

.REPT 0

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

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDLB-B=D
PRODUCT NAME: DL11-E ON LINE TESTS
DATE: 21-DECEMBER-1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: ROBERT WHITTON
REVISED BY: AL LOSCHAK JUN-75
M. MCNALLY DEC-75 ;SUPPORT SOFTWARE SWITCH REGISTE

THIS MAINDEC OBSOLETE MAINDEC-11-DZDLBA

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1972,1975

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

49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

1. ABSTRACT

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

THREE STARTING ADDRESSES ARE PROVIDED, THEY ARE:

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

THIS DOCUMENT DESCRIBES THE ON LINE TESTS.

THE AVAILABLE TESTS ARE:

- PRG0 SINGLE CHARACTER LINE MODE DATA TEST
- PRG1 BINARY COUNT LINE MODE DATA TEST
- PRG2 MESSAGE TRANSMIT ONLY W/W/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

92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

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)

121
122
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

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.

```
*****  
* * * * *  
* DATA SET * * * * * DATA SET *  
* * * * *  
*****  
* * * * *  
* * * * *  
* * * * *  
*****  
* * * * *  
* DL11-E * * * * * REMOTE *  
* * * * * * * * * *  
* * * * * * * * * *  
*****
```

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.).

215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251

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

3.5 LINE NUMBERS

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

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

252
253
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

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.

IT'S THE RESPONSIBILITY OF THE OPERATOR TO SET UP
THIS LOCATION PRIOR TO STARTING THE PROGRAM.

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
BITS	1/0	TRANSMIT ODD/EVEN PARITY

E. WHEN 'MAKE LINE CONNECTION' IS TYPED CALL THE DL11-E

306
307
308
309

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.

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

- 4.4 PRG3 - RECEIVE TRANSMIT MESSAGE TEST
- A. LOAD ADDRESS = 000200
 - B. START - PROGRAM WILL REQUEST PROGRAM NUMBER
 - C. THE PROGRAM WILL IDENTIFY ITSELF AND TYPE INSTRUCTIONS TO SELECT DESIRED OPTIONS.
 - D. SET IN OPTIONS AND PRESS CONTINUE.
 - E. WHEN 'MAKE LINE CONNECTION' IS TYPED CALL THE DL11-E YOU WISH TO TRANSMIT ON. WHEN THE 'HANDSHAKING' IS COMPLETED THE DL11-E WILL TRANSMIT A CRLF TO THE TERMINAL DEVICE. AT THIS TIME YOU MAY BEGIN TO SEND DATA FROM THE DEVICE TO THE DL11-E WHERE IT WILL BE ECHOED BACK TO THE TERMINAL. TYPE ANY CHARACTER TO SIGNAL START OF MESSAGE. THEN TYPE MESSAGE AND THE SAME CHARACTER TO SIGNAL END OF MESSAGE. CONTROL C WILL CAUSE THE BUFFERS CONTENTS TO BE TRANSMITTED WHEN TYPED.
 - F. IF NO ECHO IS DESIRED (ON A CHARACTER BASIS FOR EXAMPLE WHEN USING A TERMINAL THAT PRODUCES ITS OWN LOCAL COPY) SET BIT7 OF SWITCH REGISTER.
- 4.5 PRG4 - SPECIAL MESSAGE XMIT ONLY
- A. LOAD ADDRESS = 000200
 - B. OPTIONS
 - 1. BITS 0-2 = 4
 - 2. BITS 3-6 = LINE NUMBER (SEE SECT 3.5)
 - C. DEPRESS START - THE PROGRAM WILL IDENTIFY ITSELF AND TYPE INSTRUCTIONS TO SELECT DESIRED DL11-E PARAMETERS (SEE SECT 3.2)
 - D. SET IN PARAMETERS IF IT IS DESIRED TO TRANSMIT DATA WITH PARITY RAISE SR6. ALSO RAISE SR5 TO TRANSMIT ODD PARITY AND LOWER TO TRANSMIT EVEN PARITY.
 - BIT6 1/0 ENABLE/DISABLE PARITY
 - BIT5 1/0 TRANSMIT ODD/EVEN PARITY
 - E. WHEN 'MAKE LINE CONNECTION' IS TYPED CALL THE DL11-E YOU WISH TO TRANSMIT ON FROM THE TERMINAL MODEM. WHEN THE 'HANDSHAKING' IS COMPLETED A SPIRAL PATTERN WILL BE TRANSMITTED. TO TERMINATE, HANG UP.

358 5. PROGRAM DESCRIPTIONS
359
360 5.1 PRG0 - SINGLE CHARACTER LINE MODE DATA TEST
361
362 PRG0 TRANSMITS USER SPECIFIED DATA AND A CARRIAGE RETURN/LINE FEED
363 EVERY 72ND CHARACTER.
364
365 5.2 PRG1 - BINARY COUNT PATTERN LINE MODE DATA TEST
366
367 PRG1 TRANSMITS A BINARY COUNT PATTERN. THIS PROGRAM IS THE SAME
368 AS PRG0 EXCEPT FOR THE DATA TRANSMITTED.
369
370 5.3 PRG2 - SPECIAL MESSAGE TRANSMIT ONLY
371
372 PRG2 TRANSMITS THE MESSAGE
373 THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789.
374 NO DATA ERROR CHECKING IS PERFORMED BY THE PROGRAM.
375
376 5.4 PRG3 - RECEIVE/TRANSMIT MESSAGE TEST
377
378 PRG3 - RECEIVES DATA FROM A TERMINAL AND READS THE RECEIVED MESSAGE
379 BACK, AND TYPES THE MESSAGE ON THE PDP-11 TTY WHEN THE MESSAGE
380 IS TERMINATED. CHARACTERS MAY BE ECHOED BACK (IF REQUIRED) ON
381 A CHARACTER BASIS THEREBY CREATING LOCAL COPY AS THE MESSAGE
382 IS TYPED.
383 TRANSMISSION MAY BEGIN AT THE TERMINAL WHEN A CR/LF IS RECEIVED
384 AT THE TERMINAL. THIS PROGRAM IS RESTRICTED TO USE BY ONLY
385 FULL DUPLEX MODEMS.
386
387 5.5 PRG4 - SPECIAL MESSAGE TRANSMIT ONLY
388
389 PRG4 TRANSMITS A SPIRAL PATTERN.
390 NO DATA CHECKING IS PERFORMED BY THE PROGRAM.

391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440

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/R 301 WAS 321 CHAR NO 23

THIS TIMEOUT INDICATES A DATA ERROR ON LINE 1 IF CONFIGURATION 2 IS SELECTED TWO ERROR TIMEOUTS 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 TIMEOUTS WILL OCCUR AS SHOWN BELOW:

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

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

.ENDR

```

441 .TITLE OZDLBB
442 .ENABLE A88,AMA
443 ;THIS TEST CHECKS THE DL11-E USING MODEMS
444 ;REFER ALSO TO TEST OZDLA (DL11-E OFF LINE TESTS)
445 ;STARTING PROCEDURE
446 ; LOAD ADDRESS 200
447 ;
448 ; STACK POINTER IS AT 1200
449 ; PRESS START
450 ;
451 ;AVAILABLE PROGRAMS
452 ; PRG0= SINGLE CHARACTER LINE MODE DATA TEST.
453 ; PRG1= SPECIAL BINARY COUNT LINE MODE DATA TEST.
454 ; PRG2= SPECIAL MESSAGE XMIT ONLY W/W/O PARITY
455 ; PRG3= RECEIVE DATA TEST
456 ; PRG4= SPIRAL PATTERN MESSAGE XMIT ONLY W/W/O PARITY
457 ; PRG5= DATA ECHO TEST (USES FACILITY AT MAYNARD)
458
459 ;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
460 ;SR15= HALT ON ERROR.
461 ;SR14= SCOPE (NOT USED)
462 ;SR13= INHIBIT PRINTOUT
463 ;SR12= INHIBIT TRACE (NOT USED)
464 ;SR11= INHIBIT ITERATION (NOT USED)
465
466 000000 004520 ERTF ;UNASSIGNED TRAP
467 000002 000000 0
468 000004 004520 MACHERR: ERTF ;SP OVERFLOW, BUS ERROR TRAP
469 000006 000040 40
470 000010 004520 ERTF ;RESERVED INSTRUCTION TRAP
471 000012 000100 100
472 000014 004520 ERTF ;TRACE TRAP
473 000016 000140 140
474 000020 004626 MAPVEC ;TRAP TO MAP VECTOR
475 000022 000340 PRTY7
476 000024 002646 PFAIL ;POWER FAIL TRAP
477 000026 000340 PRTY7
478 000030 002222 EMTINT ;EMT TRAP
479 000032 000340 PRTY7
480 000034 004520 ERTF
481 000036 000340 340
482 000040 000042 .+2
483 000042 000000 HALT
484 .REPT 119.
485 .+2
486 4 ;TRAP TO MAP MAKER
487 .ENDR
488
489 ;EQUATE STATEMENTS
490 PSW=17776
491 STKPTR=1200
492 OPEN=0
493 MANUAL=BIT15
494 BIT15=100000
  
```

```

495 040000 BIT14=40000
496 020000 BIT13=20000
497 010000 BIT12=10000
498 004000 BIT11=4000
499 002000 BIT10=2000
500 001000 BIT9=1000
501 000400 BIT8=400
502 000200 BIT7=200
503 000100 BIT6=100
504 000040 BIT5=40
505 000020 BIT4=20
506 000010 BIT3=10
507 000004 BIT2=4
508 000002 BIT1=2
509 000001 BIT0=1
510 005726 POPSP=5726 ;POP THE STACK. SAME AS TST (6)+
511 022626 POPSP2=022626 ;POP STACK TWICE. SAME AS CMP (6)+,(6)+
512 000340 PRTY7=340 ;PRIORITY LEVEL DEFINITIONS
513 000300 PRTY6=300
514 000240 PRTY5=240
515 000200 PRTY4=200
516 000140 PRTY3=140
517 000100 PRTY2=100
518 000040 PRTY1=40
519 000000 PRTY0=0
520 104000 TYPE=EMT+0
521 104001 TYPE3=EMT+1
522 104002 STALL=EMT+2
523 104003 ERROR=EMT+3
524 104004 DATCHK=EMT+4
525 104005 CHALTY=EMT+5
526 104006 STRXV=EMT+6
527 104007 STTXV=EMT+7
528 104010 EMHLT=EMT+10
529 104011 SAVREG=EMT+11
530 104012 RSTREG=EMT+12
531 104013 ERROR1=EMT+13
532 104014 ERRTY=EMT+14
533 104015 ERRRN=EMT+15
534 104016 DELAY=EMT+16
535 000000 N=0
536 000000 A=0
537
538 .MACR CNVOA SRC,DST,COUNT
539 JBR X5,0A0V ;SD TO OCTAL TO ASCII CONVERT.
540 SRC ;SOURCE ADDR.
541 DST ;DESTINATION ADDR.
542 COUNT ;#OF DIGITS TO CONVERT.
543 .ENDM
544
545 .MACR ISR N
546 RISR'N' JMR MOV S'N',XB
547 JMR RISR ;GO TO COMMON INTERRUPT SERVICE ROUTINE
548
  
```

```

549 .ENDM
550 .MACR ISRT N
551 TISR'N': #N',X0 IPUT LINE # IN R0
552 JMP TISR IGO TO COMMON INTERRUPT SERVICE
553 .ENDM
554 .MACR RRCV N,A
555 175610+N IADDRESS OF RECEIVER LINE # 'A'
556 .ENDM
557 .MACR RBUF N,A
558 175612+N IADDRESS OF RECEIVER BUFFER LINE # 'A'
559 .ENDM
560 .MACR TXMT N,A
561 175614+N IADDRESS OF TRANSMITTER CSR LINE # 'A'
562 .ENDM
563 .MACR TBUF N,A
564 175616+N IADDRESS OF TRANSMIT BUFFER LINE # 'A'
565 .ENDM
566
567
568
569
570
571
572
573
574 .=174
575 000174 177570 SRPTR: 177570
576 000176 000000 SOFTSR: 000000
577 000200 #200
578 000200 012737 001022 000300 MOV #START,STAD ISET UP FOR NORMAL START
579 000206 000407 BR STCONT ICONTINUE
580 000210 012737 001416 000300 MOV #REMAP,STAD ISET FOR A REMAP START
581 000216 000403 BR STCONT
582 000220 012737 000204 000300 MOV #MODEV,STAD ISET TO MODIFY ON START
583 000226 STCONT:
584 000226 012706 001200 MOV #STKPTR,X6 ISET BOTTOM OF STACK
585 000232 013746 000006 MOV 6,-(SP) ISAVE CURRENT VECTOR
586 000236 013746 000004 MOV 4,-(SP)
587 000242 012737 000256 000004 MOV #18,4 ISET UP TIME OUT VECTOR
588 000250 005777 177720 TBT #SRPTR ITRY TO REFERENCE THE
589 BR 28 IHardware SWITCH REGISTER
590 000254 000404 BR 28 IBRANCH IF NO TIME OUT TRAP OCCURRS
591 000256 012737 000176 000174 131 MOV #SOFTSR,SRPTR ICHANGE THE SWITCH REGISTER
592 CMP (6)+,(6)+ ITO POINT TO A SOFTWARE SWITCH REGISTER
593 000264 022626 291 MOV (6)+,4 IRESTORE THE STACK
594 000266 012637 000004 MOV (6)+,6 IRESTORE TIME OUT VECTOR
595 000272 012637 000006 JMP #PC)+ IJUMP TO SELECTED START
596 000276 000137 STAD: 0
597 000300 000000 BR 0
598 001200 #1200
599 001200 000000 RXCSR: 0 IRECEIVER CSR UNDER TEST
600 001202 000000 RXBUF: 0 IRECEIVER BUFFER UNDER TEST
601 001204 000000 TXCSR: 0 ITRANSMITTER CSR UNDER TEST
602 001206 000000 TXBUF: 0 ITRANSMITTER BUFFER UNDER TEST
    
```

```

603 001210 000000 RXVTR: OPEN IRECEIVER VECTOR
604 001212 000200 RXLVL: PRTY4 IRECEIVER PRIORITY LEVEL
605 001214 000000 TXVTR: OPEN ITRANSMITTER VECTOR
606 001216 000200 TXLVL: PRTY4 ITRANSMITTER PRIORITY LEVEL
607 001220 177560 TK8: 177560 ILSR CSR
608 001222 177562 TK9: 177562 ILSR BUFFER
609 001224 177564 TP8: 177564 ILSR CSR
610 001226 177566 TP9: 177566 ILSR BUFFER
611 001230 000060 TKVTR: 60 ILSR INTERRUPT VECTOR
612 001232 000200 TKLVL: PRTY4 ILSR PRIORITY LEVEL
613 001234 000064 TPVTR: 64 ILSR INTERRUPT VECTOR
614 001236 000200 TPLVL: PRTY4 ILSR PRIORITY LEVEL
615 001240 000000 PRGNUM: OPEN ICONTAINS CURRENT PROGRAMS
616 001242 000000 PRGID: OPEN ICONTAINS PROGRAM INDICATORS
617 001244 006424 PRGTAB: PRG0 IPRG0 START ADDRESS
618 001246 006400 PRG1 IPRG1 START ADDRESS
619 001250 006442 PRG2 IPRG2 START ADDRESS
620 001252 006474 PRG3 IPRG3 START ADDRESS
621 001254 007474 PRG4 IPRG4 START ADDRESS
622 001256 007706 PRG5 IPRG5 START ADDRESS
623 001260 002354 INCPRG INCPRG
624 001262 002354 INCPRG INCPRG
625 001264 003122 ENHTAB: TYP IPOINTER TO TYPEOUT ROUTINE
626 001266 003244 TYP8 IPOINTER TO CHAINED MESSAGES ROUTINE
627 001270 000000 OPEN IPOINTER TO RANDOM STALL ROUTINE
628 001272 002512 ERR IPOINTER TO ERROR ROUTINE
629 001274 002412 DTCHK IPOINTER TO DATA CHECK ROUTINE
630 001276 002366 CHLT ICOMMON HALT
631 001300 003042 STRVRV IPOINTER TO ROUTINE TO SET RCVR VECTOR AND PRIORITY
632 001302 003072 STXMTV IPOINTER TO ROUTINE TO SET XMIT VECTOR AND PRIORITY
633 001304 002400 EHLT IPOINTER TO ERROR HALT ROUTINE
634 001306 002294 SAVRS IPOINTER TO SAVE REGISTERS ROUTINE
635 001310 002314 RSTRG IPOINTER TO RESTORE REGISTERS ROUTINE
636 001312 002534 ERR1 IPOINTER TO ERROR ROUTINE
637 001314 002710 TXERR IPOINTER TO XMIT ERROR ROUTINE
638 001316 002732 RXERR IPOINTER TO RCVR ERROR ROUTINE
639 001320 003466 DLY IPOINTER TO DELAY ROUTINE
640 001322 000000 PARBIT: OPEN
641 001324 000000 COUNT: OPEN
642 001326 000000 SAVE: OPEN
643 001330 000000 LINE: OPEN
644 001332 000000 CONFIG: OPEN
645 001334 000000 NUMBER: OPEN
646
647 001336 000000 OLDP8: 0
648 001340 000000 TOPC: 0
649 001342 000000 FROMPC: 0
650 001344 000000 FTITLE: 0
651 001346 000000 FNONE: 0
652 001350 000000 FMAP: 0
653 001352 000000 TEMP1: 0
654 001354 000000 LINENO: 0
655 001356 000000 RECDAT: OPEN
656 001360 000000 XMTDAT: OPEN
    
```

657	001362	000000		CARMSK:	OPFN		
658	001364	000000		CTRD:	OPFN		
659	001366	000000		TXCSR:	OPFN		
660	001370	000000		RXCSCR:	OPFN		
661	001372	000000		TEMP:	OPFN		
662	001374	000000		SRT:	OPFN		
663	001376	000000		INBUFP:	OPFN		
664	001400	000000		BUFFP:	OPFN		
665	001402	000000		CALLER:	OPFN		
666	001404	000000		CALLED:	OPFN		
667	001406	000000		OTBUFP:	OPFN		
668							
669	001410	000000		TBUFP:	OPFN		
670	001412	000000		MODEM:	OPFN		
671	001414	000000					
672							
673							
674	001416	005037	001344	REMAP:	CLR	FTITLE	
675	001422	012706	001200	START:	MOV	*STKPTR,X6);SET BOTTOM OF SP STACK.
676	001426	000005			RESET		
677	001430	005037	177776		CLR	PSW	
678	001434	012737	004520		MOV	*ERTP,MACHER	
679	001442	012737	000040		MOV	#40,MACHER+2	
680	001450	005737	001344		TST	FTITLE);TITLE PRINTED
681	001454	001145			BNE	START1);YES, SKIP THIS
682	001456	104000			TYPE		
683	001460	012012			MTITLE		
684	001462	005237	001344		INC	FTITLE);SET TITLE PRINTED FLAG
685	001466	005037	001346		CLR	FNONE);CLEAR NO DEVICE FLAG
686	001472	012737	177777		MOV	#-1,LINEND	
687	001500	012737	001590		MOV	*HAPNE,MACHER);SET UP FOR NO DEVICE ANSWER
688	001506	012737	000340		MOV	*PRTY7,MACHER+2	
689	001514	012704	010432		MOV	*RCSR,X4);SET UP DEVICE POINTER
690	001520	005237	001354		INC	LINENO	
691	001524	020427	010530		CMP	X4,*RBUF);LAST DEVICE
692	001530	001477			BEO	HAPEND);YES
693	001532	005037	177776		CLR	PSW	
694	001536	005774	000000		TST	0(4));TEST DEVICE
695	001542	000240			NOP		
696	001544	000240			NOP		
697	001546	000404			BR	HAPOK	
698	001550	062704	000002		ADD	#2,X4	
699	001554	022626			POPBP2		
700	001556	000760			BR	HAPA	
701	001560	011437	001352		MOV	(4),TEMP1);SAVE DEVICE FOR TYPING
702	001564	004537	003530		JSR	X5,OACNV	
703	001570	001352			TEMP1		
704	001572	012154			MADDR		
705	001574	000006			6		
706	001576	004537	003530		JSR	X5,OACNV	
707	001602	001354			LINENO		
708	001604	012146			MLINE		
709	001606	000002			2		
710	001610	011401			MOV	(4),X1);GET RXCSR DEVICE ADDRESS

711	001612	004737	004704		JSR	X7,FORMAD	
712	001616	052737	000001	001350	BIS	*BIT6,MFAP	
713	001624	042777	000100	177352	BIC	*BIT6,*TXCSR	
714	001632	052777	000100	177344	BIS	*BIT6,*TXCSR	
715	001640	000240			NOP		
716	001642	012737	000340	177776	MOV	*PRTY7,PSW	
717	001650	005737	001210		TST	RXVTR	
718	001654	001406			BEO	HAPOKA	
719	001656	013701	001354		MOV	LINENO,X1	
720	001662	006301			ASL	X1	
721	001664	013761	001210	010334	MOV	RXVTR,VECTAB(1));STORE VECTOR
722	001672	042777	000100	177304	HAPOKA:	BIC	*BIT6,*TXCSR
723	001700	004537	003530		JSR	X5,OACNV	
724	001704	001210			RXVTR		
725	001706	012166			MTRAP		
726	001710	000004			4		
727	001712	104000			TYPE		
728	001714	012146			MLINE		
729	001716	005237	001346		INC	FNONE	
730	001722	062704	000002		ADD	#2,X4	
731	001726	000674			BR	HAPA	
732	001730	012737	004520	000004	HAPEND:	MOV	*ERTP,MACHER
733	001736	012737	000040	000006	MOV	#40,MACHER+2	
734	001744	005737	001346		TST	FNONE	
735	001750	001007			BNE	START1	
736	001752	104000			TYPE		
737	001754	012176			MNONE		
738	001756	005037	001344		CLR	FTITLE	
739	001762	000000			HALT		
740	001764	000137	001422		JMP	START	
741	001770	005037	177776		START1:	CLR	PSW
742	001774	104000			TYPE		
743	001776	012213			MWBEL		
744	002000	004737	003314		JSR	PC,RDOCT);GET INPUT
745	002004	012600			MOV	(SP)+,X0);(SR) TO R0
746	002006	042700	177770		BIC	*17770,X0);LIMIT (SR) TO BITS 2-9
747	002012	010037	001240		MOV	X0,PRNUM);SAVE PROGRAM #
748	002016	006300			ASL	X0);R0X2
749	002020	000170	001244		JMP	*PR0TAB(0));GO TO SELECTED PROGRAM.
750							
751	002024	012706	001200		MODEV:	MOV	*STKPTR,X6
752	002030	000005			RESET		
753	002032	005037	177776		CLR	PSW	
754	002036	104000			MODEV1:	TYPE	
755	002040	014435			MMOD1		
756	002042	004737	003314		JSR	PC,RDOCT);GET INPUT
757	002046	011600			MOV	(SP),X0	
758	002050	042700	177740		BIC	*177740,X0	
759	002054	006300			ASL	X0	
760	002056	022627	000036		CMP	(SR)+,#36	
761	002062	101403			BLOS	MODEV2);BRANCH IF > 36
762	002064	104000			TYPE		
763	002066	014363			MMODX		
764	002070	000762			BR	MODEV1	

```

765 002072 104000          MODEV2: TYPE
766 002074 014553          MMD02
767 002076 004737 003314  JSR   PC,RDOCT          IGET INPUT
768 002102 032716 000001  RIT   #BIT0,(SP)
769 002106 001403          BEQ   MONEV3
770 002110 104000          TYPE
771 002112 014405          MMD0D
772 002114 000766          BR    MODEV2
773 002116 012601          MODEV3: MOV   (SP)+,X1          ISAVE DEV ADR
774 002120 010003          MOV   X0,X3
775 002122 062703 010432  ADD   #RCSR,X3
776 002126 010113          MOV   X1,(3)
777 002130 062701 000002  ADD   #2,X1          IUPDATE DEV ADR
778 002134 010003          MOV   X0,X3
779 002136 062703 010530  ADD   #RBUF,X3
780 002142 010113          MOV   X1,(3)
781 002144 062701 000002  ADD   #2,X1          IUPDATE DEV ADR
782 002150 010003          MOV   X0,X3
783 002152 062703 010626  ADD   #TCSR,X3
784 002156 010113          MOV   X1,(3)
785 002160 062701 000002  ADD   #2,X1          IUPDATE DEV ADR
786 002164 010003          MOV   X0,X3
787 002166 062703 010724  ADD   #TBUF,X3
788 002172 010113          MOV   X1,(3)
789 002174 104000          TYPE
790 002176 014621          MMD03
791 002200 004737 003314  JSR   PC,RDOCT          IGET INPUT
792 002204 022627 77777  CMP   (SP)+,#17777
793 002210 001712          BEQ   MONEV1
794 002212 005037 001344  CLR   FTITLE
795 002216 000137 001422  JMP   START

;
; EMT TRAP INTERPRETER
EMTINT: MOV   %0,-(6)          IGET SAVED PC.
          SUB   #2,%0          IDECREMENT PC BY 2.
          MOV   @(%),%0
EMTAI:  ROL   %0          IEMT ARG X 2.
          BIC  #177001,%0     IREMOVE 7 MSB.
          ADD  #EMTTAB,%0     IFORM EMT RTN ADDR.
          MOV   @(%),%0
          JMP  @(%)+          IGO TO EMT ROUTINE.

;
; SAVE REGS 0 TO 4 SUBROUTINE.
SAVRG:  MOV   (6)+,SVRPC      ISAVE PC AND PSW.
          MOV   (6)+,SVRPSW
          MOV   X4,-(6)        ISAVE REGS 0 = 4
          MOV   X3,-(6)        IIN STACK.
          MOV   X2,-(6)
          MOV   X1,-(6)
          MOV   X0,-(6)
          MOV   SVRPSW,-(6)    IRESTORE PC AND PSW.
          MOV   SVRPC,-(6)
          RTI
SVRPC:  OPEN
  
```

```

819 002312 000000          SVRPSW: OPEN
820
821
822 002314 012637 002350  IRESTORE REGS 0 TO 4 SUBROUTINE.
823 002320 012637 002352  RSTRG: MOV   (6)+,RSTPC      ISAVE PC AND PSW.
824 002324 012600          MOV   (6)+,RSTPSW
825 002326 012601          MOV   (6)+,X0          IRESTORE REGS 0 = 4
826 002330 012602          MOV   (6)+,X1          IFROM STACK.
827 002332 012603          MOV   (6)+,X2
828 002334 012604          MOV   (6)+,X3
829 002336 013746 002352  MOV   (6)+,X4
830 002342 013746 002350  MOV   RSTPSW,-(6)      IRESTORE PC AND PSW.
831 002346 000002          MOV   RSTPC,-(6)
832 002350 000000          RTI
833 002352 000000          RSTPC: OPEN
834 002354 104000          RSTPSW: OPEN
835 002356 013352          INCRG: TYPE          ITYPE INCORRECT PROGRAM SELECTED.
836 002360 000000          AINPRG
837 002362 000137 001422  HALT
          JMP   START
838
839 002366 011600          ICOMMON HALT ROUTINE
840 002370 162700 000002  CHLT: MOV   %0,%0          IDEVELOP ADDRESS OF CALLER.
841 002374 000000          SUB   #2,%0
842 002376 000002          HALT
          RTI          IHALT, ADDRESS OF CALL INSTRUCTION
          IIN DATA LIGHTS.
843
844
845 002400 005777 175570  ICONDITIONAL ERROR HALT ROUTINE.
846 002404 100001          EMLT: TST  #SRPTR          ICHECK FOR HALT ON ERROR.
847 002406 000000          BPL   EMLTA          IBRANCH IF NO HALT DESIRED.
848 002410 000002          HALT
          RTI          IHALT.
          IIN DATA LIGHTS.
849
850
851 002412 043737 001362 001360  I DATA CHECK ROUTINE.
852 002420 123737 001356 001360  DTCHK: BIC  CARNK,XMTDAT    ICLEAR UNTRANSMITTED BITS
853 002426 001430          CMPB  RECDAT,XMTDAT    ICOMPARE TRANSMITTED AND RECEIVED
854 002430          BEQ  DTCHKA          ICHARS. BRANCH IF SAME.
855 002442          CNVDA  RECDAT,CWAS,3
856 002454          CNVDA  XMTDAT,C8B,3
857 002466 104013          CNVDA  RXCSR,C8RADD,6
858 002470 012357          ERROR: CORADD
859 002472 004537 004216  JSR   5,BDCNV          ICONVERT
860 002476 001364          CTRD  ICHAR #
861 002500 013343          CRNUM ITO DECIMAL
862 002502 000004          4
863 002504 104013          ERROR: I4 BITS
864 002506 013300          CERDAT
865 002510 000002          DTCHKA: RTI
          I EXIT.
866
867 002512 012737 177777 002624  I
ERR:  MOV   #=1,ERRB          ISET UP ONE MESSAGE CALL.
868 002520 012737 000240 002626  MOV   #240,ERRB+2
869 002526 005037 002644          CLR   ERRE
870 002532 000413          BR    ERRA
871 002534 011637 002624          ERR1: MOV   %0,ERRB          IDEVELOP ADD'L MESSAGE ADDR.
872 002540 017737 000060 002624  MOV   #ERRB,ERRB      ISTORE AT ERRB.
  
```

```

873 002546 012737 177777 002626 MOV #=1,ERRB+2
874 002554 012737 000002 002644 MOV #2,ERRE
875 002562 032777 020000 175404 ERRAI BIT13,#SRPTR ;INHIBIT ERROR PRINT?
876 002570 001020 BNF ERRC ;BRANCH TO INHIBIT PRINT.
877 002572 011637 002642 002642 MOV #%,ERRD ;DEVELOP CALLING ADDR.
878 002576 162737 000002 002642 SUR #2,ERRD
879 002604 CNVOA ERRO,APC,6 ;CONVERT CALL ADDR TO ASCII.
880 002616 104011 SAVREG
881 002620 104001 TYPE8
882 002622 012427 EMB ;TYPE1
883 002624 000000 ERRB: OPEN ;ERROR HEADER,
884 002626 177777 -1 ;ADDT'L ERROR MESSAGE IF ANY.
885 002630 104012 RSTREG
886 002632 104010 ERRC: EHALT ;GO ERR HALT IF DESIRED.
887 002634 063716 002644 ADD ERRE,%%
888 002640 000002 RTI ;EXIT.
889 002642 000000 ERRO: OPEN
890 002644 000000 ERRE: OPEN
891
892 ;
893 002646 012737 002656 000024 ;POWER FAIL SERVICE
894 002654 000000 PFAIL: MOV #PWRUP,##24
895 002656 012737 002646 000024 PWRUP: MOV #PFAIL,##24
896 002664 000005 RESET
897 002666 012706 001200 MOV #STKPTR,%%
898 002672 104000 TYPE
899 002674 104325 MPWRP
900 002676 013700 001240 RESTR7: MOV PRGNUM,%%
901 002702 006300 ASL %0
902 002704 000170 001244 JMP #PROTAB(0)
903
904 002710 ;TXERR: CNVOA TXCSR,ATXWAS,6 ;CONVERT CONTENTS OF TXCSR TO ASCII.
905 002722 012737 012445 003026 MOV #ATXCSR,CRXTXB
906 002730 000410 BR CRTX
907 002732 ;RXERR: CNVOA RXCSR,ARXWAS,6 ;CONVERT CONTENTS OF RXCSR TO ASCII.
908 002744 012737 012466 003026 MOV #ARXCSR,CRXTXB
909 002752 011637 003024 CRXTX: MOV %%,CRXTXA ;DEVELOP ADDR OF ADTT'L ERROR MESSAGE.
910 002756 017737 000042 003024 MOV #CRXTXA,CRXTXA
911 002764 032777 020000 175202 BIT #BIT13,#SRPTR ;INHIBIT PRINT?
912 002772 001017 BNE CRXTXC ;BRANCH TO INHIBIT PRINT.
913 002774 011637 002642 MOV #%,ERRD ;DEVELOP CALLING ADDR.
914 003000 162737 000002 002642 SUB #2,ERRD
915 003006 CNVOA ERRO,APC,6 ;CONVERT CALLING ADDR TO ASCII.
916 003020 104001 TYPE8
917 003022 012427 EMB ;TYPE ERROR MESSAGE.
918 003024 000000 CRXTXA: OPEN ;ERR HEADER
919 003026 000000 CRXTXB: OPEN ;ADDT'L ERR MESSAGE
920 003030 177777 -1 ;TXCSR OR RXCSR CONTENTS.
921 003032 104010 CRXTXC: EHALT ;GO HALT IF DESIRED.
922 003034 062716 000002 ADD #2,%%
923 003040 000002 RTI ;EXIT.
924
925 ;ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
926 003042 017637 000000 003062 STRVRV: MOV #(%),STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2

```

```

927 003050 062716 000002 ADD #2,%% ;SET UP EXIT
928 003054 013701 001210 MOV RXVTR,%%
929 003060 012721 000000 STPRA: MOV #OPEN,(%)+ ;SET VECTOR ADDRESS
930 003064 013721 001212 MOV RXLVL,(%)+ ;SET PRIORITY
931 003070 000002 RTI ;EXIT
932
933 ;ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
934 003072 017637 000000 003112 STXMYV: MOV #(%),STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
935 003100 062716 000002 ADD #2,%% ;SET UP EXIT
936 003104 013701 001214 MOV TXVTR,%%
937 003110 012721 000000 STPPA: MOV #OPEN,(%)+ ;SET VECTOR ADDRESS.
938 003114 013721 001216 MOV TXLVL,(%)+ ;SET PRIORITY
939 003120 000002 RTI ;EXIT.
940
941 ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
942 003122 010037 003312 TYP: MOV %0,SAVR0 ;SAVE R0
943 003126 011600 MOV %%,%% ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
944 003130 062716 000002 ADD #2,%% ;SET UP EXIT.
945 003134 011000 MOV %%,%% ;ADDRESS OF MESSAGE TO R0.
946 003136 112037 003242 TYP: MOV (%),TYPDAT ;GET CHARACTER
947 003142 122737 000100 003242 CMPB #100,TYPDAT ;CHECK FOR"0"CHARACTER
948 003150 001003 BNE TYPC ;BRANCH IF NOT"0".
949 003152 013700 003312 MOV SAVR0,%% ;RESTORE R0
950 003156 000002 RTI ;TERMINATOR CHAR. DONE. EXIT.
951 003160 122737 000045 003242 TYP: CMPB #45,TYPDAT ;CHECK FOR"X".
952 003166 001412 BEQ TYPF ;BRANCH IF"X".
953 003170 004737 003176 JSR #7,TYPD ;TYPE CHAR IN TYPDAT
954 003174 000760 BR TYP
955 003176 113777 003242 176022 TYPD: MOVB TYPDAT,%TPB ;OUTPUT CHARACTER TO PRINTER
956 003204 105777 176014 TSTB #TPB ;WAIT FOR DONE FLAG.
957 003210 100375 BPL .-4
958 003212 000207 RTS #7 ;EXIT
959 003214 112737 000015 003242 TYPF: MOVB #15,TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
960 003222 004737 003176 JSR #7,TYPD ;GO TYPE CHAR.
961 003226 112737 000012 003242 MOVB #12,TYPDAT ;MOVE LF CODE TO TYPDAT.
962 003234 004737 003176 JSR #7,TYPD ;GO TYPE CHAR.
963 003240 000736 BR TYP
964 003242 000000 TYPDAT: OPEN
965
966 ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
967 003244 010037 003312 TYP: MOV %0,SAVR0
968 003250 011600 TYP: MOV %%,%% ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
969 003252 062716 000002 TYP: ADD #2,%% ;UPDATE TO NEXT MESSAGE ADDRESS
970 003256 011037 003306 TYP: MOV %%,%% ;ADDRESS OF MESSAGE TO TYPB
971 003262 022737 177777 003306 CMP #=1,TYPB ;CHECK FOR TERMINATOR
972 003270 001003 BNE TYP: ;BRANCH IF NOT TERMINATOR.
973 003272 013700 003312 MOV SAVR0,%% ;RESTORE R0
974 003276 000002 RTI ;TERMINATOR, EXIT
975 003300 013700 003312 TYP: MOV SAVR0,%%
976 003304 104000 TYPE ;CALL ON TYP SUB TO TYPE MESSAGE
977 003306 000000 TYP: OPEN ;ADDRESS OF MESSAGE GOES HERE
978 003310 000757 BR TYP: ;GO PROCESS NEXT MESSAGE
979 003312 000000 SAVR0: OPEN
980 ;SUBROUTINE TO READ OCTAL DATA FROM THE TELETYPE PRINTER

```

```

081 003314 011646          RDOCT: MOV      (SP),-(SP)      ;MAKE ROOM FOR DATA WORD
082 003316 010046          MOV      %0,-(SP)          ;SAVE R0
083 003320 010146          MOV      %1,-(SP)          ;SAVE R1
084 003322 005001          INDAT: CLR      %1          ;CLEAR DATA WORD
085 003324 005037 001324   CLR      COUNT            ;SET NO. OF DIGITS = 0
086 003330 105777 175664   RDDAT: TSTB    %TKS        ;TEST TTY READ STATUS
087 003334 100375          RPL      RDDAT            ;WAIT
088 003336 117746 175660   MOVB    %TKB,-(SP)        ;PUSH DIGIT ON STACK
089 003342 002716 000200   BIC     %BIT7,(SP)        ;
090 003346 105777 175652   ECDAT: TSTB    %TPS        ;TEST TTY PRINT STATUS
091 003352 100375          RPL      ECDAT            ;WAIT
092 003354 111677 175646   MOVB    (SP),%TPB        ;ECHO CHARACTER
093 003360 122716 000015   CMPB    %15,(SP)          ;IS IT A TERMINATOR?
094 003364 001432          BEQ      RETRN            ;BR IF YES
095 003366 122716 000177   CMPB    %17,(SP)          ;IS IT A RUBOUT?
096 003372 001423          BEQ      RREAD           ;BR IF YES
097 003374 122716 000060   CMPB    %60,(SP)          ;IS IT AN OCTAL DIGIT?
098 003400 003020          BGT      RREAD           ;BR IF NO
099 003402 122716 000067   CMPB    %67,(SP)          ;TEST AGAIN
1000 003406 002415          BLT      RREAD           ;BR IF NO
1001 003410 005237 001324   INC     COUNT            ;INC NO. OF DIGITS
1002 003414 022737 000067 001324   CMP     %67,COUNT        ;MORE THAN SIX DIGITS?
1003 003422 003407          BLE      RREAD           ;BR IF YES
1004 003424 006301          ABL     %1                ;CLEAR LOWEST THREE BITS
1005 003426 006301          ABL     %1                ;JOP DATA WORD
1006 003430 006301          ABL     %1                ;
1007 003432 162716 000060   SUR     %60,(SP)          ;CONVERT TO BINARY
1008 003436 062601          ADD     (SP)+,%1          ;ADD DIGIT TO DATA WORD
1009 003440 000733          BV      RDDAT            ;GET NEXT DIGIT
1010 003442 104000          RREAD: TYPE    DTERR      ;TELL USER ABOUT ILLEGAL CHARACTER
1011 003444 013232          TST     %1                ;
1012 003446 005726          BR      INDAT            ;GET RID OF ILLEGAL CHARACTER
1013 003450 000724          BR      INDAT            ;START SUBROUTINE AGAIN
1014 003452 010166 000010   MOV     %1,%10(S)        ;STORE DATA WORD ON STACK
1015 003456 005726          TST     %1                ;INC STACK POINTER
1016 003460 012601          MOV     (SP)+,%1         ;RESTORE R1
1017 003462 012600          MOV     (SP)+,%0         ;RESTORE R0
1018 003464 000207          RTS     PC                ;RETURN
1019
1020 ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS.
1021 003466 011637 003526   DLY:  MOV     %6,DLCNT    ;GET DELAY COUNT ADDRESS.
1022 003472 062716 000002   ADD     %2,%6           ;SET UP EXIT ADDRESS
1023 003476 017746 000024   MOV     %DLCNT,-(6)     ;DELAY COUNT TO STACK
1024 003502 001407          BEQ     DLYC            ;
1025 003504 012746 000226   DLYA: MOV     %226,-(6)   ;1 MSEC COUNT TO STACK
1026 003510 005316          DLYB: DEC     %6         ;DECREMENT 1 MSEC COUNT
1027 003512 001376          BNE     DLYB           ;BRANCH IF NOT 0.
1028 003514 005726          POPSP   %6             ;ZERO, UNCOVER MSEC. COUNT.
1029 003516 005316          DEC     %6             ;DECREMENT IT
1030 003520 001371          BNE     DLYA           ;BR IF NOT DONE DELAYING
1031 003522 005726          DLYC: POPSP           ;DONE
1032 003524 000002          RTI     %6             ;EXIT.
1033 003526 000000          DLCNT: OPEN           ;CONTAINS MILLISECONDS COUNT ADDRESS.
1034

```

```

1035 ;OCTAL TO ASCII CONVERT ROUTINE
1036 003530 104011          OACNV: SAVREG           ;SAVE REGS.
1037 003532 013500          MOV     %5,%0           ;GET OCTAL VALUE.
1038 003534 012501          MOV     %5,%1           ;GET DESTINATION ADDR.
1039 003536 012502          MOV     %5,%2           ;GET CONVERT COUNT.
1040 003540 060201          ADD     %2,%1           ;DEVELOP ADDR TO STORE 1ST CHAR.
1041 003542 010003          OACNVA: MOV     %0,%3    ;
1042 003544 042703 177770   BIC     %177770,%3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
1043 003550 062703 000060   ADD     %60,%3          ;CONVERT DIGIT TO ASCII.
1044 003554 110341          MOVB    %3,-(1)         ;STORE ASCII CHARACTER.
1045 003556 042700 000007   BIC     %7,%0           ;
1046 003562 006000          ROR     %0             ;
1047 003564 006000          ROR     %0             ;
1048 003566 006000          ROR     %0             ;
1049 003570 005302          DEC     %2             ;DONE ALL DIGITS?
1050 003572 001363          BNE     OACNVA          ;BRANCH IF NOT DONE.
1051 003574 104012          RSTREG           ;RESTORE REGS.
1052 003576 000205          RTS     %5             ;DONE, EXIT.
1053 ;SUBROUTINE TO GENERATE PARITY ON DATA FOR 5,6,7,8 LEVEL CODE.
1054 ;PARITY BIT IS THE MSB OF THE CHARACTER PARITY CAN BE EITHER
1055 ;EVEN OR ODD
1056 ;GENERATES ODD/EVEN PARITY.

```

```

1057
1058 003600 032737 000200 001362 GENPAR: BIT #BIT7,CARMSK ;TEST LSR CHAR LENGTH
1059 003600 001411 REO EIGHT ;CHAR IS 8
1060 003610 032737 000100 001362 BIT #BIT6,CARMSK ;TEST MSB CHAR LENGTH
1061 003616 001427 REO SEVEN ;CHAR LENGTH IS 7
1062 003620 032737 000040 001362 BIT #BITS,CARMSK
1063 003626 001412 REO SIX
1064 003630 000433 BR FIVE
1065 003632 012737 000200 001322 EIGHT: MOV #BIT7,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1066 003640 012737 000007 001324 MOV #8,COUNT ;SET UP ROTATE COUNTER=7
1067 003646 042701 177600 BIC #177600,X1 ;MASK OFF UNUSED BITS
1068 003652 000433 BR ;GO AND GENERATE PARITY FOR 8
1069 003654 012737 000040 001322 SIX: MOV #BITS,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1070 003662 012737 000005 001324 MOV #5,COUNT ;SET UP ROTATE COUNTER=5
1071 003670 042701 177740 BIC #177740,X1 ;MASK OFF UNUSED BITS
1072 003674 000422 BR ;GO AND GENERATE PARITY FOR
1073 003676 012737 000100 001322 SEVEN: MOV #BIT6,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1074 003704 012737 000006 001324 MOV #6,COUNT ;SET UP ROTATE COUNTER=6
1075 003712 042701 177700 BIC #177700,X1 ;MASK OFF UNUSED BITS
1076 003716 000411 BR ;GO AND GENERATE PARITY FOR 7
1077 003720 012737 000020 001322 FIVE: MOV #BIT4,PARBIT ;PLACE PARITY BIT IN PROPER POSITION
1078 003726 012737 000004 001324 MOV #4,COUNT ;SET UP ROTATE COUNTER=4
1079 003734 042701 177760 BIC #177760,X1 ;MASK OFF UNUSED BITS
1080 003740 000400 BR ;GO AND GENERATE PARITY FOR
1081 003742 010137 001326 DOIT: MOV X1,SAVE ;SAVE DATA
1082 003746 006001 AGAIN: ROR X1 ;ROTATE DATA
1083 003750 103415 BCS ADD1 ;IF CARRY SET ADD IN PARBIT
1084 003752 005337 001324 RTN: COUNT ;DECREMENT COUNTER
1085 003756 001373 BNE AGAIN ;NOT DONE DO IT AGAIN
1086 003760 032737 000040 001374 BIT #BITS,BRT ;DONE EVEN OR ODD PARITY?
1087 003766 001403 BEO DONE ;IF EVEN FINISHED
1088 003770 063737 001322 001326 ADD PARBIT,SAVE ;IF ODD ADD IN ANOTHER 1
1089 003776 013701 001326 DONE: MOV SAVE,X1 ;PLACE DATA + PARITY BACK IN R1
1090 004002 000207 RTS ;AND EXIT
1091 004004 063737 001322 001326 ADD: ADD PARBIT,SAVE ;ADD PARBIT TO DATA
1092 004012 000757 RTN ;RETURN TO COUNTER
1093
1094 ;SUBROUTINE TO SELECT LINE AND LOAD VECTOR ASSIGNMENT
1095 004014 104000 LINSEL: TYPE
1096 004016 014256 LDLINE
1097 004020 004737 003314 JSR PC,RDOCT ;GET INPUT
1098 004024 012601 MOV #8P,X1 ;LOAD R1
1099 004026 042701 177407 BIC #177407,X1 ;MASK OFF ALL BUT LINE BITS
1100 004032 006201 ASR X1
1101 004034 006201 ASR X1
1102 004036 010137 001372 MOV X1,TEMP ;SAVE LINE #
1103 004042 012703 001200 MOV #RXCSR,X3 ;LOAD ADDRESS OF REGISTERS
1104 004046 012704 000004 MOV #4,X4 ;SET UP COUNTER
1105 004052 016102 010432 MOV RCSR(1),X2
1106 004056 010223 LINSA: MOV X2,(X3)+
1107 004060 062702 000002 ADD #2,X2
1108 004064 005304 DEC X4
1109 004066 001373 BNE LINSA
1110 004070 016101 010334 MOV VECTAB(1),X1 ;GET LINE VECTOR ADDRESS

```

```

1111 004074 010123 MOV X1,(X3)+ ;LOAD INTO PROG. RXVTR
1112 004076 022121 CMP (1)+,(1)+ ;ADD +4 TO RXVTR TO = TXVTR
1113 004100 005723 TST (3)+ ;POINT TO PROG TXVTR
1114 004102 010113 MOV X1,(X3) ;LOAD INTO PROG TXVTR
1115 004104 022737 000005 001240 CMP #5,PRGNUM ;RUNNING PROGRAM # 5
1116 004112 001001 BNE +4
1117 004114 000205 RTS ;RETURN TO PROG 5
1118 004116 006237 001372 ASR TEMP ;POSITION
1119 004122 CNVOA TEMP,TLINEX,2
1120 004134 104000 TYPE ;TYPE LINE # THAT
1121 004136 013645 ALINEX ;WAS CALLED
1122 004140 000205 RTS
1123
1124 ;SUBROUTINE TO LOAD BINARY COUNT PATTERN INTO OUTPUT BUFFER
1125 004142 105037 001334 INFIL: CLR# NUMBER ;INITIALIZE BINARY COUNT
1126 004146 012500 FILL: MOV (S)+,X0 ;GET ADDRESS
1127 004150 012537 001364 MOV (S)+,CTRD ;GET COUNT
1128 004154 113720 001334 FILL: MOV# NUMBER,(0)+ ;LOAD ADDRESS WITH BINARY COUNT
1129 004160 105237 001334 INCB NUMBER ;INC. BINARY COUNT
1130 004164 005337 001364 DEC CTRD ;DEC. COUNT
1131 004170 001371 BNE FILL
1132 004172 000205 RTS ;EXIT
1133
1134 ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1135 004174 104011 BMOVE: SAVREG ;SAVE REGS.
1136 004176 012501 MOV (S)+,X1 ;GET "FROM" ADDRESS
1137 004200 012502 MOV (S)+,X2 ;GET "TO" ADDRESS
1138 004202 012503 MOV (S)+,X3 ;GET COUNT
1139 004204 112122 BNOVA: MOV# (1)+,(2)+ ;MOVE BYTE
1140 004206 005303 DEC X3 ;DECREMENT COUNT
1141 004210 001375 BNE BNOVA ;BRANCH IF NOT DONE.
1142 004212 104012 RSTREG ;RESTORE REGS.
1143 004214 000205 RTS ;DONE. EXIT
1144
1145 ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
1146 004216 104011 BDCNV: SAVREG ;SAVE REGS.
1147 004220 012700 004374 MOV #DECVAL,X0 ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
1148 004224 013501 MOV #5,X1 ;BINARY VALUE TO R1.
1149 004226 012537 004304 MOV (S)+,BDCNVC ;DESTINATION ADDR TO BDCNVC.
1150 004232 012537 004306 MOV (S)+,BDCNVD ;COUNT TO BDCNVD.
1151 004236 012702 004362 MOV #ADTENP,X2 ;ADDR OF TEN POWER STRING TO R2.
1152 004242 012737 000005 004354 MOV #B.CNVCTR ;SET UP FOR 5 POWER CONVERSIONS.
1153 004250 012237 004360 MOV (0)+,TENPHR ;MOVE #POWER OF TEN VALUE TO TENPHR.
1154 004254 004737 004314 BDCNV: JBR X7,SUBTEN ;PERFORM CONVERSION
1155 004260 005337 004354 DEC CNVCTR ;DONE 5 CONVERSIONS?
1156 004264 001371 BNE BDCNVA ;BRANCH IF NOT YET 5.
1157 004266 163700 004306 SUB BDCNVD,X0 ;SET UP ADDR TO MOVE DECIMAL
1158 004272 010037 004302 MOV X0,BDCNVB ;DATA FROM.
1159 004276 004537 004174 JBR X5,BMOVE ;MOVE DECIMAL DATA TO DESTINATION.
1160 004302 000000 BDCNVB: OPEN ;SRC ADDR.
1161 004304 000000 BDCNVC: OPEN ;DEST ADDR.
1162 004306 000000 BDCNVD: OPEN ;COUNT.
1163 004310 104012 RSTREG ;RESTORE REGS.
1164 004312 000205 RTS ;YES. EXIT.

```

```

DZDL88.SRC
1165 004314 005037 004356 SUBTEN: CLR DIGIT JCLEAR DIGIT
1166 004320 163791 004360 SUBTNA: SUR TENPWR,X1 JSUBTRACT TEN POWER FROM BINARY VALUE,
1167 004324 103493 RCS SURTNB JBRANCH IF UNSUCCESSFUL SUBTRACTION.
1168 004326 005237 004356 INC DIGIT
1169 004332 007772 RR SUBTNA
1170 004334 063701 004360 SUBTNB: ADD TENPWR,X1 JRESTORE SUBTRACTED VALUE.
1171 004340 062737 000000 004356 ADD JCONVERT (DIGIT) TO ASCII
1172 004346 113720 004356 MOVB JMOVE ASCII CHAR TO DECIMAL FIELD.
1173 004352 000207 RTS JEXIT.
1174 004354 000000 CNVCTR: OPEN
1175 004356 000000 DIGIT: OPEN
1176 004360 000000 TENPWR: OPEN
1177 004362 023420 ADTENP: 10000.
1178 004364 001750 1000.
1179 004366 000144 100.
1180 004370 000012 10.
1181 004372 000001 1
1182 004374 000 040 040 DECVAL: .BYTE 040,040,040,040,040,040
1183 004377 000 040 040
1184
1185 004402 104000 JSUBROUTINE TO SET CHARACTER LENGTH PARAMETER
1186 004404 013055 SETPAR: TYPE JTYPE: SELECT PARAMETERS.
1187 004406 004737 003314 JSR PC,RDOCT JGET INPUT
1188 004412 012637 001374 MOV (SP)+,SR
1189 004416 CNVOA SR,APARM,3
1190 004430 104000 TYPE
1191 004432 013355 PARMNB
1192 004434 012737 177400 001362 TBIT1: MOV #177400,CARMSK JSET CHARACTER MASK TO 8 BITS.
1193 004442 032737 000002 001374 BIT #BIT1,SR JSEE IF SR BIT 1 IS SET.
1194 004450 001413 BEQ STPAR JBRANCH IF NOT SET.
1195 004452 012737 177700 001362 MOV #177700,CARMSK JCHANGE CHAR MASK TO 6 BITS.
1196 004460 032737 000001 001374 BIT #BIT0,SR JSEE IF SR BIT0 IS SET.
1197 004466 001403 BEQ PAREX JBRANCH IF NOT SET.
1198 004470 012737 177700 001362 MOV #177740,CARMSK JCHANGE CHAR MASK TO 5 BITS.
1199 004476 000207 PAREX: RTS JEXIT.
1200 004500 032737 000001 001374 STPAR: BIT #BIT0,SR JSEE IF SR BIT0 IS SET.
1201 004506 001773 BEQ STPAR-2 JBRANCH IF NOT SET.
1202 004510 012737 177600 001362 MOV #177600,CARMSK JCHANGE CHAR MASK TO 7 BITS.
1203 004516 000767 BR PAREX
1204
1205 JERROR TRAP HANDLER = TYPE TO AND FROM WHERE ERROR TRAP OCCURS
1206 004520 013737 177776 001336 ERTPI: MOV PSW,OLDP5 JSAVE OLDP5
1207 004526 012737 000340 177776 MOV #PRTY7,PSW
1208 004534 006237 001336 ASR OLDP5
1209 004540 006237 001336 ASR OLDP5
1210 004544 006237 001336 ASR OLDP5
1211 004550 042737 177740 001336 BIC #177740,OLDP5
1212 004556 013737 001336 001340 MOV OLDP5,TOPC
1213 004564 011637 001342 MOV #%,FROMPC
1214 004570 004537 003530 ERTPA: JSR %5,OACNV
1215 004574 001340 TOPC
1216 004576 012314 MTO
1217 004600 000006
1218 004602 004537 003530 JBR %5,OACNV

```

```

DZDL88.SRC
1219 004606 001342 FROMPC
1220 004610 012346 MFROM
1221 004612 000006
1222 004614 104000 TYPE
1223 004616 012247 MTERR
1224 004620 000000 HALT
1225 004622 000137 001422 JMP START
1226
1227 JHAPVEC = MAP VECTOR VECTOR OR REPORT ERROR DEPENDING ON PHAP FLAG
1228 004626 011637 001340 MAPVEC: MOV #%,TOPC
1229 004632 022626 POPBP2
1230 004634 011637 001342 MOV #%,FROMPC
1231 004640 162737 000004 001340 SUB #4,TOPC
1232 004646 005737 001350 TST FNAP
1233 004652 001746 BEQ ERTPA JNOT MAPPING, REPORT ERROR
1234 004654 013737 001340 001214 MOV TOPC,TVXTR JSTORE VECTOR
1235 004662 162737 000004 001340 SUB #4,TOPC
1236 004670 013737 001340 001210 MOV TOPC,RXVTR
1237 004676 005037 001350 CLR FNAP
1238 004702 000002 RTI
1239
1240 JFORMAD = FORM DEVICE AT ADDRESS
1241 004704 010137 001200 FORMAD: MOV #1,RXC5R
1242 004710 062701 000002 ADD #2,X1
1243 004714 010137 001202 MOV #1,RXB5F
1244 004720 062701 000002 ADD #2,X1
1245 004724 010137 001204 MOV #1,TXCSR
1246 004730 062701 000002 ADD #2,X1
1247 004734 010137 001206 MOV #1,TXB5F
1248 004740 000207 RTS
1249
1250 JSUBROUTINE TO MAKE LINE CONNECTION.
1251 004742 017737 174232 001370 LINCON: MOV #RXC5R,RXC5RT
1252 004750 032737 020000 001370 BIT #BIT13,RXC5RT JYES, IS CLEAR TO SEND UP
1253 004756 001046 BNE LINEUP JYES CONNECTION IS MADE.
1254 004760 042777 000146 174212 LINCA: BIC #146,RXC5R JCLEAR IE BIT AND DTR, RG TO SND
1255 004766 005777 174210 TST #RXC5F JCLEAR DONE FLAG
1256 004772 104000 TYPE JTYPE
1257 004774 013251 MAKCON JMAKE LINE CONNECTION
1258 004776 017737 174176 001370 LINCB: MOV #RXC5R,RXC5RT
1259 005004 032737 040000 001370 BIT #BIT14,RXC5RT JDID YOU RING
1260 005012 001771 BEQ LINCB JNO WAIT FOR RING
1261 005014 052777 000006 174156 BIC #6,RXC5R JSET DTR, RG TO SND
1262 005022 104016 DELAY JWAIT 10 SECONDS FOR
1263 005024 023420 100000 JCLEAR TO SEND
1264 005026 017737 174146 001370 MOV #RXC5R,RXC5RT
1265 005034 005777 174142 TST #RXC5F JCLEAR DONE
1266 005040 032737 020000 001370 BIT #BIT13,RXC5RT JIS CLEAR TO SEND UP?
1267 005046 001003 BNE LINCPI JYES, GO TO LINCPI
1268 005050 104015 ERRR JNO. PRINT ERROR MESSAGE
1269 005052 013377 LINCPI: LINCPI JCLEAR TO SEND NOT SET
1270 005054 000741 BR LINCPI JSTART OVER AGAIN
1271 005056 017737 174116 001370 LINCPI: MOV #RXC5R,RXC5RT JCLEAR ALL FLAGS
1272 005064 005777 174112 TST #RXC5F JAND DONE

```

```

1273 005070 104000          TYPE          ;TYPE MESSAGE
1274 005072 013427          LINMAD         ;CONNECTION IS MADE'
1275 005074 000205          LINEUP: RTS      ;EXIT LINE CONNECTION ROUTINE WITH
1276                                ;SUBROUTINE TO OVERLAY <CRLF> IN DATA PATTERN (EVERY 72,ND CHAR)
1277 005076 012701 014710  OVRLAY: MOV    #OUTBUF,X1 ;GET OUTPUT BUFFER ADDRESS
1278 005102 012702 000016  OVRLAY: MOV    #14,,X2 ;GET COUNTER
1279 005106 012711 105215  OVRLAY: MOV    #105215,(1) ;INSERT CRLF
1280 005112 062701 000110  ADD      #72,,X1 ;ADD OFFSET
1281 005116 005302          DEC      X2 ;DONE?
1282 005120 001372          RNE     OVRLYA
1283 005122 000207          RTS     ;EXIT
1284
1285                                ;RECEIVER INTERRUPTS COMMON HANDLER
1286 005124 000240          RISR:  NOP
1287 005126 010037 001330  MOV     #0,LINE
1288 005132 006300          ASL     X0
1289 005134 016037 010432 001200  MOV     @RCSR(0),RXC0R ;GET ADDRESS OF INTERRUPTING DL11-E'S RCSR
1290 005142 017737 174032 001370  MOV     @RXC0R,RXC0R ;GET RCSR CONTENTS
1291 005150 100570          BMI     DCERR ;CHECK INT
1292 005152 105737 001370  TSTB   @RXC0R ;TEST DONE
1293 005156 001002          BNE     RISRA
1294 005160 104003          ERROR  ;FALSE INTERRUPT
1295 005162 000002          RTI    ;EXIT
1296 005164 020037 001404  RISRA: CMP     #0,CALLED ;DID CALLED LINE INTERRUPT?
1297 005170 001020          BNE     RISRB ;BRANCH IF CALLER INTERRUPTED
1298 005172 005737 001412  TSTB   @MODEM ;CHECK MODEM TYPE
1299 005176 001403          BEQ    @RISRAA ;BRANCH IF 103
1300 005200 005770 010530  TSTB   @RBUF(0) ;READ CALLED LINES DATA
1301 005204 000002          RTI
1302 005206 011707 010530 174162  RISRAA: MOVB   @RBUF(0),@INBUFP ;STORE CHARACTER IN INPUT BUFFER
1303 005214 005237 001376  INC     INBUFP ;INCREMENT POINTER
1304 005220 022737 017024 001376  CMP     @INBUFP+100,,@INBUFP ;HAVE 100. CHARACTERS BEEN RECEIVED?
1305 005226 001430          BEQ    @RISRC ;GO CHECK DATA IF YES
1306 005230 000002          RTI    ;EXIT IF NO
1307 005232 011707 010530 174140  RISRB: MOVB   @RBUF(0),@BUFP ;STORE CHARACTER IN INTERMEDIATE DATA BUFFER
1308 005240 005237 001400  INC     BUFP ;INCREMENT POINTER
1309 005244 022737 015066 001400  CMP     @BUFP+10,,@BUFP ;HAVE 10 CHARACTERS BEEN RECEIVED
1310 005252 002401          BLT    ,+4
1311 005254 000002          RTI    ;EXIT
1312 005256 022737 000002 001332  CMP     @R,CONFIG ;RUNNING CONFIGURATION ?
1313 005264 001405          BEQ    @RISRB
1314 005266 022737 015220 001400  CMP     @BUFP+100,,@BUFP ;HAVE 100. CHARACTERS BEEN RECEIVED?
1315 005274 001405          BEQ    @RISRC ;GO CHECK DATA IF YES, OTHERWISE
1316 005276 000002          RTI    ;EXIT
1317 005300 052770 000100 010626  RISRBB: BIS    @BIT6,@TCSR(0) ;START CALLERS TRANSMITTER
1318 005306 000002          RTI    ;EXIT
1319
1320                                ;CHECK DATA CONFIGURATION #1
1321 005310 000240          RISRC: NOP
1322 005312 012737 000001 001364  MOV     #1,CTRD ;INITIALIZE CHARACTER COUNT
1323 005320 012702 015054  MOV     @RBUF,X2 ;POINT R2 TO CALLERS RECEIVED DATA BUFFER
1324 005324 012703 014710  MOV     @OUTBUF,X3 ;R3 = FIRST ADDRESS OF OUTPUT DATA BUFFER
1325 005330 010237 001400  MOV     X2,@BUFP ;RESTORE CALLERS RCVD DATA BUFFER PTR
1326 005334 022737 000001 001332  CMP     #1,CONFIG ;CHECK CONFIGURATION
1327 005340 001015          BNE     @RISRD

```

```

1327 005344 112337 001360  RISRCA: MOVB   (3)+,XMYDAT ;GET TRANSMITTED CHARACTER
1328 005350 112237 001356  MOVB   (2)+,RECDDAT ;GET RECEIVED CHARACTER
1329 005354 104004          DATCHK ;CHECK DATA
1330 005356 005237 001364  INC     CTRD ;INCREMENT CHARACTER COUNT
1331 005362 022737 000101 001364  CMP     #101,CTRD ;HAS ALL DATA BEEN CHECKED
1332 005370 001365          BNE     @RISRCA
1333 005372 000137 005476  JMP     FINISH
1334
1335                                ;CHECK DATA CONFIGURATION #2
1336 005376 000240          RISRD: NOP
1337 005400 012704 016660  MOV     @INBUF,X4 ;POINT R4 TO CALLED LINES RECEIVER
1338 005404 010437 001376  MOV     #4,@INBUFP ;DATA BUFFER & INIT. POINTER
1339 005410 012737 015054 001410  MOV     @BUFP,@BUFP ;
1340 005416 013701 001404  RISRDA: MOV     CALLED,X1
1341 005422 016137 010432 001200  MOV     @RCSR(1),RXC0R
1342 005430 112337 001360  MOVB   (3)+,XMYDAT
1343 005434 112237 001356  MOVB   (2)+,RECDDAT ;COMPARE TRANSMITTED DATA WITH DATA
1344 005440 104004          DATCHK ;RECEIVED BY CALLED LINE
1345 005442 013701 001402  MOV     CALLER,X1
1346 005446 016137 010432 001200  MOV     @RCSR(1),RXC0R
1347 005454 112437 001356  MOVB   (4)+,RECDDAT ;COMPARE TRANSMITTED DATA WITH DATA
1348 005460 104004          DATCHK ;RECEIVED BY CALLER
1349 005462 005237 001364  INC     CTRD
1350 005466 022737 000101 001364  CMP     #101,CTRD
1351 005474 001350          BNE     @RISRDA
1352 005476 000240          FINISH: NOP
1353 005500 013701 001404  MOV     CALLED,X1
1354 005504 004537 004146  JSR     $,FILL
1355 005510 014710          OUTBUF 100.
1356 005512 000144          TYPE
1357 005514 104000          ENDPAS
1358 005516 013350          BIS    @BIT6,@TCSR(1)
1359 005520 052771 000100 010626  NOP
1360 005526 000240          RTI
1361 005530 000002
1362
1363 005532 052770 100000 010530  ;ERROR SERVICE: ROUTINE
1364 005540 001402 0CERR: BIT    @BIT15,@RBUF(0) ;TEST ERROR
1365 005544 104015          BEQ    @RISRP
1366 005544 012337          ERRRX  @RADD
1367 005546 012737 014710 001406  RISRP: MOV     @OUTBUF,@OTBUFP ;SET OUTPUT BUFFER POINTER
1368 005554 012737 016660 001376  MOV     @INBUF,@INBUFP ;SET INPUT BUFFER POINTER
1369 005562 012737 015054 001400  MOV     @BUFP,@BUFP ;SET INTERMEDIATE BUFFER POINTER
1370 005570 012737 015054 001410  MOV     @BUFP,@BUFP ;SET POINTER FOR CONFIG #2 TRANSMITTER
1371 005576 032737 000000 001370  BIT    @BIT14,@RXC0R ;CHECK RING INDICATOR
1372 005604 001005          BNE     @RISREX ;BRANCH IF RING
1373 005606 007737 006234  JSR     $,DISCON ;ERROR SET = NO RING
1374 005612 104015          ERRRX  @RADD
1375 005614 012337          RTI
1376 005616 000002
1377 005620          RISREX: CNVDA LINE,TLINE,2
1378 005632 104000          TYPE
1379 005634 013715          ALTNE
1380 005636 010037 001404  MOV     #0,CALLED

```



```
1489
1490 006404 022737 015054 001406 TISRR: CMP #OUTBUF+100,,OTBUFF;HAVE 100. CHARS. BEEN SENT?
1491 006412 001350 RNF TISRAA EXIT IF NOT
1492 006414 042770 000100 010626 RIC #BIT6,#YCSR(0) I DISABLE TRANSMITTER INTERRUPT
1493 006422 000764 BR TISRRB I RESET POINTER AND EXIT
1494
1495
1496
1497
1498 006424 000240 *****
1499 006426 104000 IPRG0 = SINGLE CHARACTER LINE MODE TEST.
1500 006430 012507 *****
1501 006432 104000 PRG0: NOP I BEGIN PRG0
1502 006434 013143 TYPE I TYPE
1503 006436 004737 003314 P0TIT I PROGRAM TITLE
1504 006442 112601 SELCAR
1505 006444 010137 014710 JSR PC,RDOCT I GET INPUT
1506 006450 004537 004174 MOVB (SP)+,X1 I GET USER SPECIFIED DATA
1507 006454 014710 MOV X1,OUTBUF I AND
1508 006456 014711 JSR S,BMOVE I LOAD
1509 006460 001747 OUTBUF I INTO
1510 006462 004737 005076 990, I OUTPUT
1511 006466 004737 010274 JSR 7,OVRLAY I BUFFER
1512 006472 004737 010226 JSR 7,LDPRI I OVER LAY CR,LF#8 IN DATA
1513 006476 004737 010162 JSR 7,LDTVEC I LOAD PRIORITY LEVEL IN VECTOR+2
1514 006502 012737 000340 177776 JSR 7,LDVVEC I LOAD TRANSMITTER VECTORS
1515 006510 012702 000140 JSR 7,LDVCS I LOAD RECEIVER VECTORS
1516 006514 012701 010432 MOV #PRTY7,PSW I SET PROCESSOR PRIORITY=7
1517 006520 004537 010116 MOV #140,X2 I GET IE
1518 006524 104000 MOV #RCSR,X1 I BIT IN
1519 006526 013251 JSR S,MOVIT I CALL RECEIVERS
1520 006530 005037 177776 TYPE I TYPE
1521 006534 000001 MAKCON I MAKE LINE CONNECTION
1522 006536 000776 CLR PSW I SET PROCESSOR PRIORITY=0
1523
1524
1525
1526 006540 104000 *****
1527 006542 012553 IPRG1 = SPECIAL BINARY COUNT PATTERN LINE MODE TEST.
1528 006544 012737 105215 014710 *****
1529 006552 004537 004142 PRG1: TYPE I TYPE PROGRAM TITLE.
1530 006556 014712 P1TIT
1531 006560 001750 MOV #105215,OUTBUF I LOAD CRLF
1532 006562 012737 000100 001334 JSR S,INFIL I LOAD OUTPUT
1533 006570 004737 010274 OUTBUF+2 I WITH BINARY
1534 006574 004737 010226 1000, I COUNT PATTERN
1535 006600 004737 010162 MOV #100,NUMBER
1536 006604 012737 000340 177776 JSR 7,LDPRI I LOAD PRIORITY LEVEL IN VECTOR +2
1537 006612 012702 000140 JSR 7,LDTVEC I LOAD TRANSMITTER VECTORS
1538 006616 012701 010432 JSR 7,LDVCS I LOAD RECEIVER VECTORS
1539 006622 004537 010116 MOV #PRTY7,PSW I SET PROCESSOR PRIORITY=7
1540 006626 104000 MOV #140,X2 I GET IE BIT
1541 006630 013251 MOV #RCSR,X1 I GET FIRST CSR ADDRESS
1542 006632 005037 177776 JSR S,MOVIT I AND MOVE IT
1543
1544
1545
1546
1547
1548
1549
1550
1551 006642 104000 *****
1552 006644 012622 IPRG2=SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS
1553 006646 004537 004014 *****
1554 006652 004737 004402 I THE MESSAGE 'A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS
1555 006656 052777 000004 I BACK 1234567890.'
1556 006664 004537 004742 I
1557 006670 012702 013504 PRG2: TYPE I TYPE PROGRAM
1558 006674 112201 P2TIT I TITLE
1559 006676 020127 000045 JSR S,LINSEL I DO SET PARAMETERS
1560 006702 001772 JSR 7,SETPAR I GET REQUEST TO SEND
1561 006704 032737 000100 001374 PRG2A: JSR S,LINCON I DO MAKE LINE CONNECTION
1562 006712 001402 PRG2B: MOV #PRG2H,X2 I GET ADDRESS OF MESSAGE
1563 006714 004737 003600 PRG2C: MOVB (2)+,X1 I GET FIRST CHARACTER
1564 006720 004537 004742 CMP X1,#X I TERMINATOR CHARACTER
1565 006724 010177 172256 BEQ PRG2B I SEND MESSAGE
1566 006730 105777 172250 RIT #BIT6,SR I PARITY ENABLED
1567 006734 100375 BEQ .+6
1568 006736 000756 JSR 7,GENPAR I GENERATE PARITY
1569
1570
1571
1572 006740 104000 *****
1573 006742 012672 IPRG3=PROGRAM TO RECEIVE A MESSAGE.
1574 006744 004537 004014 *****
1575 006750 004737 004402 I
1576 006754 012706 001176 PRG3: TYPE I TYPE PROGRAM
1577 006760 052777 000004 172212 P3TIT I TITLE
1578 006766 004537 004742 JSR S,LINSEL I GET PARAMETERS
1579 006772 104006 JSR 7,SETPAR I REPOSITION STACK POINTER
1580 006774 007036 MOV #BIT2,#RCSR I GET REQUEST TO SEND
1581 006776 104007 JSR S,LINCON I MAKE LINE CONNECTION
1582 007000 007270 STRNV I SET RECEIVER INTERRUPT
1583 007002 005037 007470 RINTV I TO THIS ADDRESS
1584 007006 013700 001234 STTXV I SET TRANSMITTER INTERRUPT
1585 007012 012720 007422 TINTV I TO THIS ADDRESS
1586 007016 013710 001236 CLR WORDS
1587 007022 012701 014710 MOV TPVTR,X0
1588 007026 052737 100000 007466 MOV #TPINT,(0)+ I LOAD TELEPRINTER VECTOR
1589 007034 004737 007034 MOV TPLVL,(0) I AND PRIORITY
1590 007040 052777 000140 172132 MOV #OUTBUF,X1 I GET BUF ADD
1591 007046 005037 177776 BIS #BIT15,TPLAG I SET BIT 15
1592 007052 000001 JSR 7,TCRLF I SEND CRLF
1593 007054 000776 CLR PSW I ENABLE RECEIVER INTERRUPTS
1594 007056 017737 172116 001370 RINTV: MOV #2 I DO
1595 007064 100461 BMT ERR3A I NOTHING
1596 007066 105737 001370 Y8B RXCSR I BRANCH IF ERROR
1597
1598
1599
1600
```

```
1543 006636 000001 PRG1C: WAIT I WAIT
1544 006640 000776 BR PRG1C I HERE
1545
1546
1547
1548
1549
1550
1551 006642 104000 *****
1552 006644 012622 IPRG2=SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS
1553 006646 004537 004014 *****
1554 006652 004737 004402 I THE MESSAGE 'A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS
1555 006656 052777 000004 I BACK 1234567890.'
1556 006664 004537 004742 I
1557 006670 012702 013504 PRG2: TYPE I TYPE PROGRAM
1558 006674 112201 P2TIT I TITLE
1559 006676 020127 000045 JSR S,LINSEL I DO SET PARAMETERS
1560 006702 001772 JSR 7,SETPAR I GET REQUEST TO SEND
1561 006704 032737 000100 001374 PRG2A: JSR S,LINCON I DO MAKE LINE CONNECTION
1562 006712 001402 PRG2B: MOV #PRG2H,X2 I GET ADDRESS OF MESSAGE
1563 006714 004737 003600 PRG2C: MOVB (2)+,X1 I GET FIRST CHARACTER
1564 006720 004537 004742 CMP X1,#X I TERMINATOR CHARACTER
1565 006724 010177 172256 BEQ PRG2B I SEND MESSAGE
1566 006730 105777 172250 RIT #BIT6,SR I PARITY ENABLED
1567 006734 100375 BEQ .+6
1568 006736 000756 JSR 7,GENPAR I GENERATE PARITY
1569
1570
1571
1572 006740 104000 *****
1573 006742 012672 IPRG3=PROGRAM TO RECEIVE A MESSAGE.
1574 006744 004537 004014 *****
1575 006750 004737 004402 I
1576 006754 012706 001176 PRG3: TYPE I TYPE PROGRAM
1577 006760 052777 000004 172212 P3TIT I TITLE
1578 006766 004537 004742 JSR S,LINSEL I GET PARAMETERS
1579 006772 104006 JSR 7,SETPAR I REPOSITION STACK POINTER
1580 006774 007036 MOV #BIT2,#RCSR I GET REQUEST TO SEND
1581 006776 104007 JSR S,LINCON I MAKE LINE CONNECTION
1582 007000 007270 STRNV I SET RECEIVER INTERRUPT
1583 007002 005037 007470 RINTV I TO THIS ADDRESS
1584 007006 013700 001234 STTXV I SET TRANSMITTER INTERRUPT
1585 007012 012720 007422 TINTV I TO THIS ADDRESS
1586 007016 013710 001236 CLR WORDS
1587 007022 012701 014710 MOV TPVTR,X0
1588 007026 052737 100000 007466 MOV #TPINT,(0)+ I LOAD TELEPRINTER VECTOR
1589 007034 004737 007034 MOV TPLVL,(0) I AND PRIORITY
1590 007040 052777 000140 172132 BIS #BIT15,TPLAG I SET BIT 15
1591 007046 005037 177776 JSR 7,TCRLF I SEND CRLF
1592 007052 000001 CLR PSW I ENABLE RECEIVER INTERRUPTS
1593 007054 000776 WAIT I DO
1594 007056 017737 172116 001370 RINTV: MOV #2 I NOTHING
1595 007064 100461 BMT ERR3A I BRANCH IF ERROR
1596 007066 105737 001370 Y8B RXCSR I TEST
```

```
1597 007072 100044 RPL ERR3R
1598 007074 005237 007470 INC WORDS
1599 007100 017737 172076 007472 MOV @RXBUF,RXBUF ;GET DATA
1600 007106 113711 007472 MOV@ RXBUF,(1)
1601 007112 005737 007472 TST RXBUF
1602 007116 100455 RMI ERR3C
1603 007120 105777 171050 TST@ @SRPTR ;ECHO OPTION SELECTED
1604 007124 100405 RMI RINT3A
1605 007126 105777 172052 TST@ @TXCSR
1606 007132 100375 BPL =-4
1607 007134 111177 172046 MOV@ (1),@TXBUF ;ECHO CHARACTER
1608 007140 023727 007470 001604 RINT3A: CMP WORDS,#900. ;END OF BUFFER ALLOWED
1609 007146 001411 BEQ RINT3B ;YES EXIT
1610 007150 005737 007466 TST TFLAG ;IS THIS THE FIRST
1611 007154 100441 RMI RINT3E ;CHARACTER BRANCH IF YES.
1612 007156 121103 CMPB (1),X5 ;LAST CHARACTER RECEIVED
1613 007160 001404 BEQ RINT3B
1614 007162 122127 000203 CMPB (1),#203 ;CONTROL C
1615 007166 001401 BEQ RINT3B
1616 007170 000002 RTI ;EXIT
1617 007172 005037 007470 RINT3B: CLR WORDS
1618 007176 042777 000140 171774 BIC #140,@RXCSR ;DISABLE RECEIVER
1619 007204 012701 014711 MOV @OUTBUF+1,X1 ;INITIALIZE BUFFER POINTER
1620 007210 010102 MOV X1,X2
1621 007212 052777 000100 171764 BIS @BIT6,@TXCSR ;ENABLE TRANSMITTER
1622 007220 052777 000100 171776 BIS @BIT6,@TPS ;ENABLE TELEPRINTER
1623 007226 000002 RTI ;EXIT
1624 007230 104015 ERR3A: ERRRX ;TYPE ERROR MESSAGE
1625 007232 013605 LFAIL
1626 007234 042777 000140 171736 BIC #140,@RXCSR ;DISABLE RECEIVER
1627 007242 000644 BR PRG3A
1628 007244 104015 ERR3B: ERRRX ;TYPE
1629 007246 012366 RINTM ;ERROR MESSAGE
1630 007250 000002 RTI ;EXIT
1631 007252 104015 ERR3C: ERRRX
1632 007254 013623 ROVER
1633 007256 000002 RTI
1634
1635 007260 005037 007466 ;
1636 007264 112103 RINT3E: CLR TFLAG
1637 007266 000002 MOV@ (1),X3
1638 RTI
1639
1639 007270 017737 171710 001366 ;
1640 007276 105737 001366 TINT3: MOV @TXCSR,TXCSR ;GET TXCSR DATA
1641 007302 100016 TST@ TXCSR ;TEST
1642 007304 112177 171676 BPL TINT3B
1643 007310 005237 007470 MOV@ (1),@TXBUF ;TRANSMIT CHARACTER
1644 007314 121103 INC WORDS
1645 007316 001431 CMPB (1),X3 ;ALL CHARACTERS TRANSMITTED
1646 007320 023727 007470 001604 BEQ TINT3C
1647 007326 001425 CMP WORDS,#900.
1648 007330 121127 000203 BEQ TINT3C
1649 007334 001422 CMPB (1),#203 ;= CONTROL C
1650 007336 000002 BEQ TINT3C
RTI ;RETURN TO MAIN PROGRAM
```

```
1651 007340 017737 171660 001372 TINT3B: MOV @TPS,TEMP ;SAVE TELEPRINTER STATUS
1652 007346 005077 171652 CLR @TPS ;DISABLE INTERRUPT
1653 007352 105777 171646 TST@ @TPS ;WAIT FOR
1654 007356 100375 BPL =-4 ;TELEPRINTER TO FINISH
1655 007360 104014 ERRTX ;TYPE
1656 007362 012407 TINTM ;ERROR MESSAGE
1657 007364 105777 171634 TST@ @TPS ;WAIT FOR TELEPRINTER
1658 007370 100375 BPL =-4 ;TO FINISH
1659 007372 013777 001372 171624 MOV TEMP,@TPS ;RESTORE TELEPRINTER STATUS
1660 007400 000002 RTI ;EXIT
1661
1662 007402 042777 000100 171574 ;
1663 007410 032777 000100 171606 TINT3C: BIC @BIT6,@TXCSR ;DISABLE INTERRUPT
1664 007416 001421 BIT @BIT6,@TPS ;IS TTY ACTIVE
1665 007420 000002 BEQ PRG3EX
1666 RTI
1667
1668 007422 112277 171600 ;
1669 007426 121203 TPINT: MOV@ (2),@TPB ;TYPE CHARACTER
1670 007430 001404 CMPB (2),X3 ;WAS THIS THE LAST CHAR.
1671 007432 121227 000203 BEQ TPINTA
1672 007436 001401 CMPB (2),#203 ;= CONTROL C
1673 007440 000002 BEQ TPINTA
1674 007442 042777 000100 171550 RTI
1675 007450 032777 000100 171526 TPINTA: BIC @BIT6,@TPS ;DISABLE INTERRUPT
1676 007456 001401 BIT @BIT6,@TXCSR ;IS TRANSMITTER ACTIVE
1677 007460 000002 BEQ =+4
1678 007462 000137 006754 RTI ;EXIT
1679 007466 000000 PRG3EX: JMP PRG3A
1680 007470 000000 TFLAG: 0
1681 007472 000000 WORDS: 0
1682 RXBUF: 0
1683 ;*****
1684 ;PRG0-SPECIAL MESSAGE TRANSMIT ONLY THIS PROGRAM TRANSMITS
1685 ;*****
1686 ;MESSAGE SPIRAL PATTERN
1687 ;
1688 007474 100000 PRG0: TYPE ;TYPE PROGRAM
1689 007476 012730 P4TIT ;TITLE
1690 007500 004537 JSR 5,LINSEL
1691 007504 004737 004002 JSR 7,SETPAR ;GO SET PARAMETERS
1692 007510 052777 000004 171462 BIS @BIT2,@RXCSR ;SET REQUEST TO SEND
1693 007516 012737 000110 007652 MOV #72,COLMN ;INIT PAGE WIDTH
1694 007524 012703 000000 MOV #40,X3 ;SET LINE START CHAR
1695 007530 004537 004742 PRG0A: JSR 5,LINCON ;GO MAKE LINE CONNECTION
1696 007534 004737 007654 JSR 7,TCRLF
1697 007540 010302 PRG0B: MOV X3,X2 ;GET FIRST CHARACTER
1698 007542 110201 PRG0C: MOV@ X2,X1 ;GET CHARACTER
1699 007544 120127 000136 CMPB X1,#136 ;TERMINATOR CHARACTER
1700 007550 001003 BNE PRG0D ;RESEND MESSAGE
1701 007552 012702 000000 MOV #40,X2
1702 007556 000771 BR PRG0C
1703 007560 032737 000100 001374 PRG0D: BIT @BIT6,SRT ;PARITY ENABLED
1704 007566 001402 BEQ =+6
1705 007570 000737 003600 JSR 7,GENPAR ;GENERATE PARITY
```

```
1705 007574 004537 000742 JSR 5,LINCON ;CHECK LINE CONNECTION
1706 007600 010177 171402 MOV X1,@TXBUF ;LOAD BUFFER
1707 007604 105777 171374 TSTB @TXCSR ;AND WAIT FOR CHARACTER
1708 007610 100375 BPL ,-4 ;TO BE TRANSMITTED
1709 007612 005202 INC X2 ;SET FOR NEXT CHAR
1710 007614 005337 007652 DEC COLMN ;ALL COLUMNS PRINTED?
1711 007620 001350 BNE PRG4C ;NO, GET NEXT CHAR
1712 007622 012737 000110 007652 MOV #72,COLMN ;RESET COLUMN COUNTER
1713 007630 000737 007634 JSR 7,TCRLF
1714 007634 005203 INC X3
1715 007636 120527 000136 ;UPDATE LINE START CHAR
1716 007642 001336 CMPB #X3,#136 ;LAST IN SET
1717 007644 012703 000040 BNE PRG4B ;NO
1718 007650 000733 MOV #40,X3 ;YES, RESET
1719 007652 000000 BR PRG4B ;GET NEXT CHARACTER.
;
; COLUMN: 0
;
; SEND CR LF
;
1722 1723 007654 112777 000015 171324 TCRLF: MOVB #15,@TXBUF ;SEND CR,LF
1724 007662 105777 171316 TSTB @TXCSR
1725 007666 100375 BPL ,-4
1726 007670 112777 000012 171310 MOVB #12,@TXBUF
1727 007676 105777 171302 TSTB @TXCSR
1728 007702 100375 BPL ,-4
1729 007704 000207 RTS X7
;*****
;PROGRAM 5
;*****
PRG5: TYPE
PSYIT
JSR 5,LINSEL
RESET
JSR 7,SETPAR
BIS #6,@RXCSR ;SET DTR RQ TO SND
;TYPE MESSAGE TO MAKE
;LINE CONNECTION
;WAIT FOR USER TO MAKE LINE CONNECTION
;READ BUFFER
;TEST FOR CLEAR TO SEND
PRG5A: TYPE
MAKCON
HALT
TST @RXBUF
BIT13,@RXCSR ;TEST FOR CLEAR TO SEND
PRG5B: TYPE
LINCMM
BR PRG5A ;AND TRY AGAIN
PRG5C: TYPE
LINMAD
CLR ERRCNT
MOV #PRG2M,X2 ;GET BASE ADDRESS OF DATA TO BE TRANSMITTED
MOV #2,@X1 ;GET A CHARACTER
CMP X1,#*% ;WAS IT THE TERMINATOR?
BEQ PRG5E
BIT6,BRT ;WAS PARITY OPTION SELECTED?
BEQ PRG5D ;BRANCH IF NO PARITY DESIRED
JSR 7,GENPAR ;GENERATE PARITY ON CHAR. IN R1
BIT13,@RXCSR ;CHECK CLEAR TO SEND
1730
1731
1732
1733 007706 104000
1734 007710 012777
1735 007712 004537 004014
1736 007716 000005
1737 007720 004737 004402
1738 007724 052777 000006 171246
1739 007732 104000
1740 007734 013251
1741 007736 000000
1742 007740 005777 171236
1743 007744 032777 020000 171226
1744 007752 001003
1745 007754 104000
1746 007756 013377
1747 007760 000764
1748 007762 104000
1749 007764 013427
1750 007766 005037 010114
1751 007772 012702 013504
1752 007776 112201
1753 010000 020127 000045
1754 010004 001440
1755 010006 032737 000100 001374
1756 010014 001402
1757 010016 004737 003600
1758 010022 032777 020000 171150
```

```
1759 010030 001751 BEQ PRG5AA ;TYPE ERROR MSG. IF NOT SET
1760 010032 010177 171150 MOV X1,@TXBUF ;TRANSMIT THE CHARACTER
1761 010036 005777 171140 TSTB @RXBUF ;ANY ERROR FLAG?
1762 010042 100001 BPL ,-4 ;BRANCH IF NO ERROR FLAG
1763 010044 104003 ERROR ;ERROR! SOME ERROR FLAG IS SET
1764 010046 105777 171126 TSTB @RXCSR ;WAIT FOR THE RECEIVER TO RECEIVE
1765 010052 100375 BPL ,-4 ;THE TRANSMITTED CHARACTER
1766 010054 117703 171122 MOVB @RXBUF,X3 ;SAVE IT IN R3
1767 010060 043701 001362 BIC CARMASK,X1 ;CLEAR NON-TRANSMITTED BITS
1768 010064 120103 CMPB X1,X3 ;WAS RECEIVED & TRANSMITTED DATA THE SAME
1769 010066 001403 BEQ PRG5D
1770 010070 104003 ERROR ;ERROR! DATA ERROR
1771 010072 005237 010114 INC ERRCNT
1772 010076 105777 171102 PRG5D: TSTB @TXCSR ;WAIT FOR TRANSMITTER TO FINISH
1773 010102 100375 BPL ,-4
1774 010104 000734 BR PRG5C
1775 010106 104000 PRG5E: TYPE
1776 010110 013350 ENDPAS
1777 010112 000727 BR PRG5B
1778 010114 000000 ERRCNT: OPEN
;THIS ROUTINE MOVES THE CONTENTS OF R2 TO THE ADDRESS SPECIFIED
;BY R1
1781 010116 012737 000006 000004 MOVIT: MOV #6,4 ;SET UP FOR RETURN
1782 010124 012737 000002 000006 MOV #2,6
1783 010132 012700 000037 MOV #31,%0 ;GET COUNTER
1784 010136 010231 MOVIT: MOV #2,@(1)+ ;MOVE THE DATA
1785 010140 005300 DEC %0 ;ALL DATA MOVED?
1786 010142 001375 BNE MOVITA ;NO, RETURN
1787 010144 012737 004520 000004 MOV #ERTP,MACHER
1788 010152 012737 000040 000006 MOV #40,MACHER+2
1789 010160 000205 RTS 5 ;RETURN
;
; SUBROUTINE TO LOAD ALL VECTORS
1792 010162 012701 011022 LDVECB: MOV #R18R0,X1
1793 010166 012702 010334 MOV #VECTAB,X2
1794 010172 012703 000010 MOV #10,X3
1795 010176 012704 000037 MOV #31,X4
1796 010202 032712 000001 LDVECB: BIT #BIT0,(2) ;DOES THIS VECTOR EXIST
1797 010206 001002 BNE LDVEC1 ;NO, SKIP LOADING
1798 010210 010172 000000 MOV #1,@(2) ;LOAD VECTOR
1799 010214 060301 LDVEC1: ADD X3,X1
1800 010216 005722 TST (2)+
1801 010220 005304 DEC X4
1802 010222 001367 BNE LDVECB
1803 010224 000207 RTS 7
;
1804
1805 010226 012701 011412 LDTVEC: MOV #T18R0,X1
1806 010232 012702 010334 MOV #VECTAB,X2
1807 010236 012703 000010 MOV #10,X3
1808 010242 012704 000037 MOV #31,X4
1809 010246 032712 000001 LDTVEC: BIT #BIT0,(2) ;DOES THIS VECTOR EXIST
1810 010252 001003 BNE LDVEC2 ;NO, SKIP LOADING
1811 010254 011200 MOV #2,X0
1812 010256 010160 000004 MOV #1,4(0)
```

```

1813 010262 060101 LDVECT1 ADN K3,X1
1814 010264 005722 T3T (2)+
1815 010266 005106 DEC K4
1816 010270 001366 RNF LOTVED
1817 010272 000207 RTS 7
1818
1819
1820 010274 012701 1 ROUTINE TO LOAD PRIORITY LEVEL
1821 010276 012702 LOPRT1 MOV #VECTAB,X1 7 IN VECTOR +2
1822 010278 012703 MOV #300,X2 IGET BASE VECTOR
1823 010280 032711 000057 MOV #31,X3 IGET LEVEL 7
1824 010282 032711 000001 LOPR1A BIT #BIT0,(1) LOAD COUNTER
1825 010284 011100 RNF LOPR1A JDOES THIS VECTOR EXIST
1826 010286 010286 MOV (1),K4 JND SKIP LOADING
1827 010288 005721 MOV #2,X2(4) LOAD VECTOR +2
1828 010290 005721 LOPR1X TST K3(4)
1829 010292 005721 DEC (1)+
1830 010294 001367 BNE LOPR1A JPOINT TO NEXT VECTOR
1831 010296 000207 RTS INCREMENT COUNTER
1832
1833 010324 000301 /VECTOR ASSIGNMENT TABLE
1834 010326 000301 VECTAB: 301
1835 010328 000301 311
1836 010330 000301 321
1837 010332 000301 331
1838 010334 000301 341
1839 010336 000301 351
1840 010338 000301 361
1841 010340 000301 371
1842 010342 000301 381
1843 010344 000301 391
1844 010346 000301 401
1845 010348 000301 411
1846 010350 000301 421
1847 010352 000301 431
1848 010354 000301 441
1849 010356 000301 451
1850 010358 000301 461
1851 010360 000301 471
1852 010362 000301 481
1853 010364 000301 491
1854 010366 000301 501
1855 010368 000301 511
1856 010370 000301 521
1857 010372 000301 531
1858 010374 000301 541
1859 010376 000301 551
1860 010378 000301 561
1861 010380 000301 571
1862 010382 000301 581
1863 010384 000301 591
1864 010386 000301 601
1865 010388 000301 611
1866 010390 000301 621
1867 010392 000301 631
1868 010394 000301 641
1869 010396 000301 651
1870 010398 000301 661
1871 010400 000301 671
1872 010402 000301 681
1873 010404 000301 691
1874 010406 000301 701
1875 010408 000301 711
1876 010410 000301 721
1877 010412 000301 731
1878 010414 000301 741
1879 010416 000301 751
1880 010418 000301 761
1881 010420 000301 771
1882 010422 000301 781
1883 010424 000301 791
1884 010426 000301 801
1885 010428 000301 811
1886 010430 000301 821
1887 010432 000301 831
1888 010434 000301 841
1889 010436 000301 851
1890 010438 000301 861
1891 010440 000301 871
1892 010442 000301 881
1893 010444 000301 891
1894 010446 000301 901
1895 010448 000301 911
1896 010450 000301 921
1897 010452 000301 931
1898 010454 000301 941
1899 010456 000301 951
1900 010458 000301 961
1901 010460 000301 971
1902 010462 000301 981
1903 010464 000301 991
1904 010466 000301 000
1905 010468 000301 000
1906 010470 000301 000
1907 010472 000301 000
1908 010474 000301 000
1909 010476 000301 000
1910 010478 000301 000
1911 010480 000301 000
1912 010482 000301 000
1913 010484 000301 000
1914 010486 000301 000
1915 010488 000301 000
1916 010490 000301 000
1917 010492 000301 000
1918 010494 000301 000
1919 010496 000301 000
1920 010498 000301 000

```

OZDLB8 MACV11.P7(657) 17-NOV-75 14:50 PAGE 39
 OZDLB8.SRC

```

1867 RCVR N,N,A
1868 NAN+10
1869 ASA+1
1870 .ENDR
1871 N#0
1872
1873 RBUP1 31,
1874 RBUPP N,N,A
1875 NAN+10
1876 ASA+1
1877 .ENDR
1878 N#0
1879 TCOR1 31,
1880 .REPT N,N,A
1881 TXYT
1882 NAN+10
1883 ASA+1
1884 .ENDR
1885 N#0
1886 TBUP1 31,
1887 .REPT N,N,A
1888 TBUPP
1889 NAN+10
1890 ASA+1
1891 .ENDR
1892 N#0
1893 .REPT 31,
1894 TOR N
1895 .REPT N
1896 N#0
1897 .ENDR
1898 TOR
1899 N#0
1900 .REPT 38,
1901 TOST N
1902 .ENDR
1903
1904 JMS9AGE5
1905 HTTLE1 .ASCII 'DL11=8 ON LINE TEST = MAINDEC=11-OZDLB-8*'
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920

```

HLINER .ASCII ' '
 MDADR .ASCII ' '
 TRAP ATRO'

2029	013246	022501	100						
2030	013251	045	040515	042513	MAKCON:	.ASCII	'XMAKE LINE CONNECTION.0'		
2031	013256	046040	047111	020105					
2032	013264	047503	047116	041505					
2033	013272	044524	047117	040056					
2034	013300	042040	052101	020101	CERDAT:	.ASCII	' DATA S/B: '		
2035	013306	027523	035102	040					
2036	013313	040	020040	020040	CSB:	.ASCII	' WAS: '		
2037	013320	040527	035123	040					
2038	013325	040	020040	020040	CWAS:	.ASCII	' CHAR NO. '		
2039	013332	044103	051101	047040					
2040	013340	027117	040						
2041	013343	040	020040	040040	CRNUM:	.ASCII	' 0'		
2042	013350	007			ENDPAS:	.BYTE	007		
2043	013351	100				.ASCII	'0'		
2044	013352	037445	100		AINPRG:	.ASCII	'X?'		
2045	013355	045	040520	040522	PARMTS:	.ASCII	'XPARAMETERS '		
2046	013362	042515	042524	051522					
2047	013370	036440	040						
2048	013373	040	020040	100	APARM:	.ASCII	' 0'		
2049	013377	040	046103	040505	LINCHM:	.ASCII	' CLEAR TO SEND NOT SET.0'		
2050	013404	020122	047524	051440					
2051	013412	047105	020104	047516					
2052	013420	020124	042523	027124					
2053	013426	100							
2054	013427	045	044514	042516	LINMAD:	.ASCII	'XLINE CONNECTION MADE.0'		
2055	013434	041040	047117	042516					
2056	013442	052103	047511	020116					
2057	013450	040515	042504	040056					
2058	013456	041045	047117	044506	ACONFIG:	.ASCII	'XCONFIGURATION # '		
2059	013464	052507	040522	044524					
2060	013472	047117	021440	040					
2061	013477	040	020040	100	TCONFIG:	.ASCII	' 0'		
2062	013504					.EVEN			
2063	013504	015	012		PRG2M:	.BYTE	015,012		
2064	013506	044124	020105	052521		.ASCII	'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK'		
2065	013514	041511	020113	051102					
2066	013522	053517	020116	047506					
2067	013530	020130	052512	050115					
2068	013536	042105	047440	042526					
2069	013544	020122	044124	020105					
2070	013552	040514	054532	042040					
2071	013560	043517	020123	040502					
2072	013566	045503							
2073	013570	030040	031061	032063		.ASCII	' 0123456789.X'		
2074	013576	033065	034067	027071					
2075	013604	045							
2076	013605	045	044514	042516	LFAIL:	.ASCII	'XLINE FAILED0'		
2077	013612	043040	044501	042514					
2078	013620	022504	100						
2079	013623	045	053117	051105	ROVER:	.ASCII	'XOVERRUN, FRAME, OR PARITY ERRORX0'		
2080	013630	052522	026116	043040					
2081	013636	040522	042515	020054					
2082	013644	051117	050040	051101					

2083	013652	052111	020131	051105					
2084	013660	047522	022522	100					
2085	013665	045	044514	042516	ALINEX:	.ASCII	'XLINE # '		
2086	013672	021440	040						
2087	013675	040	020040	040527	TLINEX:	.ASCII	' WAS SELECTED0'		
2088	013702	020123	042523	042514					
2089	013710	052103	042105	100					
2090	013715	045	044514	042516	ALINE:	.ASCII	'XLINE # '		
2091	013722	021440	040						
2092	013725	040	020040	040527	TLINE:	.ASCII	' WAS CALLED'		
2093	013732	020123	040503	046114					
2094	013740	042105							
2095	013742	052045	050131	020105		.ASCII	'XTYPE IN OCTAL: CONFIGURATION IN BITS-1 & MODEM TYPE'		
2096	013750	047111	047440	052103					
2097	013756	046101	020072	047503					
2098	013764	043116	043511	051125					
2099	013772	052101	047511	020116					
2100	014000	047111	041040	052111					
2101	014006	026460	020061	020046					
2102	014014	047515	042504	020115					
2103	014022	054524	042520						
2104	014026	044045	020116	044502		.ASCII	'XIN BITS (0=103, 1=202) 0'		
2105	014034	031124	020040	030050					
2106	014042	030475	031460	020054					
2107	014050	036461	030062	024462					
2108	014056	020040	020040	020040					
2109	014064	100							
2110	014065	045	054524	042520	WRU:	.ASCII	'XTYPE IN OCTAL IN BITS=4 THE LINE # YOU ARE CALLING FROM 0'		
2111	014072	044440	020116	041517					
2112	014100	040524	020114	047111					
2113	014106	041040	052111	026460					
2114	014114	020064	044124	020105					
2115	014122	044514	042516	021440					
2116	014130	054440	052517	040440					
2117	014136	042522	041440	046101					
2118	014144	044514	043516	043040					
2119	014152	047522	020115	020040					
2120	014160	040040							
2121	014162	054445	052517	041440	URI:	.ASCII	'XYOU CALLED FROM LINE # '		
2122	014170	046101	042514	020104					
2123	014176	051106	046517	046040					
2124	014204	047111	020105	020043					
2125	014212	020040	100						
2126	014215	045	051120	051505	URA:	.ASCII	' 0'		
2127	014222	020123	040504	040524	BUTTON:	.ASCII	'XPRESS DATA BUTTON ON DATA PHONE0'		
2128	014230	041040	052125	047524					
2129	014236	020116	047117	042040					
2130	014244	052101	020101	044120					
2131	014252	047117	040105						
2132	014256	052045	050131	020105	LDLINE:	.ASCII	'XTYPE LINE NO. IN OCTAL IN BIT 3=7 0'		
2133	014264	044514	042516	047040					
2134	014272	027117	044440	020116					
2135	014300	041917	040524	020114					
2136	014306	047111	041040	052111					

2137	014314	031400	033455	020040	
2138	014322	022040	100		
2139	014325	045	042522	047503	MPWRFI .ASCII 'XRECOVERED FROM POWER FAILURE'
2140	014332	042526	042522	020104	
2141	014340	051106	046517	050040	
2142	014346	053517	051105	043040	
2143	014354	044501	052514	042522	
2144	014362	100			
2145	014363	045	046111	042514	MH0DXI .ASCII 'XILLEGAL LINE NO.0'
2146	014370	040507	020114	044514	
2147	014376	042516	047040	027117	
2148	014404	100			
2149	014405	045	046111	042514	MH0DDI .ASCII 'XILLEGAL DEVICE ADDRESS0'
2150	014412	040507	020114	042504	
2151	014420	044526	042503	040440	
2152	014426	042104	042522	051523	
2153	014434	100			
2154	014435	045	054524	042520	MH0D1I .ASCII 'XTYPE IN OCTAL I BIT 0-5 THE LINE NUMBER OF DEVICE ADDRESS TO BE'
2155	014442	044440	020116	041517	
2156	014450	040524	020114	020072	
2157	014456	044502	020124	026460	
2158	014464	020065	044124	020105	
2159	014472	044514	042516	047040	
2160	014500	046525	042502	020122	
2161	014506	043117	042040	053105	
2162	014514	041511	020105	042101	
2163	014522	051104	051505	020123	
2164	014530	047524	041040	105	
2165	014535	040	047515	044504	.ASCII ' MODIFIED 0'
2166	014542	044506	042105	020040	
2167	014550	020040	100		
2168	014553	045	054524	042520	MH0D2I .ASCII 'XTYPE IN NEW RXCSR DEVICE ADDRESS 0'
2169	014560	044440	020116	042516	
2170	014566	020127	054122	051503	
2171	014574	020122	042504	044526	
2172	014602	042503	040440	042104	
2173	014610	042522	051523	020040	
2174	014616	020040	100		
2175	014621	045	054524	042520	MH0D3I .ASCII 'XTYPE IN 177777 TO CHANGE ANOTHER'
2176	014626	044440	020116	033461	
2177	014634	033467	033467	020040	
2178	014642	047524	041440	040510	
2179	014650	043516	020105	047101	
2180	014656	052117	042510	122	
2181	014663	040	042504	044526	.ASCII ' DEVICE ADDRESS 0'
2182	014670	042503	040440	042104	
2183	014676	042522	051523	020040	
2184	014704	020040	100		
2185		014710			.EVEN
2186	014710	020000			OUTBUF: OPEN
2187		016660			.=OUTBUF+1000.
2188	016660	020000			INBUF: OPEN
2189		020630			.=INBUF+1000.
2190		015054			BUFF=OUTBUF+100.

2191 020630 000001 DEND: .END

A	= 000037	ACONFI	013456	ADN1	004004	ADTENP	004362
AGAIN	003746	AINPRG	013352	ALTNF	013715	ALINEX	013665
APARM	013373	APC	012434	ARXCSR	012466	ARXWAS	012500
ATXCSR	012445	ATXNAS	012457	RDCNV	004216	RDCNVA	004250
RDCNVB	004302	RDCNVC	004304	RDCNVV	004306	RIT0	= 000001
BIT1	= 000002	RIT10	= 002000	RIT11	= 004000	RIT12	= 010000
BIT13	= 020000	RIT14	= 040000	RIT15	= 100000	RIT2	= 000004
BIT3	= 000010	RIT4	= 000020	RIT5	= 000040	RIT6	= 000100
BIT7	= 000200	RIT8	= 000400	RIT9	= 001000	RMOVA	004204
MOVE	004174	RUFF	= 015054	RUFFP	001400	BUTTON	014215
CALLED	001404	CALLER	001402	CARMSK	001362	CERDAT	013300
CHALT	= 000004	CHLT	002366	CNVCTR	004354	COLMN	007652
CONFIG	001332	CONN	006250	COUNT	001324	CRNUM	013343
CRXTX	002752	CRXTXA	003024	CRXTXB	003026	CRXYXC	003032
CSB	013313	CSRADD	012357	CTRD	001364	CWAS	013325
DATCHK	= 104004	DCERR	005532	DECVL	004374	DELAY	= 104016
DEND	020630	DIGIT	004356	DIRCON	006234	DLCNT	003526
DLY	003466	DLYA	003504	OLVB	003510	OLYC	003522
DOIT	003742	DONE	003776	DTCHK	002412	DTCHKA	002510
DTERR	013232	FCDAT	003346	EHALT	= 104010	EHLT	002400
EMLTA	002410	FIGHT	003632	EMTA	002234	EMTINT	002222
EMTTAB	001264	FM0	012427	FNOP8	013350	ERR	002512
ERRA	002562	FRB	002624	ERRC	002632	ERRCNT	0101.4
ERRD	002642	FRRE	002644	ERROR	= 104003	ERROR1	= 104013
ERRRX	= 104015	FRRTX	= 104014	ERR1	002534	ERR3A	007230
ERR3B	007244	FR3C	007252	ERTP	004520	ERTPA	004570
FILL	004146	FILLA	004154	FINISH	005476	FIVE	003720
FNAP	001350	FNONE	001346	FORMAD	004704	FROMPC	001342
FTITLE	001344	GENPAR	003600	INBUF	016660	INBUFF	001376
INCPRG	002354	INDAT	003322	INFL	004142	LDLINE	014256
LDPRI	010274	LDPRIA	010310	LDPRIX	010324	LDVVEC	010226
LDTVED	010246	LDVECB	010202	LDVECS	010162	LDVEC1	010214
LDVECD	010262	LFAIL	013605	LINCA	004760	LINCB	004776
LINCF	005056	LINCMM	013377	LINCON	004742	LINE	001330
LINENO	001354	LINEUP	005074	LINMAD	013427	LINSA	004056
LINSEL	004014	MACHER	000004	MAKCON	013251	MANUAL	= 100000
MAPA	001520	MAPEND	001730	MAPERR	001752	MAPNE	001550
MAPOK	001560	MAPOKA	001672	MAPVEC	004626	MDADR	012154
MFROM	012346	MAPLINE	012146	MMOD0	014405	MMODX	014363
MMOD1	014435	MMOD2	014553	MMOD3	014621	MMONE	012176
MODEM	001412	MODEV	002024	MODEV1	002036	MODEV2	002072
MODEV3	002116	MOVIT	010116	MOVITA	010136	MPWRF	014325
MSWSEL	012213	MOTERR	012247	MYTILE	012012	MT0	012314
MTRAP	012166	N	= 000040	NUMBER	001334	OACNV	003530
OACNVA	003542	NLDP	001336	OPEN	= 000000	OTBUFP	001406
OUTBUF	014710	OVRLAY	005076	OVRLYA	005106	PARBIT	001322
PAREX	004476	PARMTS	013355	PC	=X000007	PFAL	002646
POPSP	= 005726	POPSP2	= 022626	PRGID	001242	PRGNUM	001240
PRGTAB	001244	PRG0	006424	PRG0A	006534	PRG1	006540
PRG1C	006636	PRG2	006642	PRG2A	006664	PRG2B	006670
PRG2C	006674	PRG2M	013504	PRG3	006740	PRG3A	006754
PRG3EX	007462	PRG4	007474	PRG4A	007530	PRG4B	007540
PRG4C	007542	PRG4D	007560	PRG5	007706	PRG5A	007732
PRG5AA	007754	PRG5B	007762	PRG5BB	007772	PRG5C	007776

PRG5D	010076	PRG5E	010106	PRTY0	= 000000	PRTY1	= 000040
PRTY2	= 000100	PRTY3	= 000140	PRTY4	= 000200	PRTY5	= 000240
PRTY6	= 000300	PRTY7	= 000340	PSW	= 177776	PWRUP	002656
PQTIT	012507	PITIT	012553	P2TIT	012622	P3TIT	012672
P4TIT	012730	P5TIT	012777	RBUF	010530	RCSR	010432
RDDAT	003330	RDOCT	003314	RECDAT	001356	REMAP	001416
RESTRT	002676	RETRN	003452	RINTM	012366	RINT3	007056
RINT3A	007140	RINT3B	007172	RINT3E	007260	RISR	005124
RISRA	005164	RISRAA	005206	RISRB	005232	RISRBB	005300
RISRC	005310	RISRCA	005344	RISRD	005376	RISRDA	005416
RISREX	005620	RISRF	005546	RISRFB	005734	RISRFC	006000
RISRFD	006110	RISRFE	006152	RISRFF	006124	RISRFG	006164
RISRFH	006202	RISR0	011022	RISR1	011032	RISR10	011122
RISR11	011132	RISR12	011142	RISR13	011152	RISR14	011162
RISR15	011172	RISR16	011202	RISR17	011212	RISR2	011042
RISR20	011222	RISR21	011232	RISR22	011242	RISR23	011252
RISR24	011262	RISR25	011272	RISR26	011302	RISR27	011312
RISR3	011052	RISR30	011322	RISR31	011332	RISR32	011342
RISR33	011352	RISR34	011362	RISR35	011372	RISR36	011402
RISR4	011062	RISR5	011072	RISR6	011102	RISR7	011112
ROVER	013623	RREAD	003442	RSTPC	002350	RSTP0W	002352
RSTREG	= 104012	RSTRG	002314	RTN	003752	RXBUFF	001202
RXBUFF	007472	RXC3R	001200	RXC3RT	001370	RXERR	002732
RXLVL	001212	RKVTR	001210	R0	=X000000	R1	=X000001
R2	=X000002	R3	=X000003	R4	=X000004	R5	=X000005
SAVE	001326	SAVREG	= 104011	SAVR0	002254	SAVR0	003312
SELCAR	013143	SELPAR	013055	SETPAR	004402	SEVEN	003676
SIX	003654	SOFTSR	000176	SP	=X000006	SRPRT	000174
SRT	001374	STAD	000300	STALL	= 104002	START	001422
START1	001770	STCONT	000226	STKPTR	= 001200	STPARA	004500
STPPA	003110	STPRA	003060	STRVRV	003042	STRXV	= 104006
STTVX	= 104007	STXMTV	003072	SURTEN	004314	SUBTNA	004320
SUBTNB	004334	SVRPC	002310	SVRPCW	002312	TBIT1	004434
TBUF	010724	TBUFPF	001410	TCONFI	013477	TCRLF	007654
TCSR	010626	TEMP	001372	TEMP1	001352	TENPWR	004360
TFLAG	007466	TINTM	012407	TINT3	007270	YINT3B	007340
TINT3C	007402	TISR	006260	TISRA	006276	TISRAA	006274
TISRB	006404	TISRBB	006374	TISRC	006304	TISR0	011412
TISR1	011422	TISR10	011512	TISR11	011522	TISR12	011532
TISR13	011542	TISR14	011552	TISR15	011562	TISR16	011572
TISR17	011602	TISR2	011432	TISR20	011612	TISR21	011622
TISR22	011632	TISR23	011642	TISR24	011652	TISR25	011662
TISR26	011672	TISR27	011702	TISR3	011442	TISR30	011712
TISR31	011722	TISR32	011732	TISR33	011742	TISR34	011752
TISR35	011762	TISR36	011772	TISR37	012002	TISRA	011452
TISR5	011462	TISR6	011472	TISR7	011502	TKR	001222
TKLVL	001232	TKS	001220	TKVTR	001230	TLINE	013725
TLINEX	013675	TOPC	001340	TPR	001226	TPINT	007422
TPINTA	007442	TPLVL	001236	TPS	001224	TPVTR	001234
TXBUFF	001206	TXCSR	001204	TXC3RT	001366	TXERR	002710
TKLVL	001216	TKVTR	001214	TYP	003122	TYPA	003136
TYPC	003160	TYPD	003176	TYPDAT	003242	TYPE	= 104000
TYPS	= 104001	TYPF	003214	TYPS	003244	TYPSA	003300
TYPSAA	003250	TYPSB	003306	UR	014162	URA	014212

DZDLBB MACY11 27(657) 17-NOV-75 14:50 PAGE 48
DZDLBB.SRC SYMROL TABLE

VECTAB 010334 WORDS 007470 WRU 014065 XMTDAT 001360
020630

ERRORS DETECTED: 0

*DZDLBB,DZDLBB=DZDLBB.SRC/SOL
RUN-TIME: 10 17 0 SECONDS
CORE USED: 10K