TXMT   \N,\A
2065                                N=N+10
2066                                A=A+1
2067                                  .ENDR
2068                                N=0
2069                                A=0
2070 011510 000037                TBUF:  .REPT   31.
2071                                TBUF   \N,\A
2072                                N=N+10
2073                                A=A+1
2074                                  .ENDR
2075                                N=0
2076                                .REPT   31.
2077                                ISP    \N
2078                                N=N+1
2079                                  .ENDR
2080                                N=0
2081                                .REPT   32.
2082                                ISRT   \N
2083                                N=N+1
2084                                  .ENDR
2085                                ;MESSAGES
2086 012576 041445 042132 041114  MTITLE: .ASCII '%CZDLBCO DL11-E ON LINE TEST%'
2087 012604 030103 042040 030514
2088 012612 026461 020105 047117
2089 012620 046040 047111 020105
2090 012626 042524 052123    045
2091 012633    045 040515 020120    .ASCII '%MAP OF DEVICES PRESENT%'
2092 012640 043117 042040 053105
2093 012646 041511 051505 050040
2094 012654 042522 042523 052116
2095 012662    045
2096 012663    045 044514 042516    .ASCII '%LINE D-ADR TRAP AT%a%'
2097 012670 020040 026504 042101
2098 012676 020122 020040 020040
2099 012704 051124 050101 040440
2100 012712 022524    100
2101 012715    040 020040 020040  MLINE: .ASCII ' '
2102 012722    040
2103 012723    040 020040 020040  MDADR: .ASCII ' '
2104 012730 020040 020040    040
2105 012735    040 020040 020040  MTRAP: .ASCII ' %a'
2106 012742 022440    100
2107 012745    045 047516 042516  MNONE: .ASCII '%NONE FOUND%a%'
2108 012752 043040 052517 042116
2109 012760 040045
2110 012762 052045 050131 020105  MSWSEL: .ASCII '%TYPE IN PROGRAM NUMBER %a'
2111 012770 047111 050040 047522
2112 012776 051107 046501 047040
2113 013004 046525 042502 020122
2114 013012 020040 040040
2115 013016 042445 051122 051117  MTERR: .ASCII '%ERROR - UNEXPECTED TRAP'

```

| | | | | | | | |
|------|--------|--------|--------|--------|---------|---------------------|---|
| 2116 | 013024 | 026440 | 052440 | 042516 | | | |
| 2117 | 013032 | 050130 | 041505 | 042524 | | | |
| 2118 | 013040 | 020104 | 051124 | 050101 | | | |
| 2119 | 013046 | 052045 | 040522 | 050120 | .ASCII | '%TRAPPED TO ' | |
| 2120 | 013054 | 042105 | 052040 | 020117 | | | |
| 2121 | 013062 | 040 | | | | | |
| 2122 | 013063 | 040 | 020040 | 020040 | MTO: | .ASCII | ' ' |
| 2123 | 013070 | 020040 | 040 | | | | |
| 2124 | 013073 | 045 | 051124 | 050101 | .ASCII | '%TRAPPED FROM PC ' | |
| 2125 | 013100 | 042520 | 020104 | 051106 | | | |
| 2126 | 013106 | 046517 | 050040 | 020103 | | | |
| 2127 | 013114 | 040 | | | | | |
| 2128 | 013115 | 040 | 020040 | 020040 | MFROM: | .ASCII | ' @' |
| 2129 | 013122 | 020040 | 040040 | | | | |
| 2130 | 013126 | 020040 | 020040 | 020040 | CSRADD: | .ASCII | ' @' |
| 2131 | 013134 | 100 | | | | | |
| 2132 | 013135 | 045 | 040506 | 051514 | RINTM: | .ASCII | '%FALSE INT. RCVR@' |
| 2133 | 013142 | 020105 | 047111 | 027124 | | | |
| 2134 | 013150 | 051040 | 053103 | 040122 | | | |
| 2135 | 013156 | 043045 | 046101 | 042523 | TINTM: | .ASCII | '%FALSE INT XMIT@' |
| 2136 | 013164 | 044440 | 052116 | 054040 | | | |
| 2137 | 013172 | 044515 | 040124 | | | | |
| 2138 | 013176 | 050045 | 036503 | 040 | EMO: | .ASCII | '%PC= ' |
| 2139 | 013203 | 040 | 020040 | 020040 | APC: | .ASCII | ' @' |
| 2140 | 013210 | 020040 | 040040 | | | | |
| 2141 | 013214 | 020040 | 054124 | 051503 | ATXCSR: | .ASCII | ' TXCSR = ' |
| 2142 | 013222 | 020122 | 020075 | | | | |
| 2143 | 013226 | 020040 | 020040 | 020040 | ATXWAS: | .ASCII | ' @' |
| 2144 | 013234 | 100 | | | | | |
| 2145 | 013235 | 040 | 051040 | 041530 | ARXCSR: | .ASCII | ' RXCSR = ' |
| 2146 | 013242 | 051123 | 036440 | 040 | | | |
| 2147 | 013247 | 040 | 020040 | 020040 | ARXWAS: | .ASCII | ' @' |
| 2148 | 013254 | 040040 | | | | | |
| 2149 | 013256 | 022445 | 051120 | 030107 | POTIT: | .ASCII | '%PRGO - SINGLE CHAR LINE MODE TEST@' |
| 2150 | 013264 | 026440 | 051440 | 047111 | | | |
| 2151 | 013272 | 046107 | 020105 | 044103 | | | |
| 2152 | 013300 | 051101 | 046040 | 047111 | | | |
| 2153 | 013306 | 020105 | 047515 | 042504 | | | |
| 2154 | 013314 | 052040 | 051505 | 040124 | | | |
| 2155 | 013322 | 022445 | 051120 | 030507 | P1TIT: | .ASCII | '%PRG1 - SPEC BIN COUNT LINE MODE TEST@' |
| 2156 | 013330 | 026440 | 051440 | 042520 | | | |
| 2157 | 013336 | 020103 | 044502 | 020116 | | | |
| 2158 | 013344 | 047503 | 047125 | 020124 | | | |
| 2159 | 013352 | 044514 | 042516 | 046440 | | | |
| 2160 | 013360 | 042117 | 020105 | 042524 | | | |
| 2161 | 013366 | 052123 | 100 | | | | |
| 2162 | 013371 | 045 | 050045 | 043522 | P2TIT: | .ASCII | '%PRG2 - SPECIAL MESSAGE LINE MODE TEST@' |
| 2163 | 013376 | 020062 | 020055 | 050123 | | | |
| 2164 | 013404 | 041505 | 040511 | 020114 | | | |
| 2165 | 013412 | 042515 | 051523 | 043501 | | | |
| 2166 | 013420 | 020105 | 044514 | 042516 | | | |
| 2167 | 013426 | 046440 | 042117 | 020105 | | | |
| 2168 | 013434 | 042524 | 052123 | 100 | | | |
| 2169 | 013441 | 045 | 050045 | 043522 | P3TIT: | .ASCII | '%PRG3 - RECEIVE MESSAGE TEST@' |
| 2170 | 013446 | 020063 | 020055 | 042522 | | | |
| 2171 | 013454 | 042503 | 053111 | 020105 | | | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 2172 | 013462 | 042515 | 051523 | 043501 | |
| 2173 | 013470 | 020105 | 042524 | 052123 | |
| 2174 | 013476 | 100 | | | |
| 2175 | 013477 | 045 | 050045 | 043522 | P4TIT: .ASCII '%PRG4 - SPECIAL MESSAGE TEST (SPIRAL)@' |
| 2176 | 013504 | 020064 | 020055 | 050123 | |
| 2177 | 013512 | 041505 | 040511 | 020114 | |
| 2178 | 013520 | 042515 | 051523 | 043501 | |
| 2179 | 013526 | 020105 | 042524 | 052123 | |
| 2180 | 013534 | 024040 | 050123 | 051111 | |
| 2181 | 013542 | 046101 | 040051 | | |
| 2182 | 013546 | 050045 | 043522 | 020065 | P5TIT: .ASCII '%PRG5 - DATA ECHO TEST USING MAYNARD FACILITY@' |
| 2183 | 013554 | 020055 | 040504 | 040524 | |
| 2184 | 013562 | 042440 | 044103 | 020117 | |
| 2185 | 013570 | 042524 | 052123 | 052440 | |
| 2186 | 013576 | 044523 | 043516 | 046440 | |
| 2187 | 013604 | 054501 | 040516 | 042122 | |
| 2188 | 013612 | 043040 | 041501 | 046111 | |
| 2189 | 013620 | 052111 | 040131 | | |
| 2190 | 013624 | 052045 | 050131 | 020105 | SELPAR: .ASCII '%TYPE IN PARAMETERS AS FOLLOWS:' |
| 2191 | 013632 | 047111 | 050040 | 051101 | |
| 2192 | 013640 | 046501 | 052105 | 051105 | |
| 2193 | 013646 | 020123 | 051501 | 043040 | |
| 2194 | 013654 | 046117 | 047514 | 051527 | |
| 2195 | 013662 | 072 | | | |
| 2196 | 013663 | 045 | 044502 | 030524 | .ASCII '%BIT1-0 = CHAR LENGTH@a' |
| 2197 | 013670 | 030055 | 036440 | 041440 | |
| 2198 | 013676 | 040510 | 020122 | 042514 | |
| 2199 | 013704 | 043516 | 044124 | 040045 | |
| 2200 | 013712 | 052045 | 050131 | 020105 | SELCAR: .ASCII '%TYPE TEST CHAR CODE IN BIT7-BITO OF AN OCTAL BYTE @' |
| 2201 | 013720 | 042524 | 052123 | 041440 | |
| 2202 | 013726 | 040510 | 020122 | 047503 | |
| 2203 | 013734 | 042504 | 044440 | 020116 | |
| 2204 | 013742 | 044502 | 033524 | 041055 | |
| 2205 | 013750 | 052111 | 020060 | 043117 | |
| 2206 | 013756 | 040440 | 020116 | 041517 | |
| 2207 | 013764 | 040524 | 020114 | 054502 | |
| 2208 | 013772 | 042524 | 020040 | 020040 | |
| 2209 | 014000 | 100 | | | |
| 2210 | 014001 | 045 | 046111 | 042514 | DTERR: .ASCII '%ILLEGAL DATA@a' |
| 2211 | 014006 | 040507 | 020114 | 040504 | |
| 2212 | 014014 | 040524 | 040045 | | |
| 2213 | 014020 | 046445 | 045501 | 020105 | MAKCON: .ASCII '%MAKE LINE CONNECTION.@' |
| 2214 | 014026 | 044514 | 042516 | 041440 | |
| 2215 | 014034 | 047117 | 042516 | 052103 | |
| 2216 | 014042 | 047511 | 027116 | 100 | |
| 2217 | 014047 | 040 | 040504 | 040524 | CERDAT: .ASCII ' DATA S/B: ' |
| 2218 | 014054 | 051440 | 041057 | 020072 | |
| 2219 | 014062 | 020040 | 020040 | 053440 | CSB: .ASCII ' WAS: ' |
| 2220 | 014070 | 051501 | 020072 | | |
| 2221 | 014074 | 020040 | 020040 | 041440 | CWAS: .ASCII ' CHAR NO. ' |
| 2222 | 014102 | 040510 | 020122 | 047516 | |
| 2223 | 014110 | 020056 | | | |
| 2224 | 014112 | 020040 | 020040 | 100 | CRNUM: .ASCII ' @' |
| 2225 | 014117 | 007 | | | ENDPAS: .BYTE 007 |
| 2226 | 014120 | 100 | | | .ASCII '@' |
| 2227 | 014121 | 045 | 040077 | | AINPRC: .ASCII '%?@' |

| | | | | | | |
|------|--------|--------|--------|--------|-----------------|--|
| 2228 | 014124 | 050045 | 051101 | 046501 | PARMTS: .ASCII | '%PARAMETERS - ' |
| 2229 | 014132 | 052105 | 051105 | 020123 | | |
| 2230 | 014140 | 020075 | | | | |
| 2231 | 014142 | 020040 | 040040 | | APARM: .ASCII | ' @' |
| 2232 | 014146 | 041440 | 042514 | 051101 | LINCHM: .ASCII | ' CLEAR TO SEND NOT SET.@' |
| 2233 | 014154 | 052040 | 020117 | 042523 | | |
| 2234 | 014162 | 042116 | 047040 | 052117 | | |
| 2235 | 014170 | 051440 | 052105 | 040056 | | |
| 2236 | 014176 | 046045 | 047111 | 020105 | LINMAD: .ASCII | '%LINE CONNECTION MADE.@' |
| 2237 | 014204 | 047503 | 047116 | 041505 | | |
| 2238 | 014212 | 044524 | 047117 | 046440 | | |
| 2239 | 014220 | 042101 | 027105 | 100 | | |
| 2240 | 014225 | 045 | 047503 | 043116 | ACONFIG: .ASCII | '%CONFIGURATION # ' |
| 2241 | 014232 | 043511 | 051125 | 052101 | | |
| 2242 | 014240 | 047511 | 020116 | 020043 | | |
| 2243 | 014246 | 020040 | 040040 | | TCONFIG: .ASCII | ' @' |
| 2244 | | | | | | |
| 2245 | 014252 | 015 | 012 | | PRG2M: .EVEN | |
| 2246 | 014254 | 044124 | 020105 | 052521 | .BYTE | 015,012 |
| 2247 | 014262 | 041511 | 020113 | 051102 | .ASCII | 'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK' |
| 2248 | 014270 | 053517 | 020116 | 047506 | | |
| 2249 | 014276 | 020130 | 052512 | 050115 | | |
| 2250 | 014304 | 042105 | 047440 | 042526 | | |
| 2251 | 014312 | 020122 | 044124 | 020105 | | |
| 2252 | 014320 | 040514 | 054532 | 042040 | | |
| 2253 | 014326 | 043517 | 020123 | 040502 | | |
| 2254 | 014334 | 045503 | | | | |
| 2255 | 014336 | 030040 | 031061 | 032063 | .ASCII | ' 0123456789.%' |
| 2256 | 014344 | 033065 | 034067 | 027071 | | |
| 2257 | 014352 | 045 | | | | |
| 2258 | 014353 | 045 | 044514 | 042516 | LFAIL: .ASCII | '%LINE FAILED%@' |
| 2259 | 014360 | 043040 | 044501 | 042514 | | |
| 2260 | 014366 | 022504 | 100 | | | |
| 2261 | 014371 | 045 | 053117 | 051105 | ROVER: .ASCII | '%OVERRUN, FRAME, OR PARITY ERROR%@' |
| 2262 | 014376 | 052522 | 026116 | 043040 | | |
| 2263 | 014404 | 040522 | 042515 | 020054 | | |
| 2264 | 014412 | 051117 | 050040 | 051101 | | |
| 2265 | 014420 | 052111 | 020131 | 051105 | | |
| 2266 | 014426 | 047522 | 022522 | 100 | | |
| 2267 | 014433 | 045 | 044514 | 042516 | ALINEX: .ASCII | '%LINE # ' |
| 2268 | 014440 | 021440 | 040 | | | |
| 2269 | 014443 | 040 | 020040 | 040527 | TLINEX: .ASCII | ' WAS SELECTED@' |
| 2270 | 014450 | 020123 | 042523 | 042514 | | |
| 2271 | 014456 | 052103 | 042105 | 100 | | |
| 2272 | 014463 | 045 | 044514 | 042516 | ALINE: .ASCII | '%LINE # ' |
| 2273 | 014470 | 021440 | 040 | | | |
| 2274 | 014473 | 040 | 020040 | 040527 | TLINE: .ASCII | ' WAS CALLED' |
| 2275 | 014500 | 020123 | 040503 | 046114 | | |
| 2276 | 014506 | 042105 | | | | |
| 2277 | 014510 | 052045 | 050131 | 020105 | .ASCII | '%TYPE IN OCTAL: CONFIGURATION IN BIT0-1 & MODEM TYPE' |
| 2278 | 014516 | 047111 | 047440 | 052103 | | |
| 2279 | 014524 | 046101 | 020072 | 047503 | | |
| 2280 | 014532 | 043116 | 043511 | 051125 | | |
| 2281 | 014540 | 052101 | 047511 | 020116 | | |
| 2282 | 014546 | 047111 | 041040 | 052111 | | |
| 2283 | 014554 | 026460 | 020061 | 020046 | | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| 2284 | 014562 | 047515 | 042504 | 020115 | |
| 2285 | 014570 | 054524 | 042520 | | |
| 2286 | 014574 | 044445 | 020116 | 044502 | .ASCII '%IN BIT2 (0=103, 1=202) @' |
| 2287 | 014602 | 031124 | 020040 | 030050 | |
| 2288 | 014610 | 030475 | 031460 | 020054 | |
| 2289 | 014616 | 036461 | 030062 | 024462 | |
| 2290 | 014624 | 020040 | 020040 | 020040 | |
| 2291 | 014632 | 100 | | | |
| 2292 | 014633 | 045 | 054524 | 042520 | WRU: .ASCII '%TYPE IN OCTAL IN BIT0-4 THE LINE # YOU ARE CALLING FROM @' |
| 2293 | 014640 | 044440 | 020116 | 041517 | |
| 2294 | 014646 | 040524 | 020114 | 047111 | |
| 2295 | 014654 | 041040 | 052111 | 026460 | |
| 2296 | 014662 | 020064 | 044124 | 020105 | |
| 2297 | 014670 | 044514 | 042516 | 021440 | |
| 2298 | 014676 | 054440 | 052517 | 040440 | |
| 2299 | 014704 | 042522 | 041440 | 046'01 | |
| 2300 | 014712 | 044514 | 043516 | 043040 | |
| 2301 | 014720 | 047522 | 020115 | 020040 | |
| 2302 | 014726 | 040040 | | | |
| 2303 | 014730 | 054445 | 052517 | 041440 | UR: .ASCII '%YOU CALLED FROM LINE # ' |
| 2304 | 014736 | 046101 | 042514 | 020104 | |
| 2305 | 014744 | 051106 | 046517 | 046040 | |
| 2306 | 014752 | 047111 | 020105 | 020043 | |
| 2307 | 014760 | 020040 | 100 | | URA: .ASCII '@' |
| 2308 | 014763 | 045 | 051120 | 051505 | BUTTON: .ASCII '%PRESS DATA BUTTON ON DATA PHONE@' |
| 2309 | 014770 | 020123 | 040504 | 040524 | |
| 2310 | 014776 | 041040 | 052125 | 047524 | |
| 2311 | 015004 | 020116 | 047117 | 042040 | |
| 2312 | 015012 | 052101 | 020101 | 044120 | |
| 2313 | 015020 | 047117 | 040105 | | |
| 2314 | 015024 | 052045 | 050131 | 020105 | LDLINE: .ASCII '%TYPE LINE NO. IN OCTAL IN BIT 3-7 @' |
| 2315 | 015032 | 044514 | 042516 | 047040 | |
| 2316 | 015040 | 027117 | 044440 | 020116 | |
| 2317 | 015046 | 041517 | 040524 | 020114 | |
| 2318 | 015054 | 047111 | 041040 | 052111 | |
| 2319 | 015062 | 031440 | 033455 | 020040 | |
| 2320 | 015070 | 020040 | 100 | | |
| 2321 | 015073 | 045 | 042522 | 047503 | MPWRF: .ASCII '%RECOVERED FROM POWER FAILURE@' |
| 2322 | 015100 | 042526 | 042522 | 020104 | |
| 2323 | 015106 | 051106 | 046517 | 050040 | |
| 2324 | 015114 | 053517 | 051105 | 043040 | |
| 2325 | 015122 | 044501 | 052514 | 042522 | |
| 2326 | 015130 | 100 | | | |
| 2327 | 015131 | 045 | 046111 | 042514 | MMODX: .ASCII '%ILLEGAL LINE NO.@' |
| 2328 | 015136 | 040507 | 020114 | 044514 | |
| 2329 | 015144 | 042516 | 047040 | 027117 | |
| 2330 | 015152 | 100 | | | |
| 2331 | 015153 | 045 | 046111 | 042514 | MMODD: .ASCII '%ILLEGAL DEVICE ADDRESS@' |
| 2332 | 015160 | 040507 | 020114 | 042504 | |
| 2333 | 015166 | 044526 | 042503 | 040440 | |
| 2334 | 015174 | 042104 | 042522 | 051523 | |
| 2335 | 015202 | 100 | | | |
| 2336 | 015203 | 045 | 054524 | 042520 | MMOD1: .ASCII '%TYPE IN OCTAL : BIT 0-5 THE LINE NUMBER OF DEVICE ADDRESS TO BE' |
| 2337 | 015210 | 044440 | 020116 | 041517 | |
| 2338 | 015216 | 040524 | 020114 | 026072 | |
| 2339 | 015224 | 044502 | 020124 | 026460 | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 2340 | 015232 | 020065 | 044124 | 020105 | |
| 2341 | 015240 | 044514 | 042516 | 047040 | |
| 2342 | 015246 | 046525 | 042502 | 020122 | |
| 2343 | 015254 | 043117 | 042040 | 053105 | |
| 2344 | 015262 | 041511 | 020105 | 042101 | |
| 2345 | 015270 | 051104 | 051505 | 020123 | |
| 2346 | 015276 | 047524 | 041040 | 105 | |
| 2347 | 015303 | 040 | 047515 | 044504 | .ASCII ' MODIFIED @' |
| 2348 | 015310 | 044506 | 042105 | 020040 | |
| 2349 | 015316 | 020040 | 100 | | |
| 2350 | 015321 | 045 | 054524 | 042520 | MMOD2: .ASCII '%TYPE IN NEW RXCSR DEVICE ADDRESS @' |
| 2351 | 015326 | 044440 | 020116 | 042516 | |
| 2352 | 015334 | 020127 | 054122 | 051503 | |
| 2353 | 015342 | 020122 | 042504 | 044526 | |
| 2354 | 015350 | 042503 | 040440 | 042104 | |
| 2355 | 015356 | 042522 | 051523 | 020040 | |
| 2356 | 015364 | 020040 | 100 | | |
| 2357 | 015367 | 045 | 054524 | 042520 | MMOD3: .ASCII '%TYPE IN 177777 TO CHANGE ANOTHER' |
| 2358 | 015374 | 044440 | 020116 | 033461 | |
| 2359 | 015402 | 033467 | 033467 | 020040 | |
| 2360 | 015410 | 047524 | 041440 | 040510 | |
| 2361 | 015416 | 043516 | 020105 | 047101 | |
| 2362 | 015424 | 052117 | 042510 | 122 | |
| 2363 | 015431 | 040 | 042504 | 044526 | .ASCII ' DEVICE ADDRESS @' |
| 2364 | 015436 | 042503 | 040440 | 042104 | |
| 2365 | 015444 | 042522 | 051523 | 020040 | |
| 2366 | 015452 | 020040 | 100 | | |
| 2367 | | | | | ;***** CHGCS ***** |
| 2368 | 015455 | 040 | 057040 | 020107 | CNTG: .ASCII / ^G @/ |
| 2369 | 015462 | 100 | | | |
| 2370 | 015463 | 040 | 051440 | 051127 | SWR: .ASCII / SWR= @/ |
| 2371 | 015470 | 020075 | 100 | | |
| 2372 | 015473 | 040 | 047040 | 053505 | NEW: .ASCII / NEW- @/ |
| 2373 | 015500 | 020075 | 100 | | |
| 2374 | 015503 | 040 | 037440 | 040040 | QUEST: .ASCII / ? @/ |
| 2375 | 015510 | 040045 | | | SCRLF: .ASCII /%@/ |
| 2376 | | | | | ;***** |
| 2377 | | | | | .EVEN |
| 2378 | 015512 | 000000 | | | OUTBUF: OPEN |
| 2379 | | 017462 | | | . =OUTBUF+1000. |
| 2380 | 017462 | 000000 | | | INBUF: OPEN |
| 2381 | | 021432 | | | . =INBUF+1000. |
| 2382 | | 015656 | | | BUFF=OUTBUF+100. |
| 2383 | 021432 | 000001 | | | DEND: .END |

| | | | | | | | | | | |
|----------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CONFIG | 001332 | 674# | 1363 | 1376 | 1435* | 1436* | 1455 | 1485 | 1522 | 1534 |
| CONN | 006326 | 1438 | 1472 | 1513# | | | | | | |
| COUNT | 001324 | 671# | 1036* | 1052* | 1053 | 1117* | 1121* | 1125* | 1129* | 1135* |
| COUNT1 | 001416 | 704# | 1923* | 1938 | 1952* | | | | | |
| CRNUM | 014112 | 912 | 2224# | | | | | | | |
| CRXTX | 003030 | 957 | 960# | | | | | | | |
| CRXTXA | 003102 | 960* | 961* | 969# | | | | | | |
| CRXTXB | 003104 | 956* | 959* | 970# | | | | | | |
| CRXTXC | 003110 | 963 | 972# | | | | | | | |
| CSB | 014062 | 907 | 2219# | | | | | | | |
| CSRADD | 013126 | 908 | 909 | 1417 | 1426 | 2130# | | | | |
| CTRD | 001364 | 688# | 911 | 1178* | 1181* | 1372* | 1381* | 1382 | 1400* | 1401 |
| CWAS | 014074 | 906 | 2221# | | | | | | | |
| DATCHK= | 104004 | 540# | 1380 | 1395 | 1399 | | | | | |
| DCERR | 005610 | 1342 | 1414# | | | | | | | |
| DECVAL | 004452 | 1198 | 1233# | | | | | | | |
| DELAY = | 104016 | 550# | 1313 | 1441 | 1476 | | | | | |
| DEND | 021432 | 2383# | | | | | | | | |
| DIGIT | 004434 | 1216* | 1219* | 1222* | 1223 | 1226# | | | | |
| DISCON | 006312 | 1424 | 1448 | 1482 | 1508# | | | | | |
| DISPLA | 000174 | 597# | 622 | | | | | | | |
| DISPRE | 000172 | 596# | 622* | | | | | | | |
| DLCNT | 003604 | 1072* | 1074 | 1084# | | | | | | |
| DLY | 003544 | 669 | 1072# | | | | | | | |
| DLYA | 003562 | 1076# | 1081 | | | | | | | |
| DLYB | 003566 | 1077# | 1078 | | | | | | | |
| DLYC | 003600 | 1075 | 1082# | | | | | | | |
| DOIT | 004020 | 1119 | 1123 | 1127 | 1131 | 1132# | | | | |
| DONE | 004054 | 1138 | 1140# | | | | | | | |
| DTCHK | 002470 | 659 | 902# | | | | | | | |
| DTCHKA | 002566 | 904 | 916# | | | | | | | |
| DTERR | 014001 | 1062 | 2210# | | | | | | | |
| ECDAT | 003424 | 1041# | 1042 | | | | | | | |
| EHALT = | 104010 | 544# | 937 | 972 | | | | | | |
| EHLT | 002456 | 663 | 896# | | | | | | | |
| EHLTA | 002466 | 897 | 899# | | | | | | | |
| EIGHT | 003710 | 1110 | 1116# | | | | | | | |
| EMTA | 002312 | 852# | | | | | | | | |
| EMTINT | 002300 | 494 | 849# | | | | | | | |
| EMITAB | 001264 | 655# | 854 | | | | | | | |
| EMO | 013176 | 933 | 968 | 2138# | | | | | | |
| ENDPAS | 014117 | 1409 | 1827 | 2225# | | | | | | |
| ERR | 002570 | 658 | 918# | | | | | | | |
| ERRA | 002640 | 921 | 926# | | | | | | | |
| ERRB | 002702 | 918* | 919* | 922* | 923* | 924* | 934# | | | |
| ERRC | 002710 | 927 | 937# | | | | | | | |
| ERRCNT | 010172 | 1801* | 1822* | 1829# | | | | | | |
| ERRD | 002720 | 928* | 929* | 931 | 940# | 964* | 965* | 967 | | |
| ERRE | 002722 | 920* | 925* | 938 | 941# | | | | | |
| ERROR = | 104003 | 539# | 1345 | 1446 | 1481 | 1489 | 1498 | 1520 | 1814 | 1821 |
| ERROR1 = | 104013 | 547# | 908 | 914 | | | | | | |
| ERRRX = | 104015 | 549# | 1319 | 1416 | 1425 | 1675 | 1679 | 1682 | | |
| ERRTX = | 104014 | 548# | 1706 | | | | | | | |
| ERR1 | 002612 | 666 | 922# | | | | | | | |
| ERR3A | 007306 | 1646 | 1675# | | | | | | | |
| ERR3B | 007322 | 1648 | 1679# | | | | | | | |

| | | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|-------|-------|------|-------|-------|
| ERR3C | 007330 | 1653 | 1682# | | | | | | | |
| ERTP | 004576 | 482 | 484 | 486 | 488 | 496 | 718 | 772 | 1257# | 1838 |
| ERTPA | 004646 | 1265# | 1284 | | | | | | | |
| FILL | 004224 | 1177# | 1405 | | | | | | | |
| FILLA | 004232 | 1179# | 1182 | | | | | | | |
| FILL1 | 001422 | 706# | 1969* | 1979 | | | | | | |
| FINISH | 005554 | 1384 | 1403# | | | | | | | |
| FIVE | 003776 | 1115 | 1128# | | | | | | | |
| FMAP | 001350 | 682# | 752* | 1283 | 1288* | | | | | |
| FNONE | 001346 | 681# | 725* | 769* | 774 | | | | | |
| FORMAD | 004762 | 751 | 1292# | | | | | | | |
| FROMPC | 001342 | 679# | 1264* | 1270 | 1281* | | | | | |
| FTITLE | 001344 | 680# | 714* | 720 | 724* | 778* | 845* | | | |
| GENPAR | 003656 | 1109# | 1614 | 1755 | 1808 | | | | | |
| INBUF | 017462 | 1355 | 1388 | 1419 | 2380# | 2381 | | | | |
| INBUFP | 001376 | 693# | 1353* | 1354* | 1355 | 1389* | 1419* | | | |
| INCPRG | 002432 | 653 | 654 | 885# | | | | | | |
| INDAT | 003400 | 1035# | 1064 | | | | | | | |
| INFIL | 004220 | 1176# | 1580 | | | | | | | |
| LDLINE | 015024 | 1147 | 2314# | | | | | | | |
| LDPRI | 010352 | 1562 | 1584 | 1871# | | | | | | |
| LDPRIA | 010366 | 1874# | 1880 | | | | | | | |
| LDPRIX | 010402 | 1875 | 1878# | | | | | | | |
| LDTVEC | 010304 | 1563 | 1585 | 1856# | | | | | | |
| LDTVED | 010324 | 1860# | 1867 | | | | | | | |
| LDVECB | 010260 | 1847# | 1853 | | | | | | | |
| LDVECS | 010240 | 1564 | 1586 | 1843# | | | | | | |
| LDVEC1 | 010272 | 1848 | 1850# | | | | | | | |
| LDVEC2 | 010340 | 1861 | 1864# | | | | | | | |
| LFAIL | 014353 | 1676 | 2258# | | | | | | | |
| LINCA | 005036 | 1305# | 1321 | | | | | | | |
| LINCB | 005054 | 1309# | 1311 | | | | | | | |
| LINCF | 005134 | 1318 | 1322# | | | | | | | |
| LINCHM | 014146 | 1320 | 1797 | 2232# | | | | | | |
| LINCON | 005020 | 1302# | 1607 | 1615 | 1629 | 1745 | 1756 | | | |
| LINE | 001330 | 673# | 1338* | 1429 | 1466* | 1468 | | | | |
| LINENO | 001354 | 684# | 726* | 730* | 747 | 759 | | | | |
| LINEUP | 005152 | 1304 | 1326# | | | | | | | |
| LINMAD | 014176 | 1325 | 1453 | 1800 | 2236# | | | | | |
| LINSA | 004134 | 1157# | 1160 | | | | | | | |
| LINSEL | 004072 | 1146# | 1604 | 1625 | 1740 | 1786 | | | | |
| MACHER | 000004 | 484# | 718* | 719* | 727* | 728* | 772* | 773* | 1838* | 1839* |
| MAKCON | 014020 | 1308 | 1570 | 1592 | 1791 | 2213# | | | | |
| MANUAL | 100000 | 509# | | | | | | | | |
| MAPA | 001534 | 730# | 740 | 771 | | | | | | |
| MAPEND | 001744 | 732 | 772# | | | | | | | |
| MAPERR | 001766 | 776# | | | | | | | | |
| MAPNE | 001564 | 727 | 738# | | | | | | | |
| MAPOK | 001574 | 737 | 741# | | | | | | | |
| MAPOKA | 001706 | 758 | 762# | | | | | | | |
| MAPVEC | 004704 | 490 | 1279# | | | | | | | |
| MDADR | 012723 | 744 | 2103# | | | | | | | |
| MFROM | 013115 | 1271 | 2128# | | | | | | | |
| MLINE | 012715 | 748 | 768 | 2101# | | | | | | |
| MMODD | 015153 | 822 | 2331# | | | | | | | |
| MMODX | 015131 | 814 | 2327# | | | | | | | |

| | | | | | | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| TISR11 | 012306 | 2085# | | | | | | | | | | | | |
| TISR12 | 012316 | 2085# | | | | | | | | | | | | |
| TISR13 | 012326 | 2085# | | | | | | | | | | | | |
| TISR14 | 012336 | 2085# | | | | | | | | | | | | |
| TISR15 | 012346 | 2085# | | | | | | | | | | | | |
| TISR16 | 012356 | 2085# | | | | | | | | | | | | |
| TISR17 | 012366 | 2085# | | | | | | | | | | | | |
| TISR2 | 012216 | 2085# | | | | | | | | | | | | |
| TISR20 | 012376 | 2085# | | | | | | | | | | | | |
| TISR21 | 012406 | 2085# | | | | | | | | | | | | |
| TISR22 | 012416 | 2085# | | | | | | | | | | | | |
| TISR23 | 012426 | 2085# | | | | | | | | | | | | |
| TISR24 | 012436 | 2085# | | | | | | | | | | | | |
| TISR25 | 012446 | 2085# | | | | | | | | | | | | |
| TISR26 | 012456 | 2085# | | | | | | | | | | | | |
| TISR27 | 012466 | 2085# | | | | | | | | | | | | |
| TISR3 | 012226 | 2085# | | | | | | | | | | | | |
| TISR30 | 012476 | 2085# | | | | | | | | | | | | |
| TISR31 | 012506 | 2085# | | | | | | | | | | | | |
| TISR32 | 012516 | 2085# | | | | | | | | | | | | |
| TISR33 | 012526 | 2085# | | | | | | | | | | | | |
| TISR34 | 012536 | 2085# | | | | | | | | | | | | |
| TISR35 | 012546 | 2085# | | | | | | | | | | | | |
| TISR36 | 012556 | 2085# | | | | | | | | | | | | |
| TISR37 | 012566 | 2085# | | | | | | | | | | | | |
| TISR4 | 012236 | 2085# | | | | | | | | | | | | |
| TISR5 | 012246 | 2085# | | | | | | | | | | | | |
| TISR6 | 012256 | 2085# | | | | | | | | | | | | |
| TISR7 | 012266 | 2085# | | | | | | | | | | | | |
| TKB | 001222 | 638# | 789 | 1039 | 1908 | 1926 | | | | | | | | |
| TKLVL | 001232 | 642# | 788 | | | | | | | | | | | |
| TKS | 001220 | 637# | 790* | 1037 | 1906 | 1924 | | | | | | | | |
| TKVTR | 001230 | 641# | 786 | | | | | | | | | | | |
| TLINE | 014473 | 1429 | 2274# | | | | | | | | | | | |
| TLINEX | 014443 | 1171 | 2269# | | | | | | | | | | | |
| TOPC | 001340 | 678# | 1263* | 1266 | 1279* | 1282* | 1285 | 1286* | 1287 | | | | | |
| TPB | 001226 | 640# | 1006* | 1043* | 1719* | 1914* | 1929* | 1999* | | | | | | |
| TPINT | 007500 | 1636 | 1719# | | | | | | | | | | | |
| TPINTA | 007520 | 1721 | 1723 | 1725# | | | | | | | | | | |
| TPLVL | 001236 | 644# | 1637 | | | | | | | | | | | |
| TPS | 001224 | 639# | 1007 | 1041 | 1673* | 1702 | 1703* | 1704 | 1708 | 1710* | 1714 | 1725* | 1927 | 1997 |
| | | 2000 | | | | | | | | | | | | |
| TPVTR | 001234 | 643# | 1635 | | | | | | | | | | | |
| TTINTS | 010712 | 787 | 1960# | | | | | | | | | | | |
| TXBUF | 001206 | 632# | 1298* | 1616* | 1658* | 1693* | 1757* | 1774* | 1777* | 1811* | | | | |
| TXCSR | 001204 | 631# | 753* | 754* | 762* | 1296* | 1450* | 1501* | 1617 | 1656 | 1672* | 1690 | 1713* | 1726 |
| | | 1758 | 1775 | 1778 | 1823 | | | | | | | | | |
| TXCSRT | 001366 | 689# | 956 | 1690* | 1691 | | | | | | | | | |
| TXERR | 002766 | 667 | 955# | | | | | | | | | | | |
| TXLVL | 001216 | 636# | 989 | | | | | | | | | | | |
| TXVTR | 001214 | 635# | 987 | 1285* | | | | | | | | | | |
| TYP | 003200 | 655 | 993# | | | | | | | | | | | |
| TYPA | 003214 | 997# | 1005 | 1014 | | | | | | | | | | |
| TYPC | 003236 | 999 | 1002# | | | | | | | | | | | |
| TYPD | 003254 | 1004 | 1006# | 1011 | 1013 | | | | | | | | | |
| TYPDAT | 003320 | 997* | 998 | 1002 | 1006 | 1010* | 1012* | 1015# | | | | | | |

(ZDLBCO DL1*-E ON LINE TSET
CZDLBC.P11 19-JUL-79 15:06

MACY11 30A(1052) 19-JUL-79 15:10 PAGE 59
(CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0057

| | | | | | | | | | | | | | |
|-------|------|------|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CNVOA | 554# | 905 | 906 | 907 | 930 | 955 | 958 | 966 | 1170 | 1240 | 1428 | 1454 | 1467 |
| ISR | 561# | 2080 | | | | | | | | | | | |
| ISRT | 567# | 2085 | | | | | | | | | | | |
| RBUF | 577# | 2061 | | | | | | | | | | | |
| RRCV | 573# | 2054 | | | | | | | | | | | |
| TBUF | 585# | 2075 | | | | | | | | | | | |
| TXMT | 581# | 2068 | | | | | | | | | | | |

. ABS. 021432 000

ERRORS DETECTED: 0

.CZDLBC.LST/CRF/SOL/NL:TOC=CZDLBC.P11
RUN-TIME. 10 17 3 SECONDS
RUN-TIME RATIO: 91/31=2.9
CORE USED: 10K (19 PAGES)