

VT62

VT62/DZ11 SELF-TEST/LOOPBACK
MD-11-DZDZC-A

EP-DZDZC-A-DL

COPYRIGHT © 1977

FICHE 1 OF 1

JAN 1978

digital

MADE IN USA

000000

REPT 0

IDENTIFICATION

PRODUCT CODE	MAINDEC-11-DZDZC-A-D
PRODUCT NAME	VT62/DZ11 SELFTEST/LOOPBACK
DATE	1-SEP-77
MAINTAINER	DIAGNOSTIC ENGINEERING
AUTHOR	MICHAEL DENSMORE

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 WITHIN.

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) 1977 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS

REVISION HISTORY

REVISION
A

BY
M DENSMORE ORIGINAL RELEASE

TABLE OF CONTENTS

1	GENERAL	PG	4
1 1	ABSTRACT	PG	4
1 2	DEFINITIONS	PG	4
1 3	DESIGN NOTES	PG	4
1 4	DDCMP	PG	5
2	EQUIPMENT REQUIREMENTS	PG	5
3	LOAD PROCEDURE	PG	5
4	STARTING AND RUNNING THE PROGRAM	PG	5
4 1	START	PG	6
4 2	RESTART	PG	9
4 3	CONTINUE	PG	10
4 4	ZERO FLAGS	PG	10
4 5	ADD UNITS	PG	10
4 6	DROP UNITS	PG	10
4 7	DISPLAY PTABLE	PG	11
4 8	SWITCH OPTIONS	PG	11
5	ERROR AND INFORMATIONAL MESSAGES	PG	12
6	RESTRICTIONS	PG	13
7	MISC	PG	13
8	TEST DESCRIPTIONS	PG	13
8 1	MICRODIAGNOSTIC	PG	13
8 2	LOOPBACK	PG	13
9	INDEX	PG	15

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

1 0 GENERAL
1 1 ABSTRACT
MAINDEC-11-DZDZC RUNS THE SELFTEST FOR THE VT62 TERMINAL AND EXERCISES
THE LINK BETWEEN THE TERMINAL AND THE HOST THIS PARTICULAR VERSION
SUPPORTS MULTIDROP, FULL DUPLEX COMMUNICATION VIA THE DZ-11 COMMUNICATION
INTERFACE OTHER VERSIONS (MAINDEC-11-DZDPG AND MAINDEC-11-DZDVP)
SUPPORT THE DUP-11 AND DV-11 COMMUNICATION INTERFACES RESPECTIVELY THE
PROGRAM IS XXDP COMPATIBLE AND SUPPORTS SEQUENTIAL TESTING OF UP TO 8
VT62'S PLEASE READ ALL SECTIONS OF THIS DOCUMENT CAREFULLY BEFORE USING
THE PROGRAM PAY PARTICULAR ATTENTION TO SECTIONS 3 0 THROUGH 6 0

1 2 DEFINITIONS
MODEM - EQUIPMENT, ALSO KNOWN AS A "DATASET", WHICH PERFORMS THE
MODULATION/DEMULATION OF DATA SIGNALS THIS UNIT PROVIDES TIMING
FOR SYNCHRONOUS COMMUNICATION DEVICES
MODEM ELIMINATOR - EQUIPMENT WHICH CONNECTS TWO SYNCHRONOUS DEVICES USING
A CABLE RATHER THAN TELEPHONE VOICE LINES THE ELIMINATOR HAS A
CLOCK FOR TIMING
DDCMP PROTOCOL - DEC'S COMMUNICATION LINE PROTOCOL
UUT - UNIT UNDER TEST
PTABLE - A TABLE OF INFORMATION, WITH ENTRIES FOR EACH UNIT UNDER TEST,
THAT DESCRIBES THE HARDWARE (BUS ADDRESS, VECTOR, ETC)
DUP - SINGLE LINE, SYNCHRONOUS COMMUNICATION INTERFACE
DZ - EIGHT LINE, MULTIPLEXED, SYNCHRONOUS COMMUNICATION INTERFACE
DV - SIXTEEN LINE, MULTIPLEXED, SYNCHRONOUS COMMUNICATION INTERFACE
CRC CHECKSUM - CHECKSUM USED IN DEC NETWORK PROTOCOL (CYCLIC REDUNDANCY
CHECK)
MICRODIAGNOSTIC - THE MICROCODE IN THE VT62 THAT PERFORMS THE DIAGNOSTIC
CHECKOUT OF THE HARDWARE (ALSO KNOWN AS "SELF TEST")
SELF TEST - SEE "MICRODIAGNOSTIC"

1 3 DESIGN NOTES
THIS PROGRAM WAS DESIGNED AND IMPLEMENTED USING STRUCTURED TECHNIQUES
THE PROCEDURE CAUSED THE PROGRAM TO HAVE SOME OPERATIONAL CHARACTERISTICS
WHICH ARE OF INTEREST TO THE USER THESE CHARACTERISTICS ARE DISCUSSED
IN THIS SECTION THIS DISCUSSION IS NOT VITAL TO LEARNING HOW TO OPERATE
THE PROGRAM HOWEVER

THE MOST IMPORTANT OPERATIONAL CHARACTERISTIC OF STRUCTURED DESIGN, AS IT
AFFECTS THE USER, IS HOW ERRORS ARE REPORTED THE TOP-DOWN DESIGN OF THE
PROGRAM RESULTS IN A BOTTOM-UP REPORT OF ERROR INFORMATION SEVERAL
LINES OF ERROR INFORMATION ARE PRINTED AS EACH PROGRAM SECTION REPORTS
ITS LEVEL OF INFORMATION AN EXAMPLE OF THIS TYPE OF ERROR LOG WOULD BE
A DZ ERROR THE FIRST ERROR MESSAGE WOULD GIVE SPECIFIC DEVICE
INFORMATION SUCH AS REGISTER CONTENTS AND THEN RETURN CONTROL TO THE NEXT
LEVEL AT THE NEXT LEVEL, THE LOGICAL DEVICE NUMBER OR DEVICE SERIAL
NUMBER WOULD BE REPORTED AT THE LEVEL ABOVE THAT, STILL MORE GENERAL
INFORMATION WOULD BE GIVEN THE REPORTS CONTINUE UNTIL THE HIGHEST LEVEL
IS REACHED

COMMENT FIELDS IN THE LISTING WITH SQUARE BRACKETS () REFER TO

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
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194

FUNCTIONAL BLOCKS IN THE DESIGN AND HAVE NO SIGNIFICANCE TO THE USER
THEY ARE RETAINED FOR USE BY PROGRAM MAINTAINERS

1 4

DDCMP

THIS SECTION GIVES A VERY BRIEF DESCRIPTION OF DDCMP AS USED BY THIS
DIAGNOSTIC THE DESCRIPTION IS BY NO MEANS COMPLETE. IT IS INTENDED TO
HELP THE USER UNDERSTAND ERROR REPORTS

THERE ARE THREE TYPES OF DDCMP MESSAGES CONTROL, DATA AND BOOTSTRAP
THE BOOTSTRAP MESSAGE IS NOT USED BY THIS DIAGNOSTIC THE DATA MESSAGES
CONSIST OF A 6-BYTE HEADER, A 2-BYTE CRC FOR THE HEADER, THE DATA ITSELF
AND A 2-BYTE CRC FOR THE DATA CONTROL MESSAGES CONSIST OF THE HEADER
(AND ITS CRC) ONLY ALL CONTROL MESSAGES HAVE AN "ENQ" CHARACTER AS THE
FIRST BYTE ALL DATA MESSAGES ARE NUMBERED SO THAT EACH STATION ON THE
LINK CAN KEEP TRACK OF WHICH MESSAGES ARE TRANSMITTED PROPERLY ALL
MESSAGES HAVE A STATION ADDRESS IN THE HEADER SO THAT MESSAGES ARE
DIRECTED TO AND FROM THE PROPER STATIONS ON A MULTIDROP LINE

THERE ARE FOUR CONTROL MESSAGES USED BY THIS DIAGNOSTIC ACK, NAK, STRT
AND STACK ACK IS AN ACKNOWLEDGEMENT THAT THE LAST MESSAGE SENT WAS
PROPERLY RECEIVED NORMALLY, MESSAGES ARE ACKNOWLEDGED BY SENDING A DATA
MESSAGE WITH THE NUMBER OF THE LAST MESSAGE IN THE HEADER ACK'S ARE
USED WHEN THERE IS NO DATA MESSAGE TO SEND BACK NAK IS A NEGATIVE
ACKNOWLEDGEMENT FOR THE LAST MESSAGE SENT AND IT INDICATES THE REASON
FOR THE ERROR STRT IS USED TO INITIATE COMMUNICATION BETWEEN TWO
STATIONS THE RECEIVING STATION SETS ITSELF UP AND ACKNOWLEDGES THE STRT
WITH A STACK, OR START ACKNOWLEDGE THE STRT-STACK SEQUENCE IS USED
BEFORE AND AFTER EACH TEST ALL MESSAGES ARE REQUIRED TO BE ACKNOWLEDGED
WITH EITHER A DATA MESSAGE OR AN ACK

2 2

EQUIPMENT REQUIREMENTS

THIS PROGRAM REQUIRES A PDP-11 WITH A MINIMUM OF 16K CORE AND A CONSOLE
TERMINAL A CLOCK OPTION IS NOT REQUIRED, BUT ONLY ROUGH TIMING IS
AVAILABLE WITHOUT ONE THE PDP-11 CONSOLE SWITCH REGISTER IS NOT USED
ONE OR MORE DZ-11'S MUST BE CONNECTED TO THE TERMINALS UNDER TEST VIA
COMMUNICATION LINES THROUGH EITHER MODEM'S OR NULL MODEM'S THE VT62'S
MUST BE IN ASYNCHRONOUS MODE AND ON-LINE THE TERMINAL ADDRESS SWITCHES
ON THE CONTROL BOARD MUST BE SET TO MATCH THE TERMINAL ADDRESS ASSIGNED
TO THE LOGICAL UNIT (SEE SECTION 4.0)

3 1

LOADING PROCEDURE

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER IT
MAY ALSO BE LOADED FROM PAPER TAPE OR DISK USING XXDP PROCEDURES

4 1

STARTING AND RUNNING THE PROGRAM

THE STARTING ADDRESS FOR THE PROGRAM IS 200 THE PROGRAM WILL FIRST ASK
IF A CLOCK (KW11-L OR KW11-P) IS AVAILABLE IT WILL THEN ASK IF A LINE
PRINTER IS AVAILABLE IF A LINE PRINTER IS SPECIFIED, THE PROGRAM WILL

136
137
138
139
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249

ASK FOR ITS ADDRESS (DEFAULT = 177514) THE PROGRAM ALSO ASKS FOR MEMORY SIZE, Q-BUS OPTION AND LINE FREQUENCY IF NEITHER CLOCK IS SPECIFIED, THE PROGRAM WILL ASK THE USER TO TYPE TWO CHARACTERS, FIVE SECONDS APART, IN ORDER TO PROVIDE ROUGH TIMING.

THE PROGRAM WILL THEN PROMPT THE USER BY TYPING A ">" NOW THE USER MAY TYPE ONE OF SEVEN COMMANDS STA(RT), RES(TART), CON(TINUE), ZFL(AG), ADD, DRO(P) OR DIS(PLAY), WHERE THE LETTERS IN PARENTHESIS NEED NOT BE TYPED IF A TYPING MISTAKE IS MADE, THE RUBOUT KEY IS USED TO DELETE THE LAST CHARACTER(S) TYPED THE PROGRAM WILL TYPE A MESSAGE IF THE COMMAND IS INVALID EACH OF THESE COMMANDS IS EXPLAINED IN THE SECTIONS BELOW SWITCHES ARE USED TO SET SOFTWARE FLAGS SUCH AS "LOOP ON ERROR" THE FLAGS USED BY THIS PROGRAM ARE DESCRIBED IN SECTION 4 8

4 1

START
 THE START COMMAND (STA) ZEROS ALL FLAGS (PRINT ALL ERROR MESSAGES AND DO NOT LOOP WITHIN TESTS) AND INITIATES THE OPERATOR DIALOGUE AT THIS POINT, THE DEVICE TABLE (PTABLE) IS EMPTY AND THE OPERATOR MUST USE THE DIALOGUE TO DESCRIBE ALL UUT'S AS SHOWN BELOW THE NUMBERS IN SQUARE BRACKETS, , RELATE THE TYPEOUT TO THE EXPLANATION BELOW AND ARE NOT TYPED IN ACTUAL OPERATION THE PROCEDURES FOR HANDLING TYPING ERRORS ARE ALSO DESCRIBED BELOW CR REFERS TO THE RETURN KEY (CARRIAGE RETURN)

DIALOGUE

UNITS ? (DECIMAL) 1
 DZ ADDRESS ? (OCTAL) NNNNNN 2
 DZ VECTOR ? (OCTAL) NNN 3
 DZ BUS PRIORITY ? (OCTAL) N 4
 TERMINAL ADDRESS ? (OCTAL) N 5
 LINE NUMBER ? (OCTAL) N 6
 BAUD RATE ? (OCTAL) N 7
 STOP CODE/CHAR LENGTH ? (OCTAL) N 8
 CHANGE S W ? (Y OR N) 9
 HOW MANY INITIATE TRIES BEFORE ABORT ? (OCTAL) 10 10
 USE ALL ZERO'S FOR LOOPBACK DATA ? (Y OR N) Y 11

EXPLANATION

1 TYPE THE TOTAL NO OF UUT'S IN DECIMAL QUESTIONS 2
 THROUGH 5 WILL BE REPEATED UNTIL ALL UUT'S HAVE BEEN

251
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

- DESCRIBED THE USER DOES NOT HAVE TO ANSWER THESE QUESTIONS FOR EACH DZ HOWEVER, IF THERE ARE MULTIPLE UUT'S PER DZ OR PER DZ LINE THE EXPLANATION FOR QUESTION 5 SHOWS HOW TO ENTER MULTIPLE UUT'S
- 2 TYPE THE BUS ADDRESS OF THE DZ THIS MUST BE AN EVEN OCTAL NUMBER NOT GREATER THAN 177776 THE USER MAY TYPE CR TO USE THE DEFAULT VALUE NNNNNN
 - 3 TYPE THE VECTOR ADDRESS FOR THE DZ THIS MUST BE AN EVEN OCTAL NUMBER NOT GREATER THAN 376 THE USER MAY TYPE CR TO USE THE DEFAULT VALUE NNN
 - 4 TYPE THE BUS PRIORITY LEVEL OF THE DZ THIS MUST BE A DIGIT FROM 0 TO 7 AGAIN, THE USER MAY TYPE CR TO GET THE DEFAULT VALUE N
 - 5 TYPE THE TERMINAL ADDRESS FOR EACH TERMINAL ATTACHED TO THE DZ LINE SEPARATE EACH ADDRESS WITH A COMMA THIS IS THE ADDRESS SELECTED BY THE SWITCHES ON THE I/O BOARD IN THE TERMINAL AND IS AN OCTAL NUMBER BETWEEN 0 AND 377 THE USER MAY TYPE CR TO GET THE DEFAULT VALUE N
 - 6 TYPE THE DZ LINE NUMBER (0-7) TO WHICH THE TERMINAL IS ATTACHED TYPE CR TO GET THE DEFAULT OF N
 - 7 TYPE THE CODE FOR THE BAUD RATE DESIRED THE CODES ARE GIVEN IN THE TABLE BELOW TYPE CR TO GET THE DEFAULT N

BAUD RATE TABLE	
TYPE	TO GET THIS RATE
0	50
1	75
2	110
3	134 5
4	150
5	300
6	600
7	1200
10	1800
11	2000
12	2400
13	3600
14	4800
15	7200
16	9600
17	19,200

FOR FASTER TESTING, USE AS HIGH A BAUD RATE AS PRACTICAL A BAUD RATE OF 19,200 IS NOT AVAILABLE ON THE VT62.

- 8 TYPE A CODE TO SELECT THE STOP CODE AND CHARACTER LENGTH THE CODES ARE GIVEN IN THE TABLE BELOW

STOP CODE/CHARACTER LENGTH	
TYPE	STOP CODE LENGTH
0	1 BIT 5 BITS

306
307
308
309
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
358

1	"	6	"
2	"	7	"
3	"	8	"
4	2 8 TS	5	"
5	"	6	"
6	"	7	"
7	"	8	"

- 9 IF THE USER WISHES TO CHANGE EITHER OF THE TWO SOFTWARE PARAMETERS DESCRIBED FOR QUESTIONS 7 OR 8, HE MUST ANSWER THIS QUESTION BY TYPING "Y" IF THE USER WISHES TO BYPASS QUESTIONS 7 AND 8, HE TYPES "N"
- 10 TYPE THE NUMBER OF TIMES THE PROGRAM IS TO ATTEMPT TO INITIALIZE A DUP BEFORE ABORTING THE UNIT THE USER MAY TYPE ANY OCTAL NUMBER BETWEEN 0 AND 177777 TYPE CR TO DEFAULT TO 10 (OCTAL) ATTEMPTS
- 11 IF THE USER TYPES "Y", THE LOOPBACK DATA WILL BE 255 ZERO BYTES IF THE USER TYPES "N", THE LOOPBACK DATA WILL BE A PATTERN OF ALTERNATING 1 AND 0 BITS SEE SECTION 8 2 FOR DETAILS ON THE LOOPBACK TEST

EXAMPLE OF DIALOGUE

ASSUME THE FOLLOWING HARDWARE CONFIGURATION A DZ (160000,300,3) WITH TWO TERMINALS (ADDRESSES 0 AND 1) ON LINE 0, A SECOND DZ (160010,320,3) WITH A SINGLE TERMINAL (ADDRESS 5) ON LINE 2 AND A THIRD DZ (170000,340,4) WITH THREE TERMINALS (ADDRESSES 0) ON LINES 1, 2 AND 3 NOTE THAT VARIABLE BAUD RATES CAN BE SPECIFIED THE USER WOULD USE THE FOLLOWING DIALOGUE TO SET UP THE PROGRAM

\STA

UNITS ? (DECIMAL) 6
 DZ ADDRESS ? (OCTAL) 160000
 DZ VECTOR ? (OCTAL) 310 300
 DZ BUS PRIORITY ? (OCTAL) 4 3
 TERMINAL ADDRESS ? (OCTAL) 0 0.1
 LINE NUMBER ? (OCTAL) 0
 BAUD RATE ? (OCTAL) 0 16
 STOP CODE/CHAR LENGTH (SEE DOC) ? (OCTAL) 0 7
 DZ ADDRESS ? (OCTAL) 160000 160010
 DZ VECTOR ? (OCTAL) 300 320

360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413

DZ BUS PRIORITY ? (OCTAL) 3

TERMINAL ADDRESS ? (OCTAL) 1 5

LINE NUMBER ? (OCTAL) 0 2

BAUD RATE ? (OCTAL) 16 7

STOP CODE/CHAR LENGTH (SEE DOC) ? (OCTAL) 7

DZ ADDRESS ? (OCTAL) 160010 170000

DZ VECTOR ? (OCTAL) 320 340

DZ BUS PRIORITY ? (OCTAL) 3 5

TERMINAL ADDRESS ? (OCTAL) 5 0

LINE NUMBER ? (OCTAL) 2 1-3

BAUD RATE ? (OCTAL) 7 7,16,7

STOP CODE/CHAR LENGTH (SEE DOC) ? (OCTAL) 7

CHANGE S W ? (Y OR N) N

(NOTE WHERE CR WAS USED TO ACCEPT THE DEFAULT)

TYPING ERRORS

- (1) THE RUBOUT KEY IS USED TO DELETE ANY CHARACTERS WHICH ARE IN ERROR
- (2) IF THE RESPONSE IS NOT WITHIN LIMITS OR IS IN THE WRONG FORMAT, THE PROGRAM WILL NOTIFY THE USER AND ASK THE QUESTION AGAIN
- (3) WHEN ENTERING MULTIPLE TERMINAL ADDRESSES FOR A SINGLE DZ, THE USER MAY NEGLECT TO TYPE ONE OR MORE ADDRESSES. IN SUCH A CASE, THE USER MERELY ENTERS THE SAME DZ ADDRESS, VECTOR AND PRIORITY IN THE NEXT SERIES OF QUESTIONS AND THEN ENTERS THE REMAINING TERMINAL ADDRESSES. REMEMBER, THE SERIES OF QUESTIONS (2 THROUGH 5) WILL BE RE-ASKED UNTIL ALL UUT'S HAVE BEEN ACCOUNTED FOR

4 2

RESTART

THE RESTART COMMAND (RES) STARTS THE PROGRAM WITHOUT CLEARING THE PTABLE. ALL FLAGS REVERT TO THEIR INITIAL STATE UNLESS THE FLAG SWITCH IS USED TO CHANGE THEM (SECTION 4 8). THE OPERATOR DIALOGUE DESCRIBED IN 4 1 DOES NOT TAKE PLACE UNLESS REQUESTED. WHEN THIS COMMAND IS USED, THE PROGRAM WILL ASK IF THE USER WISHES TO CHANGE THE HARDWARE TABLE AND IF HE WISHES TO CHANGE THE SOFTWARE TABLE. IF THE USER ANSWERS YES (BY TYPING "Y") TO THE HARDWARE TABLE QUESTION, THAT PORTION OF THE DIALOGUE DESCRIBED IN

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
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468

SECTION 4.1 BEGINNING WITH QUESTION 1 ENDING WITH QUESTION 5 WILL TAKE PLACE IF THE USER ANSWERS YES TO THE SOFTWARE TABLE QUESTION, HE WILL BE ASKED QUESTIONS 7 AND 8.

THE MOST COMMON USE OF THIS COMMAND IS TO CHANGE RUN-TIME FLAGS OR TO ADD/DROP UNITS. THESE FLAGS ARE DESCRIBED IN SECTION 4.8 - SWITCH OPTIONS. TYPING A "C" WILL FORCE THE SUPERVISOR TO ISSUE A PROMPT CHARACTER (">") SO THAT THE USER MAY USE THE RESTART COMMAND FOR THIS PURPOSE. NOTE IF THE USER USES THE START COMMAND AFTER A C, HE WILL HAVE TO REENTER ALL DEVICE INFORMATION.

4 3 CONTINUE
THE CONTINUE COMMAND WILL RESTART THE PROGRAM AT THE TEST WHICH WAS INTERRUPTED. UNLIKE THE RESTART COMMAND, THIS COMMAND DOES NOT FORCE FLAGS BACK TO THE INITIAL STATE. THE FLAGS WILL REMAIN AS SET UP DURING THE LAST START OR RESTART. THE USER HAS THE SAME DIALOGUE AND FLAG OPTIONS AS DESCRIBED FOR THE RESTART COMMAND.

4 4 ZERO FLAGS
THIS COMMAND WILL ZERO ALL FLAGS (SECTION 4.8) AND TYPE A PROMPT. SECTION 4.8 DESCRIBES THE EFFECTS OF CLEARING ALL FLAGS, BUT ESSENTIALLY, ALL LOOPING AND HALTS WILL BE DISABLED AND ALL ERROR INFORMATION WILL BE PRINTED IN THE EVENT OF AN ERROR.

4 5 ADD UNITS
WHILE TESTING MULTIPLE UNITS, THE USER MAY WISH TO DROP A PARTICULAR UNIT FROM THE TEST PROCEDURE AND, LATER ON, BEGIN TESTING THAT UNIT AGAIN. IN THIS CASE, THE USER NEED NOT STOP THE DIAGNOSTIC AND REBUILD THE PTABLE. AFTER A UNIT HAS BEEN DROPPED (SEE SECTION 4.6), THE USER MAY ADD THE UNIT BY TYPING "C" AND USING THE ADD COMMAND AS FOLLOWS:

ADD/UNI N

WHERE N IS A DECIMAL UNIT NUMBER THAT SPECIFIES THE UNIT TO BE ADDED. UNITS ARE NUMBERED FROM 1 TO 32 ACCORDING TO THE ORDER IN WHICH THEY WERE SPECIFIED FOR THE HARDWARE TABLE. MULTIPLE UNITS MAY BE ADDED BY TYPING EACH UNIT NUMBER SEPARATED BY COLONS (I.E., ADD/UNI 2-4-22) OR BY TYPING THE FIRST AND LAST UNIT NUMBERS OF A CONTIGUOUS GROUP OF UNITS SEPARATED BY A DASH (I.E., ADD/UNI 1-10 WILL ADD THE FIRST TEN UNITS). NOTE AFTER A START COMMAND, ALL UNITS ARE CONSIDERED ADDED. THERE IS NO NEED TO USE THE ADD COMMAND UNLESS THE UNIT WAS DROPPED.

4 6 DROP UNITS
TO REMOVE A UNIT FROM TESTING, THE DROP COMMAND (DRO) IS USED. TO USE THIS COMMAND, TYPE "C" AND DROP THE UNIT(S) USING THE DROP COMMAND AS FOLLOWS:

DRO/UNI N

WHERE "N" IS A DECIMAL UNIT NUMBER WHICH SPECIFIES THE UNIT TO BE DROPPED. UNITS ARE NUMBERED FROM 1 TO 32 ACCORDING TO THE ORDER IN WHICH THEY WERE SPECIFIED FOR THE HARDWARE TABLE. MULTIPLE UNITS MAY BE DROPPED BY TYPING EACH UNIT NUMBER SEPARATED BY COLONS (I.E., DRO/UNI 1-3-7) OR BY TYPING THE FIRST AND LAST UNIT NUMBERS OF A CONTIGUOUS GROUP SEPARATED BY A DASH (I.E., DRO/UNI 1-9 CAUSES THE FIRST

470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523

NINE UNITS TO BE DROPPED). AFTER THE PROGRAM IS RESTARTED, THE SPECIFIED UNIT(S) WILL NOT BE INCLUDED FOR TESTING

4 7 DISPLAY PTABLE
WHEN THE DISPLAY COMMAND IS TYPED, THE ENTIRE CONTENTS OF THE PTABLE WILL BE PRINTED. UNITS THAT HAVE BEEN DROPPED WILL BE SO MARKED THIS COMMAND IS USEFUL FOR VERIFYING THE PTABLE CONTENTS

4 8 SWITCH OPTIONS
THERE ARE THREE SWITCH OPTIONS AVAILABLE FOR USE WITH THE COMMANDS /TES, /PAS AND /FLA. THE TEST SWITCH, /TES, IS USED ONLY WITH THE START COMMAND AND THE RESTART COMMAND. IT IS USED TO SPECIFY THE INDIVIDUAL TEST(S) TO BE RUN IN CASES WHERE ALL TESTS ARE NOT TO BE RUN. FOR EXAMPLE, IF THE USER WISHED TO RUN THE LOOPBACK TEST ONLY, HE WOULD TYPE

>STA/TES 2

IF THIS SWITCH IS NOT USED, ALL TESTS WILL BE RUN AS THE DEFAULT CONDITION

THE PASS SWITCH IS USED TO SPECIFY THE NUMBER OF PASSES TO BE RUN TO RUN THE LOOPBACK TEST TWENTY TIMES, THE USER WOULD TYPE

>STA/TES 2/PAS 20

THIS SWITCH MAY BE USED WITH THE START, RESTART AND CONTINUE COMMANDS. THE DEFAULT CONDITION IS ESSENTIALLY AN INFINITE NUMBER OF PASSES

THE FLAG SWITCH IS USED TO SET FLAGS WHICH CONTROL PROGRAM OPERATION AND ERROR PRINTOUTS. THIS SWITCH MAY BE USED WITH THE CONTINUE COMMAND AS WELL AS THE START COMMAND AND RESTART COMMAND. THE AVAILABLE FLAGS ARE SHOWN IN THE TABLE BELOW

FLAG	EFFECT IF SET
LOE	LOOP ON ERROR
HLT	HALT ON ERROR
IER	INHIBIT ERROR REPORTS
IBE	INHIBIT BASIC ERROR REPORT*
IXE	INHIBIT EXTENDED ERROR REPORT*
PRI	SEND ALL TYPEOUTS TO THE PRINTER
BOE	BELL ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
PNT	PRINT TEST NUMBER

*SEE SECTION 5 0

EXAMPLE
IF THE USER WISHES TO RUN THE PROGRAM WITHOUT ERROR MESSAGES AND HAVE IT LOOP ON ERROR, HE WOULD TYPE
>RES/FLA LOE IER

EXAMPLE
IF THE USER LATER WANTED TO STOP THE LOOP ON ERROR FEATURE, HE WOULD TYPE

>CON/FLA LOE=0

NOTE THE ERROR REPORTS WOULD STILL BE INHIBITED

ALL FLAGS ARE DEFAULTED TO ZERO (NOT SET)

5 0

ERROR AND INFORMATIONAL MESSAGES

AS NOTED IN THE DESIGN NOTES (SECTION 1 3), A SINGLE ERROR MAY RESULT IN SEVERAL MESSAGES BEING TYPED. THIS IS DUE TO THE STRUCTURE OF THE PROGRAM. EACH SECTION, IN THE EVENT OF AN ERROR, REPORTS THE INFORMATION AVAILABLE TO IT. THE FINAL MESSAGE FOR EACH ERROR CONTAINS THE UNIT NUMBER (DECIMAL 1 1 TO 32 - SEE SECTIONS 4 5 AND 4 6) AND THE PHRASE ""END OF ERROR REPORT""

THERE ARE THREE LEVELS OF ERROR REPORTS. BY USING THE FLAG SWITCH (SECTION 4 8), ONE OR MORE LEVELS OF REPORT MAY BE INHIBITED. THE FIRST LEVEL CONSISTS OF AN ERROR NUMBER, PC CONTENTS AND ERROR TYPE (HARD, SOFT OR FATAL). THE ERROR NUMBER CAN BE RELATED TO THE TABLE AT THE END OF THIS SECTION. THE PC CONTENTS GIVE THE ADDRESS WHERE THE ERROR REPORT MECHANISM WAS INVOKED. THE NEXT LEVEL (BASIC ERROR REPORT) GIVES AN ENGLISH DESCRIPTION OF THE ERROR. AN EXAMPLE OF THIS TYPE OF MESSAGE IS "HEADER HAS CRC ERROR". THE FINAL LEVEL (EXTENDED ERROR REPORT) GIVES ADDITIONAL DATA CONCERNING THE ERROR, SUCH AS REGISTER CONTENTS

TO INHIBIT SOME OR ALL ERROR MESSAGES, USE THE FLAGS DESCRIBED IN SECTION 4 8. THE INHIBIT-EXTENDED-ERROR-REPORT FLAG (IXE) INHIBITS THAT LEVEL ONLY. THE INHIBIT-BASIC-ERROR-REPORT FLAG (IBE) INHIBITS BOTH BASIC AND EXTENDED ERROR REPORTS. FINALLY, THE INHIBIT-ERROR-REPORT FLAG (IER) INHIBITS ALL TYPEOUTS

TABLE OF ERROR NUMBERS

NO	ERROR DESCRIPTION
00	CANNOT INITIALIZE DZ
01	REPLY MESSAGE FROM TERM WAS NOT AN ENQ
02	REPLY FROM TERM WAS NOT EXPECTED TYPE OF CONTROL MESSAGE
03	TERM ADDRESS IN RESPONSE WAS NOT COORECT
04	RESPONSE HAD CRC ERROR IN THE HEADER
20	DZ TRANSMIT ERROR
21	DZ RECEIVE ERROR
22	MESSAGE TO TERMINAL WAS NOT ACK'ED
23	ERROR OCCURRED WHILE INITIALIZING DZ
24	ERROR OCCURRED DURING TRANSMISSION
25	DDCMP STRT MESSAGE DID NOT GET A STACK
26	ERROR DURING RECEPTION
31	TIMEOUT WHILE WAITING FOR RESPONSE
40	SEE 21
41	SEE 04
42	SEE 03
43	MESSAGE DATA HAD CRC ERROR
50	ERROR OCCURRED DURING SELFTEST (TEST #1)
51	ERROR OCCURRED DURING LOOPBACK (TEST #2)
52	ERROR IN RECEIVED DATA

580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633

70 SELFTEST RESPONSE IN ERROR (NOT RECOGNIZED)
71 TERMINAL MEMORY ERROR
72 TERMINAL CONTROL-ROM ERROR

6 0 RESTRICTIONS
THIS DIAGNOSTIC DOES NOT ATTEMPT TO TEST THE DZ11 IF A LARGE NUMBER OF
COMM ERRORS OCCUR, OR IF ATTEMPTS TO SEND OR RECEIVE DATA TO AND FROM THE
TERMINAL FAIL; USE THE SPECIFIC DZ11 DIAGNOSTICS TO VERIFY THE DZ11

7 0 MISCELLANEOUS
PLEASE REPORT ALL PROGRAM PROBLEMS VIA THE AIDS REPORTING SYSTEM REFER
ALL COMMENTS REGARDING DOCUMENTATION TO DIAGNOSTIC ENGINEERING TERMINAL
GROUP IN MAYNARD, MASS

8 0 TEST DESCRIPTIONS
8 1 MICRODIAGNOSTIC (TEST #1)
THE MICRODIAGNOSTIC, OR SELFTEST, IS ROM RESIDENT CODE IN THE VT62 THAT
IS EXECUTED UPON COMMAND FROM THE HOST COMPUTER OR DURING POWER UP A
FULL DESCRIPTION OF THE TEST IS AVAILABLE IN THE HARDWARE MANUAL FOR THE
VT62 ESSENTIALLY, THE TEST VERIFIES THE MICROPROCESSOR, CHECKS ROM
MEMORY, EXERCISES RAM MEMORY AND EXERCISES THE SCREEN. THE HOST MAY
REQUEST THE RESULTS OF THE TEST IN ANY CASE, THE RESULTS ARE ALWAYS
DISPLAYED ON THE LAST LINE OF THE SCREEN THE FIRST CHARACTERS ARE A
CODED ERROR REPORT FOR THE HOST COMPUTER AND THE REMAINDER OF THE LINE IS
AN ENGLISH DESCRIPTION OF THE ERROR STATUS

8 2 LOOPBACK (TEST #2)
IN THIS TEST, THE TERMINAL IS COMMANDED TO ENTER LOOPBACK MODE AND A
SINGLE, 256(DECIMAL) BYTE DATA MESSAGE IS SENT TO THE TERMINAL THIS
DATA MESSAGE WILL HAVE A DDCMP PROTOCOL ENVELOPE WHEN THE DATA IS
RETURNED, THE TERMINAL IS TAKEN OUT OF LOOPBACK MODE USING A DDCMP "STRT"
SEQUENCE THE CRC CHECKSUMS ARE VALIDATED FOR BOTH THE MESSAGE HEADER
AND THE MESSAGE DATA THE DATA IS THEN CHECKED, BYTE BY BYTE THUS
THERE ARE THREE POSSIBLE ERRORS DETECTED NO TERMINAL RESPONSE, BAD CRC
OR BAD DATA

INTERPRETATION OF ANY ERROR MESSAGES REQUIRES SOME EDUCATED GUESSWORK
BECAUSE OF THE NUMBER OF COMPONENTS IN THE COMMUNICATION LINK
INTERMITTENT ERRORS WOULD GENERALLY INDICATE LINE PROBLEMS UNLESS THE
LINE IS PHYSICALLY SHORT OR THE ENVIRONMENT IS RELATIVELY NOISE FREE IF
EITHER OF THESE CONDITIONS IS THE CASE OR IF THE ERROR IS NOT
INTERMITTENT, THEN THE TERMINAL I/O BOARD, THE MODEM AND THE DUP ARE
SUSPECT IF THERE IS NO TERMINAL RESPONSE, RUN THE TERMINAL SELF TEST
OFF-LINE THIS WILL SHOW WHETHER THE TERMINAL IS COMPLETELY "DEAD" OR
THE PORTIONS OF THE COMMUNICATION LINK MENTIONED ABOVE ARE BROKEN

THERE ARE TWO DATA PATTERNS AVAILABLE FOR LOOPBACK ALL ZERO'S AND
ALTERNATING 1/0 BITS (I E , BYTES OF 252,125 252,125,) THE ZERO

635
636
637
638
639
640

PATTERN IS NOISE-SUSCEPTIBLE HOWEVER THE ALTERNATING BITS PATTERN
EXERCISES THE DATA PATHS BETTER SECTION 4 D SPECIFIES HOW TO SELECT THE
PATTERN

642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686

9 0

INDEX

ACK	5, 12
ADD COMMAND	10
BAUD RATE	7
CONTINUE COMMAND	10-11
CRC	4-5, 12
DDCMP	4-5
DISPLAY COMMAND	11
DROP COMMAND	10
END	5, 12
ERROR NUMBER	12
ERROR REPORT	12
FLAGS	6, 9-12
LOOPBACK	8, 12-13
MULTIPLE UNITS	10
MULTIPLE UUT'S	7
NAK	5
PTABLE	4, 10-11
RESTART COMMAND	9-11
STACK	5, 12
START COMMAND	6, 10-11
STOP CODE	7
STRT	5, 12
SWITCH OPTIONS	10
SWITCH OPTIONS	11
SWITCHES	6
UNIT NUMBER	10, 12

ENDR

5789	002000	=2000
5790		
5791		HEADER - VT62 ACCEPTANCE TEST
5792		
5805	002000	104 ASCII 202
(5)	002001	132 ASCII 222
(5)	002002	104 ASCII 202
(5)	002003	132 ASCII 222
(5)	002004	103 ASCII 202
(6)	002005	000 BYTE 0
(6)	002006	000 BYTE 0
(5)	002007	000 BYTE 0
(4)	002010	101 ASCII 2A2
(4)	002011	060 ASCII 202
(4)	002012	001 BYTE CSREVISION
(4)	002013	006 BYTE CSEDIT
(4)	002014	000000 WORD 0
(4)	002016	000000 WORD 0
(4)	002020	000000 WORD 0
(4)	002022	000000 WORD 0
(4)	002024	000000 WORD 0
(5)	002026	000000 WORD 0
(4)	002030	000000 WORD 0
(4)	002032	000000 WORD 0
(4)	002034	000000 WORD 0
(4)	002036	000000 WORD 0
(4)	002040	011462 WORD LDISPATCH
(4)	002042	011474 WORD LSINIT
(4)	002044	012066 WORD LSCLEAN
(4)	002046	021422 WORD LSHARD
(4)	002050	021722 WORD LSSOFT
(4)	002052	002222 WORD LSDUTYP
(4)	002054	011466 WORD LSRPT
(4)	002056	011442 WORD LSHW
(4)	002060	011454 WORD LSSH
(4)	002062	002236 WORD LSDR
(4)	002064	002242 WORD LSDPST
(4)	002066	000000 WORD 0
(4)	002070	011470 WORD LSAU
(4)	002072	011472 WORD LSDU
(4)	002074	000000 WORD 0
(4)	002076	022054 WORD LSLAST


```

5825      000004      EFO4== 4
5826      000003      EFO3== 3
5827      000002      EFO2== 2
5828      000001      EFO1== 1
5829
5830      /
5831      / PRIORITY LEVEL DEFINITIONS
5832      /
5833      000340      PRI07== 340
5834      000300      PRI06== 300
5835      000240      PRI05== 240
5836      000200      PRI04== 200
5837      000140      PRI03== 140
5838      000100      PRI02== 100
5839      000040      PRI01== 40
5840      000000      PRI00== 0
5841
5842      /
5843      / VT62 DIAGNOSTIC EQUATES
5844      /
5845      000201      SOH=201      / START OF HEADER      (DOCMP)
5846      000001      ACK=1      / ACKNOWLEDGE      "
5847      000002      NAK=2      / NEGATIVE ACKNOWLEDGE "
5848      000003      REP=3      / REPLY      "
5849      000005      ENQ=5      / ENQUIRY      "
5850      000006      STRT=6      / START      "
5851      000007      STACK=7      / START ACKNOWLEDGE "
5852      000377      SYNC=377      / SYNCHRONIZATION CHARACTER
5853      000220      DLE=220      / DATA LINK ESCAPE
5854      000024      OSOP=24      / LOOPBACK
5855      000200      PARITY=200      / PARITY BIT
5856      000002      TSTS=2      / NO OF TESTS
5857      000000      RCVCSR=0      / OFFSETS RECEIVER CSR
5858      000002      RCVBUF=2      / RECEIVER BUFFER REG
5859      000004      XMTCSR=4      / TRANSMITTER CSR
5860      000006      XMTBUF=6      / TRANSMITTER BUFFER REG.
5861
5862      /
5863      / HARDWARE REGISTER BIT DEFINITIONS
5864      /
5865      000020      MASCLR=20      / MASTER CLEAR
5866      000040      MSCNEN=40      / MASTER SCAN ENABLE
5867      000100      RXINTE=100      / RECEIVE INTERRUPT ENABLE
5868      040000      TXINTE=40000      / TRANSMIT INTERRUPT ENABLE
5869      100000      INV DAT=100000      / INVALID DATA
5870      010000      RENABL=10000      / RECEIVE ENABLE
5871      000017      LINEMK=17      / LINE BITS MASK
5872      177400      HYBYTE=177400      / MASK FOR HIGH BYTE OF A WORD
5873      000377      LOBYTE=377      / MASK FOR LOW BYTE OF A WORD

```

```

5964 ,*****
5965 ,      GLOBAL DATA      *
5966 ,*****
5967 002100 000000 R5STCK 0 ;FLAG TO INDICATE IF R5 STACK IS SET UP
5968 002102 000000 SCRATCH 0 ;SCRATCH LOCATIONS
5969 002104 000000 TEMP 0
5970 002106 000000 ABORT 0 ;TEST ABORT FLAG
5971 002110 020000 TIMLIM 20000 ;TIMEOUT LIMIT (IN 100 MICROSEC INCREMENTS)
5972 002112 000000 STIME 0 ;SHORT TIMEOUT LIMIT (APPROX 20 MICROSEC INCREM )
5973 002114 000010 ILIMIT 10 ;LIMIT ON DEVICE INITIALIZATION ATTEMPTS
5974 002116 000000 INERR 0 ;INPUT ERROR
5975 002120 000000 OUTERR 0 ;OUTPUT ERROR
5976 002122 000001 PASCNT 1
5977 002124 000000 EOMSG 0 ;END_OF_MESSAGES FLAG
5978 002126 000000 LOGDEV 0 ;LOGICAL DEVICE NUMBER
5982 002130 000000 EXPINP 0 ;EXPECT_INPUT FLAG
5983 002132 000000 OUTMSG 0 ;OUTPUT MESSAGE POINTER AND SIZE
5984 002134 000000 0
5985 002136 000000 INMSG 0 ;INPUT MESSAGE POINTER AND SIZE
5986 002140 000000 0
5987 002142 000000 OUTBUF 0 ;OUTPUT BUFFER POINTER AND SIZE
5988 002144 000000 OUTSIZ 0
5989 002146 000000 NBUF 0 ;INPUT BUFFER POINTER AND SIZE
5990 002150 000000 INSIZ 0
5991
5992 002152 MSGAP
5993 002152 000000 OUT 0 ;TEST MESSAGE AREA POINTER
5994 002154 000000 INITMA 0 ;TEMPORARY MESSAGE POINTER AND SIZE
5995 002156 000000 INITMS 0 ;USED FOR INITIALIZATION OF OUTPUT ROUTINES
5996 002160 000000 BAPNT 0 ;GENERAL BUFFER AREA POINTER AND SIZE
5997 002162 000000 BASIZE 0
5998 002164 000000 MSGNO 0 ;MESSAGE NUMBER (DDCMP)
5999 002166 000000 RMSGNO 0
6006 002170 000000 PTABLE 0 ;PRESENT UNIT PTABLE DATA
6007 002172 000000 0
6008 002174 000000 0
6009 002176 000000 0
6013 002200 PTEND
6014 002200 000000 DEVPRI 0 ;DEVICE PRIORITY
6016 002202 LINMSK 0 ;MASKS FOR LINE SELECTION BITS (DZ11)
6017 002202 000001 1 ;LINE 0
6018 002204 000002 2 ;LINE 1
6019 002206 000004 4 ;LINE 2
6020 002210 000010 10 ;LINE 3
6021 002212 000020 20 ;LINE 4
6022 002214 000040 40 ;LINE 5
6023 002216 000100 100 ;LINE 6
6024 002220 000200 200 ;LINE 7
6026
6032 002222 052126 031066 042054 ASCII OUT62,DZ110
(3) 002230 030532 000061
(2)
6033 002234 000001 EVEN
(6) 002236 000017 WORD 17
(4) 002240 000004 WORD 4
(2) 002242 000004 BLKH 4
  
```

```

6040 ,*****
6041 ; GLOBAL TEXT *
6042 ,*****
6043
6044 002252 040503 020116 047516 MSG00 ASCIZ /CAN NOT INITIALIZE COMM DEVICE/
      002260 020124 047111 052111
      002266 040511 044514 042532
      002274 041440 046517 020115
      002302 042504 044526 042503
      002310 000
6045 002311 122 050105 054514 MSG01 ASCIZ /REPLY WAS NOT AN ENG/
      002316 053440 051501 047040
      002324 052117 040440 020116
      002332 047105 000121
6046 002336 047111 047503 051122 MSG02 ASCIZ /INCORRECT REPLY TYPE/
      002344 041507 020124 042522
      002352 0461 J 020131 054524
      002360 042520 000
6047 002363
6048 002363 122 050105 054514 MSG42
      002370 044040 051501 044440 MSG03 ASCIZ /REPLY HAS INCORRECT STATION ADDRESS/
      002376 041516 051117 042522
      002404 052103 051440 040524
      002412 044524 047117 040440
      002420 042104 042522 051523
      002426 000
6049 002427
6050 002427 110 040505 042504 MSG41
      002434 020122 040510 020123 MSG04 ASCIZ /HEADER HAS CRC ERROR/
      002442 051103 020103 051105
      002450 047522 000122
6051 002454 051105 047522 020122 MSG10 ASCIZ /ERROR DURING KEY OAPD TEST ** END OF REPORT **/
      002462 052504 044522 043516
      002470 045440 054505 047502
      002476 051101 020104 042524
      002504 052123 020040 025052
      002512 042440 042116 047440
      002520 020106 042522 047520
      002526 052122 025040 000052
6052 002534 040502 020104 042522 MSG11 ASCIZ /BAD RESPONSE FROM KEYBOARD/
      002542 050123 047117 042523
      002550 043040 047522 020115
      002556 042513 041131 040517
      002564 042122 000
6053 002567 124 047517 046440 MSG12 ASCIZ /TOO MANY CHARACTERS SENT FROM KEYBOARD/
      002574 047101 020131 044103
      002602 051101 041501 042524
      002610 051522 051440 047105
      002616 020124 051106 046517
      002624 045440 054505 047502
      002632 051101 000104
6054 002636 051124 047101 046523 MSG20 ASCIZ /TRANSMIT ERROR/
      002644 052111 042440 051122
      002652 051117 000
6055 002655
6056 002655 122 041505 050105 MSG40
      MSG21 ASCIZ /RECEPTION ERROR
  
```

	002662	044524	047117	042440		
	002670	051122	051117	000		
6057	002675	116	020117	041501	MSG22	ASCIZ /NO ACK RECEIVED AFTER TRANSMISSION/
	002702	020113	042522	042503		
	002710	053111	042105	040440		
	002716	052106	051105	052040		
	002724	040522	051516	044515		
	002732	051523	047511	000116		
6058	002740	051105	047522	020127	MSG23	ASCIZ /ERROR WHILE INITIALIZING DEVICE/
	002746	044127	046111	020105		
	002754	047111	052111	040511		
	002762	044514	044532	043516		
	002770	042040	053105	041511		
	002776	000105				
6059	003000	51105	047522	020122	MSG24	ASCIZ /ERROR OCCURRED DURING TRANSMISSION/
	003006	01517	052503	051122		
	003014	042105	042040	051125		
	003022	047111	020107	051124		
	003030	047101	046523	051511		
	003036	044523	047117	000		
6060	003043	116	020117	052123	MSG25	ASCIZ /NO STACK IN RESPONSE TO STRT/
	003050	041501	020113	047111		
	003056	051040	051505	047520		
	003064	051516	020105	047524		
	003072	051440	051124	000124		
6061	003100	051105	047522	020122	MSG26	ASCIZ /ERROR WHILE RECEIVING BUFFER/
	003106	044127	046111	020105		
	003114	042522	042503	053111		
	003122	047111	020107	052502		
	003130	043106	051105	000		
6062	003135	103	047101	052047	MSG20	ASCIZ /CAN'T GET CLEAR TO SEND.
	003142	043440	052105	041440		
	003150	042514	051101	052040		
	003156	020117	042523	042116		
	003164	000				
6063	003165	124	046511	020105	MSG31	ASCIZ /TIME OUT WHILE WAITING FOR TERMINAL RESPONSE/
	003172	052517	020124	044127		
	003200	046111	020105	040527		
	003206	052111	047111	020107		
	003214	047506	020122	042524		
	003222	046522	047111	046101		
	003230	051040	051505	047520		
	003236	051516	000105			
6064	003242	040504	040524	041440	MSG43	ASCIZ /DATA CRC ERROR
	003250	041522	042440	051122		
	003256	051117	000			
6065	003261	116	044517	042523	MSG45	ASCIZ /NOISE PATTERN SUBTEST/
	003266	050040	052101	042524		
	003274	047122	051440	041125		
	003302	042524	052123	000		
6066	003307	105	051122	051117	MSG50	ASCIZ /ERROR DURING MICRODIAGNOSTIC/
	003314	042040	051125	047111		
	003322	020107	044515	051103		
	003330	042117	040511	047107		
	003336	051517	044524	000103		
6067	003344	051105	047522	020122	MSG51	ASCIZ /ERROR DURING LOOPBACK/

	003352	052504	044522	043516		
	003360	046040	047517	041120		
	003366	041501	000113			
6068	003372	040504	040524	042440	MSG52	ASCIZ /DATA ERROR/
	003400	051122	051117	000		
6069	003405	111	046114	043505	MSG70	ASCIZ /ILLEGAL SELF TEST RESPONSE/
	003412	046101	051440	046105		
	003420	020106	042524	052123		
	003426	051040	051505	047520		
	003434	051516	000105			
6070	003440	042515	047515	054522	MSG71	ASCII /MEMORY ERROR TYPE /
	003446	042440	051122	051117		
	003454	020040	052040	050131		
	003462	020105				
6071	003464	000	000		MTY	BYTE 0,0
6072	003466	020115	020055	040502		ASCII /M - BANK #/
	003474	045516	021440			
6073	003500	000	000		BNO	BYTE 0,0
6074						EVEN
6075	003502	051103	046517	042440	MSG72	ASCIZ /CPM ERROR/
	003510	051122	051117	000		
6076		003516				EVEN
6077						
6078	003516	005			STRTB	BYTE ENQ
6079	003517	006				BYTE STRT
6080	003520	300				BYTE 300
6081	003521	000				BYTE 0
6082	003522	000				BYTE 0
6083	003523	000			STRTSA	BYTE 0
6084	003524	000000			STRTC	WORD 0
6085						
6086						*****
6087						GLOBAL ERROR REPORT
6088						*****
6090	003526	016746	176362			MOV ILIMIT, -(SP)
(7)	003532	012746	003554			MOV #FORM00, -(SP)
(6)	003536	012746	000002			MOV #2, -(SP)
(3)	003542	010600				MOV SP, R0
(4)	003544	104015				EMT C\$PNTX
(4)	003546	062706	000006			ADD #6, SP
6091	003552				L10000	
(3)	003552	104023				EMT C\$MSG
6092	003554	040445	043101	042524	FORM00	ASCIZ /%AFTER %06%A ATTEMPTS/N
	003562	020122	047445	022466		
	003570	020101	052101	042524		
	003576	050115	051524	047045		
	003604	000				
6093		003606				EVEN
6094						
6096	003606				L10001	
(3)	003606	104023				EMT C\$MSG
6097						
6099	003610	016746	000030			MOV A02B, -(SP)
(8)	003614	016746	000022			MOV A02A, -(SP)
(7)	003620	012746	003646			MOV #FORM02, -(SP)
(6)	003624	012746	000003			MOV #3, -(SP)

(3)	003630	010600			MOV	SP, RO
(4)	003632	104015			EMT	C\$PNTX
(4)	003634	062706	000010		ADD	#10, SP
6100	003640			L10002		
(3)	003640	104023			EMT	C\$MSG
6101	003642	000000		A02A	0	
6102	003644	000000		A02B	0	
6103	003646	040445	042522	042503	FORM02	ASCIZ /%ARECEIVED %03%A -- EXPECTED %03%N/
	003654	053111	042105	022440		
	003662	031517	040445	026440		
	003670	020055	054105	042520		
	003676	052103	042105	022440		
	003704	031517	047045	000		
6104		003712				EVEN
6105						
6107	003712			AREA42		
6108	003712	016746	176164		MOV	SCPTCH, -(SP)
(8)	003716	016746	176162		MOV	TEMP, -(SP)
(7)	003722	012746	003646		MOV	#FORM02, -(SP)
(6)	003726	012746	000C03		MOV	#3, -(SP)
(3)	003732	010600			MOV	SP, RO
(4)	003734	104015			EMT	C\$PNTX
(4)	003736	062706	000010		ADD	#10, SP
6109	003742			L10003		
(3)	003742	104023			EMT	C\$MSG
6110						
6112	003744			AREA41		
6113	003744			AREA43		
6114	003744	016746	000030		MOV	A04EX, -(SP)
(8)	003750	016746	000022		MOV	A04RC, -(SP)
(7)	003754	012746	004002		MOV	#FORM04, -(SP)
(6)	003760	012746	000003		MOV	#3, -(SP)
(3)	003764	010600			MOV	SP, RO
(4)	003766	104015			EMT	C\$PNTX
(4)	003770	062706	000010		ADD	#10, SP
6115	003774			L10004		
(3)	003774	104023			EMT	C\$MSG
6116	003776	000000		A04RC	0	
6117	004000	000000		A04EX	0	
6118	004002	040445	042522	042503	FORM04	ASCIZ /%ARECEIVED %06%A -- EXPECTED %06%N/
	004010	053111	042105	022440		
	004016	033117	040445	026440		
	004024	020055	054105	042520		
	004032	052103	042105	022440		
	004040	033117	047045	000		
6119		004046				EVEN
6120						
6122	004046	016746	000030		MOV	KEX, -(SP)
(8)	004052	016746	000022		MOV	KRC, -(SP)
(7)	004056	012746	004002		MOV	#FORM04, -(SP)
(6)	004062	012746	000003		MOV	#3, -(SP)
(3)	004066	010600			MOV	SP, RO
(4)	004070	104015			EMT	C\$PNTX
(4)	004072	062706	000010		ADD	#10, SP
6123	004076			L10005		
(3)	004076	104023			EMT	C\$MSG

6124	004100	000000			KRC	0		
6125	004102	000000			KEX	0		
6127	004104				L10006			
(3)	004104	104023			EMT		C\$MSG	
6129	004106				L10007			
(3)	004106	104023			EMT		C\$MSG	
6131	004110				AREA24			
6132	004110				AREA40			
6133	004110	016767	176012	000064	MOV		LOGDEV, TEMP2	
6134	004116	005767	000060		TST		TEMP2	
(9)	004122	002002			BGE		500005	
6135	004124	005067	000052		CLR		TEMP2	
6136	004130				500005			
6137	004130	005267	000046		INC		TEMP2	
6138	004134	016746	000042		MOV		TEMP2, -(SP)	
(7)	004140	012746	004204		MOV		#FORM23, -(SP)	
(6)	004144	012746	000002		MOV		#2, -(SP)	
(3)	004150	010600			MOV		SP, R0	
(4)	004152	104015			EMT		C\$PNTX	
(4)	004154	062706	000006		ADD		#6, SP	
6139	004160	012746	004224		MOV		#FORM40, -(SP)	
(6)	004164	012746	000001		MOV		#1, -(SP)	
(3)	004170	010600			MOV		SP, R0	
(4)	004172	104015			EMT		C\$PNTX	
(4)	004174	062706	000004		ADD		#4, SP	
6140	004200				L10010			
(3)	004200	104023			EMT		C\$MSG	
6141	004202	000000			TEMP2	0		
6142	004204	040445	042504	044526	FORM23	ASCIZ	/%ADEVICE %D3%N	
	004212	042503	020072	042045				
	004220	022463	000116					
6143	004224	040445	020040	020040	FORM40	ASCIZ	RA ***** END OF ERROR REPORT *****N/	
	004232	020040	020040	025040				
	004240	025052	025052	042440				
	004246	042116	047440	020106				
	004254	051105	047522	020122				
	004262	042522	047520	052122				
	004270	025040	025052	025052				
	004276	047045	000					
6144		004302						
6145					EVEN			
6147	004302				L10011			
(3)	004302	104023			EMT		C\$MSG	
6148								
6168	004304				AREA31			
6169	004304	010146			MOV		R1, -(SP)	
6170	004306	012701	002242		MOV		#DZREG, R1	
6171	004312	016146	000000		MOV		RCVCSR(R1), -(SP)	
(7)	004316	012746	004436		MOV		#FORM30, -(SP)	
(6)	004322	012746	000002		MOV		#2, -(SP)	
(3)	004326	010600			MOV		SP, R0	
(4)	004330	104015			EMT		C\$PNTX	
(4)	004332	062706	000006		ADD		#6, SP	
6172	004336	016146	000002		MOV		RCVBUF(R1), -(SP)	
(7)	004342	012746	004457		MOV		#FRM30A, -(SP)	
(6)	004346	012746	000002		MOV		#2, -(SP)	

(3)	004352	010600				MOV	SP, R0
(4)	004354	104015				EMT	C\$PNTX
(4)	004356	062706	000006			ADD	#6, SP
6173	004362	016146	000004			MOV	XMTCSR(R1), -(SP)
(7)	004366	012746	004501			MOV	#FRM30B, -(SP)
(6)	004372	012746	000002			MOV	#2, -(SP)
(3)	004376	010600				MOV	SP, R0
(4)	004400	104015				EMT	C\$PNTX
(4)	004402	062706	000006			ADD	#6, SP
6174	004406	016146	000006			MOV	XMTBUF(R1), -(SP)
(7)	004412	012746	004522			MOV	#FRM30C, -(SP)
(6)	004416	012746	000002			MOV	#2, -(SP)
(3)	004422	010600				MOV	SP, R0
(4)	004424	104015				EMT	C\$PNTX
(4)	004426	062706	000006			ADD	#6, SP
6175	004432	012601				MOV	(SP)+, R1
6176	004434				L10012		
(3)	004434	104023				EMT	C\$MSG
6177	004436	040445	055104	051040	FORM30	ASCIZ	/%ADZ RCSR %06%N/
	004444	051503	035122	022440			
	004452	033117	047045	000			
6178	004457	045	042101	020132	FRM30A	ASCIZ	/%ADZ RBUFF %06%N/
	004464	041122	043125	035106			
	004472	022440	033117	047045			
	004500	000					
6179	004501	045	042101	020132	FRM30B	ASCIZ	/%ADZ TCSP: %06%N/
	004506	041524	051123	020072			
	004514	047445	022466	000116			
6180	004522	040445	055104	052040	FRM30C	ASCIZ	/%ADZ TBUF: %06%N/
	004530	052502	043106	020072			
	004536	047445	022466	000116			
6181						EVEN	
6204							
6206	004544	012746	004566			MOV	#FORM60, -(SP)
(6)	004550	012746	000001			MOV	#1, -(SP)
(3)	004554	010600				MOV	SP, R0
(4)	004556	104015				EMT	C\$PNTX
(4)	004560	062706	000004			ADD	#4, SP
6207	004564				L10013		
(3)	004564	104023				EMT	C\$MSG
6208	004566	040445	040503	023516	FORM60	ASCIZ	/%ACAN'T GET INTO LOOPBACK%.N/
	004574	020124	042507	020124			
	004602	047111	047524	046040			
	004610	047517	041120	041501			
	004616	022513	000116				
6209						EVEN	
6211	004622	012746	004714			MOV	#FORM61, -(SP)
(6)	004626	012746	000001			MOV	#1, -(SP)
(3)	004632	010600				MOV	SP, R0
(4)	004634	104015				EMT	C\$PNTX
(4)	004636	062706	000004			ADD	#4, SP
6212	004642	016767	175260	177332		MOV	LOGDEV, TEMP2
6213	004650	005767	177326			TST	TEMP2
(9)	004654	002002				BGE	500015
6214	004656	005067	177320			CLR	TEMP2
6215	004662				500015		

6216	004662	005267	177314			INC	TEMP2
6217	004666	016746	177310			MOV	TEMP2, -(SP)
(7)	004672	012746	004204			MOV	#FORM23, -(SP)
(6)	004676	012746	000002			MOV	#2, -(SP)
(3)	004702	010600				MOV	SP, R0
(4)	004704	104015				EMT	C\$PNTX
(4)	004706	062706	000006			ADD	#6, SP
6218	004712				L10014		
(3)	004712	104023				EMT	C\$MSG
6219	004714	040445	040503	023516	FORM61	ASCIZ	/%ACAN'T GET OUT OF LOOPBACK%N/
	004722	020124	042507	020124			
	004730	052517	020124	043117			
	004736	046040	047517	041120			
	004744	041501	022513	000116			
6220						EVEN	
6222	004752	016746	000074			MOV	A52B, -(SP)
(8)	004756	016746	000066			MOV	A52A, -(SP)
(7)	004762	012746	005004			MOV	#FORM52, -(SP)
(6)	004766	012746	000003			MOV	#3, -(SP)
(3)	004772	010600				MOV	SP, R0
(4)	004774	104015				EMT	C\$PNTX
(4)	004776	062706	000010			ADD	#10, SP
6223	005002				L10015		
(3)	005002	104023				EMT	C\$MSG
6224	005004	040445	054105	042520	FORM52	ASCIZ	/%AEXPECTED. %03%A -RECEIVED: %03%N/
	005012	052103	042105	020072			
	005020	047445	022463	020101			
	005026	026440	042522	042503			
	005034	053111	042105	020072			
	005042	047445	022463	000116			
6225						EVEN	
6226	005050	000000			A52A	0	
6227	005052	000000			A52B	0	
6228							
6230	005054				L10016		
(3)	005054	104023				EMT	C\$MSG
6231							
6233	005056				L10017		
(3)	005056	104023				EMT	C\$MSG
6234							
6235	005060				TEMPBUF		
6236	005060	001452				BLKW	810
6237	010204				TMPBUF		
6238	010204	000416				BLKW	270
6239	011240	000100				BLKW	100
6240	011440				SSTACK		
6241							

STACK FOR SUBROUTINE LINKAGE

```
6250 ,*****
6251 ,      DEFAULT HARDWARE P-TABLE AND SOFTWARE P-TABLE      *
6252 ,*****
6253 011440 000004      WORD  L10020-L$HW/2
6254 011442 160010      160010
6255 011444 000320      320
6256 011446 000140      140
6257 011450 000000      0
6261 011452 L10020      WORD  L10021-L$SW/2
6262 011452 000002      000010
6263 011454 000010      PATTERN 000001
6264 011456 000001      L10021
6268 011460
6269
6270 ,*****
6271 ,      DISPATCH TABLE      *
6272 ,*****
6273 011460 000002      WORD  TSTS
    (6) 011462 017126      WORD  T1
    (6) 011464 020130      WORD  T2
```

```

6464
6466 ;*****
6467 ;STUB AREAS FOR SUPERVISOR COMPATIBILITY *
6468 ;*****
6470 011466 L10022
(3) 011466 104025 EMT CSRPT
6471
6473 011470 L10023
(3) 011470 104054 EMT CSAU
6474
6476 011472 L10024
(3) 011472 104055 EMT CSOU
6478
6479 ;*****
6480 ; INITIALIZE CODING *
6481 ;*****
6483 ;
6484 ;SAVE R1
6485 ;
6486 011474 010146 MOV R1, -(SP)
6487 ;
6488 ;CHECK START EVENT FLAG TO SEE IF WE GOT HERE BECAUSE OF A START
6489 ;
6490 011476 012700 000040 MOV #EF START, R0
(3) 011502 104050 EMT CSREFG
6491 ;
6492 ; IF A START, THEN.
6493 ;
6494 011504 103005 BCC 500025
6522 ;
6523 ; INITIALIZE THE SUBROUTINE LINKAGE STACK
6524 ;
6525 011506 012705 011440 MOV #SSTACK, R5
6526 011512 005267 170362 INC R5STCK
6527 011516 000414 BR 500035
(3) 011520 500025
6528 ;
6529 ; IF NOT START THEN
6530 ;
6531 ;CHECK FOR RESTART AND IF IT IS
6532 ;UPDATE LOGICAL DEVICE AND STAT TABLE ENTRY POINTER
6533 ;BUT DON'T EXCEED THE NUMBER OF UNITS SPECIFIED
6534 ;
6535 011520 012700 000037 MOV #EF RESTART, R0
(3) 011524 104050 EMT CSREFG
6536 011526 103410 BCS 500045
6537 011530 005267 170372 INC LOGDEV
6541 011534 026767 170366 170252 CMP LOGDEV, LSUNIT
(9) 011542 001002 BNE 500055
6542 011544 005067 170356 CLR LOGDEV
6546 011550 500055
6547 011550 500045
6548 011550 500035
6549 ;
6550 ;CHECK FOR NEW PASS IF IT IS, FORCE LOGICAL UNIT TO ZERO
6551 ;

```

6552	011550	012700	000035		MOV	#EF NEW, R0
(3)	011554	104050			EMT	C\$REFG
6553	011556	103002			BCC	50006\$
6554	011560	005067	170342		CLR	LOGDEV
6558	011564			50006\$		
6559						
6560						
6561						
6562						
6563	011564	005067	003604		CLR	INIT
6564	011570	005067	170312		CLR	ABORT
6565						
6566						
6567						
6568	011574	012767	015420	003614	MOV	#RQUET, RQUE
6569	011602	016767	003610	003604	MOV	RQUE, RQUEUE
6570	011610	012767	015436	003616	MOV	#RBQUET, RBQUE
6571	011616	016701	003612		MOV	RBQUE, R1
6572	011622	012767	005060	170252	MOV	#TEMBUF, SCRTCH
6573	011630				50007\$	
6574	011630	016721	170246		MOV	SCRTCH, (R1)+
6575	011634	062767	000416	170240	ADD	#270, SCRTCH
6576	011642	020127	015442		CMP	R1, #RBQUEB
(6)	011646	003770			BLE	50007\$
6593						
6594						
6595						
6596	011650				50010\$	
6597	011650	005201			INC	R1
6598	011652	016700	170250		MOV	LOGDEV, R0
(3)	011656	104042			EMT	C\$GPHRD
(3)	011660	010067	170216		MOV	R0, SCRTCH
6599						
6600						
6601						
6602	011664	103411			BCS	50011\$
6603	011666	005267	170234		INC	LOGDEV
6607	011672	026767	170230	170114	CMP	LOGDEV, L\$UNIT
(9)	011700	001002			BNE	50012\$
6608	011702	005067	170220		CLR	LOGDEV
6612	011706				50012\$	
6613	011706	005001			CLR	R1
6614	011710				50011\$	
6615	011710	005701			TST	R1
(6)	011712	001756			BEQ	50010\$
6628						
6629						
6630						
6631						
6632	011714	012701	002170		MOV	#PTABLE, R1
6633	011720				50013\$	
6634	011720	017711	170156		MOV	@SCRTCH, (R1)
6635	011724	062701	000002		ADD	#2, R1
6636	011730	062767	000002	170144	ADD	#2, SCRTCH
6637	011736	020127	002200		CMP	R1, #PTEND
(6)	011742	001366			BNE	50013\$

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 186-2
 DZDZCA P11 21-NOV-77 14 55

SEQ 0030

```

6639
6640      GET THE LINE NUMBER AND ISOLATE IT FOR FUTURE USE
6641
6642      011744 012701 002170      MOV      #PTABLE,R1
        (6) 011750 062701 000006      ADD      #6,R1
6643      011754 011101      MOV      (R1),R1
6644      011756 042701 177770      BIC      #177770,R1
6645      011762 006301      ASL      R1
6646      011764 016167 002202 003366      MOV      LINMSK(R1),$LINE
6656
6657      RESTORE R1 AND EXIT INIT CODE
6658
6659      011772 012601      MOV      (SP)+,R1
6660      011774      L10025
        (3) 011774 104011      EMT      C$INIT
6665      011776 047045 040445 052126      VMSG      ASCII /%N:AVT62 ACCEPTANCE TEST  --  VERSION 10 TERMINALS%N/
        012004 031066 040440 041503
        012012 050105 040524 041516
        012020 020105 042524 052123
        012026 020040 026440 020055
        012034 020040 042526 051522
        012042 047511 020116 030061
        012050 052040 051105 044515
        012056 040516 051514 047045
        012064      000
6666      012066      EVEN
6667
6668      *****
6669      CLEAN-UP CODING
6670      *****
6672
6673      MAKE SURE DEVICE IS SHUT DOWN
6674
6682      012066 010146      MOV      R1,-(SP)
6683      012070 016701 170074      MOV      #PTABLE,R1
6684      012074 005061 000004      CLR      XMTCSR(R1)
6685      012100 052761 000020 000000      BIS      #MASCLR,RC(CSP(P1))
6686      012106 012601      MOV      (SP)+,R1
6688
6689      UPDATE PASS COUNT
6690
6691      012110 005267 170006      INC      PASCNT
6706
6707      EXIT CLEAN-UP CODE
6708
6709      012114      L10026
        (3) 012114 104012      EMT      C$CLEAN

```

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 187
 DZDZCA P11 21-NOV-77 14 55

SEQ 0031

```

6713  , *****
6714  , GLOBAL SUBROUTINES *
6715  , *****
6716  ,
6717  , ++++++
6718  , ROUTINE TO PROCESS TERMINAL MESSAGES FOR ALL TESTS +
6719  , RETURNS POINTER TO MESSAGE ("OUTMSG") AND ITS SIZE +
6720  , IN BYTES ("OUTSIZ") +
6721  , +
6722  , SIZE MAY BE ANY NUMBER OF BYTES +
6723  , +
6724  , SETS A FLAG ("EXPINP") IF RESPONSE OTHER THAN "ACK" IS +
6725  , TO BE EXPECTED +
6726  , +
6727  , SETS ANOTHER FLAG ("EOMSG") IF THIS IS THE LAST MESSAGE +
6728  , ++++++
6729  MSG
6730
6733  012116 010446      MOV      R4, -(SP)
        (2) 012120 010604      MOV      SP, R4
        (3) 012122 162706 000002  SUB      #1, SP
6734
6735  , GET MESSAGE POINTER
6736
6737  012126 017764 170020 177776      MOV      @MSGAP, A(R4)
6738
6739  , IF MESSAGE POINTER IS ODD,
6740  , SET END_OF_MESSAGE INDICATOR
6741
6742  012134 016467 177776 167740      MOV      A(R4), SCRTCH
        (6) 012142 016746 167734      MOV      SCRTCH, -(SP)
        (6) 012146 042716 000001      BIC      #1, (SP)
        (6) 012152 042667 167724      BIC      (SP)+, SCRTCH
6743  012156 005767 167720      TST      SCRTCH
        (9) 012162 001405      BEQ      500025
6744  012164 005267 167734      INC      EOMSG
6745  012170 042764 000001 177776      BIC      #1, A(R4)
6746  012176      500025
6747
6748  , IF MESSAGE_POINTER IS NEGATIVE
6749  , SET EXPECT_INPUT INDICATOR
6750
6751  012176 005764 177776      TST      A(R4)
        (9) 012202 002006      BGE      500035
6752  012204 005267 167720      INC      EXPINP
6753  012210 042764 100000 177776      BIC      #100000, A(R4)
6754
6755  , OTHERWISE CLEAR EXPECT_INPUT
6756
6757  012216 000402      BR      500045
        (3) 012220      500035
6758  012220 005067 167704      CLR      EXPINP
6759  012224      500045
6760
6761  , SET MESSAGE_ADDRESS AND
6762  , MESSAGE_SIZE

```

```

6763
6764 012224 016467 177776 167700      MOV      A(R4), OUTMSG
6765 012232 062767 000002 167712      ADD      #2, MSGAP
6766 012240 017767 167706 167666      MOV      @MSGAP, OUTMSG+2
6767
6768      UPDATE MESSAGE_AREA_POINTER
6769
6770 012246 067767 167700 167676      ADD      @MSGAP, MSGAP
6771 012254 062767 000002 167670      ADD      #2, MSGAP
6772
6773      IF MESSAGE_AREA_POINTER IS ODD
6774      ADD 1 TO MAKE IT EVEN
6775
6776 012262 016767 167664 167612      MOV      MSGAP, SCRTCH
6777 012270 016746 167606      MOV      SCRTCH, -(SP)
6778 012274 042716 000001      BIC      #1, (SP)
6779 012300 042667 167576      BIC      (SP)+, SCRTCH
6780 012304 005767 167572      TST      SCRTCH
6781 012310 001402      BEQ      50005$
6782 012312 005267 167624      INC      MSGAP
6783
6784      50005$
6785      50000$
6786      50001$
6787
6788      MOV      R4, SP
6789      MOV      (SF)+, R4
6790      RTS      PC
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
  
```

```

*****
ROUTINE TO HANDLE OUTPUT OF MESSAGES TO THE TERMINAL
UNDER TEST  MESSAGES MAY BE OF ANY LENGTH SINCE
ROUTINES USED BY THIS ROUTINE SPLIT THE MESSAGES INTO
BUFFERS OF APPROPRIATE LENGTH FOR ODCMP
*****
  
```

```

TEROUT:
TST      INIT
BNE      50002$
JSR      PC, XSTPT
BCC      50003$
SEC
BR       50001$
50003$
CLR      MSGNO
CLR      RMSGNO
50002$
  
```

INITIALIZE THE BUFFER ROUTINES 3 2 20 4

```

6803 012354 016767 167552 167572      MOV      OUTMSG, INITMA
6804 012362 012767 002132 167512      MOV      @OUTMSG, SCRTCH
6805 012370 062767 000002 167504      ADD      #2, SCRTCH
6806 012376 017767 167500 167552      MOV      @SCRTCH, INITMS
6807 012404 012767 010204 167546      BUFFER #1, BAPNT
      MOV      @TMPBUF, BAPNT
  
```

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 187-2
 DZDZCA P11 21-NOV-77 14 55

SEQ 0033

```

6808 012412 012767 000412 167542      MOV      #266, BASIZE
6809
6810      GET A BUFFER OF THE MESSAGE
6811
6812 012420 162705 000004      SUB      #2*2, R5
(3) 012424 004767 000410      JSR      PC, GET08
(4) 012430 012567 167506      MOV      (R5)+, OUTBUF
(4) 012434 012567 167504      MOV      (R5)+, OUTSIZ
6813
6814      PERFORM THE FOLLOWING LOOP WHILE
6815      A AND ABORT ARE CLEAR AND OUTSIZ
6816      IS GREATER THAN 0
6817
6818 012440      50004$
(6) 012440 005767 167500      TST      OUTSIZ
(9) 012444 003442      BLE      50005$
(6) 012446 005767 167434      TST      ABORT
(9) 012452 001037      BNE      50005$
6819
6820      ADD PROTOCOL ENVELOPE AND TRANSMIT
6821
6822 012454 162705 000004      SUB      #2*2, R5
(3) 012460 010546      MOV      R5, -(SP)
(5) 012462 016745 167456      MOV      OUTSIZ, -(R5)
(4) 012466 016745 167450      MOV      OUTBUF, -(R5)
(3) 012472 004767 000470      JSR      PC, PROBUF
(3) 012476 012605      MOV      (SP)+, R5
(4) 012500 012567 002674      MOV      (R5)+, XBUFFER
(4) 012504 012567 002672      MOV      (R5)+, XSIZE
6823 012510 004767 001756      JSR      PC, XMIT
6824
6825      IF TRANSMIT ERROR, REPORT IT AND SET ERR=MT
6826
6827 012514 103005      BCC      50006$
6828 012516 104423      TRAP     T$ERRCODE
(5) 012520 000024      WORD     20
(5) 012522 002636      WORD     MSG20
6829 012524 000261      SEC
(4) 012526 000412      BR       50001$
6830 012530      50006$
6831
6832      GET THE NEXT BUFFER
6833
6834 012530 162705 000004      SUB      #2*2, R5
(3) 012534 004767 000300      JSR      PC, GET08
(4) 012540 012567 167376      MOV      (R5)+, OUTBUF
(4) 012544 012567 167374      MOV      (R5)+, OUTSIZ
6835
6836      END OF LOOP
6837
6838 012550 000733      BR       50004$
(3) 012552      50005$
6839
6840      IF ERROR OCCURRED DO AN ERROR RETURN
6841
6842 012552      50000$

```

(2)	012552	000241			CLC
(3)	012554			500015	
(2)	012554	000207			RTS PC
6843	012556	000000			TYPE 0 , TYPE OF MESSAGE EXPECTED
6844					
6845					
6846					
6847					
6848					
6849					
6850					
6851					
6852					
6853	012560				
6854	012560	010446			
(2)	012562	010604			
(3)	012564	162706	000002		
6857					
6858					
6859					
6860					
6861					
6862	012570	012767	000010	002604	
6863	012576	012767	002516	002574	
6864					
6865					
6866					
6867	012604	012764	002170	177776	
(6)	012612	062764	000005	177776	
6868	012620	117467	177776	170675	
6869					
6870					
6871					
6872					
6873	012626	162705	000002		
(3)	012632	010546			
(5)	012634	012745	000006		
(4)	012640	016745	002534		
(2)	012644	004767	003702		
(3)	012650	012605			
(4)	012652	012567	170646		
6874					
6875					
6876					
6877	012656	005067	000154		
6878	012662				
6879	012662	004767	001604		
6880					
6881					
6882					
6883	012666	103005			
6884	012670	104423			
(5)	012672	000024			
(5)	012674	002636			
6885	012676	000261			
(4)	012700	000453			


```

*****
ROUTINE TO SEND A "STRT" MESSAGE TO THE TERMINAL
A "STACK" MUST BE RETURNED BY THE TERMINAL
*****
XSTRT
MOV R4, -(SP)
MOV SP, R4
SUB #2, SP

THE DDCMP STRT MESSAGE IS 10 BYTES LONG (INCLUDING CRC) AND IS IN
A BUFFER STARTING AT "STRTB" -- PUT THIS INFORMATION
IN THE TRANSMIT ROUTINE'S POINTERS

MOV #10, XSIZE
MOV #STRTB, XBUFFER

GET THE TERMINAL ADDRESS AND PUT INTO THE STRT MESSAGE

MOV #PTABLE, A(R4)
ADD #5, A(R4)
MOVB @A(R4), STRTSA

COMPUTE THE CRC FOR THE STRT MESSAGE AND PUT IN THE LAST TWO BYTES
OF THE BUFFER ("STRTC")

SUB #1*2, R5
MOV R5, -(SP)
MOV #6, -(R5)
MOV XBUFFER, -(R5)
JSR PC, CRC
MOV (SP)+, R5
MOV (R5)+, STRTC

TRANSMIT THE MESSAGE

CLR DDSTRY
50002$ JSR PC, XMIT

IF AN ERROR OCCURS, REPORT IT AND EXIT WITH THE ERROR FLAG SET

BCC 50003$
TRAP T$ERCODE
WORD 20
WORD MSG20
SEC
BR 50001$
  
```

6886	012702			50003\$	
6887					
6888					OTHERWISE, GET THE TERMINAL'S RESPONSE
6889					
6890	012702	010546		MOV	R5, -(SP)
(4)	012704	012745	000001	MOV	#1, -(R5)
(3)	012710	004767	003344	JSR	PC, RCVE
(3)	012714	012605		MOV	(SP)+, R5
6891					
6892					IF RECEPTION ERROR, REPORT IT AND EXIT WITH THE ERROR FLAG SET
6893					
6894	012716	103003		BCC	50004\$
6895	012720	005267	000112	INC	DDSTRY
6896	012724	000402		BR	50005\$
(3)	012726			50004\$	
6897	012726	005067	000104	CLR	DDSTRY
6898	012732			50005\$	
6899	012732	005767	000100	TST	DDSTRY
(6)	012736	001404		BEQ	50006\$
(4)	012740	026727	000072 000003	CMP	DDSTRY, #3
(7)	012746	001345		BNE	50002\$
(4)	012750			50006\$	
6900	012750	005767	000062	TST	DDSTRY
(9)	012754	001405		BEQ	50007\$
6901	012756	104423		TRAP	TSEPCODE
(5)	012760	000025		WORD	21
(5)	012762	002655		WORD	MSG21
6902	012764	000261		SEC	
(4)	012766	000420		BR	50001\$
6903	012770			50007\$	
6904					
6905					OTHERWISE, SET TYPE TO "STACK" AND CHECK TO SEE THAT IT WAS THE REPLY
6906					
6907	012770	012767	000007 177560	MOV	#STACK, TYPE
6908	012776	010546		MOV	R5, -(SP)
(4)	013000	016745	177552	MOV	TYPE, -(R5)
(3)	013004	004767	001072	JSR	PC, CHKREP
(3)	013010	012605		MOV	(SP)+, R5
6909					
6910					IF REPLY WASN'T A "STACK", REPORT THE ERROR AND EXIT WITH THE ERROR FLAG SET
6911					
6912	013012	103005		BCC	50010\$
6913	013014	104423		TRAP	TSEPCODE
(5)	013016	000031		WORD	25
(5)	013020	003043		WORD	MSG25
6914	013022	000261		SEC	
(4)	013024	000401		BR	50001\$
6915	013026			50010\$	
6916					
6917					END OF XSTR ROUTINE
6918					
6919	013026			50000\$	
(2)	013026	000241		CLC	
(3)	013030			50001\$	
(3)	013030	010406		MOV	R4, SP
(3)	013032	012604		MOV	(SP)+, R4

```

DDSTRY 0      RTS      PC
;
;*****
; THIS ROUTINE SPLITS A MESSAGE OF ANY LENGTH INTO
; SEGMENTS OF A SIZE SUITABLE FOR TRANSMISSION OVER A DATA
; LINK -- RETURNS BUFFER ADDRESS, BUFFER SIZE AND END_OF_MESSAGE+
; (BUFFER SIZE = 0 IS END_OF_MESSAGE)
;*****
GETOB      MOV      R4, -(SP)
           MOV      SP, R4
           SUB      #2, SP
;
; LEAVE ENOUGH ROOM FOR A DDCMP HEADER
;
           MOV      BAPNT, A(R5)
           ADD      #10, A(R5)
;
; INIT SIZE TO ZERO
           CLR      B(R5)
;
; SCRTCH HAS MAXIMUM DDCMP MESSAGE SIZE
           MOV      BASIZE, SCRTCH
           SUB      #12, SCRTCH
;
; GET A COPY OF THE BUFFER POINTER
           MOV      A(R5), C(R4)
;
; REPEAT LOOP UNTIL MAX SIZE OR END OF DATA
50002$
           CMP      B(R5), SCRTCH
           BGE      50003$
           TST      INITMS
           BLE      50003$
;
; MOVE DATA INTO MESSAGE BUFFER
           MOVB     @INITMA, @C(R4)
           DEC      INITMS
           INC      INITMA
           INC      B(R5)
           INC      C(R4)
           BR       50002$
50003$
;
; END OF LOOP

```

6968	013160			50000\$	
(3)	013160			50001\$	
(3)	013160	010406		MOV	R4, SP
(3)	013162	012604		MOV	(SP)+, R4
(2)	013164	000207		RTS	PC
6969					
6970					
6971					
6972					
6973					
6974					
6975					
6976					
6977					
6978					
6979	013166				
6980	013166	010146			
6981	013170	010246			
6982					
6983					
6984					
6985	013172	016502	000000		
(6)	013176	162702	000010		
6986					
6987					
6988					
6989	013202	112722	000201		
6990					
6991					
6992					
6993	013206	016501	000002		
6994	013212	110122			
6995	013214	000301			
6996	013216	110112			
6997					
6998					
6999					
7000	013220	152712	000200		
7001	013224	005202			
7002					
7003					
7004					
7005	013226	116722	166734		
7006					
7007					
7008					
7009	013232	005267	166726		
7010	013236	116722	166722		
7011					
7012					
7013					
7014	013242	012701	002170		
(6)	013246	062701	000005		
7015	013252	111122			
7016					
7017					


```

*****
ROUTINE TO PROVIDE LINE PROTOCOL HEADER AND CHECKSUMS
CALL WITH DATA AREA ADDRESS AND SIZE (IN BYTES)
RETURN WITH MODIFIED ADDRESS AND SIZE
*****
PROBUF
MOV R1, -(SP)
MOV R2, -(SP)

CHANGE POINTER TO INCLUDE DDCMP HEADER

MOV A(R5), R2
SUB #10, R2

FIRST BYTE IS SOH (START OF HEADER)

MOVB #SOH, (R2)+

NEXT TWO BYTES GET MESSAGE SIZE

MOV B(R5), R1
MOVB R1, (R2)+
SWAB R1
MOVB R1, (R2)

SET SELECT BIT

BISB #200, (R2)
INC R2

NEXT BYTE IS LAST GOOD RESPONSE

MOVB RMSGNO, (R2)+

NEXT BYTE GETS MESSAGE NUMBER --- POP NUMBER FOR NEXT MESSAGE (IF ANY)

INC MSGNO
MOVB MSGNO, (R2)+

GET THE TERMINAL ADDRESS FOR THE NEXT HEADER BYTE

MOV #PTABLE R1
ADD #5, R1
MOVB (R1), (R2)+

GET HEADER START ADDRESS AND SIZE IN BYTES

```

```

7018
7019 013254 016567 000000 166620      MOV      A(R5), SCRTCH
    (6) 013262 162767 000010 166612      SUB      #10, SCRTCH
7020 013270 012767 000006 166606      MOV      #6, TEMP
7021
7022      COMPUTE HEADER CRC
7023
7024 013276 162705 000002      SUB      #1*2, R5
    (3) 013302 010546      MOV      R5, -(SP)
    (5) 013304 016745 166574      MOV      TEMP, -(R5)
    (4) 013310 016745 166566      MOV      SCRTCH, -(R5)
    (3) 013314 004767 003232      JSR      PC, CRC
    (3) 013320 012605      MOV      (SP)+, R5
    (4) 013322 012567 166556      MOV      (R5)+, TEMP
7025
7026      PUT CRC INTO BUFFER
7027
7028 013326 016712 166552      MOV      TEMP, (R2)
7029
7030      GET ADDRESS OF MESSAGE AND MESSAGE SIZE IN BYTES
7031
7032 013332 016567 000000 166542      MOV      A(R5), SCRTCH
7033 013340 016567 000002 166536      MOV      B(R5), TEMP
7034
7035      COMPUTE DATA CRC
7036
7037 013346 162705 000002      SUB      #1*2, R5
    (3) 013352 010546      MOV      R5, -(SP)
    (5) 013354 016745 166524      MOV      TEMP, -(R5)
    (4) 013360 016745 166516      MOV      SCRTCH, -(R5)
    (3) 013364 004767 003162      JSR      PC, CRC
    (3) 013370 012605      MOV      (SP)+, R5
    (4) 013372 012567 166506      MOV      (R5)+, TEMP
7038
7039      PUT DATA CRC IN LAST TWO BYTES OF BUFFER
7040
7041 013376 016502 000000      MOV      A(R5), R2
    (6) 013402 066502 000002      ADD      B(R5), R2
7042 013406 116722 166472      MOVB     TEMP, (R2)+
7043 013412 000367 166466      SWAB     TEMP
7044 013416 116712 166462      MOVB     TEMP, (R2)
7045
7046      RETURN WITH MODIFIED ADDRESS AND SIZE
7047
7048 013422 016565 000002 000006      MOV      B(R5), D(R5)
    (6) 013430 062765 000012 000006      ADD      #12, D(R5)
7049 013436 016565 000000 000004      MOV      A(R5), C(R5)
    (6) 013444 162765 000010 000004      SUB      #10, C(R5)
7050 013452 012602      MOV      (SP)+, R2
7051 013454 012601      MOV      (SP)+, R1
7052 013456      50000$
    (3) 013456      50001$
    (2) 013456 000207      RTS      PC
7053
7054
7055
  
```

7056						
7057						
7058						
7059						ROUTINE TO GET TERMINAL INPUT
7060						
7061	013460					TERIN:
7062						
7063						SAVE R1
7064						
7065	013460	010146				MOV R1, -(SP)
7066						
7067						GET TERMINAL RESPONSE
7068						
7069	013462	010546				MOV R5, -(SP)
(4)	013464	012745	000000			MOV #0, -(R5)
(3)	013470	004767	002564			JSR PC, RCVE
(3)	013474	012605				MOV (SP)+, R5
7070						
7071						IF RECEIVE ERROR, REPORT IT AND EXIT WITH THE ERROR FLAG SET
7072						
7073	013476	103007				BCC 50002\$
7074	013500	104463				TRAP T\$ERCODE
(5)	013502	000050				WORD 40
(5)	013504	002655				WORD MSG40
(5)	013506	004110				WORD AREA40
7075						
7076						RESTORE R1 FIRST
7077						
7078	013510	012601				MOV (SP)+, R1
7079	013512	000261				SEC
(4)	013514	000571				BR 50001\$
7080	013516					50002\$
7081						
7082						IF MESSAGE IS ONLY 10 BYTES, SEE IF IT IS AN ACK
7083						IT'S OK TO BE AN ACK, BUT RECEPTION IS INCOMPLETE (CLEAR FLAG)
7084						
7085	013516	026727	166426	000010		CMP INSIZ, #10
(9)	013524	001027				BNE 50003\$
7086	013526	005065	000000			CLR A(R5)
7087	013532	012767	000001	177016		MOV #ACK, TYPE
7088	013540	010546				MOV R5, -(SP)
(4)	013542	016745	177010			MOV TYPE, -(R5)
(3)	013546	004767	000330			JSR PC, CHKREP
(3)	013552	012605				MOV (SP)+, R5
7089	013554	103003				BCC 50004\$
7090	013556	012601				MOV (SP)+, R1
7091	013560	000261				SEC
(4)	013562	000546				BR 50001\$
7092	013564					50004\$
7093	013564	016701	166356			MOV INBUF, R1
(6)	013570	062701	000004			ADD #4, R1
7094	013574	111167	166366			MOVB (R1), RMSGNO
7095	013600	012601				MOV (SP)+, R1
7096	013602	000535				BR 50000\$
7097	013604					50003\$
7098						

```

7099      , IF RECEIVE OK, CHECK THE HEADER CRC
7100      , COMPUTE EXPECTED CRC
7101      ,
7102      013604 162705 000002      SUB      #1*2,R5
      (3) 013610 010546      MOV      R5,-(SP)
      (5) 013612 012745 000006      MOV      #6,-(R5)
      (4) 013616 016745 166324      MOV      INBUF,-(R5)
      (3) 013622 004767 002724      JSR      PC,CRC
      (3) 013626 012605      MOV      (SP)+,R5
      (4) 013630 012567 170144      MOV      (R5)+,A04EX
7103      ,
7104      , GET THE CRC RECEIVED
7105      ,
7106      013634 016701 166306      MOV      INBUF,R1
      (6) 013640 062701 000010      ADD      #10,R1
7107      013644 114167 170126      MOV      -(R1),A04RC
7108      013650 000367 170122      SWAB     A04RC
7109      013654 114167 170116      MOV      -(R1),A04RC
7110      ,
7111      , IF THEY AREN'T THE SAME, REPORT THE ERROR, RESTORE R1 AND EXIT WITH
7112      , THE ERROR FLAG SET
7113      ,
7114      013660 026767 170114 170110      CMP      A04EX,A04PC
      (9) 013666 001407      BEQ      500055
7115      013670 104463      TRAP     T$EPCODE
      (5) 013672 000051      WORD     41
      (5) 013674 002427      WORD     MSG41
      (5) 013676 003744      WORD     AREA41
7116      013700 012601      MOV      (SP)+,R1
7117      013702 000261      SEC
      (4) 013704 000475      BR       500015
7118      013706      500055
7119      ,
7120      , COMPARE TERMINAL ADDRESS RECEIVED WITH THAT EXPECTED
7121      , AND IF THEY AREN'T THE SAME, REPORT THE ERROR, RESTORE R1 AND
7122      , EXIT WITH THE ERROR FLAG SET
7123      ,
7124      013706 124167 167611      CMPB     -(R1),STRTSA
      (9) 013712 001407      BEQ      500065
7125      013714 104463      TRAP     T$EPCODE
      (5) 013716 000052      WORD     42
      (5) 013720 002363      WORD     MSG42
      (5) 013722 003712      WORD     AREA42
7126      013724 012601      MOV      (SP)+,R1
7127      013726 000261      SEC
      (4) 013730 000463      BR       500015
7128      013732      500065
7129      ,
7130      , MOVE THE POINTERS PAST THE HEADER AND COMPUTE THE DATA CRC
7131      ,
7132      013732 062767 000010 166206      ADD      #10,INBUF
7133      013740 162767 000012 166202      SUB      #12,INSIZ
7134      013746 162705 000002      SUB      #1*2,R5
      (3) 013752 010546      MOV      R5,-(SP)
      (5) 013754 016745 166170      MOV      INSIZ,-(R5)
      (4) 013760 016745 166162      MOV      INBUF,-(R5)
  
```

(3)	013764	004767	002562	JSR	PC, CRC
(3)	013770	012605		MOV	(SP)+, R5
(4)	013772	012567	170002	MOV	(R5)+, A04EX
7135					
7136					GET THE DATA CRC RECEIVED
7137					
7138	013776	016701	166144	MOV	INBUF, R1
(6)	014002	066701	166142	ADD	INSIZ, R1
7139	014006	062701	000002	ADD	#2, R1
7140	014012	114167	167760	MOVB	-(R1), A04RC
7141	014016	000367	167754	SWAB	A04RC
7142	014022	114167	167750	MOVB	-(R1), A04RC
7143					
7144					IF THEY AREN'T THE SAME, REPORT THE ERROR, RESTORE R1 AND
7145					EXIT WITH THE ERROR FLAG SET
7146					
7147	014026	026767	167744 167744	CMP	A04RC, A04EX
(9)	014034	001407		BEQ	500075
7148	014036	104463		TRAP	T\$ERCODE
(5)	014040	000053		WORD	43
(5)	014042	003242		WORD	MSG43
(5)	014044	003744		WORD	AREA43
7149	014046	012601		MOV	(SP)+, R1
7150	014050	000261		SEC	
(4)	014052	000412		BR	500015
7151	014054				500075
7152					
7153					OTHERWISE, SET COMPLETE FLAG, RESTORE P1 AND EXIT NORMALLY
7154					
7155	014054	005265	000000	INC	A(R5)
7156	014060	016701	166062	MOV	INBUF, R1
(6)	014064	162701	000004	SUB	#4, R1
7157	014070	111167	166072	MOVB	(R1), RMSGNO
7158	014074	012601		MOV	(SP)+, P1
7159	014076				500005
(2)	014076	000241		CLC	
(3)	014100				500015
(2)	014100	000207		RTS	PC
7160					
7161					
7162					
7163					
7164					
7165					
7166					*****
7167					ROUTINE TO CHECK AN INPUT BUFFER FOR A STANDARD
7168					RESPONSE ("ACK" OR "STACK")
7169					
7170					LOCATION "TYPE" CONTAINS THE RESPONSE CODE EXPECTED
7171					
7172					INPUT BUFFER POINTER IS IN "INBUF"
7173					
7174					INPUT BUFFER SIZE (IN BYTES) IS IN "INSIZ"
7175					*****
7176	014102			CHKREP	
7179	014102	010446		MOV	R4, -(SP)

(2)	014104	010604			MOV	SP, R4
(3)	014106	162706	000002		SUB	#2, SP
7180						
7181					; IF MESSAGE IS NOT 10 BYTES LONG, IT CAN'T BE A CONTROL MESSAGE	
7182					; TYPE AN ERROR MESSAGE AND EXIT WITH ERROR SET	
7183						
7184	014112	026727	166032	000010	CMP	INSIZ, #10
(9)	014120	001406			BEQ	50002\$
7185	014122	104463			TRAP	T\$ERCODE
(5)	014124	000001			WORD	1
(5)	014126	002311			WORD	MSG01
(5)	014130	003606			WORD	AREA01
7186	014132	000261			SEC	
(4)	014134	000553			BR	50001\$
7187	014136					50002\$
7188						
7189					; OTHERWISE, COMPUTE THE CRC AND CHECK IT AGAINST THAT SENT	
7190						
7191	014136	162705	000002		SUB	#1*2, R5
(3)	014142	010546			MOV	R5, -(SP)
(5)	014144	012745	000006		MOV	#6, -(R5)
(4)	014150	016745	165772		MOV	INBUF, -(R5)
(3)	014154	004767	002372		JSR	PC, CRC
(3)	014160	012605			MOV	(SP)+, R5
(4)	014162	012567	165716		MOV	(R5)+, TEMP
7192	014166	016764	165754	177776	MOV	INBUF, A(R4)
(6)	014174	062764	000007	177776	ADD	#7, A(R4)
7193	014202	117467	177776	165672	MOVB	@A(R4), SCRTCH
7194	014210	000367	165666		SWAB	SCRTCH
7195	014214	005364	177776		DEC	A(R4)
7196	014220	117467	177776	165654	MOVB	@A(R4), SCRTCH
7197						
7198					; IF CRC'S MATCH GO ON OTHERWISE SKIP FOLLOWING CODE	
7199						
7200	014226	026767	165652	165646	CMP	TEMP, SCRTCH
(9)	014234	001076			BNE	50004\$
7201						
7202					; COMPARE CONTROL MESSAGE TYPE WITH THAT PASSED AS AN ARGUMENT	
7203						
7204	014236	016764	165704	177776	MOV	INBUF, A(R4)
(6)	014244	005264	177776		INC	A(R4)
7205	014250	117464	177776	177776	MOVB	@A(R4), A(R4)
7206	014256	042764	177400	177776	BIC	#HYBYTE, A(R4)
7207						
7208					; IF NOT THE EXPECTED TYPE, PRINT ERROR MESSAGE AND EXIT WITH ERROR SET	
7209						
7210	014264	026465	177776	000000	CMP	A(R4), B(R5)
(9)	014272	001414			BEQ	50005\$
7211	014274	016467	177776	167340	MOV	A(R4), A02A
7212	014302	016567	000000	167334	MOV	B(R5), A02B
7213	014310	104463			TRAP	T\$ERCODE
(5)	014312	000002			WORD	2
(5)	014314	002336			WORD	MSG02
(5)	014316	003610			WORD	AREA02
7214	014320	000261			SEC	
(4)	014322	000460			BR	50001\$

50005\$

```
MOV      #PTABLE, SCRTCH
ADD      #5, SCRTCH
MOVB     @SCRTCH, SCRTCH
BIC      #HYBYTE, SCRTCH
MOV      INBUF, TEMP
ADD      #5, TEMP
MOVB     @TEMP, TEMP
BIC      #HYBYTE, TEMP
```

CMP	TEMP, SCRTCH
BEQ	50007\$
TRAP	T\$ERCODE
WORD	3
WORD	MSG03
WORD	AREAO3
SEC	
BR	50001\$

50007\$
50006\$

```

, IF CRC WAS IN ERROR, SKIP ABOVE CODE, COME TO HEPE, PEPOT ERROR AND
EXIT WITH ERROR SET

```

50004\$ BR 50010\$

```

MOV      SCRATCH, A04RC
MOV      TEMP, A04EX
TRAP     T$ERCODE
        WORD    4
        WORD    MSG04
        WORD    APEA04
SEC
BR        500015

```

50010\$
50003\$

NORMAL RETURN

50000\$

500015

```
CLC
MOV     P4, SP
MOV     (SP)+, P4
RTS     PC
```

7254				*****
7255				THIS ROUTINE SETS UP AND CHECKS FOR COMPLETE TRANSMISSION +
7256				RETURNS AN ERROR FLAG IF ANY ERROR OCCURS DURING TRANSMISSION +
7257				SET "ABORT" IF INITIALIZATION OF THE DEVICE FAILS +
7258				*****
7259	014472			XMIT
7262	014472	010446		MOV R4, -(SP)
(2)	014474	010604		MOV SP, R4
(3)	014476	162706	000002	SUB #2, SP
7263				
7264				CLEAR ERROR COUNT, TRANSMIT DONE FLAG AND RECEIVE DONE FLAG
7265				
7266	014502	005064	177776	CLR B(R4)
7267	014506	005067	000270	CLR XDONE
7268	014512	005067	000266	CLR RDONE
7269				
7270				START OF LOOP
7271				
7272	014516			50002\$
7273				
7274				TRY TO INIT DEVICE
7275				
7276	014516	004767	000264	JSR PC.DEVINIT
7277				
7278				IF ERROR OCCURRED
7279				
7280	014522	103033		BCC 50003\$
7281	014524	010146		MOV R1, -(SP)
7282	014526	010246		MOV R2, -(SP)
7283				
7284				GET A COPY OF THE DEVICE REGISTERS
7285				
7286	014530	016701	165434	MOV PTABLE, R1
7291	014534	012702	002242	MOV #DZREG, R2
7294	014540	016162	000000 000000	MOV RCVCSR(R1), RCVCSR(R2)
7295	014546	016162	000002 000002	MOV RCVBUF(R1), RCVBUF(R2)
7296	014554	016162	000004 000004	MOV XMTCSR(R1), XMTCSR(R2)
7297	014562	016162	000006 000006	MOV XMTBUF(R1), XMTBUF(R2)
7308	014570	012602		MOV (SP)+, R2
7309	014572	012601		MOV (SP)+, R1
7310				
7311				REPORT THE ERROR AND UPDATE THE ERROR COUNT
7312				
7313	014574	104463		TRAP T\$ERCODE
(5)	014576	000027		WORD 23
(5)	014600	002740		WORD MSG23
(5)	014602	004110		WORD AREA23
7314	014604	005264	177776	INC B(R4)
7315				
7316				OTHERWISE, CLEAR THE ERROR COUNT
7317				
7318	014610	000402		BR 50004\$
(3)	014612			50003\$
7319	014612	005064	177776	CLR B(R4)
7320	014616			50004\$
7321				

```

7322      , IF ERROR COUNT HAS EXCEEDED LIMITS, ABORT AND NEGATE THE COUNT
7323
7324      014616 026467 177776 165270      CMP      B(R4), ILIMIT
(9)      014624 001010                      BNE      50005$
7325      014626 104462                      TRAP     TSERCODE
(5)      014630 000000                      WORD     0
(5)      014632 002252                      WORD     MSG00
(5)      014634 003526                      WORD     AREA00
7326      014636 005267 165244              INC      ABORT
7327      014642 005464 177776              NEG      B(R4)
7328      014646                      50005$
7329
7330                      END OF LOOP
7331      , EXIT LOOP WHEN ERROR COUNT IS ZERO OR NEGATIVE (ABORT)
7332
7333      014646 005764 177776              TST      B(R4)
(6)      014652 003321                      BGT      50002$
7334
7335      , IF NOT ABORT THEN EXECUTE THE FOLLOWING CODE
7336      , OTHERWISE SKIP AND RETURN WITH ERROR SET
7337
7338      014654 005764 177776              TST      B(R4)
(9)      014660 001042                      BNE      50006$
7339
7340      , WAIT FOR TRANSMIT DONE FLAG TO SET
7341
7342      014662                      50007$
7343      014662 104022                      EMT      CSBRK
7344      014664 005767 000112              TST      XDONE
(6)      014670 001774                      BEQ      50007$
7345
7346      , IF TRANSMIT FLAG IS NEGATIVE, AN ERROR OCCURRED
7347
7348      014672 005767 000104              TST      XDONE
(9)      014676 002032                      BGE      50010$
7349      014700 010146                      MOV      R1, -(SP)
7350      014702 010246                      MOV      R2, -(SP)
7351
7352      , IF ERROR, GET COPY OF DEVICE REGISTERS PRINT MESSAGE AND EXIT WITH ERROR SET
7353      , OTHERWISE DO A NORMAL EXIT
7354
7355      014704 016701 165260              MOV      PTABLE, R1
7356      014710 012702 002242              MOV      #DZREG, R2
7357      014714 016162 000000 000000        MOV      RCUCSR(R1), RCUCSR(R2)
7358      014722 016162 000002 000002        MOV      RCVBUF(R1), RCVBUF(R2)
7359      014730 016162 000004 000004        MOV      XMTCSR(R1), XMTCSR(R2)
7360      014736 016162 000006 000006        MOV      XMTBUF(R1), XMTBUF(R2)
7361      014744 012602                      MOV      (SP)+, R2
7362      014746 012601                      MOV      (SP)+, R1
7363      014750 104463                      TRAP     TSERCODE
(5)      014752 000030                      WORD     24
(5)      014754 003000                      WORD     MSG24
(5)      014756 004110                      WORD     AREA24
7364      014760 000261                      SEC
(4)      014762 000404                      BR       50001$
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
  
```

7382	014764	000402		BR	500115
(3)	014766		500065	SEC	
7383	014766	000261		BR	500015
(4)	014770	000401			
7384	014772		500115		
7385	014772		500005		
(2)	014772	000241		CLC	
(3)	014774		500015		
(3)	014774	010406		MOV	R4, SP
(3)	014776	012604		MOV	(SP)+, R4
(2)	015000	000207		RTS	PC
7386	015002	000000	XDONE	0	
7387	015004	000000	RDONE	0	

 ROUTINE TO INITIALIZE A DEVICE
 IF "INIT" IS NON-ZERO, THE ROUTINE WILL EXECUTE AND SET IT
 RETURNS WITH AN ERROR IF CTS CANNOT BE SET
 RETURNS WITH ERROR AND "ABORT" SET IF THE DEVICE DOES
 NOT EXIST

7504			
7505	015006		
7508	015006	010446	
(2)	015010	010604	
(3)	015012	162706	000002
7509	015016	010146	
7510	015020	010246	

DEINIT
 MOV R4, -(SP)
 MOV SP, R4
 SUB #2, SP
 MOV R1, -(SP)
 MOV R2, -(SP)

7511			
7512			
7513			
7514	015022	016701	165142

GET DEVICE ADDRESS
 MOV PTABLE, R1

7515			
7516			
7517			
7518	015026	052761	000020 000000
7519	015034		
7520	015034	032761	000020 000000
(6)	015042	001374	

DO A MASTER CLEAR ON THE DZ AND WAIT FOR THE CLEAR BIT TO GO DOWN
 BIS #MASCLR, RCUCSR(R1)
 500025
 BIT #MASCLR, RCUCSR(R1)
 BNE 500025

7521			
7522			
7523			
7524	015044	016767	000310 165030
7525	015052	000367	165024
7526	015056	056761	165020 000004

GET THE LINE MASK AND SET THE RIGHT DATA TERMINAL READY BIT IN THE DZ
 MOV \$LINE, SCRATCH
 SWAB SCRATCH
 BIS SCRATCH, XMTCSR(R1)

7527			
7528			
7529			
7530			

GET THE LINE NUMBER, STOP CODE, CHARACTER LENGTH AND BAUD RATE FROM
 THE PTABLE AND CLEAR EXTRANEOUS BITS SET THE RECEIVED ENABLE BIT
 AND LOAD THE DZ LINE CONTROL REGISTER

```

7531
7532 015064 012702 002170      MOV    #PTABLE,R2
    (6) 015070 062702 000006      ADD    #6,R2
7533 015074 011202      MOV    (R2),R2
7534 015076 042702 000300      BIC    #300,R2
7535 015102 052702 010000      BIS    #10000,R2
7536 015106 010261 000002      MOV    R2,RCVBUF(R1)
7537
7538      GET THE DEVICE PRIORITY AND SET UP THE INTERRUPT HANDLERS AND
7539      CLEAR THE NEW MESSAGE FLAG (NMSG)
7540
7541 015112 012767 002170 165060      MOV    #PTABLE,DEVPR1
    (6) 015120 062767 000004 165052      ADD    #4,DEVPR1
7542 015126 117767 165046 165044      MOVB   @DEVPR1,DEVPR1
7543 015134 042767 177400 165036      BIC    #HYBYTE,DEVPR1
7544 015142 012764 002170 177776      MOV    #PTABLE,A(R4)
    (6) 015150 062764 000002 177776      ADD    #2,A(R4)
7545 015156 016746 165016      MOV    DEVPR1,-(SP)
    (6) 015162 012746 015612      MOV    #RINT,-(SP)
    (5) 015166 017446 177776      MOV    @A(R4),-(SP)
    (4) 015172 012746 000003      MOV    #3,-(SP)
    (3) 015176 104037      EMT     C$SVEC
    (2) 015200 062706 000010      ADD    #10,SP
7546 015204 005067 000174      CLR    NMSG
7547 015210 017464 177776 177776      MOV    @A(P4),A(R4)
    (6) 015216 062764 000004 177776      ADD    #4,A(R4)
7548 015224 016746 164750      MOV    DEVPR1,-(SP)
    (6) 015230 012746 015444      MOV    #XINT,-(SP)
    (5) 015234 016446 177776      MOV    A(R4),-(SP)
    (4) 015240 012746 000003      MOV    #3,-(SP)
    (3) 015244 104037      EMT     C$SVEC
    (2) 015246 062706 000010      ADD    #10,SP
7549
7550      GET COPIES OF THE DEVICE BUFFER REG STEP ADDRESSES
7551
7552 015252 010167 000104      MOV    R1,XMTADD
    (6) 015256 062767 000006 000076      ADD    #6,XMTADD
7553 015264 010167 000074      MOV    R1,RCVADD
    (6) 015270 062767 000002 000066      ADD    #2,RCVADD
7554
7555      SET COUNTS FOR 4 SYNC'S AND FOUR FILL CHARACTERS
7556
7557 015276 012767 000004 000064      MOV    #4,SYN_CNT
7558 015304 012767 000004 000060      MOV    #4,FLL_CNT
7559
7560      USE LINE MASK TO ENABLE THE TRANSMITTER      ENABLE THE MASTER SCAN
7561      ENABLE TRANSMIT AND RECEIVE INTERRUPTS
7562
7563 015312 056761 000042 000004      BIS    $LINE,XMTCSR(R1)
7564 015320 052761 000040 000000      BIS    #MSCNEN,RCVCSR(R1)
7565 015326 052761 040000 000000      BIS    #TXINTE,RCVCSR(R1)
7566 015334 052761 000100 000000      BIS    #RXINTE,RCVCSR(R1)
7567 015342 012602      MOV    (SP)+,R2
7568 015344 012601      MOV    (SP)+,R1
7569
7570      SET INIT COMPLETE FLAG
  
```

7571						
7572	015346	005267	000022		INC	INIT
7573	015352			50000\$		
(3)	015352			50001\$		
(3)	015352	010406			MOV	R4, SP
(3)	015354	012604			MOV	(SP)+, R4
(2)	015356	000207			RTS	PC
7574	015360	000000		\$LINE	0	
7648						
7649						
7650	015362	000000		XMTADD	0	, TRANSMIT BUFFER REGISTER ADDRESS
7651	015364	000000		RCVADD	0	, RECEIVE BUFFER REGISTER ADDRESS
7652	015366	000000		TIMER	0	, TIMER FOR CLEAR TO SEND OPERATION
7653	015370	000000		SYNCT	0	, NUMBER OF SYNC CHARACTERS INSERTED BEFORE DDCMP MESSAGE
7654	015372	000000		FLLCNT	0	, NUMBER OF FILL CHARACTERS AFTER MESSAGE
7655	015374	000000		INIT	0	, INITIALIZE COMPLETE FLAG
7656	015376	000377		FIL	377	, FILL CHARACTER
7657	015400	000000		XBUFFER	0	, CURRENT TRANSMIT BUFFER POINTER
7658	015402	000000		XSIZE	0	, CURRENT TRANSMIT BUFFER SIZE
7659	015404	000000		NMSG	0	, NEW MESSAGE EXPECTED FLAG
7660	015406	000000		RTEMP1	0	, CURRENT RECEIVE BUFFER POINTER
7661	015410	000000		RTEMP2	0	, CURRENT RECEIVE BUFFER SIZE
7662	015412	000000		CNTFLG	0	, CONTROL FLAG FOR INPUTTING DDCMP MESSAGES
7663						
7664						
7665						
7666	015414	000000		PQUEUE	0	, QUEUE POINTER FOR BUFFER PROCESSING
7667	015416	000000		PQUE	0	, QUEUE POINTER FOR NEXT AVAILABLE BUFFER
7668	015420	000000		RQUET	0	, QUEUE OF BUFFER ADDRESSES AND SIZES
7669	015422	000000			0	
7670	015424	000000			0	
7671	015426	000000			0	
7672	015430	000000			0	
7673	015432	000000		RQUEB	0	
7674	015434	000000		RBQUE	0	QUEUE POINTER FOR NEXT AVAILABLE BUFFER AREA
7675	015436	000000		RBQUET	0	QUEUE OF BUFFER AREAS
7676	015440	000000			0	
7677	015442	000000		RBQUEB	0	

```

7679
7680      TRANSMISSION INTERRUPT HANDLER
7681
7684
7685      SET PROCESSOR PRIORITY LEVEL
7686
7687      015444 016700 164530      MOV      DEVPRI,R0
(3) 7688      015450 104041      EMT      C$SPRI
7689
7690      IF SYNC COUNT EQUAL TO ZERO
7691      015452 005767 177712      TST      SYNCNT
(9) 7692      015456 001047      BNE      50003$
7693
7694      CHECK BUFFER SIZE
7695      IF GREATER THAN ZERO, THEN
7696      015460 010146      MOV      R1,-(SP)
7697      015462 005767 177714      TST      XSIZE
(9) 7698      015466 003413      BLE      50004$
7699
7700      SEND THE NEXT CHARACTER AND UPDATE POINTER AND COUNT
7701
7702      015470 117701 177704      MOVB     @XBUFFER,R1
7703      015474 042701 177400      BIC      #HYBYTE,R1
7704      015500 110177 177656      MOVB     R1,@XMTADD
7705      015504 005267 177670      INC      XBUFFER
7706      015510 005367 177666      DEC      XSIZE
7707
7708
7709
7710
7711
7712      IF ALL CHARACTERS SENT, CHECK FILL COUNT
7713
7714      015514 000426      BR      50005$
(3) 7715      015516      50004$
7716      015516 005767 177650      TST      FLLCNT
(9) 7717      015522 001406      BEQ      50006$
7718
7719
7720      IF FILL COUNT ISN'T ZERO SEND A FILL CHAP AND DECREMENT COUNT
7721
7722      015524 056777 177646 177630      BIS      FIL,@XMTADD
7723      015532 005367 177634      DEC      FLLCNT
7724      015536 000415      BR      50007$
(3) 7725      015540      50006$
7726
7727      IF ALL TRANSMIT DONE, DISABLE TRANSMISSION AND SET DONE FLAG
7728
7729      015540 016701 177616      MOV      XMTADD,R1
(6) 7730      015544 162701 000006      SUB      #6,R1
7731      015550 005267 177226      INC      XDONE
7732      015554 104021      EMT      C$ABRT
7733      015556 046761 177576 000004      BIC      $LINE,XMTC$R(R1)
7734      015564 042761 040000 000000      BIC      #TXINTE,RCVCSR(R1)
7735      015572      50007$
7736      015572      50005$
7737      015572 012001      MOV      (SP)+,R1
7738
7739
7740      OTHERWISE SEND A SYNC AND DECREMENT THE SYNC COUNT
7741

```

```

7742
7743 015574 000405          BR      50010$
(3) 015576
7749 015576 112777 000377 177556 50003$  MOVB  #SYNC, @XMTADD
7752 015604 005367 177560          DEC  SYNCNT
7753 015610          50010$
7754
7755          DO A SUPERVISOR-TYPE RTI
7756
7757 015610          L10027
(2) 015610 000002          RTI
7759
7760
7761
7762
7763
7764
7765          RECEPTION INTERRUPT ROUTINE
7766
7768
7769          SET PROCESSOR PRIORITY
7770
7771 015612 016700 164262          MOV  DEVPRI, R0
(3) 015616 104041          EMT   CSSPRI
7772 015620 010146          MOV  R1, -(SP)
7773
7774          IF NEW MESSAGE FLAG IS RESET
7775
7776 015622 005767 177556          TST   NMSG
(9) 015626 001111          BNE   50011$
7777
7778          OBTAIN A BUFFER FROM THE QUEUE AND GET THE RECEIVED BYTE
7779
7780 015630 017767 177600 177550          MOV  @RBQUE, RTEMP1
7781 015636 016777 177544 177552          MOV  RTEMP1, @RQUE
7782 015644 117777 177514 177534          MOVB  @RCVADD, @RTEMP1
7783 015652 117701 177530          MOVB  @RTEMP1, R1
7784
7785          CHECK FOR SYNC/FILL CHARACTER
7786          (LOOKING FOR FIRST NON-SYNC/NON-FILL)
7787
7788 015656 120127 000377          CMPB  R1, #SYNC
(9) 015662 001472          BEQ   50012$
(6) 015664 120167 177506          CMPB  R1, FIL
(9) 015670 001467          BEQ   50012$
7789
7790          NOT FILL OR SYNC CHECK FOR ENQ (DDCMP CONTROL MESSAGE)
7791
7792 015672 120127 000005          CMPB  R1, #ENQ
(9) 015676 001027          BNE   50013$
7793
7794          IF IT IS AN ENQ, SET EXPECTED BYTE COUNT TO 7, SET MESSAGE SIZE
7795          TO 10 (OCTAL) AND UPDATE THE BUFFER QUEUES
7796
7797 015700 012767 000007 177502          MOV  #7, RTEMP2
7798 015706 062767 000002 177502          ADD  #2, RQUE
  
```

7799	015714	012777	000010	177474	MOV	#10, RQUE
7800	015722	062767	000002	177466	ADD	#2, RQUE
7801	015730	026727	177462	015432	CMP	RQUE, #RQUEB
(9)	015736	003403			BLE	50014\$
7802	015740	012767	015420	177450	MOV	#RQUET, RQUE
7803	015746					50014\$
7804						
7805						SET COUNTER FLAG TO -1 (MESSAGE SIZE COUNT COMPLETE)
7806						
7807	015746	012767	177777	177436	MOV	#-1, CNTFLG
7808	015754	000417			BR	50015\$
(3)	015756					50013\$
7809						
7810						NOT FILL, SYNC OR ENQ CHECK FOR SOH OR DLE (DDCMP DATA MESSAGE)
7811						
7812	015756	120127	000201		CMPB	R1, #SOH
(8)	015762	001403			BEQ	50016\$
(6)	015764	120127	000220		CMPB	R1, #DLE
(9)	015770	001007			PNE	50017\$
(6)	015772					50016\$
7813						
7814						MOVE QUEUE POINTER TO SIZE ENTRY AND SET COUNTER FLAG TO 1 (EXPECT
7815						FIRST COUNT BYTE)
7816						
7817	015772	062767	000002	177416	ADD	#2, PQUE
7818	016000	012767	000001	177404	MOV	#1, CNTFLG
7819	016006	000402			BR	50020\$
(3)	016010					50017\$
7820						
7821						BYTE WASN'T ANY OF THE LEGAL CHARACTERS FORCE
7822						A RESTART BY FORCING THE NEW MESSAGE FLAG TO RESET ON EXIT
7823						
7824	016010	005367	177370		DEC	NMSG
7825	016014					50020\$
7826	016014					50015\$
7827						
7828						COUNT THE CHARACTER AND UPDATE THE FREE BUFFER QUEUE
7829						
7830	016014	005267	177366		INC	RTEMP1
7831	016020	062767	000002	177406	ADD	#2, RBQUE
7832	016026	026727	177402	015442	CMP	RBQUE, #RBQUEB
(9)	016034	003403			BLE	50021\$
7833	016036	012767	015436	177370	MOV	#RBQUET, RBQUE
7834	016044					50021\$
7835						
7836						SET THE NEW MESSAGE FLAG
7837						EXIT ROUTINE
7838						
7839	016044	005267	177334		INC	NMSG
7840	016050					50012\$
7841						
7842						NEW MESSAGE FLAG IS SET CONTINUE BUILDING MESSAGE
7843						
7844	016050	000501			BR	50022\$
(3)	016052					50011\$
7845						

```

7846      ,NOT A NEW MESSAGE .. GET BYTE AND STORE IT
7847
7848 016052 117777 177306 177326      MOVB  @RCVADD,@RTEMP1
7849
7850      , IF COUNTER FLAG IS GREATER THAN ZERO, BYTE IS ALSO THE FIRST COUNT BYTE
7851
7852 016060 005767 177326      TST  CNTFLG
(9) 016064 003413      BLE  50023$
7853
7854      , PUT THE BYTE IN RTEMP2 (EXPECTED COUNT), UPDATE THE BUFFER POINTER
7855      , DECREMENT THE COUNTER FLAG (SET TO GET SECOND COUNT BYTE) AND
7856      , GO DO AN RTI
7857
7858 016066 117767 177314 177314      MOVB  @RTEMP1,RTEMP2
7859 016074 042767 177400 177306      BIC  #HYBYTE,RTEMP2
7860 016102 005267 177300      INC  RTEMP1
7861 016106 005367 177300      DEC  CNTFLG
7862 016112 000460      BR    50024$
(3) 016114      50023$
7863
7864      , COUNTER FLAG NOT GREATER THAN ZERO, IS IT ZERO?
7865
7866 016114 005767 177272      TST  CNTFLG
(9) 016120 001041      BNE  50025$
7867
7868      , IF ZERO, BYTE IS ALSO SECOND BYTE OF COUNT (6 HIGH ORDER BITS)
7869      , GET IT, MASK IT AND PUT IT INTO RTEMP2 (EXPECTED COUNT)
7870
7871 016122 010146      MOV  R1,-(SP)
7872 016124 117701 177256      MOVB  @RTEMP1,R1
7873 016130 005267 177252      INC  RTEMP1
7874 016134 000301      SWAB  R1
7875 016136 042701 140377      BIC  #140377,R1
7876 016142 060167 177242      ADD  R1,RTEMP2
7877 016146 012601      MOV  (SP)+,R1
7878
7879      , PUT THE COMPLETE SIZE IN THE QUEUE (EXPECTED COUNT +
7880      , 12 (OCTAL) BYTES OF HEADER AND CRC)
7881
7882 016150 016777 177234 177240      MOV  RTEMP2,@RQUE
7883 016156 062777 000012 177232      ADD  #12,@RQUE
7884
7885      , ADJUST EXPECTED COUNT TO INCLUDE THE REMAINDER OF THE HEADER AND
7886      , THE DATA CRC
7887
7888 016164 062767 000007 177216      ADD  #7,RTEMP2
7889
7890      , UPDATE THE QUEUE AND DECREMENT THE COUNTER FLAG (COUNT COMPLETE)
7891
7892 016172 062767 000002 177216      ADD  #2,RQUE
7893 016200 026727 177212 015432      CMP  RQUE,#RQUEB
(9) 016206 003403      BLE  50026$
7894 016210 012767 015420 177200      MOV  #RQUET,RQUE
7895 016216      50026$
7896 016216 005367 177170      DEC  CNTFLG
7897

```

```

7898      GO DO AN RTI
7899
7900 016222 000414      BR      50027$
(3) 016224      50025$.
7901
7902      COUNTER FLAG IS LESS THAN ZERO . THIS IS A DATA BYTE
7903      UPDATE POINTER AND COUNT
7904
7905 016224 005267 177156      INC      RTEMP1
7906 016230 005367 177154      DEC      RTEMP2
7907
7908      IF COUNT HAS REACHED ZERO, END CF MESSAGE SET DONE FLAG
7909 016234 005767 177150      TST      RTEMP2
(9) 016240 001005      BNE      50030$
7910 016242 005267 176536      INC      RDONE
7919
7920      ABORT ANY WAITS AND RESET NEW MESSAGE FLAG
7921
7922 016246 104021      EMT      CSABRT
7923 016250 005067 177130      CLR      NMSG
7924 016254      50030$.
7925 016254      50027$.
7926 016254      50024$.
7927 016254      50022$.
7928
7929      DO RTI
7930
7931 016254 012601      MOV      (SP)+,P1
7932 016256      L10030
(2) 016256 000002      RTI
7933
7934
7935
7936
7937      ++++++
7938      ROUTINE TO GET AN INPUT BUFFER FROM THE INPUT QUEUES      +
7939      WILL RETURN AN ERROR IF TERMINAL DOES NOT RESPOND      +
7940      ARGUMENT (IF SET) INHIBITS ERROR PRINTOUT THIS ALLOWS      +
7941      THE PROGRAM TO RETPY THE STRT-STACK SEQUENCE WITHOUT HAVING      +
7942      AN ERROR MESSAGE EACH TIME      +
7943      ++++++
7944
7945      RCVE
7946
7947      RESET TIMER
7948 016260
7949
7950      MOV      0,TIMER
7951
7952 016260 016767 161514 177100      GO TO SUPERVISOR FOR AWHILE AND INCREMENT TIMER
7953
7954      50002$.
7955
7956 016266      MOV      #100,R0
7957 016266 012700 000100      EMT      CSWTU
(3) 016272 104027
  
```

```

7958 016274 005267 177066          INC    TIMER
7959
7960          LOOP UNTIL TIME LIMIT EXCEEDED OR RECEIVER DONE FLAG IS SET
7961
7962 016300 026767 177062 163602    CMP    TIMER,TIMLIM
      (6) 016306 003003              BGT    50003$
      (4) 016310 005767 176470      TST    RDONE
      (7) 016314 001764              BEQ    50002$
      (4) 016316                    50003$
7963
7964          RESET THE NEW MESSAGE FLAG
7965
7966 016316 005067 177062          CLR    NMSG
7967
7968          IF THE TIME LIMIT WASN'T EXCEEDED
7969
7970 016322 026767 177040 163560    CMP    TIMER,TIMLIM
      (9) 016330 002044              BGE    50004$
7971
7972          RESET THE TIMER
7973          CHECK FOR ERROR (RDONE LESS THAN ZERO)
7974
7975 016332 005067 177030          CLR    TIMER
7976 016336 005767 176442          TST    RDONE
      (9) 016342 003426              BLE    50005$
7977
7978          NOT ERROR, GET BUFFER ADDRESS AND SIZE FROM QUEUE AND UPDATE QUEUE
7979
7980 016344 017767 177044 163574    MOV    @RQUEUE,INBUF
7981 016352 062767 000002 177034    ADD    #2,RQUEUE
7982 016360 017767 177030 163562    MOV    @RQUEUE,INSIZ
7983 016366 062767 000002 177020    ADD    #2,RQUEUE
7984 016374 026727 177014 015432    CMP    RQUEUE,#RQUEB
      (9) 016402 003403              BLE    50006$
7985 016404 012767 015420 177002    MOV    #RQUET,RQUEUE
7986 016412                    50006$
7987
7988          RESET DONE FLAG AND LEAVE
7989
7990 016412 005067 176366          CLR    RDONE
7991 016416 000410              BR     50007$
      (3) 016420                    50005$
7992
7993          ERROR DID OCCUR  RESET THE DONE FLAG  REPORT EPROP. RETURN WITH ERROR
7994
7995 016420 005067 176360          CLR    RDONE
7996 016424 104463              TRAP    T$ERCODE
      (5) 016426 000032              WORD    26
      (5) 016430 003100              WORD    MSG26
      (5) 016432 004302              WORD    AREA26
7997 016434 000261              SEC
      (4) 016436 000443              BR     50001$
7998 016440                    50007$
7999 016440 000441              BR     50010$
      (3) 016442                    50004$
8000

```

8001 TIME LIMIT EXCEEDED. RESET TIMER, GET REGISTER CONTENTS

8002					
8003	016442	005067	176720		CLR TIMER
8004	016446	010146			MOV R1, -(SP)
8005	016450	016701	176710		MOV RCVADD, R1
(6)	016454	162701	000002		SUB #2, R1
8006	016460	010246			MOV R2, -(SP)
8011	016462	012702	002242		MOV #DZREG, R2
8013	016466	016162	000000	000000	MOV RCVCSR(R1), RCVCSR(R2)
8014	016474	016162	000002	000002	MOV RCVBUF(R1), RCVBUF(R2)
8015	016502	016162	000004	000004	MOV XMTCSR(R1), XMTCSR(R2)
8016	016510	016162	000006	000006	MOV XMTBUF(R1), XMTBUF(R2)
8017	016516	012602			MOV (SP)+, R2
8018	016520	012601			MOV (SP)+, R1

8019 IF NOT INHIBITED, PRINT ERROR MESSAGE

8020					
8021					
8022	016522	005765	000000		TST DDSTRT(R5)
(9)	016526	001004			BNE 50011\$
8023	016530	104463			TRAP T\$ERCODE
(5)	016532	000037			WORD 31
(5)	016534	003165			WORD MSG31
(5)	016536	004704			WORD AREA31

50011\$

8024 RETURN WITH ERROR IN ANY CASE

8025					
8026					
8027					
8028	016540	000261			SEC
(4)	016542	000401			BR 50001\$
8029	016544				50010\$
8030	016544				50000\$
(2)	016544	000241			CLC
(3)	016546				50001\$
(2)	016546	000207			RTS PC

8031 ++++++

8032 DUMMY ROUTINE FOR COMPATIBILITY

8033 ++++++

8034					
8035					
8037					
8038					
8039					
8040					
8041					
8042	016550				INSERR
8043	016550				50000\$
(3)	016550				50001\$
(2)	016550	000207			RTS PC

8106					
8107					
8108					
8109					
8110					
8111					
8112					
8113					

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 188-7
 DZDZCA P11 21-NOV-77 14 55

SEQ 0056

```

8114      ,      THIS ROUTINE COMPUTES CRC-16 AND REQUIRES THREE ARGUMENTS      +
8115      ,      DATA ADDRESS, DATA SIZE (IN BYTES) AND A WORD LOCATION      +
8116      ,      FOR THE COMPUTED CRC      +
8117      ,      3 2 1 2 1, 3.2 1 2 4, 3 2.20 1 2, 3.2 20 3 1,      +
8118      ,      3 2 20 6 2, 3.2 20 6 4 AND 3 2 20 10 1      +
8119      ,      +
8120      ,      THIS IS A STANDARD ROUTINE FROM DEC'S NETWORK GROUP      +
8121      ,      +
8122      ,      ++++++
8123      CRC
8126      016552      010446      MOV      R4, -(SP)
      (2) 016554      010604      MOV      SP, R4
      (3) 016556      162706      000004      SUB      #4, SP
8127      016562      010146      MOV      R1, -(SP)
8128      016564      010246      MOV      R2, -(SP)
8129      016566      010346      MOV      R3, -(SP)
8130      016570      016501      000000      MOV      A(R5), R1
8131      016574      016502      000002      MOV      B(R5), R2
8132      016600      005003      CLR      R3
8133      016602      50002$      TST      R2
      (6) 016602      005702      BLE      50003$
      (9) 016604      003500      MOV8     (R1)+, D(R4)
8134      016606      112164      177776      BIC      #HYBYTE, D(R4)
8135      016612      042764      177400      177776      DEC      R2
8136      016620      005302      MOV      R3, -(SP)
8137      016622      010346      BIC      D(R4), (SP)
      (6) 016624      046416      177776      BIC      R3, D(R4)
      (6) 016630      040364      177776      BIS      (SP)+, D(R4)
      (6) 016634      052664      177776      MOV      D(R4), E(R4)
8138      016640      016464      177776      177774      BIC      #177760, D(R4)
8139      016646      042764      177760      177776      ASL      D(R4)
8140      016654      006364      177776      ADD      #CRCTAB, D(R4)
8141      016660      062764      017026      177776      MOV      @D(R4), D(R4)
8142      016666      017464      177776      177776      BIC      #177417, E(P4)
8143      016674      042764      177417      177774      ASR      E(R4)
8144      016702      006264      177774      ASR      E(R4)
      (7) 016706      006264      177774      ASR      E(R4)
      (7) 016712      006264      177774      ADD      #CRCTAB, E(R4)
8145      016716      062764      017026      177774      ADD      #32, E(R4)
8146      016724      062764      000040      177774      MOV      @E(R4), E(R4)
8147      016732      017464      177774      177774      MOV      E(R4), -(SP)
8148      016740      016446      177774      BIC      D(R4), (SP)
      (6) 016744      046416      177776      BIC      E(R4), D(R4)
      (6) 016750      046464      177774      177776      BIS      (SP)+, D(R4)
      (6) 016756      052664      177776      BIC      #LOBYTE, R3
8149      016762      042703      000377      SWAB     R3
8150      016766      000303      MOV      D(R4), -(SP)
8151      016770      016446      177776      BIC      R3, (SP)
      (6) 016774      040316      BIC      D(R4), R3
      (6) 016776      046403      177776      BIS      (SP)+, R3
      (6) 017002      052603      BR      50002$
8152      017004      000676      50003$      MOV      R3, C(R5)
      (3) 017006      010365      000004      MOV      (SP)+, R3
8153      017006      010365      000004      MOV      (SP)+, P2
8154      017012      012603
8155      017014      012602

```

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 188-8
 DZDZCA P11 21-NOV-77 14 55

SEQ 0057

8156	017016	012601		MOV	(SP)+, R1
8157	017020		500005		
(3)	017020		500015		
(3)	017020	010406		MOV	R4, SP
(3)	017022	012604		MOV	(SP)+, R4
(2)	017024	000207		RTS	PC
8158					
8159	017026	000000	CRCTAB	000000	
8160	017030	140301		140301	
8161	017032	140601		140601	
8162	017034	000500		000500	
8163	017036	141401		141401	
8164	017040	001700		001700	
8165	017042	001200		001200	
8166	017044	141101		141101	
8167	017046	143001		143001	
8168	017050	003300		003300	
8169	017052	003600		003600	
8170	017054	143501		143501	
8171	017056	002400		002400	
8172	017060	142701		142701	
8173	017062	142201		142201	
8174	017064	002100		002100	
8175	017066	000000		000000	
8176	017070	146001		146001	
8177	017072	154001		154001	
8178	017074	012000		012000	
8179	017076	170001		170001	
8180	017100	036000		036000	
8181	017102	024000		024000	
8182	017104	162001		162001	
8183	017106	120001		120001	
8184	017110	066000		066000	
8185	017112	074000		074000	
8186	017114	132001		132001	
8187	017116	050000		050000	
8188	017120	116001		116001	
8189	017122	104001		104001	
8190	017124	042000		042000	

```

8194
8195
8196
8197
8199
8200
8201
8202
8203
8205
8206
8207
8208 017126 005067 176242
8209
8210
8211
8212 017132 016767 000742 162772
8213 017140 042767 100001 162764
8214 017146 016767 000730 162760
8215
8216
8217 017154 004767 173144
8218
8219
8220
8221 017160 103015
8222
8223
8224
8225
8226
8227 017162 010546
8228 017164 012745 000000
8229 017170 004767 177354
8230 017174 012605
8231 017176 104463
8232 017200 000062
8233 017202 003307
8234 017204 004110
8235 017206 104006
8236 017210 104032
8237 017212 000364
8238 017214
8239
8240
8241 017230 103015
  
```

```

/
/      HARDWARE TESTS 3 2
/
/
/
/
/      SELF TEST 3 2C
/
/
/
/
/      FORCE INIT OF COMM DEVICE AND LINK
/
/      CLR      INIT
/
/      GET MESSAGE ADDRESS AND SIZE
/
/      MOV      MDMSG,OUTMSG
/      BIC      #100001,OUTMSG
/      MOV      MDMSG+2,OUTMSG+2
/
/      SEND THE MESSAGE
/
/      JSR      PC,TEROUT
/
/      F ERROR
/
/      BCC      50004$
/
/      INSERT ERROR INTO STATISTICS TABLE (MANUFACTURING VERSIONS ONLY),
/      REPORT THE ERROR, REQUEST SUPERVISOR TO LOOP
/      IF LOOPING ENABLED (EXIT TEST IF NO LOOPING)
/
/      MOV      R5,-(SP)
/      MOV      #0,-(R5)
/      JSR      PC,INSERT
/      MOV      (SP)+,R5
/      TRAP     T$ERRCODE
/      WORD     50
/      WORD     MSG50
/      WORD     APEA40
/      ENT      C$CLP1
/      ENT      C$EXIT
/      WORD     L10031-
/
/      50004$
/
/      LOOK FOR ACK
/
/      SUB      #1,R5
/      JSR      PC,TERIN
/      MOV      (R5)+,MCOMP
/
/      IF ERROR INSERT ERROR INTO TABLE (MANUFACTURING TEST ONLY),
/      REPORT THE ERROR, REQUEST LOOP IF ENABLED EXIT TEST
/      IF NO LOOPING)
/
/      BCC      50005$
  
```

8242 017232 010546
 (4) 017234 012745 000000
 (3) 017240 004767 177304
 (3) 017244 012605
 8243 017246 104463
 (5) 017250 000062
 (5) 017252 003307
 (5) 017254 004110
 8244 017256 104006
 8245 017260 104032
 (3) 017262 000314
 8246 017264

MOV R5, -(SP)
 MOV #0, -(R5)
 JSR PC, INSERR
 MOV (SP)+, R5
 TRAP T\$ERCODE
 WORD 50
 WORD MSG50
 WORD AREA40
 EMT C\$CLP1
 EMT C\$EXIT
 WORD L10031-

50005\$

IF MESSAGE WAS NOT AN ACK ERROR
 REPORT ERROR, LOOP IF ENABLED, EXIT TEST IF NOT

8251 017264 005767 000606
 (9) 017270 001407
 8252 017272 104463
 (5) 017274 000062
 (5) 017276 003307
 (5) 017300 004110
 8253 017302 104006
 8254 017304 104032
 (3) 017306 000270
 8255 017310

TST MCOMP
 BEQ 50006\$
 TRAP T\$ERCODE
 WORD 50
 WORD MSG50
 WORD AREA40
 EMT C\$CLP1
 EMT C\$EXIT
 WORD L10031-

50006\$

FORCE A RESTART ON THE LINK AND WAIT FOR SELFTEST TO COMPLETE

8259 017310 005067 176060
 8260 017314 012700 000400
 (3) 017320 104026

CLR INIT
 MOV #400, R0
 EMT C\$WTH

SET UP MESSAGE ASKING FOR RESULTS AND SEND IT

8264 017322 016767 000562 162602
 (6) 017330 042767 100001 162574
 8265 017336 016767 000550 162570
 8266 017344 004767 172754

MOV MDMSG1, OUTMSG
 BIC #100001, OUTMSG
 MOV MDMSG1+2, OUTMSG+2
 JSR PC, TEROUT

IF ERROR INSERT ERROR IN TABLE (MANUF TESTS ONLY) REPORT ERROR,
 LOOP IF ENABLED, EXIT TEST IF NOT LOOPING

8271 017350 103015
 8272 017352 010546
 (4) 017354 012745 000000
 (3) 017360 004767 177164
 (3) 017364 012605
 8273 017366 104463
 (5) 017370 000062
 (5) 017372 003307
 (5) 017374 004110
 8274 017376 104006
 8275 017400 104032
 (3) 017402 000174
 8276 017404

BCC 50007\$
 MOV R5, -(SP)
 MOV #0, -(R5)
 JSR PC, INSERR
 MOV (SP)+, R5
 TRAP T\$ERCODE
 WORD 50
 WORD MSG50
 WORD AREA40
 EMT C\$CLP1
 EMT C\$EXIT
 WORD L10031-

50007\$

8277				
8278				ACK TERMINAL AND GET ITS RESPONSE
8279				
8280	017404			50010\$
8281	017404	004767	000170	JSR PC,OUTREP
8282				
8283				HANDLE ERROR IF ONE OCCURED WHILE ACK'ING
8284				
8285	017410	103023		BCC 50011\$
8286	017412	010546		MOV R5,-(SP)
(4)	017414	012745	000000	MOV #0,-(R5)
(3)	017420	004767	177124	JSR PC,INSERR
(2)	017424	012605		MOV (SP)+,R5
8287	017426	010546		MOV R5,-(SP)
(4)	017430	012745	000001	MOV #1,-(R5)
(3)	017434	004767	177110	JSR PC,INSERR
(2)	017440	012605		MOV (SP)+,R5
8288	017442	104463		TRAP T\$ERCODE
(5)	017444	000062		WORD 50
(5)	017446	003307		WORD MSG50
(5)	017450	004110		WORD AREA40
8289	017452	104006		EMT C\$CLP1
8290	017454	104032		EMT C\$EXIT
(3)	017456	000120		WORD L10031-
8291	017460			50011\$
8292				
8293				GET THE RESPONSE
8294				
8295	017460	162705	000002	SUB #1&2,R5
(3)	017464	004767	173770	JSR PC,TEPIN
(4)	017470	012567	000402	MOV (R5)+,MCOMP
8296				
8297				HANDLE ERROR IF ONE OCCURED
8298				
8299	017474	103015		BCC 50012\$
8300	017476	010546		MOV R5,-(SP)
(4)	017500	012745	000000	MOV #0,-(R5)
(3)	017504	004767	177040	JSR PC,INSERR
(2)	017510	012605		MOV (SP)+,R5
8301	017512	104463		TRAP T\$ERCODE
(5)	017514	000062		WORD 50
(5)	017516	003307		WORD MSG50
(5)	017520	004110		WORD AREA40
8302	017522	104006		EMT C\$CLP1
8303	017524	104032		EMT C\$EXIT
(3)	017526	000050		WORD L10031-
8304	017530			50012\$
8305				
8306				REPEAT THE LOOP UNTIL A MESSAGE IS RECEIVED
8307				
8308	017530	005767	000342	TST MCOMP
(6)	017534	001723		BEQ 50010\$
8309				
8310				PROCESS THE MESSAGE RECEIVED
8311				
8312	017536	004767	000146	JSR PC,STINPT

```

8313
8314      IF IN ERROR  INSERT ERROR INTO TABLE (MANUF TESTS ONLY, REPORT
8315      ERROR, LOOP IF ENABLED, EXIT TEST IF NOT
8316
8317      017542  103015      BCC      50013$
8318      017544  010546      MOV      R5, -(SP)
      (4) 017546  012745  000001      MOV      #1, -(R5)
      (3) 017552  004767  176772      JSR      PC, INSERR
      (3) 017556  012605      MOV      (SP)+, R5
8319      017560  104463      TRAP     T$ERCODE
      (5) 017562  000062      WORD     50
      (5) 017564  003307      WORD     MSG50
      (5) 017566  004110      WORD     AREA40
8320      017570  104006      EMT      ($CLP1
8321      017572  104032      EMT      ($EXIT
      (3) 017574  000002      WORD     L10031-
8322      017576      50013$
8323
8324      END OF TEST
8325
8326      017576  104001      L10031  EMT      ($SETST
      (3) 017576
8327
8328
8329
8330
8331
8332      ++++++
8333      ROUTINE TO SEND AND ACK IN ORDER TO POLL A TERMINAL
8334      ++++++
8335
8336      OUTREP
8337      017600
8338
8339      INSERT MESSAGE NUMBER (LAST MESSAGE SENT) AND RESPONSE NUMBER
8340      (LAST MESSAGE RECEIVED)  INSERT STATION ADDRESS
8341
8342      017600  116767  162360  000316      MOV      MSGNO, MSGN
8343      017606  116767  162354  000307      MOV      RMSGNO, REPN
8344      017614  116767  163703  000303      MOV      STATSA, REPA
8345
8346      COMPUTE AND INSERT CRC
8347
8348      017622  162705  000002      SUB      #1&2, R5
      (3) 017626  010546      MOV      R5, -(SP)
      (5) 017630  012745  000006      MOV      #6, -(R5)
      (4) 017634  012745  020120      MOV      #REPH, -(R5)
      (3) 017640  004767  176706      JSR      PC, CRC
      (3) 017644  012605      MOV      (SP)+, R5
      (4) 017646  012567  000254      MOV      (R5)+, REPCRC
8349
8350      SET UP BUFFER POINTER AND SIZE AND SEND THE MESSAGE
8351
8352      017652  012767  020120  175520      MOV      #REPH, YBUFFER
8353      017660  012767  000010  175514      MOV      #10, XSIZE
8354      017666  004767  174600      JSR      PC, XMIT
  
```

```

8355
8356      IF AN ERROR OCCURRED, RETURN WITH ERROR OTHERWISE RETURN NORMALLY
8357
8358      017672 103004
8359      017674 005267 162220
8360      017700 000261
      (4) 017702 000401
8361      017704 500025
8362      017704 500005
      (2) 017704 000241
      (3) 017706 500015
      (2) 017706 000207
8363
8364
8365
8366
8367
8368      ++++++
8369      ROUTINE TO PROCESS TERMINAL SELFTEST RESULTS
8370      ++++++
8371      STINPT
8372
8373      017710      MOV      R1, -(SP)
8374      017710 010146      MOV      R2, -(SP)
8375      017712 010246      MOV      R3, -(SP)
8376      017714 010346
8377
8378      SKIP RESPONSE HEADER AND CHECK FIRST DATA BYTE
8379
8380      017716 016701 162224      MOV      INBUF, R1
      (6) 017722 062701 000004      ADD      #4, R1
8381
8382      IF IT IS "A", GOOD RESPONSE OTHERWISE
8383
8384      017726 121127 000101      CMPB      (R1), #101
      (9) 017732 001454      BEQ      500025
8385
8386      GET THE BYTE AND CHECK TO SEE IF IT IS "M" (MEMORY ERROR)
8387
8388      017734 112103      MOVB      (R1)+, R3
8389      017736 120327 000115      CMPB      R3, #115
      (9) 017742 001025      BNE      500035
8390
8391      MEMORY ERROR GET THE TYPE AND BANK NUMBER
8392      INSERT INTO ERROR MESSAGE, REPORT THE ERROR AND
8393      RETURN WITH ERROR
8394      (NOTE ERROR REPORTING PROTOCOL DELIMITERS ARE
8395      IGNORED THIS PROGRAM WILL NOT ACCEPT INPUT FROM
8396      ANY OTHER INTELLIGENT TERMINAL PROPERLY)
8397
8398      017744 062701 000002      ADD      #2, R1
8399      017750 012702 003464      MOV      #MTY, R2
8400      017754 112122      MOVB      (R1)+, (R2)+
8401      017756 112122      MOVB      (R1)+, (R2)+
8402      017760 012702 003500      MOV      #BNO, R2
8403      017764 005201      INC      R1
  
```

8404	017766	111103		MOVB	(R1), R3
8405	017770	062703	000060	ADD	#60, R3
8406	017774	110312		MOVB	R3, (R2)
8407	017776	104423		TRAP	TSERCODE
(5)	020000	000107		WORD	71
(5)	020002	003440		WORD	MSG71
8408	020004	012603		MOV	(SP)+, R3
8409	020006	012602		MOV	(SP)+, R2
8410	020010	012601		MOV	(SP)+, R1
8411	020012	000261		SEC	
(4)	020014	000427		BR	500015
8412	020016		500035		
8413					
8414				NOT A MEMORY ERROR OR PASS	CHECK FOR "P" (PROCESSOR ERROR)
8415					
8416	020016	120327	000120	CMPB	R3, #120
(9)	020022	001010		BNE	500055
8417					
8418				PROCESSOR ERROR	REPORT IT AND EXIT WITH ERROR
8419					
8420	020024	104423		TRAP	TSERCODE
(5)	020026	000110		WORD	72
(5)	020030	003502		WORD	MSG72
8421	020032	012603		MOV	(SP)+, R3
8422	020034	012602		MOV	(SP)+, R2
8423	020036	012601		MOV	(SP)+, R1
8424	020040	000261		SEC	
(4)	020042	000414		BR	500015
8425	020044		500055		
8426					
8427				RESPONSE WAS NOT LEGAL	REPORT IT AND RETURN WITH ERROR
8428					
8429	020044	104423		TRAP	TSERCODE
(5)	020046	000106		WORD	70
(5)	020050	003405		WORD	MSG70
8430	020052	012603		MOV	(SP)+, R3
8431	020054	012602		MOV	(SP)+, R2
8432	020056	012601		MOV	(SP)+, R1
8433	020060	000261		SEC	
(4)	020062	000404		BR	500015
8434	020064		500065		
8435	020064		500045		
8436	020064		500025		
8437	020064	012603		MOV	(SP)+, R3
8438	020066	012602		MOV	(SP)+, R2
8439	020070	012601		MOV	(SP)+, R1
8440	020072		500005		
(2)	020072	000241		CLC	
(3)	020074		500015		
(2)	020074	000207		RTS	PC
8441					
8442					
8443				MESSAGE AREA FOR SELFTEST	
8444					
8445	020076	000000		MLCMP	0
8446	020100	020104		MDMSG	+4

8522	020214	012701	021016		MOV	#LOOPB,R1
8523	020220	005767	171232		TST	PATTERN
(9)	020224	001403			BEQ	50010\$
8524	020226	005067	161650		CLR	SCRCH
8525	020232	000403			BR	50011\$
(3)	020234			50010\$		
8526	020234	012767	125125	161640	MOV	#125125,SCRCH
8527	020242			50011\$		
8528	020242			50012\$		
8529	020242	016721	161634		MOV	SCRCH,(R1)+
8530	020246	020127	021416		CMP	R1,#LOOPB
(6)	020252	001373			BNE	50012\$

; PUT A 1 AT THE END TO ASSURE THAT THE FINAL BYTE IS NOT NULL (0)

8531						
8532						
8533						
8534	020254	012701	021416		MOV	#LOOPB,R1
(6)	020260	005301			DEC	R1
8535	020262	112711	000001		MOVB	#1,(R1)

; GET THE STATION ADDRESS AND INSERT IT IN THE HEADER

8538						
8539	020266	012701	002170		MOV	#PTABLE,R1
(6)	020272	062701	000005		ADD	#5,R1
8540	020276	111167	000511		MOVB	(R1),LOOSTA
8541	020302	012601			MOV	(SP)+,R1

; INSERT LOOP DATA TYPE

8542						
8543						
8544						
8545	020304	112767	000024	000504	MOVB	#OSOP,LOOPB

; SET MESSAGE NUMBER

8546						
8547						
8548						
8549	020312	112767	000001	000472	MOVB	#1,LOOPN

; COMPUTE AND INSERT HEADER CRC

8550						
8551						
8552						
8553	020320	162705	000002		SUB	#1*2,R5
(3)	020324	010546			MOV	R5,-(SP)
(5)	020326	012745	000006		MOV	#6,-(R5)
(4)	020332	012745	021006		MOV	#LOOPH,-(R5)
(3)	020336	004767	176210		JSR	PC,CRC
(3)	020342	012605			MOV	(SP)+,R5
(4)	020344	012567	000444		MOV	(R5)+,LCRCH

; COMPUTE AND INSERT DATA CRC

8554						
8555						
8556						
8557	020350	162705	000002		SUB	#1*2,R5
(3)	020354	010546			MOV	R5,-(SP)
(5)	020356	012745	000400		MOV	#256,-(R5)
(4)	020362	012745	021016		MOV	#LOOPB,-(R5)
(3)	020366	004767	176160		JSR	PC,CRC
(3)	020372	012605			MOV	(SP)+,R5
(4)	020374	012567	001016		MOV	(R5)+,LCRCD

SET UP BUFFER POINTER AND SIZE SEND MESSAGE

8558
8559
8560

```

8561 020400 012767 021006 174772      MOV    #LOOPH,XBUFFER
8562 020406 012767 000412 174766      MOV    #266,XSIZE
8563 020414 004767 174052      JSR    PC,XMIT
8564
8565      IF ERROR OCCURRED . INSERT THE ERROR INTO TABLE (MANUF TESTS ONLY),
8566      REPORT THE ERROR, LOOP IF ENABLED
8567
8568 020420 103012      BCC    50013$
8569 020422 010546      MOV    R5,-(SP)
      (4) 020424 012745 000000      MOV    #0,-(R5)
      (3) 020430 004767 176114      JSR    PC,INSERR
      (3) 020434 012605      MOV    (SP)+,R5
8570 020436 104463      TRAP   T$ERCODE
      (5) 020440 000063      WORD   51
      (5) 020442 003344      WORD   MSG51
      (5) 020444 004110      WORD   AREA40
8571 020446      50013$
8572 020446 104006      EMT     C$CLP1
8573
8574      SEND ACK'S TO TEFMINAL UNTIL IT RESPONDS
8575
8576 020450      50014$
8577 020450 004767 177124      JSR    PC,OUTREP
8578
8579      IF AN ERROR OCCURS . INSERT IN TABLE (MAN ONLY), REPORT IT, LOOP
8580      IF ENABLED - EXIT TEST IF NOT
8581
8582 020454 103023      BCC    50015$
8583 020456 010546      MOV    R5,-(SP)
      (4) 020460 012745 000000      MOV    #0,-(R5)
      (3) 020464 004767 176060      JSR    PC,INSERR
      (3) 020470 012605      MOV    (SP)+,R5
8584 020472 010546      MOV    R5,-(SP)
      (4) 020474 012745 000002      MOV    #2,-(R5)
      (3) 020500 004767 176044      JSR    PC,INSERR
      (3) 020504 012605      MOV    (SP)+,R5
8585 020506 104463      TRAP   T$ERCODE
      (5) 020510 000063      WORD   51
      (5) 020512 003344      WORD   MSG51
      (5) 020514 004110      WORD   AREA40
8586 020516 104006      EMT     C$CLP1
8587 020520 104032      EMT     C$EXIT
      (3) 020522 000134      WORD   L10032-
8588 020524      50015$
8589
8590      GET RESPONSE
8591
8592 020524 162705 000002      SUB    #1*2,R5
      (3) 020530 004767 172724      JSR    PC,TERIN
      (4) 020534 012567 000240      MOV    (R5)+,LCOMP
8593
8594      IF ERROR OCCURS . INSERT IN TABLE (MAN ONLY) REPORT IT. LOOP IF
8595      ENABLED. EXIT TEST IF NOT
8596
8597 020540 103012      BCC    50016$
8598 020542 010546      MOV    R5,-(SP)

```

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 189-9
 DZDZCA P11 21-NOV-77 14 55

SEQ 0067

```

(4) 020544 012745 000000      MOV      #0, -(R5)
(3) 020550 004767 175774      JSR      PC, INSERR
(3) 020554 012605              MOV      (SP)+, R5
8599 020556 104463              TRAP     T$ERCODE
(5) 020560 000063              WORD     51
(5) 020562 003344              WORD     MSG51
(5) 020564 004110              WORD     AREA40
8600 020566                    50016$ EMT      C$CLP1
8601 020566 104006
8602
8603      REPEAT LOOP UNTIL TERMINAL SENDS LOOPBACK DATA
8604
8605 020570 005767 000204      TST      LCOMP
(6) 020574 001725              BEQ      50014$
8606
8607      PROCESS THE LOOPBACK DATA
8608
8609 020576 004767 000056      JSR      PC, PROLOP
8610 020602 104004              EMT      C$BSEG
8611
8612      REINIT THE LINK
8613
8614 020604 004767 171750      JSR      PC, XSTRT
8615
8616      IF ERROR OCCURS, INSERT ERROR INTO TABLE (MANUF. ONLY), REPORT ERROR,
8617      LOOP ON REINIT IF ERROR LOOPING ENABLED  EXIT TEST IF NO LOOPING
8618
8619 020610 103020              BCC      50017$
8620 020612 010546              MOV      R5, -(SP)
(4) 020614 012745 000000      MOV      #0, -(R5)
(3) 020620 004767 175724      JSR      PC, INSERR
(3) 020624 012605              MOV      (SP)+, R5
8621 020626 010546              MOV      R5, -(SP)
(4) 020630 012745 000002      MOV      #2, -(R5)
(3) 020634 004767 175710      JSR      PC, INSERR
(3) 020640 012605              MOV      (SP)+, R5
8622 020642 104463              TRAP     T$ERCODE
(5) 020644 000063              WORD     51
(5) 020646 003344              WORD     MSG51
(5) 020650 004622              WORD     AREA61
8623 020652                    50017$ EMT      C$CLP1
8624 020652 104006
8625 020654                    10000$ EMT      C$ESEG
(3) 020654 104005
8626
8627      END OF TEST
8628
8629 020656                    L10032 EMT      C$ETST
(3) 020656 104001
8630
8631
8632
8633
8634
8635      ++++++
8636

```

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 189-10
 DZDZCA P11 21-NOV-77 14 55

SEQ 0068

```

8637          ROUTINE TO PROCESS LOOPBACK DATA
8638
8639          ++++++
8640 020660      PROLOP
8641 020660      MOV      R1, -(SP)
8642 020662      MOV      R2, -(SP)
8643 020664      MOV      R3, -(SP)
8644
8645          R1 = POINTER TO EXPECTED DATA
8646          R2 = COUNT OF DATA EXPECTED
8647          R3 = POINTER TO DATA FROM TERMINAL
8648
8649 020666      MOV      #LOOPB, R1
8650 020672      MOV      #128, R2
8651 020676      MOV      INBUF, R3
8652
8653          CHECK DATA BYTE-BY-BYTE
8654
8655 020702      500025
8656
8657          IF ERROR, INSERT ERROR INTO TABLE (MANUF. ONLY), REPORT ERROR
8658          (INCLUDING RECEIVED/EXPECTED BYTES), RETURN WITH ERROR
8659
8660 020702      CMPB      (R1), (R3)
8661 (9) 020704      BEQ      500035
8662 020706      MOV      (R1), A52A
8663 020712      MOV      (R3), A52B
8664 020716      MOV      R5, -(SP)
8665 (4) 020720      MOV      #2, -(R5)
8666 (3) 020724      JSR      PC, INSERT
8667 (3) 020730      MOV      (SP)+, R5
8668 020732      TRAP      T5ERCODE
8669 (5) 020734      WORD      52
8670 (5) 020736      WORD      MSG52
8671 (5) 020740      WORD      AREA52
8672 020742      MOV      (SP)+, R3
8673 020744      MOV      (SP)+, R2
8674 020746      MOV      (SP)+, R1
8675 020750      SEC
8676 (4) 020752      BR      500015
8677 020754      500035
8678
8679          UPDATE COUNT AND POINTERS
8680
8681          INC      R1
8682          DEC      R2
8683          INC      R3
8684
8685          REPEAT UNTIL ALL DATA HAS BEEN CHECKED
8686
8687          TST      R2
8688 (6) 020764      BNE      500025
8689 020766      MOV      (SP)+, R3
8690 020770      MOV      (SP)+, R2
8691 020772      MOV      (SP)+, R1
8692 020774      500005

```

```
(2) 020774 000241          CLC
(3) 020776          50001$
(2) 020776 000207          RTS    PC
8684
8685
8686          / LOOPBACK MESSAGE AREA
8687          /
8688 021000 000000          LCOMP  0
8689 021002 121007          LOOPMG +100005
8690 021004 000410          ELOOPB-2-
8691 021006 220          000 201 LOOPH  BYTE DLE, 0, 201, 0
      021011 000
8692 021012 000          LOOPN  BYTE 0
8693 021013 000          LOOSTA  BYTE  0
8694 021014 000000          LCRCH  0
8695 021016 000200          LOOPB  BLKW  128
8696 021416
8697 021416 000000          ELLOOP
8698          LCRCD  0
```

8954						
8955						HARDWARE PARAMETER CODING
8956						
8957	021420	000041				WORD L10033-LSHARD/2
8958	021422	000031				WORD TSCODE
(4)	021424	021524				WORD DEVICE
(4)	021426	000000				WORD TSLOLIM
(4)	021430	177776				WORD TSHILIM
8959	021432	001031				WORD TSCODE
(4)	021434	021541				WORD VECTOR
(4)	021436	000000				WORD TSLOLIM
(4)	021440	000400				WORD TSHILIM
8960	021442	002032				WORD TSCODE
(4)	021444	021555				WORD PRIOR
(4)	021446	000340				WORD 340
(4)	021450	000000				WORD TSLOLIM
(4)	021452	000007				WORD TSHILIM
8961	021454	002032				WORD TSCODE
(4)	021456	021577				WORD STATION
(4)	021460	177400				WORD 177400
(4)	021462	000000				WORD TSLOLIM
(4)	021464	000377				WORD TSHILIM
8966	021466	003032				WORD TSCODE
(4)	021470	021622				WORD LINE
(4)	021472	000007				WORD 7
(4)	021474	000000				WORD TSLOLIM
(4)	021476	000007				WORD TSHILIM
8969	021500	003032				WORD TSCODE
(4)	021502	021640				WORD BAUD
(4)	021504	007400				WORD 7400
(4)	021506	000000				WORD TSLOLIM
(4)	021510	000017				WORD TSHILIM
8970	021512	003032				WORD TSCODE
(4)	021514	021654				WORD FLAGZ
(4)	021516	000070				WORD 70
(4)	021520	000000				WORD TSLOLIM
(4)	021522	000007				WORD TSHILIM
8972						EVEN
(4)	021524					
8973						
8980	021524	055104	040440	042104	DEVICE	ASCIZ '02 ADDRESS
	021532	042522	051523	020072		
	021540	000				
8981	021541	104	020132	042526	VECTOR	ASCIZ '02 VECTOR
	021546	052103	051117	020072		
	021554	000				
8982	021555	104	020132	052502	PRIOR	ASCIZ '02 BUS PRIORITY
	021562	020123	051120	047511		
	021570	044522	054524	020072		
	021576	000				
8990	021577	124	051105	044515	STATION	ASCIZ 'TERMINAL ADDRESS
	021604	040516	020114	042101		
	021612	051104	051505	035123		
	021620	000340				
8992	021622	044514	042516	047040	LINE	ASCIZ 'LINE NUMBER
	021630	046525	042502	035122		

8995	021636	000040				
	021640	040502	042125	051040	BAUD	ASCIZ /BAUD RATE /
	021646	052101	035105	000040		
8996						
8997						BAUD RATE SELECTION.
8998					TYPE	TO GET BAUD RATE
8999						
9000						BAUD RATE SELECTION
9001					TYPE	TO GET THIS RATE
9002					0	50
9003					1	75
9004					2	110
9005					3	134 5
9006					4	150
9007					5	300
9008					6	600
9009					7	1200
9010					10	1800
9011					11	2000
9012					12	2400
9013					13	3600
9014					14	4800
9015					15	7200
9016					16	9600
9017					17	19,200

9018						
9019	021654	052123	050117	041440	FLAGZ	ASCIZ @STOP CODE/CHAR LENGTH (SEE DOC . 2)
	021662	042117	027505	044103		
	021670	051101	020056	042514		
	021676	043516	044124	024040		
	021704	042523	020105	047504		
	021712	027103	035051	000040		

9020						
9021						ALLOWABLE RESPONSES
9022					0 =	1 STOP BIT - 5 BIT CHAR
9023					1 =	" " " "
9024					2 =	" " " "
9025					3 =	" " " "
9026					4 =	2 STOP BIT - 5 " "
9027					5 =	" " " "
9028					6 =	" " " "
9029					7 =	" " " "

9030						
9032						EVEN

9033						
9034						
9035						SOFTWARE PARAMETER CODING
9036						
9037	021720	000010			WORD	L10034-L5SCFT 2
9038	021722	000032			WORD	TSCODE
(4)	021724	021742			WORD	ALIMIT
(4)	021726	177777			WORD	177777
(4)	021730	000000			WORD	TSLOLIM
(4)	021732	077777			WORD	TSHILIM
9039	021734	001130			WORD	TSCODE
(4)	021736	022011			WORD	PATRN

OUTPR MACY11 30(1046) 22-NOV-77 07 50 PAGE 190-2
DZDZCA P11 21-NOV-77 14 55

H 6

SEQ 0072

```
(4) 021740 177777 WORD 177777
9043      EVEN
(3) 021742      L10034
9044 021742 047510 020127 040515 ALIM T ASCIZ /HOW MANY INITIATE TRIES BEFORE ABORT? /
      021750 054516 044440 044516
      021756 044524 052101 020105
      021764 051124 042511 020123
      021772 042502 047506 042522
      022000 040440 047502 052122
      022006 020077 000
9045 022011 125 042523 040440 PATTPN ACC 2 'USE ALL ZERO'S FOR LOOPBACK DATA' /
      022016 046114 055040 051105
      022024 023517 020123 047506
      022032 020122 047514 050117
      022040 040502 045503 042040
      022046 052101 037501 000040

9049      EVEN
9051      EVEN
(3) 022054      LSLAST
9052      REQ DOCTOR P11 400.1066
9053      TITLE PDP-11 DIAGNOSTIC SUPERVISOR
20349      052754      END SUPV= +2
20351      071776      =71776
20352 071776 000000      WORD 0
20354      072000      YIXI=
20355      000200      END 200
```

A = 000000	BIT05 = 000040 G	CNVT 045566	CSQ10 = 000377	EF11 = 000013 G
ABOFLA 022326 G	BIT06 = 000100 G	CODE 011466 G	CSRDBU= 000007	EF12 = 000014 G
ABOPAS 022240 G	BIT07 = 000200 G	COMMTA 045406	CSREFG= 000050	EF13 = 000015 G
ABORT 002106	BIT08 = 000400 G	CONTCL 051170 G	CSREQT= 000045	EF14 = 000016 G
ABO FM 025202	BIT09 = 001000 G	CRC 016552	CSRESE= 000033	EF15 = 000017 G
ACK = 000001	BIT1 = 000002 G	CRCTAB 017026	CSREVI= 000001	EF16 = 000020 G
ALIMIT 021742	BIT10 = 002000 G	CRLF 041504	CSRPT = 000025	ELOOPB 021416
ALLOC 042772	BIT11 = 004000 G	CURR T 022254 G	CSSEFG= 000047	EMDMG1 020110
AREA00 003526 G	BIT12 = 010000 G	CSAAD 034610	CSSPRI= 000041	EMDMG2 020120
AREA01 003606 G	BIT13 = 020000 G	CSAAE 034622	CSSEVC= 000037	EMT TR 022332 G
AREA02 003610 G	BIT14 = 040000 G	CSAAK 035440	CSTPRI= 000013	END OF 030660
AREA03 003712 G	BIT15 = 100000 G	CSAAL 035550	CSUNBU= 000031	END SU= 052754
AREA04 003744 G	BIT2 = 000004 G	CSABRT= 000021	CSWTM = 000026	ENQ = 000005
AREA11 004046 G	BIT3 = 000010 G	CSADR = 000020	CSWTU = 000027	EOMSG 002124
AREA21 004104 G	BIT4 = 000020 G	CSAU = 000054	D = 177776	EOP CH 051432 G
AREA22 004106 G	BIT5 = 000040 G	CSBRK = 000022	DDSTRT= 000000	EOP FM 025216
AREA23 004110 G	BIT6 = 000100 G	CSBSEG= 000004	DDSTRY 013036	EOP IN 027270
AREA24 004110	BIT7 = 000200 G	CSBSUB= 000002	DECMG 041316	ERRFOR 035626
AREA26 004302 G	BIT8 = 000400 G	CSBUFF= 000030	DEVIN 015006	ERPHAN 034642
AREA30 004304 G	BIT9 = 001000 G	CSCEFG= 000046	DEVPRI 002200	ERR HR 035420
AREA31 004304	BLD HW 027772	CSCLER= 000012	DIAG T 022334 G	ERR SF 035424
AREA40 004110	BLOCK 045126	CSCLP1= 000006	DLE = 000220	ERR1FO 035712
AREA41 003744	BNO 003500	CSCEVC= 000036	DPDVD 052120 G	ESC PC 034634
AREA42 003712	BSAAB 031374	CSOCLN= 000044	DPMUL 052006 G	EXPINP 002130
AREA43 003744	BSAAF 031306	CSOODU= 000053	DUNIT 022244 G	FIL 015376
AREA52 004752 G	C = 000004	CSORPT= 000024	DVC FT 035410	FILL 042152
AREA60 004544 G	CALLPC= 000022	CSOU = 000055	DZMSK 002236 G	FILL C 000204 G
AREA61 004622 G	CALLPS= 000024	CSEDIT= 000006	DZREG 002242 G	FLAGS 022276 G
AREA70 005054 G	CALLSP= 000026	CSERDF= 000002	DZS = 000001	FLAGTA 045324
AREA71 005056 G	CALLTC= 000030	CSERHR= 000003	DSAG 036260	FLAGZ 021654
ASAAW 027112	CAL CL 047510	CSERSF= 000001	DSAAH 036276	FLAG I 027336
ASAAW 027126	CAL TI 047546 G	CSERSO= 000004	DSAAI 041050	FLA SE 045272
ASAAW 027134	CHKLUP 031410	CSESCA= 000010	DSAAJ 041054	FLG MA 027276
ASAAZ 027150	CHKREP 014102	CSSEGC= 000005	DSAAK 041072	FLLCNT 015372
ASABA 027160	CHKSTR 043334	CSSEGC= 000003	DSAAK 041072	FORM T 035722
A02A 003642	CHKTTY 041422	CSSETST= 000001	DSAAK 041110	FORM00 003554
A02B 003644	CHK FO 023620	CSEXIT= 000032	DSAAK 041120	FORM02 003646
A04EX 004000	CHK MA 027550	CSGMAN= 000043	E = 177774	FORM04 004002
A04RC 003776	CHK PC 034636	CSGPHR= 000042	EF CON= 000036 G	FORM23 004204
A52A 005050	CHK SW 023342	CSGPR1= 000040	EF NEW= 000035 G	FORM30 004436
A52B 005052	CHRCNT 042654	CSGTIM= 000052	EF PWR= 000034 G	FORM40 004224
B = 000002	CH FLA 027252	CSINIT= 000011	EF PES= 000037 G	FORM52 005004
BAPNT 002160	CH PAS 027274	CSINLP= 000020	EF STA= 000040 G	FORM60 004566
BASIZE 002162	CLEAR 030672	CSKWF= 000035	EF01 = 000001 G	FORM61 004714
BAUD 021640	CLKACC 022236 G	CSKWF= 000034	EF02 = 000002 G	FREE 043230
BGN SU= 022054	CLKBFR 047512	CSLOOP= 000100	EF03 = 000003 G	FRM30A 004457
BINMSG 041302	CLKCNT 022234 G	CSMANI= 000051	EF04 = 000004 G	FRM30B 004501
BIT0 = 000071 G	CLKRES 051110 G	CSMSG = 000023	EF05 = 000005 G	FRM30C 004522
BIT00 = 000071 G	CLKSER 051410 G	CSPNTB= 000014	EF06 = 000006 G	FSS = 000001
BIT01 = 000002 G	CLKSON 022300 G	CSPNTF= 000017	EF07 = 000007 G	FSAU = 000015
BIT02 = 000004 G	CLK SE 027352	CSPNTS= 000016	EF08 = 000010 G	FSGN = 000040
BIT03 = 000010 G	CLR MA 027626	CSPNTX= 000015	EF09 = 000011 G	FSCLEA= 000007
BIT04 = 000020 G	CNTFLG 015412	CSPOIN= 000040	EF10 = 000012 G	FSDU = 000016

FSEND = 000041	ININIT 022256 G	LPCNTR 022102 G	L10003 003742	MSG12 002567
FSHARD= 000004	INIT 015374	LPT AD 026672	L10004 003774	MSG20 002636
FSHW = 000013	INITIA 041332	LPT RE 026666	L10005 004076	MSG21 002655
FSINIT= 000006	INITMA 002154	LSI RE 026662	L10006 004104	MSG22 002675
FSJMP = 000050	INITMS 002156	LUP 047414	L10007 004106	MSG23 002740
FSMOD = 000000	INIT M 027674	LUP AD 034640	L10010 004200	MSG24 003000
FSMSG = 000011	INIT R 022070 G	LSAU 011470 G	L10011 004302	MSG25 003043
FSPWR = 000017	INMSG 002136	LSAUT 002070 G	L10012 004434	MSG26 003100
FSRPT = 000012	INPUTA 042260	LSCCP 002044 G	L10013 004564	MSG30 003135
FSSEG = 000003	INSERR 016550	LSCLER 012066 G	L10014 004712	MSG31 003165
FSOFT= 000005	INSIZ 002150	LSDEPO 002011 G	L10015 005002	MSG40 002655
FSSRV = 000010	INTFOR 035556	LSDEVP 002052 G	L10016 005054	MSG41 002427
FSSUB = 000002	INVAL 027002	LSDISP 011462 G	L10017 005056	MSG42 002363
FSSW = 000014	INVDAT= 100000	LSDR 002236 G	L10020 011452	MSG43 003242
FSTEST= 000001	INVINT 035450	LSDRCT 002062 G	L10021 011460	MSG45 003261
GARBAG 042656	INV SW 023276	LSDRS 002064 G	L10022 011466	MSG50 003307
GETCHR 041362	IN SUF 030644	LSDRST 002242 G	L10023 011470	MSG51 003344
GETCMN 044746	LSAU = 000041	LSDTP 002040 G	L10024 011472	MSG52 003372
GETOB 013040	LSCLN = 000041	LSDU 011472 G	L10025 011774	MSG70 003405
GETPAR 036440	LSDU = 000041	LSOUT 002072 G	L10026 012114	MSG71 003440
GETSWI 043742	LSHRD = 000041	LSDVTY 002222 G	L10027 015610	MSG72 003502
GET TW 043512	LSINIT= 000041	LSEF 002024 G	L10030 016256	MTY 003464
GLOB 002100 G	LSMOD = 000041	LSEXP1 002032 G	L10031 017576	MUL 051654 G
GSUB 012116 G	LSMSG = 000041	LSEXP2 002034 G	L10032 020656	NAK = 000002
GSEXCP= 000400	LSPWR = 000041	LSEXP3 002036 G	L10033 021524	NEWPRI 051400 G
GSHILI= 000002	LSRPT = 000041	LSHARD 021422 G	L10034 021742	NEXTAR 045510
GSLOLI= 000001	LSSEG = 000041	LSHPCP 002046 G	MAJ IN 022060 G	NMSG 015404
GSNO = 000000	LSSET = 000041	LSHPTP 002056 G	MAJ LO 047514	NO CLK 024412
GSOFFS= 000400	LSRV = 000041	LSHW 011442 G	MAJ US 022062 G	NO FLA 045304
GSOFSI= 000376	LSUB = 000041	LSICP 002042 G	MAN TI 024362	NO LPT 042622
GSPRMA= 000001	LSST = 000041	LSINIT 011474 G	MAP16 052356 G	NO PTA 027102
GSPRMD= 000002	LSJMP = 000167	LSLADP 002076 G	MASCLR= 000020	NR = 000001
GSPRML= 000000	KBPTR 022106 G	LSLAST 022054 G	MASK B 031406	NUMBIN 035746
GSRADA= 000140	KBUF 022110 G	LSMREV 002012 G	MASK W 031404	NUM LA 036114
GSRADB= 000000	KEX 004102	LSNAME 002000 G	MCOMP 020076	NUM UN 022452
GSRADD= 000040	KRC 004100	LSREPP 002054 G	MDMSG 020100	NUNITS 031362
GSRADF= 000200	LCOMP 021000	LSREV 002010 G	MDMSG1 020110	NXTFOR 045560
GSRADL= 000120	LCRCD 021416	LSRPT 011466 G	MEM SI 026702	OCTMSG 041310
GSRADO= 000020	LCRCH 021014	LSOFT 021722 G	M N IN 022054 G	OSOP = 000024
GSRADT= 000100	LINE 021622	LSGPC 002030 G	MIN US 022056 G	OUT 002152
GSEXFER= 000004	LINEMK= 000017	LSGPCP 002050 G	MODR 051720 G	OUTBUF 002142
GYES = 000010	LINE F 022330 G	LSPTP 002060 G	MSCNEN= 000040	OUTERR 002120
HCORED 027034	LINMSK 002202	LSSTA 002066 G	MSG 012116	OUTMSG 002132
HCOREQ 026714	LOAD F 027272	LSW 011454 G	MSGAP 002152	OUTREP 017600
HCORET 022266 G	LOBYTE= 000377	LSTIML 002022 G	MSGN 020124	OUTSIZ 002144
HEAD 002000 G	LOGDEV 002126	LSTIMU 002020 G	MSGNO 002164	OSAPTS= 000001
HERTZ 026654	LOGMSG 041324	LSTIMI 002016 G	MSG00 002252	OSAU = 000001
HOLDSP= 000020	LOOPB 021016	LSTSTI 002074 G	MSG01 002311	OSBGNR= 000001
HYBYTE= 177400	LOOPH 021006	LSUNIT 002014 G	MSG02 002336	OSBGNS= 000001
MSAAB 046114	LOOPMG 021002	L CLK 026640	MSG03 002363	OSDU = 000001
LIMIT 002114	LOOPN 021012	L10000 003552	MSG04 002427	OSGNSW= 000001
INBUF 002146	LOOSTA 021013	L10001 003606	MSG10 002454	OSPOIN= 000001
INERF 002116	LPBFR 022104 G	L10002 003640	MSG11 002534	OSPPW = 000001

PARITY= 000200	RMSGNO 002166	SYNC = 000377	TSTAGL= 177777	XSFALS= 000040
PARSES 045020	RQUE 015416	SYNCNT 015370	TSTAGN= 010035	X\$OFFS= 000400
PAR LA 041012	RQUEB 015432	SYS FT 035400	T\$TEMP= 000000	X\$TRUE= 000020
PASCNT 002122	RQUET 015420	S\$LSYM= 010000	T\$TEST= 000002	XIX1 = 072000
PATTER 011456	RQUEUE 015414	TABLE 011440 G	T\$TSTM= 177777	\$BGNLE= 177777
PATTRN 022011	RSTACK 051602 G	TEMBUF 005060	T\$TSTS= 000001	\$BREG 027350
PRINTC 042632	RSX FL 027266	TEMP 002104	T\$SAU = 010023	\$ENDAD 051440 G
PRINTF 046134	RTEMP1 015406	TEMP2 004202	T\$SCLE= 010026	\$ERFLG= 000400
PRIOR 021555	RTEMP2 015410	TERIN 013460	T\$SDU = 010024	\$FSAND= 000310
PP100 = 000000 G	RXINTE= 000100	TERMI 047504	T\$SHAR= 010033	\$FSBAD= 000401
PP101 = 000040 G	R\$STCK 002100	TERML1 045312	T\$SHW = 010020	\$FSBLA= 000170
PP102 = 000100 G	SCRTH 002102	TERMTA 041274	T\$SINI= 010025	\$FSCAS= 000150
PP103 = 000140 G	SEARCH 043460	TEROUT 012324	T\$MSG= 010017	\$F\$DEC= 000220
PP 04 = 000200 G	SEGSTA 022302 G	TESTS 017126 G	T\$SRPT= 010022	\$F\$DO = 000340
PP105 = 000240 G	SET MA 027462	TEST M 027204	T\$SSEG= 010000	\$F\$FAL= 000405
PP106 = 000300 G	SHIFT 052440 G	TIMER 015366	T\$SSOF= 010034	\$F\$G00= 000400
PP107 = 000340 G	SIZE C 051316 G	TIMFLG 022232 G	T\$SSRV= 010030	\$F\$IF = 000110
PPNTST 042522	SIZE M 051234 G	TIMLIM 002110	T\$SSW = 010021	\$F\$INC= 000210
PROBUF 013166	SIZ TR 051374	TIM CO 022064 G	T\$STES= 010032	\$F\$ 00= 000200
PROLOP 020660	SCH = 000201	TIM OP 035720	T1 017126 G	\$F\$NAM= 000160
PRO CM 027244	SPEC U 027172	TMPBUF 010204	T2 020130 G	\$F\$NO = 000403
PTABLE 002170	SPV SE 023674	TOO MA 041254	UNI MA 027174	\$F\$OR = 000320
PTAB S 022264 G	SSTACK 011440	TSTS = 000002	USER P 022260 G	\$F\$RTN= 000300
PTCODE 021420 G	STACK = 000007	TST AB 031520	USER T 022262 G	\$F\$SEL= 000140
PTEND 002200	STARTC 051164 G	TST TO 023324	VALID 022522	\$F\$THE= 000330
PUTCHR 041336	STATIO 021577	TXINTE= 040000	VAL LA 023252	\$F\$TRU= 000404
PWR FA 052612 G	STIME 002112	TYPE 012556	VAL SW 027310	\$F\$UNT= 000130
PWR FL 022066 G	STINPT 017710	TYPEC 041650	VECTOR 021541	\$F\$WHI= 000120
PWR MS 052740	STRCHR 042212	TYPEPC 035544	VMSG 011776	\$F\$YES= 000402
PWR SA 052734	STREQ 027014	TYPFLA 045166	WIDTH 036314	\$IFLEV= 177777
PWR UP 052736	STRT = 000006	TYPLIN 041546	XBUFFE 015400	\$ISKO = 000001
P CLK 026646	STRTB 003516	TYPNUM 041134	XDONE 015002	\$ISK1 = 000001
RBQUE 015434	STRTC 003524	TYPSTR 041566	XEQDIA 051466 G	\$ISK2 = 000001
RBQUEB 015442	STR TSA 003523	TYP ER 035430	XEQSUB 051454 G	\$ISK3 = 000001
RBQUET 015436	STR T 027250	TY UNI 030664	XEQ CL 031324	\$LINE 015360
RCVADD 015364	ST REQ 027170	T\$ARGC= 000003	XEQ CM 026632	\$LOCTA= 177777
RCVBUF= 000002	ST SET 023514	T\$CODE= 001130	XEQ IN 031006	\$LSTCN= 177777
RCVCSR= 000000	SUNIT 027256	T\$ERCO= 000064	XEQ LA 025170	\$LSTIN= 000000
RCVE 016260	SUPERV 025234	T\$ERRN= 000064	XEQ OP 031100	\$LSTST= 177777
RDONE 015004	SUPFLA 022242 G	T\$EXCP= 000000	XEQ PR 024422	\$LSTTA= 000000
READ P 047516 G	SUPV T 022420 G	T\$FLAG= 000040	XEQ TE 031144	\$MCALL= 000000
REGBAC 052342 G	SUP PR 023266	T\$HILI= 077777	XINT 015444 G	\$NESTL= 177777
REGSAV 052326 G	SVC CNT= 177777	T\$LOLI= 000000	XMIT 014472	\$NSKO = 000300
RENABL= 010000	SVCGBL= 000000	T\$LSYM= 010000	XMTADD 015362	\$NSK1 = 000130
REP = 000003	SVCHAN 031566	T\$MCAL= 177777	XMTBUF= 000006	\$NSK2 = 000110
REPA 020125	SVCINS= 000000	T\$NEST= 177777	XMTCSR= 000004	\$NSK3 = 000110
REPCRC 020126	SVCSTK= 177777	T\$NSKJ= 000000	XSIZE 015402	\$NSK4 = 000110
REPM 020120	SVC SUB= 177777	T\$NSK1= 000005	XSTART 012560	\$SAVLE= 177777
REP N 020123	SVCTAG= 000000	T\$NSK2= 000003	XTIME 050174 G	\$SAV2 052504 G
REQN P 027254	SVCTST= 177777	T\$SAVL= 177777	XTIMEN 051020	\$SAV3 052520 G
REQN T 027246	SWCHAN 027074	T\$SEGL= 177777	XTIMST 050216	\$SAV4 052536 G
RE SET 023444	SWITCH 045464	T\$SEKO= 010000	XXDP D 027042	\$SAV5 052556 G
R NT 015612 G	SW PTA 027060	T\$SUBN= 000000	X\$ALWA= 000000	\$SSKO = 050003

OUTERR MACY11 30(1046) 22-NOV-77 07 50 PAGE 191-3
DOCTOR P11 07-OCT-77 13 48 SYMBOL TABLE

L 6

SEQ 0076

\$TAGLE= 177777	\$TSK3 = 050030	\$\$ERFL= 000000	\$\$RETN= 000001	\$\$TGS2= 000000
\$TAGNU= 050004	\$\$ARGC= 000000	\$\$FLAG= 000001	\$\$RTN1= 050000	\$\$TO = 000001
\$TEMP = 000300	\$\$BYTE= 000402	\$\$FROM= 000000	\$\$RTN2= 050001	\$\$\$TAG= 050000
\$TSK0 = 050002	\$\$CASE= 000000	\$\$LOC = 020764	\$\$SRC = 000064	= 072000
\$TSK1 = 050003	\$\$DST = 000003	\$\$LOCN= 000000	\$\$TGSV= 000000	
\$TSK2 = 050006	\$\$ELOC= 000402	\$\$REG = 177777	\$\$TGS1= 000000	

ABS 072000 000

ERRORS DETECTED 0

DZDZCA=DZDZCA
RUN-TIME 95 96 9 SECONDS
RUN-TIME RATIO 733/193=3 7
CPE USED 364 (71 PAGES)