

IDENTIFICATION

Product code: AC-E099A-MC
Product Name: CZLAF0 LA36 TERM TST
Date Created: MARCH 1978
Maintainer: DIAGNOSTIC ENGINEERING
Author: RALPH A. SCHAUBER

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

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) 1978 by Digital Equipment Corporation.

TABLE OF CONTENTS

- 1.0 ABSTRACT
 - 1.1 Functional Description
- 2.0 REQUIREMENTS
 - 2.1 Equipment
 - 2.2 Options
 - 2.3 Related Programs
- 3.0 LOADING & INITIALIZATION
- 4.0 CONTROL & TEST SELECTION
 - 4.1 Switch Register Control
 - 4.2 Console Control
 - 4.3 Commands
 - 4.4 Examples & Hints
- 5.0 SCOPE OF TESTS
 - 5.1 TEST 0
 - 5.2 TEST 1
 - 5.3 TEST 2
 - 5.4 TEST 3
 - 5.5 TEST 4
 - 5.6 TEST 5
- 6.0 ERRORS
- 7.0 PROCEDURES FOR NON STD. DVCS

1.0 ABSTRACT

This diagnostic will functionally test the hardware options on the LA36 terminal.

Up to 48 terminals, including the console device, can be tested at a time. All terminals under test must be interfaced thru a DL11 type interface.

Control of this diagnostic may be thru a switch register, or via interactive console terminal commands.

1.1 Functional Description

The program will first determine if the system has a hardware switch register. If none available a software switch register at loc 176 will be assigned.

The program will then determine what device interfaces are on the system, at the standard DL11-A,B,C,D,E address and the interrupt vector of each interface present. This information will then be stored in a table.

The program will then ask the operator if he/she wants to use console control. If the response is "no" the program will halt and wait for the operator to enter control information thru the switch register.

Note : If no hardware switch register is present control is forced to console control.

If console control is selected a menu of available commands will be printed on the console terminal, and the program will wait for commands via the console.

2.0 REQUIREMENTS

2.1 EQUIPMENT

This diagnostic was written to run on all models of the PDP-11 computer, including the LSI-11. One to forty eight LA36 terminals, connected thru a single line asynchronous interface (DL11-A thru E, DL11-W, DLV11), may be tested. 8K of memory is required.

A hardware switch register is supported, but not required.

2,2 OPTIONS

The following options are tested by this diagnostic:

ASCII/APL character set,
 Selective Addressing Option,
 Auto-Answerback Option,
 Top-of-form Option,
 Horizontal & Vertical Tab Option.

2,3 RELATED PROGRAMS

Although some error checking is done, this diagnostic does not functionally test the basic LA36 terminal, nor the terminals interface.

Therefore the basic LA36 terminal diagnostic, MDEC=11-DZLAC=*, and the DL11=*/DLV=11 interface test should be successfully run prior to this diagnostic. Also any other applicable processor/memory tests.

3,0 LOADING & INITIALIZATION

This diagnostic may be loaded using the standard procedures for paper tape, or via XXDP media. This diagnostic will not run in chained mode because manual intervention is required. The absolute loader area is preserved.

This diagnostic self sizes the system as far as the interfaces, and their interrupt vectors. The only operator modifications to be made are:

1. the location 'TIMER' which is a CPU dependent TIME constant. See listing of DELAYM Routine.

```
TIMER: ,WORD 554 ;SET FOR 11/35 = 11/40
;SET TO 202 IF 11/03
; 251 11/05 = 11/10
; 314 11/15 = 11/20
; 2127 11/45 BIPOLAR = 11/55
; 1237 11/45 = MOS = 11/70
; 755 11/45
DELAYT: ,WORD 0 ;DELAY TIME BUFFER
```

2. any of the preset device address to accommodate a non-standard interface address. (See Sect 7,0 for details)
3. location 'WIDTH' if other than 132 COL PAPER is being used. (See section 4,3). (Common for all terminals)

The initial starting address is 200(8), and all restarts at 1372.

4.0 CONTROL & TEST SELECTION

The diagnostic will ask (via the console) if console control is desired. Answer 'Y' if you want to use interactive commands, otherwise type 'N' for switch register control.

If 'Y' is typed a menu of available commands is printed on the console, and the program will wait for command input. If 'N' is typed the program will print a listing of the line (interface) table, then halt for the operator to set the switches to the desired parameters.

4.1 SWITCH REGISTER CONTROL

When SWITCH REGISTER CONTROL is selected the program will halt. Set the switches to the desired mode, then press continue. The program will check the entry and if a specific test is to be run, or a specific line is to be tested the program will halt again. Enter the desired line/test data in the switches, then press continue.

To change parameters the test must be restarted at loc 1372.

ALL of the switch functions are also available under console control mode. (See sect. 4.2,4.3).

SWITCH REGISTER BIT DEFINITIONS FIRST WORD MODE SELECTION

BIT15	=1 (UP) =0 (DOWN)	HALT ON ERROR CONTINUE AFTER REPORT
BIT14	=1 (UP) =0 (DOWN)	LOOP AFTER ERROR IS DETECTED DON'T LOOP
BIT13	=1 (UP) =0 (DOWN)	INHIBIT ERROR REPORTS PRINT ERROR REPORTS
BIT12	=1 (UP) =0 (DOWN)	PRINT INTERFACE TABLE DON'T PRINT TABLE
BIT11	=1 (UP) =0 (DOWN)	INHIBIT ITERATIONS NORMAL RUN
BIT6	=1 (UP) =0 (DOWN)	RUN ALL TESTS IN SEQUENCE RUN SELECTED TEST ONLY
BIT5	=1 (UP) =0 (DOWN)	RUN ALL AVAILABLE LINES RUN SINGLE LINE ONLY

SECOND WORD LINE AND TEST NO.

BIT15 - BIT8	SELECTED LINE NUMBER (00-57)
BIT7 - BIT0	SELECTED TEST NUMBER (0-5)

4.2 CONSOLE CONTROL

When console control is selected a menu of available commands is typed on the console terminal. The program will wait for commands to be entered thru the keyboard.

Enter one command per line, followed by a carriage return. To terminate command input and start execution type an ALTmode or ESCape.

To regain control once execution has begun type a CTL-C. The program will respond with 'READY'. You can now enter the desired

commands just as in startup.

4.3 COMMANDS

There are two types of commands available, interactive commands, and static commands.

ALL static commands can only be entered while in "Command Mode", that is during startup after "READY" is printed on the console, or after the operator has invoked command mode by typing a CTL-C and the program has printed "READY".

Interactive commands can be entered at any time, and are essentially the same as the switch register bits 15,14,13.

The available Commands are:

- S (STATIC) Single Line Mode. Test a single device. Line 00 is default. Use add command to select the desired line.
- M (STATIC) Multi Line Mode. Test all lines present and not deselected.
- Q (STATIC) Sequence thru all tests, starting with test 0.
- Rn (STATIC) Run test N.
- Dn (STATIC) Deselect or Drop from testing interface line N. (see T command)
- An (STATIC) Add line N. Clear out the error count for line N, and reselect the line for testing. In single line mode sets N as current line to test.
- T (STATIC) Type out a listing of the interface lines present on the system, the vector address at which the device interrupts, and whether or not the line is selected.
- Wn (STATIC) Changes location 'width' to N. Used to control output of terminal tests. Enter N as an Octal number 32 to 204. (132 decimal)
- CTL-L (interactive) Loop on error. If an error is detected the program will start looping on the test section which caused the error, and continue to loop until a klear command is issued.
In command mode type an L.

CTL=H (interactive) Halt on error, will cause the program to halt after the error message is printed, (assuming error printouts are enabled).
In command mode type or H.

CTL=K (interactive) Klear - resets both the H and L commands (Don't halt, Don't loop).
In command mode type a C.

CTL=N (interactive) NO Error reports, inhibits the printing of normal error messages.
In command mode type an N.

CTL=P (interactive) Print error reports, ALL report messages will be printed on the console.
In command mode type a P.

CTL=G (interactive) will cause the program PCFLAGS to be displayed on the console for trouble shooting purposes. See listing for bit definitions.

ESC Exit command mode & start execution of the diagnostic test(s).

CTL=C Returns test to command mode,
All operations in progress stop.

4.4 EXAMPLES & HINTS

Test numbers 0 thru 5 may be selected to run individually on all, or any terminal.

ALL tests can be run sequentially on all or any terminal.

Tests can be run sequentially on a terminal, but there is little chance that any terminal will have more than one or two of these options installed. Sequencing all tests will probably result in numerous errors from trying to test non-existent options.

If a line gets more than 16 errors it will be deselected by the program and a message will be printed on the console. If the line is the only one being tested it will automatically be re-selected.

Example 1, the commands to select test #4 to be run on all lines.

```

READY
R04(CR) = Run test 4
M(CR)   = Multi line mode
w120(CR) = Set paper width to 120
$       = ESC = echoed as $

```


Example 2, the commands to run all tests on line no. 06, with Halt on error set.

```

READY
Q(CR)   - Sequence tests
S(CR)   - Single line mode.
A06(CR) - Add line #06
H(CR)   - Halt on error
$       - ESC - echoed as $,

```

EXAMPLE 3. How do I run tests on 10 out of 12 terminals.

First you must be in command mode. If a test is running type CTL=C. The program will respond with "Ready". Now type D nn (cr) to deselect line number nn. Repeat for each additional line to be dropped from the tests. Now select your test other parameters as in Example 1.

EXAMPLE 4. How do I restart a device which has been deselected?

In command mode type Ann (cr) where nn is the line number of the line to be added.

EXAMPLE 5. How do I loop on error.

If the test is running type a CTL=L. When an error is detected the program will start to loop on the test or subtest and continue to loop until a CTL=K or CTL=C command is issued.

If in command mode type L (CR).

EXAMPLE 6. Can I set the 'width' constant different for each terminal?

The constant 'width' is the same for all terminals, although it may be changed any time you are in command mode.

5.0 SCOPE OF TESTS.

This diagnostic will functionally test the following hardware options of the LA36.

1. Secondary character set option APL/ASCII character set selection
2. Selective addressing option
3. Auto-answer back option
4. Top of forms option

5. Horizontal and Vertical tabulation option.

The diagnostic will do cursory testing of the basic interface, and basic LA36 logic. It is assumed that the basic interface, CPU/memory, and LA36 tests have been run successfully.

Due to the nature of the hardware under test most error checking will be by visual inspection of the terminal output by the operator.

Description Of Tests

5.1 Test 0 Secondary Character Set.

This test is an output only test, no terminal feed back is received.

The test prints on each terminal alternating lines of ASCII character set, and APL character set.

Output of characters per line will consist of char codes 40 thru 176 unless the paper width limit is reached first.

Output format:

```
PRIMARY---(ASCII CHARACTER SET)
SECONDARY--(APL CHARACTER SET)
```

This output is controlled by the "width" of the paper. See W command description.

5.2 Test 1 Selective Addressing Option

This test will alternately send out a select code, followed by it's ASCII Equivalent, for all possible select codes (20 thru 177).

This test will also deslect all terminals and try to output a "this should not print" message, "All terminals should be off"

This test will also try to print "this should not print" messages after transmitting "NUL" select character sequences, and no select codes sequences.

Valid terminal output should be: Select Recognized = /NN(Group Select Code) /NN(Individual Select Code)/

where NN represents the select codes recognized by this terminal, if the group select code and individual select codes are set to the same thing them only one /NN/ should appear,

More than two /nn/ codes printed indicates a logic failure in the decoder section of the option, or possibly interface to terminal line problems.

Any of the "this should not print" message that appear on the terminal indicate a failure of the selection logic.

5.3 Test 2 Auto Answer Back Option

This test is divided into six subtests:

Subtest -1

The first subtest is actually a sizing routine. The terminal should respond to its unique selection code with an answerback when polled. This test has no way of knowing what the answer back is, nor any way of "pre-selecting" its unique selection code. Therefore subtest -1 will try all legal selection codes to see if it can cause an answer back to be transmitted. If one is received the program will store the select code associated with the response in the line table for future testing.

Subtest -2

Will see if any answer back has been received, and check its length, the message should not exceed 20 (10) characters. Subtest 2 will print the ASCII message on the terminal, and an octal representation of the characters (to verify non printables are being transmitted correctly, and as a trouble shooting aid if bad data is being sent out from the switches).

Subtest -3

Will read the answer back ten times to verify reliability of the data, and lines,

Subtest -4

Will try to cause transmission of the answer back in response to a broadcast code.

Single Line Mode - Subtests 5, and 6.

Subtest -5

The test will request the operator to press the "Here-is" key, then check for answerback,

Subtest -6

The test will request the operator to type 'cTL=E', then check for answer back.

The operator must verify that the message echoed back to the terminal is correct, by comparing it to the data switch configuration.

5.4 Test 3 Top Of Forms Option

This test is divided into two subtests, one for multi line mode, the other for single line mode. Operator intervention will be required for the single line test.

1. Multi line mode.

This subtest will assume a standard form of 11 inches being used.

The test will issue a form feed, then print a line of dashes. This FF/dashes is repeated 3 times.

The operator should verify correct operation by checking for a line of dashes at the same place on each page.

2. In single line mode, This test will require the operator to set the forms length switch to the value requested. The test will then do three form feeds at each length setting.

5.5 Test 4 Horizontal Tab Option

This test will adjust it's output to conform to the paper width. Change location "width" to the appropriate value before starting test. (Preset to 132 col.) Note: see w command description.

The test will print a reference line for visual verification. The line will look like this:V.....V.....V.....V.....

Tabs will then be set corresponding to the location of each V. The test will then issue a tab and print an X, tab then X etc until the line is complete. Three lines of X's will be printed. All X's printed should be aligned with the reference line V's.

This will be repeated for various (7) values of tab spacing.

Example of output

```
.....V.....V.....V.....V.....V.....
  X      X      X      X      X
  X      X      X      X      X
  X      X      X      X      X
```

```
.....V.....V.....V.....
  X      X      X
  X      X      X
  X      X      X
```

5.6 Test 5 Vertical Tab Option

This test is divided into two subtests, one for multi line mode, the other for single line mode. The single line mode test requires operator intervention.

Multi line mode subtest

Will set tabs at intervals of 1 line, 2 lines, 3 lines etc. up to 11 lines. The test will then issue a vert. tab then print a line of dashes, then repeat until 1 full page has been done. Three pages of output are run for 1 pass of test.

Single line mode testing involves the operator to set up the forms control to 11 inch forms, and then proceeds with the same subtest as for multi line mode.

6.0 ERROR REPORTING

There will be four basic sources of error messages. First the system sizer, second the command decoder, and third the diagnostic tests, and the I/O drivers.

6.1 Diagnostic Tests

All test error messages will be 2 lines of output. A standard format line, shown below, and a descriptive message telling what went wrong.

Std. Fmt., #ERROR XXY TEST YY LINE ZZ

where XXX = the error number local to the current test,

YY = the current test number,

ZZ = the line under test.

an example of a descriptive message :

,NO ANSWERBACK MESSAGE RECEIVED

As each error is handled a routine will update an error count for the failing line. If 16 errors are accumulated on a line, that line will be "deselected" and the following message will be displayed.

EXCESSIVE ERRORS .. LINE XX DROPPED.

If the line under test is the only line being tested the program will automatically re-select the line, zero the count, and continue testing after typing the following:

LINE RE-SELECTED FOR TEST.

6.2 I/O Driver

If the IO Driver finds no available line to test a message will be displayed and then control will return to the "ready" state.

NO LINES AVAILABLE FOR TEST.

#377 Multi line driver error.

Error messages tagged as #377 indicate a failure during an I/O driver operation, such as a failure to interrupt on transmit to a terminal with the interrupt enable set.

#376

Same as #377 except a single line driver.

6.3 Command Decoder

Console terminal command errors will be handled by a CMDERR module & will output a line of ??? if the input was invalid.

If a line selection command tries to add (re-select) an invalid or non-existent line a ==LINE INVALID message will be typed.

6.4 System Sizer

If during the sizing operation the sizing routine detects a failure of the interface to interrupt it will be reported.

ex: 'NO INTERRUPT ON TXMIT LINE 27'

7.0 PROCEDURES FOR NON-STANDARD DEVICES.

This diagnostic can be modified for use on devices that have non-standard interface addresses by replacing an unused address in the line table with the address of the interface line to be tested.

The table is preset to the standard DL11=A,B,C,D,E addresses, (775610 = 776170 & 776500 = 776670), and the console address 777560.

No modification need be made because of non-standard interrupt vector addresses. The diagnostic sizes each address for presence on the system, and inserts the interrupt vector data into the table at run time.

NOTE: The table addresses are not in ascending order, rather it has been optimized for relative system size by having the most commonly used addresses at the head of the table, DL11=A,B

and DL11=C,D,E address are merged together.

CZLAFAD LA36 TERM TST MACY11 30A(1052) 03-JAN-77 00101
CZLAFAP11 03-JAN-78 11120 TABLE OF CONTENTS

101		BASIC DEFINITIONS
212		ACT11 HOOKS
216	04200	TEST CONTROL & INITIALIZATION
507	08500	LINE CONTROL & INITIALIZATION
713	12540	SWITCH REGISTER ROUTINES
791	14840	CONSOLE TERMINAL ROUTINES
1189	22020	ERROR & REPORT ROUTINES
1378	00700	INTERFACE SIZER ROUTINES
1452	05450	EMT HANDLER
1472	06650	I/O DRIVERS
1659	11595	TRAP ROUTINES
1793	16900	CONVERSION ROUTINES
1898	00250	LA36 OPTION TESTS
2778	43150	STORAGE & CONSTANTS


```

58          01160 ;***** SWITCH REGISTER BIT DEFINITIONS *****
59          01180
60          01200 ; BIT15 =1 (UP) HALT ON ERROR
61          01220 ;           =0 (DOWN) CONTINUE AFTER REPORT
62          01240
63          01260 ; BIT14 =1 (UP) LOOP AFTER ERROR IS DETECTED
64          01280 ;           =0 (DOWN) DON'T LOOP
65          01300
66          01320 ; BIT13 =1 (UP) INHIBIT ERROR REPORTS
67          01340 ;           =0 (DOWN) PRINT ERROR REPORTS
68          01360
69          01380 ; BIT12 =1 (UP) PRINT INTERFACE TABLE
70          01400 ;           =0 (DOWN) DON'T PRINT TABLE
71          01420
72          01440 ; BIT11 =1 (UP) INHIBIT ITERATIONS
73          01460 ;           =0 (DOWN) NORMAL RUN
74          01480
75          01500 ; BIT6 =1 (UP) RUN ALL TESTS IN SEQUENCE
76          01520 ;           =0 (DOWN) RUN SELECTED TEST ONLY
77          01540
78          01560 ; BIT5 =1 (UP) RUN ALL AVAILABLE LINES
79          01580 ;           =0 (DOWN) RUN SINGLE LINE ONLY
80          01600
81          01620
82          01640 ; SECOND WORD ENTRY VIA SWITCHES
83          01660
84          01680 ; BIT15 = BIT0 SELECTED LINE NUMBER
85          01700
86          01720 ; BIT7 = BIT0 SELECTED TEST NUMBER
87          01740
88          01760
89          01780 ;***** TEST ASSIGNMENTS *****
90          01800
91          01820 ; TEST 0 SECONDARY CHARACTER SET
92          01840 ; TEST 1 SELECTIVE ADDRESSING OPTION
93          01860 ; TEST 2 AUTO ANSWERBACK OPTION
94          01880 ; TEST 3 TOP OF FORMS OPTION
95          01900 ; TEST 4 HORIZONTAL TAB OPTION
96          01920 ; TEST 5 VERTICAL TAB OPTION
    
```

```

98          01960
104         02080
105         02100 ;*****
106         02120 ; LOCAL PROGRAM EQUATES
107         02140
108         000020 02160 ABO = BIT4 ;LINE ABORT FLAG
109         000006 02180 ACK = 6
110         000004 02200 ADDC = BIT2
111         000200 02220 ATTN = BIT7
112         000015 02240 CR = 15
113         000003 02260 CTLC = 3
114         000000 02280 CTLCNT = 0
115         000007 02300 CTLG = 7
116         000010 02301 CTLM = 10
117         000013 02302 CTLN = 13
118         000014 02303 CTLL = 14
119         000016 02304 CTLN = 16
120         000020 02305 CTLP = 20
121         004000 02320 DATAIN = BIT11
122         100000 02340 DLP = BIT15 ;LINE PRESENT FLAG
123         000010 02360 DROPC = BIT3
124         000005 02380 ENQ = 5
125         004000 02400 EOL = BIT11
126         020000 02420 EOP = BIT13
127         040000 02440 EOT = BIT14
128         000033 02460 ESC = 33
129         000003 02480 ETX = 3
130         000001 02500 FLAG1 = BIT0
131         000002 02520 FLAG2 = BIT1
132         100000 02540 HALTOE = BIT15
133         000200 02560 HALTC = BIT7
134         177776 02580 ICNT = -2
135         000040 02600 INHR = BIT5
136         020000 02620 INHRPT = BIT13
137         005726 02640 ISP = 5726 ;INC SP 2
138         022626 02660 ISP2 = 22626 ;INC SP 4
139         000400 02680 LDONE = BIT8
140         000100 02700 LOOPC = BIT6
141         040000 02720 LOUPOE = BIT14
142         000200 02740
143         000340 02760 PRI4 = 200 ;PRIORITY 4
144         000000 02780 PRI7 = 340 ; " 7
145         000200 02800 PRI0 = 0 ; " 0
146         000240 02820 SEL = BIT7 ;LINE SELECT FLAG
147         000240 02840 NOP = 240
148         003000 02860 NOOP = 240
149         000006 02880 MAJOR = 3000
150         000002 02900 POINT = 6
151         000010 02920 PASCNT = 2
152         000340 02940 RPC = 10
153         000017 02960 NREQ = 340
154         000016 02980 SI = 17
155         000001 03000 SO = 16
156         000002 03020 SOH = 1
157         010000 03040 STX = 2
158         020000 03060 PRINT = BIT12
          ; TDONE = BIT13
    
```

159	100000	03080	MERR = BIT15
160	002000	03120	NEWST = BIT10
161	001000	03140	NEWMOD = BIT9
162	000200	03160	READY = BIT7
163	000100	03180	SEQ = BIT6
164	000040	03200	MULTI = BITS
165	000020	03220	SWCTL = BIT4
166	000020	03240	PRINT = BIT4
167	100377	03260	MERRN = 100377
168	000004	03280	NFLAGS = 4
169	177564	03281	TPS = 177564
170	177566	03282	TPB = 177566
171	177560	03283	TKS = 177560
172	177562	03284	TKB = 177562
173	000060	03285	TKV = 60
174		03300	;***** EMT CALL EQUATES
175	104000	03320	TYPE = EMT
176	104002	03340	PRITBL = EMT+2
177	104004	03360	ITRAP = EMT+4
178	104006	03380	DELAYR = EMT+6
179		03460	

181	000000	03500	,=0	;TRAP CATCHER
182	000000	03520	,+2	
183	000002	03540	HALT	
184	000004	03560	MACHER; ,+2	
185	000006	03580	HALT	
186	000010	03600	,+2	
187	000012	03620	HALT	
188	000014	03640	INTRAP	;BREAKPOINT TRAP
189	000016	03660	PRI4	;USED DURING SYSTEM SIZER
190	000020	03680	TXTRAP	;IOT TRAP
191	000022	03700	PRI4	;USED BY TXMIT I/O DRIVER
192	000024	03720	RESTRT	;POWER FAIL INAPS TO RESTART
193	000026	03740	PRI0	
194	000030	03760	EMTBOS	
195	000032	03780	PRI0	
206	000172	04000	,=172	
207	000174	04020	SWTEST; ,WORD 0	
208	000176	04040	SWLINE; ,WORD 0	
209	000178	04060	SSWR; ,WORD 0	
210	000200	04080	JMP	START
211		04100		
212			,SBTTL	ACT11 HOOKS
(1)				
(2)				
(1)			;*****	
(1)	000204		;HOOKS REQUIRED BY ACT11	
(1)	000046		SSVPC;	;SAVE PC
(1)	000052		,=46	
(1)	000052		START	;;)SET LOC,46 TO ADDRESS OF START
(1)	000052		,=52	
(1)	000204		,WORD 20000	;;)SET LOC,52 TO 20000
(1)	000204		,=SSVPC	;; RESTORE PC
213	001100	04140	,=1100	
214	001100	04160	NOP	

```

216          04200          ,SBTTL TEST CONTROL & INITIALIZATION
221          04300          ;*****
222          04320          ;*****
223 001102    04340    START1;
225 001102    000005  04380    RESET          ;***** TEST MONITOR *****
226          04400          ;*****
227          04420          ; PROGRAM INITIALIZATION SECTION
228          04440          ;
229 001104    012706    001100    MOV     #STACK,SP
230 001110    005037    016152    CLR     NEXT
231 001114    005037    016154    CLR     INTEST
232 001120    005037    016162    CLR     NXTLIN
233 001124    005037    016160    CLR     ONLIN
234 001130    012737    016114    016164    MOV     #INBUF,PTR
235 001136    012705    010352    MOV     #T00BLK,R5
236          04580          ;
237          04600          ; SEE IF SYSTEM HAS A SWITCH REGISTER
238          04620          ;
239 001142    004737    003006    JSR     PC,SWRIST
240          04660          ;
241          04680          ; PRINT TEST IDENTIFICATION MESSAGE
242          04700          ;
243 001146    012700    016772    MOV     #PROGID,R0
244 001152    104000          TYPE
245          04760          ;
246          04780          ; DETERMIN SYSTEM CONFIGURATION
247          04800          ; BUILD A TABLE OF INTERFACE LINES.
248          04820          ;
249 001154    004737    005640    JSR     PC,BUILD
250          04860          ;
251          04880          ; RESTORE TRAP CATCHER FROM 100 TO 1000
252          04900          ;
253 001160    004737    006126    JSR     PC,CATCH
254 001164    104002          PRTTBL
255          04940          ;
256          04960          ; FIND OUT IF OPERATOR WANTS TO USE
257          04980          ; CONSOLE OR SWITCHES FOR CONTROL
258          05000          ;
259 001166    004737    003350    JSR     PC,GETSRC
260 001172    004737    003332    START2: JSR    PC,CONSON
261 001176    032737    000020  001364    BIT     #SWCTL,PCFLAG
262          05060          BNE     500010
263          05080          ; PRINT A MENU OF AVAILABLE COMMANDS
264          05100          ;
265 001206    012700    017051    MOV     #L3,R0
266 001212    104000          TYPE
267 001214    012700    017056    MOV     #HEADR1,R0
268 001220    104000          TYPE
269 001222    012700    017103    MOV     #COMSUM,R0
270 001226    104000          TYPE
271 001230          500016:
272 001230    004737    003332    START3: JSR    PC,CONSON
273 001234          500020:
274          05260          ;
275 001234    032737    000020  001364    BIT     #SWCTL,PCFLAG
  
```

```

(9) 001242    001403          BEQ     500048
276          05340          ; IF IN SWITCH CONTROL GET CONTENTS OF SW REG.
277 001244    004737    003130    JSR     PC,GETSWS
278 001250    000421          BR      500056
(3) 001252          500048:
279          05420          ;
280          05440          ; IN CONSOLE CONTROL SIGNIFY READY
281          05460          ; AND READ COMMANDS FROM THE CONSOLE.
282          05480          ;
283 001252    012700    020123    MOV     #RDY,R0
284 001256    104000          TYPE
285 001260          500068:
286 001260    000001          WAIT
287 001262    032737    000200  001364    BIT     #ATTN,PCFLAG
(5) 001270    001001          BNE     500078
288 001272    000772          BR      500068
(3) 001274          500078:
289          05620          ;
290          05640          ; PRINT THE LINE TABLE IF REQUESTED.
291          05660          ;
292 001274    032737    010000  001364    BIT     #PRINTI,PCFLAG
(9) 001302    001404          BEQ     500108
293 001304    104002          BR      PRTTBL
294 001306    042737    010000  001364    BIC     #PRINTI,PCFLAG
295 001314          500108:
296 001314          500058:
297          05780          ;
298          05800          ; SET UP THE I/O DRIVER AREAS
299          05820          ; SET UP & EXECUTE REQUESTED TESTS,
300          05840          ;
301 001314    004737    002110    JSR     PC,LINMON
302 001320    012700    020411    MOV     #ALLON,R0
303 001324    004737    007010    JSR     PC,MTYPE          ;ISSUE A SELECT ALL COMMAND
304          05872          ;IN CASE THERE ARE SELECTIVE
305 001330    004737    001446    JSR     PC,ISTCTL          ;TERMINALS ON LINE.
306 001334    032737    004000  002032    BIT     #EOL,CFLAGS
(5) 001342    001406          BEQ     500038
307 001344    042737    004000  002032    BIC     #EOL,CFLAGS
308 001352    004737    001372    JSR     PC,RESTRT
309 001356    000726          BR      500028
(3) 001360          500038:
310 001360    000137    001172    05990    JMP START2
311 001364          500008:
312          06020          ;
313          06040          ; *****
314          06060          ; *
315          06080          ; * NOTE... TYPING CTL=G WHILE IN CONSOLE *
316          06100          ; * CONTROL MODE WILL CAUSE THE *
317          06120          ; * PCFLAG WORD TO BE PRINTED, *
318          06140          ; *
319          06160          ; *****
320 001364    06180    CTLBLK:          ;PROGRAM CONTROL BLOCK
321 001364    000001    06200    PCFLAG: ,WORD 1          ;PROGRAM CONTROL FLAGS
322 001366    000000    06220    TESTNO1: ,WORD 0        ;TESTNO
323 001370    000000    06240    TESTNO: ,WORD 0        ;TESTNO
324          06260          ;LINE#
  
```

```

325 06280 ;*****
326 06281 ; PCFLAG BIT DEFINITIONS *
327 06282 ;*****
328 06283 ;
329 06284 ; BIT 15 HALT0E HALT ON ERROR (S=15)
330 06285 ; BIT 14 L00P0E LOOP ON ERROR (S=14)
331 06286 ; BIT 13 INH0PT INHIBIT REPORTS (S=13)
332 06287 ; BIT 12 PR0MTT PRINT TABLE (S=12)
333 06288 ; BIT 11 DAT0IN DATA IN FROM KBD
334 06289 ; BIT 10 NEWTST CHANGE IN TEST NO.
335 06290 ; BIT 9 NEWMOD CHANGE IN MODE.
336 06291 ; BIT 8 LDONE END OF LINE TABLE REACHED
337 06292 ; BIT 7 ATTN ATTENTION !!!!!!!!
338 06293 ; BIT 6 SEQ SEQUENCE TESTS MODE
339 06294 ; BIT 5 MULTI MULTI LINE MODE
340 06295 ; BIT 4 SWCTL CONTROL VIA SWITCHES.
341 06296 ; BIT 3 DR0PC DROP LINE COMMAND
342 06297 ; BIT 2 ADDC ADD LINE COMMAND
343 06298 ; BIT 1 FLAG2 MODE 0 = NO CURRENT I/O TO CONSOLE
344 06299 ; BIT 0 FLAG1 1 = IN COMMAND INPUT MODE
345 06300 ; 2 = I/O TESTING OF CONSOLE
346 06301 ; 3 = ?
347 06305
348 06306
349 06307
350 06319 ;*****
351 06320 ; RESTART
352 06340 ;*****
353 06360
354 001372 012706 001100 06380 RESTR1: MOV #STACK,SP ;REINITIALIZE EVERYTING
355 001376 005037 016152 06400 CLR NEXT
356 001402 005037 016154 06420 CLR INTST
357 001406 012737 000001 001364 06440 MOV #1,PCFLAG
358 001414 005037 016162 06460 CLR NXTLIN
359 001420 005037 016160 06480 CLR ONLIN
360 001424 012737 016114 016164 06500 MOV #INBUF,PTR
361 001432 004737 000126 06520 JSR PC,CATCH
362 001436 000240 06540 NOP
363 001440 000005 06560 RESET
364 001442 000137 001230 06580 JMP START3
    
```

```

369 06680 ;*****
370 06700 ; TSTCTL THIS SECTION CONTROLS TEST SELECTION, TEST
371 06720 ; SEQUENCING, AND INTERFACES TO ERROR AND REPORT
372 06740 ; MODULES AS REQUIRED BY THE TEST MODULES.
373 06760 ;*****
374 06780
375 001446 013737 001366 016152 06800 TSTCTL: MOV TESTNO,NEXT ;GET TEST NO.
376 001454 013737 016152 016154 06820 LOOP1: MOV NEXT,INTST ;GET CURRENT TEST NO.
377 001462 004737 002056 06840 JSR PC,SU1EST
378 001466 004777 014464 06920 LOOP2: JSR PC,@TESTAD ;START TEST
379 06940 ;
380 06942 ;
381 06944 ; CHECK FOR ERROR FLAG FROM TEST
382 06946 ;
383 001472 032737 000020 001364 06953
384 (9) 001500 001414
385 001502 017737 014416 016100
386 001510 042737 003777 016100
387 001516 042737 174000 001364
388 001524 053737 016100 001364
389 001532
390 001532 032765 100000 000004 06953
391 (9) 001540 001414
392 001542 016537 000004 002032
393 001550 016537 000006 002034
394 07002 ;
395 07004 ; CALL ERROR HANDLER ROUTINE
396 07006 ;
397 001556 004737 005124
398 001562 042765 100377 000004
399 001570 000421
400 (3) 001572
401 07062 ;
402 07064 ; SEE IF TEST IS REPORTING DONE CONDITION
403 07066 ;
404 001572 032765 020000 000004
405 (9) 001600 001415
406 07082 ;
407 07084 ; UPDATE THE PASS COUNT THEN REPORT END OF PASS
408 07086 ;
409 001602 005265 000002
410 001606 042765 020000 000004
411 001614 052737 020000 002032
412 001622 016537 000002 002036
413 001630 004737 005436
414 001634
415 07222 ;
416 07224 ; IF LOOP ON ERROR IS SET AND AN ERROR IS
417 07226 ; DETECTED THE ERROR HANDLER WILL MAKE THE
418 07228 ; RETURN ADDRESS OF THE TEST ODD
419 07230 ;
420 07232 ; CHECK FOR ODD ADDRESS,,,IN LOOP MODE,,,
421 07234 ;
422 001634 032765 000001 000010
423 BIT #BIT0,RPC(R5)
    
```

```

(9) 001042 001413          BEQ      500158
421                                07242 ;
422                                07244 ; IF THE LOOP ON ERROR IS TURNED OFF THEN
423                                07246 ; CONTINUE TEST AT THE NEXT SUBTEST,
424                                07248 ;
425 001044 032737 040000 001364          BIT      #LOOP0E,PCFLAG
(9) 001052 001006          BNE      500165
426 001054 042765 000001 000010          BIC      #B170,RPC(R5)
427 001062 016537 000010 010156          MOV      RPC(R5),TESTAD
428 001070          500165;
429 001070 000456          BR       500175
(3) 001072          500158;
430                                07342 ;
431                                07344 ; CHECK TO SEE IF THE ITERATION COUNT IS COMPLETED
432                                07346 ;
433 001072 026565 000002 000000          CMP      PASCNT(R5),CILCNT(R5)
(9) 001700 001447          BLE      500200
434 001702 052737 040000 002032          BIS      #EOL,CFLAGS
435 001710 005037 002036          CLR      TSCCNT
436                                07402 ;
437                                07404 ; REPORT END OF TEST CONDIIION
438                                07406 ;
439 001714 004737 005436          JSR      PC,REPORT
440 001720 016565 000002 000000          MOV      PASCNT(R5),CILCNT(R5)
(6) 001726 066565 177776 000000          ADD      ICNT(R5),CILCNT(R5)
441                                07442 ;
442                                07444 ; IF IN SEQUENCE TESTS MODE SET UP NEXT TEST
443                                07446 ;
444 001734 032737 000100 001364          BIT      #SEQ,PCFLAG
(9) 001742 001423          BEQ      500210
445 001744 013737 010152 010154          MOV      NEXT,INTEST
446 001752 005237 010152          INC      NEXT
447                                07520 ;
448                                07522 ;
449                                07524 ; IF NEXT IS A NON EXISTANT TEST SET EOL
450                                07526 ; AND RETURN TO MONITOR FOR NEW COMMANDS
451                                07528 ;
452 001756 023727 010154 000005          CMP      INTEST,#5
(9) 001764 003407          BLE      500220
453 001766 005037 010152          CLR      NEXT
454 001772 052737 004000 002032          BIS      #EOL,CFLAGS
455 002000 000207          07600 ;
456 002002 000402          BR       RTS      PC
(3) 002004          500220;
457 002004 004737 002056          JSR      PC,SUATEST
458 002010          500236;
459 002010 000402          BR       500246
(3) 002012          500210;
460                                07602 ;
461                                07604 ; SET UP TEST ADDRESS FOR THE SAME TEST AGAIN,
462                                07606 ;
463 002012 004737 002056          JSR      PC,SUATEST
464 002016          500246;
465 002016 000403          BR       500258
(3) 002020          500200;
466                                07742 ;
  
```

```

467                                07744 ; RETURN TO TEST VIA ADDRESS SUPPLIED BY TEST
468                                07746 ;
469 002020 016537 000010 010156          MOV      RPC(R5),TESTAD
470 002026          500250;
471 002026          500175;
472 002026 000137 001466          07020 JMP      LOOP2
473                                00160 ;
474                                00180 ;*****
475 002032 000000          00200 CFLAGS; ,WORD 0 ;FLAGS
476 002034 000000          00220 TSCPTM; ,WORD 0 ;POINTER
477 002036 000000          00240 TSCCNT; ,WORD 0 ;PASCNT
478                                00260 ;
479                                00300 ;
480 002040 010164          00320 TSTBL; TEST0 ; TABLE OF TEST ADDRESSES *****
481 002042 010444          00340 TEST1
482 002044 011520          00360 TEST2
483 002046 013700          00380 TEST3
484 002050 014652          00400 TEST4
485 002052 015354          00420 TEST5
486 002054 177777          00440 =1
487                                00441 ;
488                                00442 ;
489                                00443 ;
490                                00444 ;
491                                00445 ;*****
492 00446 ;SUATEST INITIALIZES THE TEST ADDRESS POINTER
493 00447 ; FOR TEST # IN 'INTEST'
494 00448 ;*****
495                                00449 ;
496 002056          SUATEST;
497 002056 006337 010154          ASL      INTEST
498 002062 012700 002040          MOV      #TSTBL,R0
(6) 002066 063700 010154          ADD      INTEST,R0
499 002072 011037 010156          MOV      (R0),TESTAD
500 002076 006237 010154          ASR      INTEST
501 002102 005065 000004          CLR      MFLAGS(R5)
502 002106          500005;
(3) 002106          500018;
(2) 002106 000207          RTS      PC
503                                00480 ;
504                                00485 ;
505                                00490 ;
  
```

```

507          00500          ,SBTTL  LINE CONTROL & INITIALIZATION
508          00520          ;*****
509          00540          ;THIS SECTION CONTROLS THE SELECTION AND SEQUENCING
510          00560          ;OF SINGLE OR MULTIPLE LINES FOR TESTING.
511          00580          ;*****
512          00600
513          00620
514 002110          LINM00:
515 002110 032737 001000 001364          BIT          #NEWMOD,PCFLAG
(9) 002116 001427          BEQ          500020
516
517          00680          ;
518          00700          ; INITIALIZE THE DEVICE HANDLER ;
519          00720          ; SET UP A POINTER AREA WITH THE
520          00740          ; DEVICE ADDRESSES & VECTORS ETC.
521          00760          ;
522 002120 032737 000040 001364          BIT          #MULTI,PCFLAG
(9) 002126 001407          BEQ          500030
523 002130 004737 002332          JSR          PC,GVL
524 002134 004737 002432          JSR          PC,MTW
525 002140 004737 002556          JSR          PC,GNL
(3) 002146 000410          BR          500040
526          500030:
527          00880          ;
528          00900          ; GET SELECTED LINE NUMBER AND
529          00920          ; PULL THE DATA FROM THE TABLE.
530          00940          ;
531 002146 013737 001370 016160          MOV          LINENO,ONLIN
532 002154 004737 002432          JSR          PC,MTW
533 002160 013737 016160 016162          MOV          ONLIN,NXTLIN
534          500040:
535 002174 000402          BIC          #NEWMOD,PCFLAG
(3) 002176          BR          500050
536          500020:
537          00900          ;
538          00910          ; DO LINESEL SECTION FOR EACH DEVICE
539          00920          ; TO BE TESTED.
540          00940          ;
541 002176 004737 002204          JSR          PC,LINESEL
542 002202
543 002202
544 002202
545 002202
546 002202
547 002202
548 002202
549 002202          RTS          PC
550          00920          ;
551          00940          ;
552          00960          ; MULTIPLE LINES UNDER TEST ?
553          00980          ;
554 002204 032737 000040 001364          BIT          #MULTI,PCFLAG
(9) 002212 001426          BEQ          500020
555          00940          ;
556          00960          ; SET UP POINTER AREA WITH DATA FOR

```

```

556          00980          ; THE NEXT DEVICE TO BE TESTED
557          00990          ;
558 002214 013737 016160 016160          MOV          NXTLIN,ONLIN
559          009540          ;
560          009560          ; RESET EVERYTHING IF AT THE END OF OUR DEVICE LIST.
561          009580          ;
562 002222 023727 016160 177777          CMP          ONLIN,#=1
(9) 002230 001012          BNE          500030
563 002232 052737 000400 001364          BIS          #LDONE,PCFLAG
564 002240 004737 002332          JSR          PC,GVL
565 002244 004737 002432          JSR          PC,MTW
566 002250 004737 002556          JSR          PC,GNL
567 002254 000404          BR          500040
(3) 002256
568          009720          ;
569          009750          ; SET UP POINTER AREA FOR LINE = 'ONLIN'.
570          009780          ;
571 002256 004737 002432          JSR          PC,MTW
572 002262 004737 002556          JSR          PC,GNL
573 002266
574 002266 000420          BR          500050
(3) 002270          500020:
575 002270 004737 002432          JSR          PC,MTW
576 002274 032737 000200 016126          BIT          #SEL,DLFLAG
(9) 002302 001404          BEQ          500060
577          009920          ;
578          009940          ; CHECK TO SEE IF ALL DEVICES
579          009960          ; HAVE BEEN TESTED YET, SET LDONE FLAG.
580          009980          ;
581 002304 052737 000400 001364          BIS          #LDONE,PCFLAG
582 002312 000406          BR          500070
(3) 002314          500060:
583          10040          ;
584          10060          ; MAKE SHURE THAT WHEN TESTING A SINGLE
585          10080          ; DEVICE , IT DOESN'T GET DROPPED
586          10100          ; BECAUSE OF EXCESSIVE ERRORS.
587          10120          ;
588 002314 052737 000200 016126          BIS          #SEL,DLFLAG
589 002322 012700 020345          MOV          #E20,R0
590 002326 104000          TYPE
591 002330          500070:
592 002330          500050:
593 002330          500000:
(3) 002330          500010:
(2) 002330 000207          RTS          PC
594          10200          ;
595          10280          ;*****
596          10300          ; GVL THIS ROUTINE FINDS A VALID LINE FOR TESTING
597          10320          ;*****
598          10340
599          10360          GVL:
(4) 002332 010346          MOV          R3,=(SP)
600 002334 012700 016170          MOV          #LIN00,R3          ;GET ADDH OF LINE TABLE
601 002340 005713          TST          (R3)          ;LIN PRESENT?
602 002342 100412          BMI          G1D          ;YES BRANCH?
603 002344 062703 000010          ADD          #10,R3          ;POINT TO OTHER WORD

```

```
004 002350 020027 016770 10460      CMP      R0,#TABEND      ;END OF TABLE ?
005 002354 001371 10480      BNE      G1A            ;NO BRANCH
006 002356 012700 020306 10500      MOV      #E19,R0       ;NOTIFY OPERATOR = NO LINES
007 002362 104000 10520      G1C:    TYPE          ;
008 002364 000137 001372 10540      JMP      RESTR         ;
009 002370 105713 10560      G1D:    TSTB         (R3) ;LINE SELECTED?
010 002372 100364 10580      BPL     G1B           ;NO TRY ANOTHER LINE
011 002374 062703 000006 10600      ADD     #6,R3         ;POINT TO OTHER WORD
012 002400 011337 016160 10620      MOV     (R3),ONLIN   ;GET DATA FROM TABLE
013 002404 000337 016160 10640      SWAB   ONLIN        ;
014 002410 105037 016161 10660      CLRB  ONLIN+1       ;
015 002414 005037 000174 10680      CLR   SWLINE       ;
016 002420 113737 010160 000174 10700      MOVB  ONLIN,SWLINE ;
017 002426 012603 10740      MOV   (SP)+,R3     ;EXIT
018 002430 000207 10760      RTS   PC
019 10780      ;*****
020 10780      ; MTH THIS ROUTINE TRANSFERS TABLE DATA TO THE WORK AREA
021 10800      ;*****
022 10820
023 002432 10840      MTH:
024 002432 010346 10860      MOV   R3,=(SP)
025 002440 006303 016160 10880      MOV   ONLIN,R3      ;GET LINE NO,
026 002442 006303 10900      ASL  R3
027 002444 006303 10920      ASL  R3
028 002446 062703 016170 10940      ASL  R3             ;X8 FOR OFFSET
029 002452 012337 016126 10960      ADD  #LIN00,R3     ;ADD IN BASE ADDR
030 002456 012337 016130 10980      MOV  (R3)+,DLFLAG  ;GET FLAG WORD
031 002462 012337 016132 11000      MOV  (R3)+,DLADR   ;GET ADDRESS
032 002466 011337 016134 11020      MOV  (R3)+,DLVEC   ;GET VECTOR
033 002472 013737 016130 016136 11040      MOV  (R3),DLOTH    ;GET "OTHER WORD"
034 002500 062737 000002 016136 11060      ADD  #2,DVCRXB
035 002506 013737 016136 016140 11080      MOV  DVCRXB,DVCTAS
036 002514 062737 000002 016140 11100      ADD  #2,DVCTAS
037 002522 013737 016140 016142 11120      MOV  DVCTAS,DVCTXB
038 002530 013737 016132 016144 11140      MOV  DLVEC,TXVEC
039 002536 062737 000004 016144 11160      ADD  #4,TXVEC
040 002544 062737 000002 016142 11180      ADD  #2,DVCTXB
041 002552 012603 11200      MOV  (SP)+,R3
042 002554 000207 11240      RTS   PC
043 11260      ;*****
044 11260      ; GNL THIS ROUTINE FINDS THE NEXT VALID LINE TO TEST
045 11280      ;*****
046 11300
047 002556 11320      GNL:
048 002556 010346 11340      MOV   R3,=(SP)
049 002560 013703 016160 11360      MOV   ONLIN,R3      ;GET CURRENT LINE
050 002564 005203 11380      INC  R3             ;CURRENT +1
051 002570 006303 11400      ASL  R3
052 002572 006303 11420      ASL  R3
053 002574 062703 016170 11440      ASL  R3             ;X8 FOR OFFSET
054 002600 005713 11460      ADD  #LIN00,R3     ;ADD IN BASE ADDR OF TABLE
055 002602 100403 11480      GNL1: TST  (R3)      ;LINE PRESENT?
056 002604 062703 000010 11500      BMI  GN3           ;YES = BRANCH
057 002610 000773 11520      GN2:  ADD  #10,R3   ;POINT TO NEXT LINE ENTRY
058 002612 105713 11540      BR   GN1           ;CHECK NEXT
```

```
058 002612 105713 11540      GN3:  TSTB  (R3)      ;LINE SELECTED?
059 002614 100373 11560      BPL  GN2          ;NO TRY ANOTHER
060 002616 021327 177777 11580      CMP  (R3),#-1     ;END OF TABLE?
061 002622 001412 11600      BEQ  GN5          ;YES = BRANCH
062 002624 062703 000006 11620      ADD  #6,R3       ;GET "OTHER WORD"
063 002630 011337 016162 11640      MOV  (R3),NXTLIN ; = NEXT AVAILABLE LINE
064 002634 000337 016162 11660      SWAB NXTLIN
065 002640 105037 016163 11680      CLRB NXTLIN+1
066 002644 11700      GN4:
067 002646 000207 11720      MOV  (SP)+,R3
068 002650 012737 177777 016162 11740      RTS  PC          ;EXIT
069 002656 000772 11760      GN5:  MOV  #-1,NXTLIN ;SET NXTLIN TO -1 = NO SELECT
070 11780      BR   GN4
071 11800
072 11820      ;*****
073 11840      ;UPDATE ROUTINE TO UPDATE INTERFACE TABLE FROM COMMANDS
074 11860      ;*****
075 11880
076 002660 11900      UPDATE:
077 11902      ; SHIFT THE CONVERTED LINE NO. FOR AN
078 11904      ; OFFSET TO THE LINE TABLE,
079 11906      ;
080 002660 006337 004776 11930      ASL  DATA
081 002664 006337 004776 11950      ASL  DATA
082 002670 006337 004776 11970      ASL  DATA
083 11990      ; ADD IN THE BASE ADDRESS OF THE TABLE,
084 12000      ADD  #LIN00,DATA
085 12020      ;
086 12020      ; IF THE LINE SELECTED DOESN'T EXIST =
087 002702 032777 100000 002066 12040      ; SEND AN ERROR MESSAGE,
088 002710 001003 12060      ;
089 002712 004737 005100 500020:  HIT  #DLP,DATA
090 002716 000430 500028  HNE  500028
091 002720 500038  BR   PC,SELEERR
092 12100      ;
093 12120      ; ADDING A LINE SETS IT'S "SELECTED" FLAG
094 12140      ; AND CLEARS OUT THE ERROR COUNT FOR THAT LINE
095 12160      ;
096 002720 032737 000004 005002 500040:  BIT  #ADDC,TEMPF
097 002726 001415 500048  BEQ  500048
098 002730 052777 000200 002040 500050  BIS  #SEL,DATA
099 002736 042777 000037 002032 500052  BIC  #ABO1#17,DATA
100 002744 062737 000007 004776 500054  ADD  #7,DATA
101 002752 117737 002020 001370 500056  MOVB #DATA,LINENO
102 002760 000407 500058  BR   500058
103 12300      ;
104 12320      ; DROPPING A LINE JUST RESETS IT'S "SELECTED" FLAG
105 12340      ;
106 002762 032737 000010 005002 500060:  BIT  #DROPC,TEMPF
107 002770 001403 500068  BEQ  500068
108 002772 042777 000200 001776 500070  BIC  #SEL,DATA
109 003000 500060:  ;
```


706	003000				500058:	
707	003000				500038:	
708	003000	005037	004776		CLR	DATA
709	003004				500008:	
(1)	003004				500018:	
(2)	003004	000207			RTS	PC
710						
711						

12480
12500

713					12540	,SBTTL SWITCH REGISTER ROUTINES
714					12560	;*****
715					12580	;SWRTST TESTS FOR HARDWARE SWITCH REGISTER
716					12600	;*****
717	003006	012737	003040	000004	12620	SWRTST: MOV #4S,MACHER ;SU NXM TRAP TO 4S
718	003014	012737	000340	000006	12640	MOV #PRI7,MACHER+2
719	003022	005777	013076		12660	TST #SWR ;ACCESS SWITCH REG.
720	003026	000240			12680	NOP
721	003030	012737	177570	016124	12700	MOV #177570,SWR ;RETAIN HARDWARE POINTER
722	003036	000404			12720	BR 6S ;
723	003040	012737	000176	016124	12740	4S: MOV #SSWR,SWR ;SU FOR SOFTWARE SWITCH REG.
724	003046	022626			12760	ISP2 ;CLEAN THE STACK
725	003050	012737	000006	000004	12780	6S: MOV #6,MACHER ;RESET TRAP CATCHER
726	003056	005037	000006		12800	CLR MACHER+2
727	003062	000207			12820	RTS PC ;EXIT
728					12840	
729					12860	;*****
730					12880	; CTLGX THIS ROUTINE PRINTS THE PROGRAM CONTROL FLAGS ON THE CONSOLE.
731					12900	;*****
732					12920	
733	003064					CTLGX:
734	003064	013746	001364			MOV PCFLAG,=(SP)
735	003070	012746	000006			MOV #6,=(SP)
736	003074	012746	020230			MOV #SW+11,=(SP)
737	003100	004737	007566			JSR PC,O2ASC
738	003104	142737	000006	020226		BICB #6,SW+11
739	003112	012700	020215			MOV #6,SW+R0
740	003116	104000			13080	TYPE
741	003120	012700	017730			MOV #L1,R0
742	003124	104000			13120	TYPE
743	003126					500008:
(3)	003126					500018:
(2)	003126	000207				RTS PC
744					13160	
745					13180	;*****
746					13200	;GETSWS THIS ROUTINE READS THE SWITCH REGISTER AND
747					13220	; CONVERTS THE DATA TO THE APPROPRIATE CONTROL
748					13240	; FLAGS OR POINTERS.
749					13260	;*****
750					13280	
751	003130					GETSWS:
752					13320	;
753					13340	; STOP HERE FOR OPERATOR TO ENTER CONTROL SWITCHES
754					13360	;
755	003130	000000			13380	HALT
756	003132	017737	012766	016100		MOV #SWR,TEMP
757					13420	;
758					13440	; IF SWITCHES INDICATE A SINGLE LINE OR A SINGLE TEST
759					13460	; TO BE DONE STOP SU OPERATOR CAN ENTER LINE/TEST DATA
760					13480	;
761	003140	032737	000100	016100		BIT #SEQ,TEMP
(8)	003146	001404				BEQ 500028
(6)	003150	032737	000040	016100		BIT #MULTI,TEMP
(9)	003156	001016				BNE 500038
(6)	003160					500028:
762	003160	000000			13520	HALT

763	003162	017737	012736	016102		MOV	05WR,TEMP+2
764	003170	005037	001366			CLR	TESTNO
765	003174	113737	016102	001366		MOVH	TEMP+2,TESTNO
766	003202	005037	001370			CLR	LINENO
767	003206	113737	016103	001370		MOVH	TEMP+3,LINENO
768	003214				500038:		
769	003214	032737	000100	016100		BIT	#SEQ,TEMP
(9)	003222	001406				BEQ	500048
770	003224	052737	000100	001364		BIS	#SEQ,PCFLAG
771	003232	005037	001366			CLR	TESTNO
772	003236	000403				BR	500056
(3)	003240				500048:		
773	003240	042737	000100	001364		BIC	#SEQ,PCFLAG
774	003246				500058:		
775				13780			
776	003246	032737	000040	016100		BIT	#MULTI,TEMP
(9)	003254	001406				BEQ	500068
777	003256	052737	000040	001364		BIS	#MULTI,PCFLAG
778	003264	005037	001370			CLR	LINENO
779	003270	000403				BR	500078
(3)	003272				500068:		
780	003272	042737	000040	001364		BIC	#MULTI,PCFLAG
781	003300				500078:		
782				13940			
783	003300	052737	003200	001364		BIS	#ATTN1#NEWMOD1#NEWTST,PCFLAG
784				13980			
785	003306	042737	174037	001364		BIC	#174037,PCFLAG
786	003314	042737	003777	016100		BIC	#3777,TEMP
787	003322	053737	016100	001364		BIS	TEMP,PCFLAG
788	003330				500088:		
(3)	003330				500018:		
(2)	003330	000207			RTS	PC	
789				14100			

791				14840			,SBTTL CONSOLE TERMINAL ROUTINES
792				14860			*****
793				14880			;CONSON== ROUTINE TO INITIALIZE CONSOLE VECTOR AREA
794				14900			*****
795				14920			
796	003332	012737	003466	000060	14940	CONSON: MOV	#READKB,#TKV ;INTERRUPT TO "READKB"
797	003340	012737	000101	177560	14960	MOV	#01,#TKS
798	003346	000207			14980	RTS	PC
799				15000			
800				15020			*****
801				15040			; GETSRC
802				15060			THIS ROUTINE ASKS THE OPERATOR IF HE/SHE
803				15080			WANTS TO USE CONSOLE CONTROL, THEN SELTS
804				15100			A CONTROL FLAG ACCORDINGLY,
805				15120			*****
806				15140			
807	003350					GETSRC:	
808	003350	005077	012610			CLR	#PTR
809	003354	012700	020370			MOV	#CTLM,R0
810	003360	104000		15220		TYPE	
811	003362	012737	000001	177560		MOV	#1,#TKS
812	003370				500028:		
813	003370	032737	000200	177560		BIT	#READY,#TKS
(9)	003376	001410				BEQ	500048
814	003400	113777	177562	012556		MOVH	#TKB,#PTR
815	003406	004737	005044			JSR	PC,ECHO
816	003412	012700	017730			MOV	#L1,R0
817	003416	104000		15360		TYPE	
818	003420				500048:		
819	003420	005777	012540			TST	#PTR
(5)	003424	001001				BNE	500038
820	003426	000760				BR	500028
(3)	003430				500038:		
821	003430	142777	000200	012526		BICR	#200,#PTR
822	003436	027727	012522	000116		CMP	#PTR,#N
(9)	003444	001001				BNE	500056
(6)	003446	023727	016124	000176		CMP	#WR,#SSWR
(9)	003454	001403				BEQ	500058
823	003456	052737	000020	001364		BIS	#SWCTL,PCFLAG
824	003464				500056:		
825	003464				500088:		
(3)	003464				500018:		
(2)	003464	000207			RTS	PC	

```

827      15540 ;*****
828      15560 ; READKB THIS MODULE IS AN INTERRUPT HANDLER
829      15580 ;          FOR THE CONSOLE TERMINAL,
830      15600 ;*****
831      15620
832 003466 READKB:
833 003466 010046 MOV R0,=(SP)
834
835      15650 ;
836      15655 ; GET CHAR FROM KEYBOARD BUFFER REG.
837      15660 ; CLEAR PARITY BIT IF SET.
838      15665 ;
839 003470 013737 177562 003776 MOV #0TKB,RDSAV
840 003476 142737 000200 003776 BIC #200,RDSAV
841
842      15680 ;
843      15685 ; CHECK FOR DEVICE ERROR
844      15690 ;
845 003504 032737 100000 003776 BIT #MERR,RDSAV
846 (9) 003512 001400 BEQ 500020
847 003514 004737 005062 JSR PC,CMDERR
848 003520 005037 177560 CLR #0TKS
849 003524 000516 BR 500030
850 (3) 003520 500020:
851
852      15715 ;
853      15720 ; IF CMD CHAR WAS A CTL-G DO THE CTLGX ROUTINE.
854      15725 ; PRINT OUT PCFLAGS ON CONSOLE.
855      15730 ;
856 003526 123727 003776 000007 CMPB RDSAV,#CTLG
857 (9) 003534 001006 BNE 500040
858 003536 004737 003064 JSR PC,CTLGX
859 003542 012700 017730 MOV #L1,R0
860 003546 104000 15750 TYPE
861 003550 000504 BR 500050
862 (3) 003552 500040:
863
864      15756 ;
865      15757 ; IF IN I/O MODE PUT DATA IN I/O BUFFER
866      15758 ;
867 003552 032737 000002 001364 BIT #FLAG2,PCFLAG
868 (9) 003560 001410 BEQ 500060
869 003562 113711 003776 MOVB RDSAV,(R1)
870 003566 052737 004000 001364 BIS #DATAIN,PCFLAG
871 003574 005037 007564 CLR DELAYT
872 (3) 003602 000470 BR 500070
873
874      15776 ;
875      15777 ; IF IN COMMAND MODE PUT DATA IN INBUF
876      15778 ; AND CALL INTERPRITER
877      15779 ;
878 003602 032737 000001 001364 BIT #FLAG1,PCFLAG
879 (9) 003610 001406 BEQ 500100
880 003612 113777 003776 MOVB RDSAV,#PTR
881 003620 004737 004000 JSR PC,CSI
882 (3) 003626 000456 BR 500110
883
884      15796 ;
885      15797 ; CLEAR AND GO TO READY STATE.
886      15798 ;
  
```

```

875 003626 123727 003776 000003 CMPB RDSAV,#CTLC
876 (9) 003634 001003 BNE 500120
877 003636 004737 001372 JSR PC,RESTR1
878 003642 000447 BR 500130
879 (3) 003644 500120:
880
881      15811 ;
882      15812 ; CHECK FOR PRINT REPORTS COMMAND
883      15813 ;
884 003644 123727 003776 000020 CMPB RDSAV,#CTLP
885 (9) 003652 001004 BNE 500140
886 003654 042737 020000 001364 BIC #INHRPT,PCFLAG
887 (3) 003662 000437 BR 500150
888
889      15826 ;
890      15827 ; CHECK FOR NO REPORTS COMMAND
891      15828 ;
892 003664 123727 003776 000016 CMPB RDSAV,#CTLN
893 (9) 003672 001004 BNE 500160
894 003674 052737 020000 001364 BIS #INHRPT,PCFLAG
895 003702 000427 BR 500170
896 (3) 003704 500160:
897
898      15841 ;
899      15842 ; CHECK FOR HALT ON ERROR COMMAND
900      15843 ;
901 003704 123727 003776 000010 CMPB RDSAV,#CTLH
902 (9) 003712 001004 BNE 500200
903 003714 052737 100000 001364 BIS #HALTOE,PCFLAG
904 003722 000417 BR 500210
905 (3) 003724 500200:
906
907      15856 ;
908      15857 ; CHECK FOR LOOP ON ERROR COMMAND
909      15858 ;
910 003724 123727 003776 000014 CMPB RDSAV,#CTLL
911 (9) 003732 001004 BNE 500220
912 003734 052737 040000 001364 BIS #LOOPOE,PCFLAG
913 003742 000407 BR 500230
914 (3) 003744 500220:
915
916      15871 ;
917      15872 ; CHECK FOR CLEAR COMMAND
918      15873 ;
919 003744 123727 003776 000013 CMPB RDSAV,#CTLK
920 (9) 003752 001003 BNE 500240
921 003754 042737 140000 001364 BIC #HALTOE,#LOOPOE,PCFLAG
922
923      500240:
924      500230:
925      500215:
926      500175:
927      500150:
928      500130:
929      500110:
930      500070:
931      500050:
932      500030:
933
934      15935 ;
935      15940 ; TURN CONSOLE BACK ON & EXIT.
936      15945 ;
  
```

```

920 003762 012737 000101 177560          MOV    #101,0#TKS
921 003770 012600          MOV    (SP)+,R0
922 003772 000002          15960  RTI
923 003774          5000001
(3) 003774          5000101
(2) 003774 000207          RTS    PC
924          16500
925 003776 000000          16600  RDSAVI ,WORD 0
  
```

```

927          16640 ;*****
928          16660 ; CSI  COMMAND STRING INTERPRETER
929          16680 ;*****
930          16700
931 004000          CSI:
932          16740 ;
933          16760 ; IF CMD CHAR IS AN ESCAPE  ECHO A '$'
934          16780 ; AND SET MODE BACK TO 0,
935          16800 ;
936 004000 123727 003776 000033          CMPB  RDSAV,#ESC
(9) 004006 001030          BNE  500028
937 004010 112777 000044 012146          MOVB  #'$,0PTR
938 004016 004737 005044          JSR   PC,ECHO
939 004022 012700 017051          MOV   #L3,R0
940 004026 104000          16900  TYPE
941 004030 042737 000001 005002          BIC   #FLAG1,TEMPF
942          17100 ;
943          17120 ; MOVE NEW CONTROL FLAGS TO THE PCFLAG WORD,
944          17140 ; RESET THE BUFFER POINTER,
945          17160 ;
946 004036 013737 005002 001364          MOV   TEMPF,PCFLAG
947 004044 013737 005004 001366          MOV   TEMPI,TESTNO
948 004052 012737 016114 016164          MOV   #INBUF,PIR
949          17235 ; CLEAR ATTENTION FLAGS FROM TEMPF
950 004060 042737 017603 005002          BIC   #17603,TEMPF
951          17260
952 004066 000457          BR    500038
(3) 004070          5000281
953          17300 ;
954          17320 ; IF CMD CHAR WAS A DELETE  RESET THE BUFFER
955          17340 ; POINTER AND ECHO A CR/LF,
956          17360 ;
957 004070 123737 003776 016166          CMPB  RDSAV,DEL
(9) 004076 001007          BNE  500048
958 004100 012737 016114 016164          MOV   #INBUF,PIR
959 004106 012700 017730          MOV   #L1,R0
960          17440 TYPE
961 004112 104000          BR    500038
(3) 004116          5000481
962          17480 ;
963          17500 ; IF CMD CHAR WAS A RETURN ECHO A CR/LF
964          17520 ; AND CALL THE DECODER,
965          17540 ;
966 004116 123727 003776 000015          CMPB  RDSAV,#CR
(9) 004124 001021          BNE  500068
967 004126 012700 017730          MOV   #L1,R0
968 004132 104000          17600  TYPE
969 004134 004737 004230          JSR   PC,DECODE
970 004140 123727 016114 000121          CMPB  INBUF,#0
(9) 004146 001007          BNE  500078
971 004150 005037 001366          CLR  TESTNO
972 004154 005037 005004          CLR  TEMPT
973 004160 012737 000001 016152          MOV   #1,NEXT
974 004166          5000781
975 004166 000417          BR    500108
(3) 004170          5000681
  
```

```

976          17800 ;
977          17820 ; IF CMD CHAR WAS A '?' RETYPE THE COMMAND
978          17840 ; SUMMARY & GO TO READY CONDITION,
979          17860 ;
980 004170 123727 003776 000077          CMPB  RDSAV,#'?
(9) 004176 001007          BNE   500118
981 004200 012700 017103          MOV   #COMSUM,R0
982 004204 104000          17920 TYPE
983 004206 012700 020123          MOV   #RDY,R0
984 004212 104000          17960 TYPE
985 004214 000404          BR    500129
(3) 004216          500118;
986          18000 ;
987          18020 ; ECHO THE INPUT CHARACTER,
988          18040 ;
989 004216 004737 005044          JSR   PC,ECHO
990 004222 005237 016164          INC   PTR
991 004226          500129;
992 004226          500108;
993 004226          500058;
994 004226          500038;
995 004226          500008;
(3) 004226          500018;
(2) 004226 000207          RTS   PC
996          18220
997          18240
  
```

```

999          18280 ;*****
1000         18300 ;DECODE THIS SECTION DECODES THE COMMAND STRING FROM THE
1001         18320 ; CONSOLE, AND SETS THE APPROPRIATE CONTROL FLAGS.
1002         18340 ;*****
1003         18360
1004 004230          18400 DECODE:
1005
1006 004230 010046          MOV   R0,=(SP)
1007 004232 010146          MOV   R1,=(SP)
1008 004234 010246          MOV   R2,=(SP)
1009 004236 012702 001364          MOV   #CTLBLK,R2
1010 004242 012700 004656          MOV   #DECTL,R0
1011 004246 012701 016114          MOV   #INBUF,R1
1012 004252          500029;
1013         18560 ;
1014         18580 ; COMPARE CHAR IN TO FIRST BYTE OF TABLE
1015         18600 ;
1016 004252 121110          CMPB  (R1),(R0)
(9) 004254 001145          BNE   500048
1017         18640 ;
1018         18660 ; IF SAME GET FLAGS FROM THE TABLE TO TEMP:
1019         18680 ;
1020 004256 116037 000001 004774          MOVB  1(R0),DECSAV
1021 004264 056037 000002 005002          HIS  2(R0),TEMPF
1022 004272 046037 000004 005002          BIC  4(R0),TEMPF
1023 004300 005037 004776          CLR  DATA
1024 004304 005037 005000          CLR  DATA2
1025         18800 ;
1026         18820 ; SEE IF THIS COMMAND REQUIRES ADDITIONAL DATA
1027         18840 ;
1028 004310 032737 000340 004774          BIT   #REQ,DECSAV
(9) 004316 001520          BEQ   500058
1029 004320 126127 000001 000015          CMPB  1(R1),#CR
(9) 004326 001006          BNE   500068
1030         18900 ;
1031         18920 ; DATA REQUIRED BUT NOT PRESENT...ERROR
1032         18940 ;
1033 004330 004737 005062          JSR   PC,CHDERR
1034         18980 ;
1035         19000 ; IF A OR D COMMAND USE DATA FOR LINE NO,
1036         19020 ;
1037 004334 012737 016114 016164          MOV   #INBUF,PTR
1038 004342 000505          BR    500078
(3) 004344          500068;
1039         19080 ;
1040         19100 ; CONVERT THE CHARS TO OCTAL...DATA
1041         19120 ;
1042 004344 012746 004776          MOV   #DATA,=(SP)
1043 004350 116137 000001 004776          MOVB  1(R1),DATA
1044 004356 126127 000002 000015          CMPB  2(R1),#CR
(9) 004364 001003          BNE   500108
1045 004366 012746 000001          MOV   #1,=(SP)
1046 004372 000417          BR    500118
(3) 004374          500108;
1047 004374 116137 000002 004777          MOVB  2(R1),DATA+1
1048 004402 126127 000003 000015          CMPB  3(R1),#CR
  
```

```

(9) 004410 001003          BNE 500128
1049 004412 012746 000002    MOV #2,-(SP)
1050 004416 000405          BR 500138
(3) 004420          500128: MOVB 3(R1),DATA2
1051 004420 116137 000003 005000    MOV #3,-(SP)
1052 004426 012746 000003          500138: MOV #3,-(SP)
1053 004432          500118: MOV #DATA,-(SP)
1054 004432          JSR PC,A2BIN
1055 004432 012746 004776          19440 ;
1056 004436 004737 007676          19460 ; IF N COMMAND USE DATA AS A TEST NO.
1057          19480 ;
1058          ;
1059          ;
1060 004442 121027 000122          CMPB (R0),#*N
(9) 004446 001020          BNE 500148
1061          19520 ;
1062          19540 ; CHECK THE LIMITS FOR VALID TEST NO.
1063          19560 ;
1064          19580 ;***** #5 BELOW IS HIGHEST TEST NO THIS DIAGNOSTIC *****
1065 004450 005737 004776          TST DATA
(8) 004454 002404          BLT 500158
(6) 004456 023727 004776 000005    CMP DATA,#5
(9) 004464 003403          BLE 500168
(6) 004466          500158: JSR PC,SELERR
1066 004466 004737 005100          ;
1067          19640 ;
1068          19660 ; OUT OF RANGE ERROR,
1069          19680 ;
1070 004472 000403          BR 500178
(3) 004474          500168: MOV DATA,TEMPT
1071 004474 013737 004776 005004    500178: BIS *MERR,DATA2
1072 004502          500148:
1073 004502 052737 100000 005000    19800 ;
1074 004510          19820 ; IF W COMMAND USE DATA AS WIDTH
1075          19840 ;
1076          ;
1077          ;
1078 004510 121027 000127          CMPB (R0),#*W
(9) 004514 001005          BNE 500208
1079          19880 ;
1080          19900 ; GO CHECK FOR VALID LIMITS ON WIDTH ENTRY.
1081          19920 ;
1082 004516 004737 005006          JSR PC,CHKW
1083 004522 052737 100000 005000    BIS *MERR,DATA2
1084 004530          500208:
1085          20000 ;
1086          20020 ; IF ADDING OR DROPPING A LINE CALL UPDATE ROUTINE
1087          20040 ;
1088 004530 121027 000101          CMPB (R0),#*A
(8) 004534 001403          BEQ 500218
(6) 004536 121027 000104          CMPB (R0),#*D
(9) 004542 001005          BNE 500228
(6) 004544          500218:
1089          20080 ;
1090          20100 ; TAKE LINE NO. AND UPDATE INTERFACE TABLE
1091          20120 ;
  
```

```

1092 004544 004737 002660          JSR PC,UPDATE
1093 004550 052737 100000 005000    BIS *MERR,DATA2
1094 004556          500228:
1095 004556          500078: BR 500238
1096 004556 000403          BR 500238
(3) 004560          500058: BIS *MERR,DATA2
1097 004560 052737 100000 005000    500238: BR 500248
1098 004566          500048: ADD #6,R0
1099 004566 000415          BR 500248
(3) 004570          500048:
1100 004570 062700 000006          ;
1101          20320 ;
1102          20340 ; IF THE CHAR IN DOESN'T COMPARE TO ANY
1103          20360 ; TABLE ENTRY THE COMMAND IS INVALID
1104          20380 ;
1105 004574 020027 004774          CMP R0,#DTEMD
(9) 004600 001010          BNE 500258
1106 004602 004737 005002          JSR PC,CMDERR
1107 004606 012737 016114 016164    MOV #INBUF,PTR
1108 004614 052737 100000 005000    BIS *MERR,DATA2
1109 004622          500258:
1110 004622          500248:
1111          20520 ;
1112          20540 ; KEEP LOOKING AT CHAR UNTIL IT'S
1113          20560 ; DECODED, OR END OF TABLE (ERROR).
1114          20580 ;
1115 004622 032737 100000 005000    BIT *MERR,DATA2
(5) 004630 001001          BNE 500038
1116 004632 000607          BR 500028
(3) 004634          500038: CLR DATA2
1117 004634 005037 005000          ;
1118          20660 ;
1119          20680 ; RESET THE INPUT BUFFER POINTER
1120          20700 ;
1121 004640 012737 016114 016164    MOV #INBUF,PTR
1122 004646 012602          MOV (SP)+,R2
1123 004650 012601          MOV (SP)+,R1
1124 004652 012600          MOV (SP)+,R0
1125 004654          500008:
(3) 004654          500018: RTS PC
(2) 004654 000207          ;
1126          20820 ;
1127          20840 ;*****
1128 004656 123 000          DECTBLI ,BYTE *S,0 ;DECODE TABLE
1129 004660 001200 000040          ,WORD ATTN1NEWMOD,MULTI
1130 004664 115 000          ,BYTE *M,0 ;FIRST = CHAR TO BE DECODED
1131 004666 001240 000000          ,WORD ATTNMULTI1NEWMOD,0
1132 004672 121 000          ,BYTE *Q,0 ;SECOND = CONTROL BITS
1133 004674 000300 000000          ,WORD ATTN1SEQ,0
1134 004700 122 200          ,BYTE *R,200 ;THIRD = SET MASK
1135 004702 000200 000100          ,WORD ATTN,SEQ
1136 004706 104 100          ,BYTE *D,100 ;FOURTH = CLEAR MASK
1137 004710 002010 000004          ,WORD DROPC1NEWTST,ADDC
1138 004714 101 100          ,BYTE *A,100
1139 004716 002004 000010          ,WORD ADDC1NEWTST,DROPC
1140 004722 124 000          ,BYTE *T,0 ;CONTROL BITS
  
```

```

1141 004724 010000 000000 21120 ,WORD PRINTI,0
1142 004730 114 000 21140 ,BYTE 'L,0
1143 004732 040000 000000 21160 ,WORD LOOPOE,0
1144 004736 110 000 21180 ,BYTE 'N,0 ;5 = GET WIDTH
1145 004740 100000 000000 21200 ,WORD HALTOE,0
1146 004744 116 000 21220 ,BYTE 'N,0 ;6 = GET LINE N
1147 004746 020000 000000 21240 ,WORD INHRRPT,0
1148 004752 120 000 21260 ,BYTE 'P,0 ;7 = GEN TEST NO.
1149 004754 000000 020000 21280 ,WORD 0,INHRRPT
1150 004760 103 000 21300 ,BYTE 'C,0
1151 004762 000000 140000 21320 ,WORD 0,HALTOE|LOOPOE
1152 004766 127 040 000 21340 ,BYTE 'W,40,0,0,0,0
1153 004774 000 000 21360 DTENDI
1154 004774 000000 21380 DECSAVI ,WORD 0
1155 004776 000000 21400 DATA1 ,WORD 0
1156 005000 000000 21420 DATA2 ,WORD 0
1157 005002 000000 21440 TEMPF1 ,WORD 0 ;TEMPORARY PCFLAG WORD
1158 005004 000000 21460 TEMPT1 ,WORD 0 ;TEMPORARY TEST NO.
1159 21480
1160 21500
1161 21520
1162 21540 ;*****
1163 21560 ;CHKW THIS ROUTINE VALIDATES A "W" COMMAND
1164 21580 ;*****
1165 21600
1166 005006 CHKWI
1167 21621 ;
1168 21622 ; RANGE OF 26 THRU 132 CHARACTERS IS VALID
1169 21623 ;
1170 005006 023727 004776 000032 CMP DATA,#32
(8) 005014 002404 BLT 500028
(6) 005016 023727 004776 000204 CMP DATA,#132.
(9) 005024 003403 BLE 500038
(6) 005026 500028 JSR PC,SELERR
1171 005026 004737 005100 BR 500044
1172 005032 000403 500038 MOV DATA,#WIDTH
(3) 005034 013737 004776 016146 500048;
1174 005042 500006;
1175 005042 500015;
(3) 005042 000207 RTS PC
1176 21760
1177 21780
  
```

```

1179 21820 ;*****
1180 21840 ;ECHO CONSOLE KEYBOARD ECHO ROUTINE; PIR HAS ADDR OF CHAR
1181 21860 ;*****
1182 21880
1183 005044 105737 177564 ECHOI TSTB @PIB
1184 005050 100375 21900 BPL ECHO
1185 005052 117737 011106 177566 21920 MOVH @PIH,@PIB
1186 005060 000207 21960 RTS PC
1187 21980
  
```

```

1189          22020          ,SBTTL  ERROR & REPORT ROUTINES
1190          22040          ;*****
1191          22060          ;CMDERR  ROUTINE TO HANDLE INVALID COMMANDS
1192          22080          ;*****
1193          22100
1194          005062          CMDEERR:
1195          005062 012700 020017          MOV     #R1,R0
1196          005066 104000          TYPE   MOV     #RDY,R0
1197          005070 012700 020123          22160
1198          005074 104000          22200          TYPE
1199          005076          500003:
1200          (3) 005076          500018:
1201          (2) 005076 000207          RTS     PC
1202          22240          ;*****
1203          22260          ;SELEERR ROUTINE TO HANDLE SELECTION ERRORS
1204          22280          ;*****
1205          22300
1206          005100          SELEERR:
1207          005100 012700 020032          MOV     #R2,R0
1208          005104 104000          22360          TYPE
1209          005106 012700 020123          MOV     #RDY,R0
1210          005112 104000          22400          TYPE
1211          (3) 005114 012737 016114 016164          MOV     #INBUF,PTK
1212          (2) 005122          500003:
1213          005122 000207          500018:
1214          22460          RTS     PC
  
```

```

1213          22500          ;*****
1214          22520          ;ERRORS ERROR LOGGER AND TYPEOUT ROUTINE
1215          22540          ;
1216          22560          ;*****
1217          22580
1218          005124          ERROR:
1219          005124 005037 005434          CLR     ERRSAV
1220          005130 032737 020000 001364          BIT     #INHRPT,PCFLAG
1221          (9) 005136 001044          BNE    500028
1222          22642          ;
1223          22644          ;CONVERT TEST NO. FOR OUTPUT
1224          22646          ;
1225          005140 013746 016154          MOV     INTST,=(SP)
1226          005144 012746 000002          MOV     #2,=(SP)
1227          005150 012746 020001          MOV     #ERR0+16,=(SP)
1228          005154 004737 007566          JSR    PC,02ASC
1229          22722          ;
1230          22724          ;CONVERT ERROR NO. FOR OUTPUT
1231          22726          ;
1232          005160 113737 002032 005434          MOVB   CFLAGS,ERRSAV
1233          005166 013746 005434          MOV     ERRSAV,=(SP)
1234          005172 012746 000003          MOV     #3,=(SP)
1235          005176 012746 017770          MOV     #ERR0+7,=(SP)
1236          005202 004737 007566          JSR    PC,02ASC
1237          22822          ;
1238          22824          ;CONVERT LINE NO. FOR OUTPUT
1239          22826          ;
1240          005206 013746 016160          MOV     ONLIN,=(SP)
1241          005212 012746 000002          MOV     #2,=(SP)
1242          005216 012746 020011          MOV     #ERR0+24,=(SP)
1243          005222 004737 007566          JSR    PC,02ASC
1244          005226 012700 017761          MOV     #R0,R0
1245          005232 104000          TYPE
1246          22940          ;
1247          22960          ;
1248          22980          ; CLEAR THE ERROR FLAG
1249          23000          ;
1250          005234 042737 100377 002032          BIC     #MEHRRN,CFLAGS
1251          23040          ;
1252          23060          ; GET THE POINTER SUPPLIED BY THE PROGRAM
1253          23080          ; AND PRINT THE ERROR DESCRIPTION MSG.
1254          23100          ;
1255          005242 013700 002034          MOV     TSCPTR,R0
1256          005246 104000          TYPE
1257          005250          500028:
1258          005250 005037 005434          CLR     ERRSAV
1259          23200          ;
1260          23220          ; UPDATE THE ERROR COUNT FOR THE FAILING LINE
1261          23240          ;
1262          005254 013737 016160 005434          MOV     ONLIN,ERRSAV
1263          005262 006337 005434          ASL    ERRSAV
1264          (7) 005266 006337 005434          ASL    ERRSAV
1265          (7) 005272 006337 005434          ASL    ERRSAV
1266          005276 006337 016170 005434          ADD    #LIN00,ERRSAV
1267          005304 005277 000124          INC    #ERRSAV
1268          23340          ;
1269          23360          ; IF LOOP ON ERROR IS SET , MAKE THE
  
```



```

1266          23300 ; RETURN ADDRESS OF THE TEST ODD.
1267          23400 ; THE TEST CONTROLLER WILL USE THE OLD
1268          23420 ; RPC TO RE-DO THE SUBTEST.
1269          23440 ;
1270 005310 032737 040000 001364          BIT #LOOPOE,PCFLAG
1271 005316 001403          BEQ 500035
1272 005320 052765 000001 000010          BIS #BIT0,RPC(R5)
1273          500038;
1274          23520 ;
1275          23540 ; SEE IF LINE ABORT FLAG IS SET
1276          23560 ;
1277 005326 032777 000020 000100          BIT #ABO,ERRSAV
1278 005334 001431          BEQ 500046
1279          23600 ;
1280          23620 ; IF ABORT IS SET DESELECT THE LINE
1281          23640 ; UNLESS IT'S THE ONLY ONE BEING TESTED
1282          23660 ;
1283 005336 032737 000040 001364          BIT #MULTI,PCFLAG
1284 005344 001417          BEQ 500056
1285 005346 042777 000377 000060          BIC #SEL1#17,ERRSAV
1286 005354 013746 016160          MOV ONLN,=(SP)
1287 005360 012746 000002          MOV #2,=(SP)
1288 005364 012746 020270          MOV #DR1,=(SP)
1289 005370 004737 007566          JSR PC,02ASC
1290          23800 ;
1291          23820 ; NOTIFY OPERATOR THAT LINE WAS DROPPED
1292          23840 ;
1293          23880 ;
1294          23900 ;
1295          23920 ;
1296 005374 012700 020242          MOV #DR0,R0
1297 005400 104000          TYPE
1298          23980 ;
1299          23920 ; IF TESTING ONLY ONE LINE DONOT ALLOW IT TO BE DESELECTED
1300          23920 ;
1301          23920 ;
1302          23920 ;
1303 005402 000406          BR 500058;
1304 005404          BIS 500065;
1305 005404 052777 000200 000022          BIS #SEL,ERRSAV
1306 005412 042777 000020 000014          BIC #ABO,ERRSAV
1307 005420          500065;
1308 005420          500046;
1309          24040 ;
1310          24060 ; HALT HERE IF HALT ON ERROR IS SET
1311          24080 ;
1312 005420 032737 100000 001364          BIT #HALT0E,PCFLAG
1313 005426 001401          BEQ 500076
1314 005430 000000          HALT
1315          24120 ;
1316          500076;
1317          500005;
1318          500018;
1319          (3)          RTS PC
1320          24180 ;
1321 005434 000000          ERRSAV,WORD 0
1322          24200 ;
1323          24220 ;
1324          24240 ;
1325          24260 ; REPORT THIS ROUTINE HANDLES END OF TEST AND
1326          24280 ; END OF PASS REPORTS.
1327          24300 ;
1328          24320 ;
1329          24320 ;
1330          24320 ;
1331          24320 ;
1332          24320 ;
1333          24320 ;
1334          24320 ;
1335          24320 ;
1336          24320 ;
1337          24320 ;
1338          24320 ;
1339          24320 ;
1340          24320 ;
1341          24320 ;
1342          24320 ;
1343          24320 ;
1344          24320 ;
1345          24320 ;
1346          24320 ;
1347          24320 ;
1348          24320 ;
1349          24320 ;
1350          24320 ;
1351          24320 ;
1352          24320 ;
1353          24320 ;
1354          24320 ;
1355          24320 ;
1356          24320 ;
1357          24320 ;
1358          24320 ;
1359          24320 ;
1360          24320 ;
1361          24320 ;
1362          24320 ;
    
```

```

1315          24345 ; CHECK FOR END OF TEST CONDITION
1316          24350 ;
1317 005436 032737 040000 002032          BIT #EOT,CFLAGS
1318 005444 001423          BEQ 500026
1319          24365 ;
1320          24370 ; CONVERT TEST NO. FOR OUTPUT
1321          24375 ;
1322 005446 013746 016154          MOV INTST,=(SP)
1323 005452 012746 000002          MOV #2,=(SP)
1324 005456 012746 020210          MOV #EOTM+10,=(SP)
1325 005462 004737 007566          JSR PC,02ASC
1326          24445 ;
1327 005466 042737 040000 002032          SEND END OF TEST MESSAGE
1328          24450 ;
1329          24455 ;
1330 005474 012700 020166          MOV #EOTM,R0
1331 005500 004737 007010          JSR PC,M1YPE
1332 005504 012700 017051          MOV #L3,R0
1333 005510 004737 007010          JSR PC,M1YPE
1334          500028;
1335          24505 ;
1336          24510 ; CHECK FOR END OF PASS CONDITION
1337          24515 ;
1338 005514 032737 020000 002032          BIT #EOP,CFLAGS
1339 005522 001425          BEQ 500035
1340 005524 013746 016154          MOV INTST,=(SP)
1341          24545 ;
1342          24550 ; CONVERT TEST NO. FOR OUTPUT
1343          24555 ;
1344 005530 012746 000002          MOV #2,=(SP)
1345 005534 012746 020161          MOV #EOPM+19,=(SP)
1346 005540 004737 007566          JSR PC,02ASC
1347          24605 ;
1348          24610 ; CONVERT PASS NUMBER FOR OUTPUT
1349          24615 ;
1350 005544 013746 002036          MOV TSCCNT,=(SP)
1351 005550 012746 020145          MOV #EOPM+7,=(SP)
1352 005554 004737 010006          JSR PC,BIN2DA
1353          24685 ;
1354          24690 ; SEND END OF PASS MESSAGE.
1355          24695 ;
1356          24695 ;
1357 005560 012700 020136          MOV #EOPM,R0
1358 005564 004737 007010          JSR PC,M1YPE
1359 005570 042737 020000 002032          BIC #EOP,CFLAGS
1360          500036;
1361          500006;
1362          500018;
1363          (3)          RTS PC
1364          24800 ;
1365          24820 ;
1366          24840 ;
1367          24840 ;
1368          24840 ;
1369          24840 ;
1370          24840 ;
1371          24840 ;
1372          24840 ;
1373          24840 ;
1374          24840 ;
1375          24840 ;
1376          24840 ;
1377          24840 ;
1378          24840 ;
1379          24840 ;
1380          24840 ;
1381          24840 ;
1382          24840 ;
1383          24840 ;
1384          24840 ;
1385          24840 ;
1386          24840 ;
1387          24840 ;
1388          24840 ;
1389          24840 ;
1390          24840 ;
1391          24840 ;
1392          24840 ;
1393          24840 ;
1394          24840 ;
1395          24840 ;
1396          24840 ;
1397          24840 ;
1398          24840 ;
1399          24840 ;
1400          24840 ;
1401          24840 ;
1402          24840 ;
1403          24840 ;
1404          24840 ;
1405          24840 ;
1406          24840 ;
1407          24840 ;
1408          24840 ;
1409          24840 ;
1410          24840 ;
1411          24840 ;
1412          24840 ;
1413          24840 ;
1414          24840 ;
1415          24840 ;
1416          24840 ;
1417          24840 ;
1418          24840 ;
1419          24840 ;
1420          24840 ;
1421          24840 ;
1422          24840 ;
1423          24840 ;
1424          24840 ;
1425          24840 ;
1426          24840 ;
1427          24840 ;
1428          24840 ;
1429          24840 ;
1430          24840 ;
1431          24840 ;
1432          24840 ;
1433          24840 ;
1434          24840 ;
1435          24840 ;
1436          24840 ;
1437          24840 ;
1438          24840 ;
1439          24840 ;
1440          24840 ;
1441          24840 ;
1442          24840 ;
1443          24840 ;
1444          24840 ;
1445          24840 ;
1446          24840 ;
1447          24840 ;
1448          24840 ;
1449          24840 ;
1450          24840 ;
1451          24840 ;
1452          24840 ;
1453          24840 ;
1454          24840 ;
1455          24840 ;
1456          24840 ;
1457          24840 ;
1458          24840 ;
1459          24840 ;
1460          24840 ;
1461          24840 ;
1462          24840 ;
1463          24840 ;
1464          24840 ;
1465          24840 ;
1466          24840 ;
1467          24840 ;
1468          24840 ;
1469          24840 ;
1470          24840 ;
1471          24840 ;
1472          24840 ;
1473          24840 ;
1474          24840 ;
1475          24840 ;
1476          24840 ;
1477          24840 ;
1478          24840 ;
1479          24840 ;
1480          24840 ;
1481          24840 ;
1482          24840 ;
1483          24840 ;
1484          24840 ;
1485          24840 ;
1486          24840 ;
1487          24840 ;
1488          24840 ;
1489          24840 ;
1490          24840 ;
1491          24840 ;
1492          24840 ;
1493          24840 ;
1494          24840 ;
1495          24840 ;
1496          24840 ;
1497          24840 ;
1498          24840 ;
1499          24840 ;
1500          24840 ;
    
```

```
1363          00050 ;*****  
1364          00070 ; SETIO ROUTINE TO SET I/O MODE  
1365          00090 ;*****  
1366          00110  
1367          00130  
1368 005600          SETIO1  
1369 005600          500028;  
1370 005600 032737 000001 001364      BIT      #FLAG1,PCFLAG  
      (9) 005606 001003                BNE     500038  
1371 005610 052737 000003 001364      BIS     #FLAG1,#FLAG2,PCFLAG  
1372 005616  
1373 005616 032737 000001 001364      500038; BIT      #FLAG1,PCFLAG  
      (7) 005624 001765                BEQ     500028  
      (4) 005626 032737 000002 001364  BIT      #FLAG2,PCFLAG  
      (7) 005634 001761                BEQ     500028  
1374 005636  
      (3) 005636  
      (2) 005636 000207  
1375          00320  
1376          00340
```

```
1378          00700          ,SBTTL INTERFACE SIZER ROUTINES  
1379          00900          ;*****  
1380          00950          ;BUILD SUBROUTINE TO BUILD THE DEVICE TABLE USED  
1381          01000          ; IN MULTI LINE MODE.  
1382          01050          ;*****  
1383 005640 012737 000003 000066 01100  BUILD; MOV      #BPT,66          ;SET UP CONSOLE TRAP  
1384 005646 012737 000120 000004 01150  MOV      #46,MACHER          ;SET UP NXM TRAP  
1385 005654 012701 016170          01200  MOV      #LIN00,K1  
1386 005666 012137 016126          01220  151  MOV      (R1)+,DLFLAG  
1387 005664 012137 016130          01240  MOV      (R1)+,DLADR  
1388 005670 012137 016132          01260  MOV      (R1)+,DLVEC  
1389 005674 012137 016134          01280  MOV      (R1)+,DLOTH  
1390 005700 013737 016136 016140 01400  MOV      DLADR,DVCTXS  
1391 005706 062737 000004 016140 01430  ADD     #4,DVCTXS  
1392 005714 013737 016140 016142 01440  MOV      DVCTXS,DVCTXB  
1393 005722 062737 000002 016142 01450  ADD     #2,DVCTXB  
1394 005730 113737 016135 016160 01500  MOV     DLOTH+1,ONLIN  
1395 005736 005777 010166          01750  TST     @DLADR  
1396 005742 052737 100000 016126 01850  BIS     #DLP,DLFLAG          ;TKY TO ACCESS DVC,  
1397 005750 012737 000300 007564 01900  MOV     #30,DELAYT          ;SET DVC PRESENT FLAG  
1398 005756 112777 000076 010156 01925  MOV     #'>,@DVCTXB          ;SET UP FOR DELAY  
1399 005764 052777 000100 010146 01950  BIS     #10,@DVCTXS          ;TXMIT A ">" CHARACTER  
1400 005772 104006          02050  DELAYR          ;SET DVC TX INTR ENABLE  
1401 005774 005737 016132          02100  TST     DLVEC          ;WAIT FOR INTERRUPT  
1402 006000 001433          02125  BEQ     2$          ;IF ZERO NO INTERRUPT OCCURED  
1403          02150          ;NO INTERRUPT = BRANCH  
1404          02200          ;OTHERWISE DLVEC=ADDR THAT  
1405 006002 052737 000200 016126 02300  BIS     #SEL,DLFLAG          ;DVC INTERRUPTED TO VIA INMAP  
1406 006010 013741 016134          02350  381  MOV     DLOTH,=(R1)          ;SET SELECTED FLAG  
1407 006014 013741 016132          02400  MOV     DLVEC,=(R1)          ;PUT NEW INFORMATION  
1408 006020 013741 016130          02450  MOV     DLADR,=(R1)          ;INTO LINE TABLE  
1409 006024 013741 016126          02500  MOV     DLFLAG,=(R1)  
1410          02550  
1411 006030 062701 000010          02600  ADD     #10,R1          ;JUMP POINTER TO NEXT LINE  
1412 006034 020127 016770          02650  581  CMP     R1,#TABEND          ;ALL DONE ?  
1413 006040 001307          02700  BNE     1$          ;NO = DO NEXT LINE  
1414 006042 162701 000010          02750  681  SUB     #10,R1          ;CHECK LAST ENTRY  
1415 006046 005711          02800  TST     (R1)          ;FOR LINE PRESENT  
1416 006050 100403          02850  BMI     7$  
1417 006052 012711 177777          02900  MOV     #-1,(R1)          ;IF NOT SET IT TO END  
1418 006056 000771          02950  BR     6$  
1419 006060 012737 000006 000004 03100  781  MOV     #6,MACHER          ;OF TABLE  
1420 006066 000207          03150  RTS     PC          ;RESET TRAP CATCHER  
1421 006070 052737 000020 016126 03200  281  BIS     #ABO,DLFLAG          ;SET ABORT FLAG  
1422 006076 042737 000200 016126 03250  BIC     #SEL,DLFLAG          ;MAKE SURE LINE IS DESELECTED  
1423 006104 004737 005124          JBR     PC,ERROR  
1424 006110 012700 020055          03350  MOV     #ER7,R0          ;SU ERROR MSG  
1425 006114 104000          03400  TYPE          ;TYPE MSG ON CONSOLE  
1426 006116 000734          03450  HR     3$          ;FIX TABLE ENTRIES  
1427          03500  
1428 006120 062706 000004 03550  481  ADD     #4,SP          ;ERASE INTR FROM STACK  
1429 006124 000743          03600  BR     5$          ;GET NEXT LINE ENTRY  
1430          03650  
1431          03700          ;*****  
1432          03750          ; CATCH REPLACES TRAP CATCHER FROM 100 TO 1000 .  
1433          03800          ;*****
```

```

1434 006126 012700 001000 03850 CATCH: MOV #1000,R0 ;START AT 1000
1435 006132 005040 03900 18: CLR =(R0) ;PUT HALI IN PC+2
1436 006134 010037 016100 03950 MOV R0,TEMP
1437 006140 013740 016100 04000 MOV TEMP,=(R0) ;PUT PC+2 IN PC
1438 006144 020027 000100 04050 CMP R0,#100 ;FIN?
1439 006150 002370 04100 BGE 18 ;NO = DO MORE
1440 006152 012737 003466 000060 04200 MOV #READKB,0#TKV ;SU CONSOLE
1441 006160 000207 04250 RTS PC
1442 04300
1446 04500
  
```

```

1451 05400
1452 05450
1453 05500 ;*****
1454 05550 ;THIS SECTION CONTAINS THE HANDLER AND MOSI ROUTINES ACCESSED
1455 05600 ;BY TRAPS THROUGH LOCATION 30.
1456 05650 ;*****
1457 05700
1458 05750
1459 006162 011637 016112 05800 EMITROS: MOV (SP),TEMP+12
1460 006174 017737 007712 016110 05850 SUB #2,TEMP+12 ;GET REAL PC
1461 006202 042737 104400 016110 05950 MOV @TEMP+12,TEMP+10 ;GET EMT INSTRUCTION
1462 006210 082737 006230 016110 06000 BIC #104400,TEMP+10 ;MASK INSTR BITS
1463 006216 017737 007666 016112 06050 ADD #EMTABL,TEMP+10 ;ADD TABLE ADDR
1464 006224 000177 007662 06100 MOV @TEMP+10,TEMP+12
1465 06150
1466 06200
1467 006230 006240 06250 ;EVEN
1468 006232 007212 06300 EMTABL: ETYPE ;INSOLE TYPE ROUTINE
1469 006234 007162 06350 PRTLTB ;LINE TABLE PRINTER
1470 006236 007530 06400 INTRAP ;DL INTERRUPT CAICHER
        DELAYM ;DELAY ROUTINE
  
```

```

1472          06650          ,SBTTL I/O DRIVERS
1473          06700          ;*****
1474          06750          ;ETYPE CONSOLE OUTPUT ROUTINE, ENTER WITH ADDRESS OF
1475          06800          ; DATA IN R0, NULL TERMINATES OUTPUT,
1476          06850          ;*****
1477          06900
1478 006240 105710          ETYPE! TSTB (R0)          ;CHECK FOR NULL
1479 006242 001406          BEQ 38          ;EXIT ROUTINE
1480 006244 105737 177564 07050 181 TSTB 0#TPS          ;CHECK FOR TRANSMIT READY
1481 006250 100375          BPL 18          ;WAIT
1482 006252 112037 177566 07150 MOVB (R0)+,0#TPB          ;TRANSMIT CHARACTER
1483 006256 000770          BR ETYPE          ;GET NEXT CHAR
1484 006260 105737 177564 07250 381 TSTB 0#TPS          ;WAIT TILL ALL DONE
1485 006264 100375          BPL 38
1486 006266 000002          RTI 38          ; EXIT,..
1487          07300
1488          07350
1489          07400          ;*****
1490          07450          ;MECHO TERMINAL OUTPUT ROUTINE - SINGLE CHAR
1491          07500          ; CHAR IN R2
1492          07550          ; INTERRUPT DRIVEN ALL LINES
1493          07600          ;*****
1494 006270 010237 006646 07650 MECHO! MOV R2,MSAVE
1495 006274 010446          MOV R4,=(SP)
1496 006276 012702 020434 07750 MOV #STACK2,R2          ;INITIALIZE STACK2
1497 006302 005037 020670 07800 CLR ENDS          ;ZERO COUNT
1498 006306 012704 020574 07850 MOV #STACK3,R4          ;INITIALIZE STACK3
1499 006312 013722 016132 07900 181 MOV DLVEC,(R2)+          ;GET THE BASE VECTOR ADDR
1500 006316 013737 016144 016150 07950 MOV TXVEC,SAVE          ;SAVE THE VECTOR
1501 006324 062737 000002 016150 08000 ADD #2,SAVE          ;
1502 006332 013777 016150 007604 08050 MOV SAVE,ATXVEC          ;PUT ADDR+2 INTO ADDR
1503 006340 012777 000004 007602 08100 MOV #IOT,MSAVE          ;PUT TRAP INTO ADDR+2
1504 006346 012737 000310 007564 08150 MOV #200,DELAYT          ;WAIT FOR 200 MS.
1505 006354 113777 006646 007560 08200 MOVB MSAVE,0DVCTXB          ;PUT CHAR IN BUF REG
1506 006362 012777 000100 007550 08250 MOV #100,0DVCTXS          ;ENABLE IX INTERRUPT
1507 006370 005237 020670 08300 INC ENDS          ;ADD 1 TO INTR PENDING COUNT
1508 006374 004737 002110          JSR PC,INMON
1509 006400 032737 000400 001364 08400 BIT #LDONE,PCFLAG          ;END OF DVC LIST 7
1510 006406 001741          BEQ 18          ;NO DO THIS LINE
1511 006410 042737 000400 001364 08500 BIC #LDONE,PCFLAG
1512 006416 010237 006650 08550 MOV R2,MSAVE+2          ;SAVE STACK2 POINTER
1513 006422 104006          DELAYR ENDS
1514 006424 005737 020670 08650 TST ENDS          ;ALL PENDING INTERRUPTS SHOULD
1515 006430 001004          BNE 38          ;BE COUNTED DOWN BY TXTRAP,
1516 006432          1281 MOV (SP)+,R4
1517 006434 013702 006646          MOV MSAVE,R2
1518 006440 000207          RTS PC          ;EXIT,..
1519 006442 010437 006652 08900 381 MOV R4,MSAVE+4          ;SAVE STACK 3 LIMIT
1520 006446 012704 020574 08950 MOV #STACK3,R4          ;RESET STACK3 POINTER
1521 006452 012702 020434 09000 MOV #STACK2,R2          ;RESET STACK2 POINTER
1522 006456 021224 09050 481 CMP (R2),(R4)+          ;VECTOR MATCH ?
1523 006460 001404          BEQ 58          ;YES = BRANCH
1524 006462 020437 006652 09150 CMP R4,MSAVE+4          ;STACK END ?
1525 006466 001403          BEQ 68          ;YES = BRANCH
1526 006470 000772          BR 48          ;COMPARE NEXT VECT,
    
```

```

1527 006472 005012 09300 581 CLR (R2)          ;ERASE VECT OUT
1528 006474 005044 09350 CLR =(R4)          ;ERASE VECT IN
1529 006476 062702 000002 09400 681 ADD #2,R2          ;MOVE STACK POINTER
1530 006502 020237 006650 09450 CMP R2,MSAVE+2          ;END OF OUT STACK ?
1531 006506 001403 09500 BEQ 78          ;YES = GO GET ODD VECTOR
1532 006510 012704 020574 09550 MOV #STACK3,R4          ;RESET STACK3 POINTER
1533 006514 000760 09600 BR 48          ;KEEP SORTING
1534 006516 012702 020434 09650 781 MOV #STACK2,R2          ;RESET STACK2 POINTER
1535 006522 005712 09700 881 TST (R2)          ;CHECK FOR NON ZERO
1536 006524 001003 09750 BNE 98
1537 006526 062702 000002 09800 ADD #2,R2
1538 006532 000773 09850 BR 88
1539 006534 012737 016174 006652 09900 981 MOV #LIN00+4,MSAVE+4          ;GET VECT FROM TABLE
1540 006542 027712 000104 09950 1081 CMP 0MSAVE+4,(R2)          ;MATCH ?
1541 006546 001404 10000 BEQ 118          ;YES THIS LINE IS N,G.
1542 006550 062737 000010 006652 10050 ADD #10,MSAVE+4          ;MOVE POINTER TO NEXT
1543 006556 000771 10100 BR 108
1544 006560 062737 000002 006652 10150 1181 ADD #2,MSAVE+4          ;GET LINE NUMBER
1545 006566 017737 000006 016160 10200 MOV 0MSAVE+4,ONLIN
1546 006574 105037 016160 10250 CLR ONLIN
1547 006600 000337 016160 10300 SWAB ONLIN          ;ERASE JUNK BITS
1548 006604 004737 020432 10350 JSR PC,MTW          ;MOVE TABLE TO WORK AREAS
1549 006610 012737 020055 002034 10400 MOV #ER7,TSCPTR          ;POINT TO ERROR MESSAGE
1550 006616 112737 000377 020232 10450 MOV#R,377,CFLAGS          ;ERROR NO.
1551 006624 052737 100000 020432 10500 BIS #MERR,CFLAGS          ;SET ERROR FLAG
1552          10551          ;*****
1553          10600          ; ERROR 377
1554          10650          ;*****
1555 006632 004737 005124 10700 JSR PC,ERROR
1556 006636 042737 100377 020232 10750 BIC #MERR,CFLAGS          ;ERASE ERROR DATA
1557 006644 000672 10800 BR 128          ;CLEAN HOUSE & EXIT
1558 006646 000000 000000 000000 10850 MSAVE1 ,WORD 0,0,0
1559          10855
1560          10856
1561          10857
1562          10860          ;*****
1563          10861          ; SECHO SINGLE LINE ECHO ROUTINE
1564          10862          ; ENTER WITH CHAR IN R2
1565          10863          ; TRANSMITS TO DVC VIA I/O DRIVER WORK AREA
1566          10864          ;*****
1567          10865
1568          10866
1569 006654          SECHO!
1570 006654 013737 016144 016150 MOV TXVEC,SAVE
1571 006662 062737 000002 016150 ADD #2,SAVE
1572 006670 012777 007510 007246 MOV #STRAP,0TXVEC
1573 006676 012777 000200 007244 MOV #PRI4,MSAVE
1574 006684 012737 000144 007564 MOV #100,DELAYT
1575 006712 110277 007224 MOV#B,R2,0DVCTXB
1576 006716 012777 000100 007214 MOV #100,0DVCTXS
1577 006724 005237 020670 INC ENDS
1578 006730 104006 10877 DELAYR
1579 006732 005737 020670 TST ENDS
1580 006736 001413 BEQ 500028
1581 006740 012737 020055 002034 MOV #ER7,TSCPTR
1581 006746 052737 100376 002032 BIS #3761#MERR,CFLAGS
    
```

```

1434 006126 012700 001000 03850 CATCH: MOV #1000,R0 ;START AT 1000
1435 006132 005040 03900 18: CLR =(R0) ;PUT HALT IN PC+2
1436 006134 010037 016100 03950 MOV R0,TEMP
1437 006140 013740 016100 04000 MOV TEMP,=(R0) ;PUT PC+2 IN PC
1438 006144 020027 000100 04050 CMP R0,#100 ;FIN?
1439 006150 002370 04100 BGE 18 ;NO = DO MORE
1440 006152 012737 003466 000060 04200 MOV #READKB,0#TKV ;SU CONSOLE
1441 006160 000207 04250 RTS PC
1442 04300
1446 04500
  
```

```

1451 05400
1452 05450 ;SBTTL EMT HANDLER
1453 05500
1454 05550 ;*****
1455 05600 ;THIS SECTION CONTAINS THE HANDLER AND MOSI ROUTINES ACCESSED
1456 05650 ;BY THAPS THROUGH LOCATION 30.
1457 05700 ;*****
1458 05750
1459 006162 011637 016112 05800 EMTROS: MOV (SP),TEMP+12
1460 006174 017737 007712 016110 05850 SUB #2,TEMP+12 ;GET REAL PC
1461 006202 042737 104400 016110 05950 MOV #TEMP+12,TEMP+10 ;GET EMT INSTRUCTION
1462 006210 082737 006230 016110 06000 BIC #104400,TEMP+10 ;MASK INSTR BITS
1463 006216 017737 007666 016112 06050 ADD #EMTABL,TEMP+10 ;ADD TABLE ADDR
1464 006224 000177 007662 06100 MOV #TEMP+10,TEMP+12
1465 06150 JMP #TEMP+12
1466 06200 ;EVEN
1467 006230 006240 06250 EMTABL: ETYPE ;NSOLE TYPE ROUTINE
1468 006232 007212 06300 PRTLTB ;LINE TABLE PRINTER
1469 006234 007162 06350 INTRAP ;DL INTERRUPT CAICHER
1470 006236 007530 06400 DELAYM ;DELAY ROUTINE
  
```

```

1472          06650          ,SBTIL I/O DRIVERS
1473          06700          ;*****
1474          06750          ;ETYPE CONSOLE OUTPUT ROUTINE, ENTER WITH ADDRESS OF
1475          06800          ; DATA IN R0, NULL TERMINATES OUTPUT.
1476          06850          ;*****
1477          06900
1478 006240 105710          06950          ETYPE!  TSTB  (R0)          ;CHECK FOR NULL
1479 006242 001406          07000          BEQ  30          ;EXIT ROUTINE
1480 006244 105737 177564 07050          101  TSTB  0#TPS          ;CHECK FOR TRANSMIT READY
1481 006250 100375          07100          BPL  10          ;WAIT
1482 006252 112037 177566 07150          MOVB  (R0)+,0#TPB          ;TRANSMIT CHARACTER
1483 006256 000770          07200          BR  ETYPE          ;GET NEXT CHAR
1484 006260 105737 177564 07250          301  TSTB  0#TPS          ;WAIT TILL ALL DONE
1485 006264 100375          07275          BPL  30
1486 006266 000002          07287          RTI  30          ; EXIT,..
1487          07300
1488          07350
1489          07400
1490          07450          ;*****
1491          07500          ;MECHO  TERMINAL OUTPUT ROUTINE - SINGLE CHAR
1492          07550          ; CHAR IN R2
1493          07600          ; INTERRUPT DRIVEN ALL LINES
1494          07650          ;*****
1494 006270 012037 006646 MECHO!  MOV  R2,MSAVE
1495 006274 010446          MOV  R4,-(SP)
1496 006276 012702 020434          MOV  #STACK2,R2          ;INITIALIZE STACK2
1497 006302 005037 020670          CLR  ENDS          ;ZERO COUNT
1498 006306 012704 020574          MOV  #STACK3,R4          ;INITIALIZE STACK3
1499 006312 013722 016132          101  MOV  DLVEC,(R2)+          ;GET THE BASE VECTOR ADDR
1500 006316 013737          MOV  TXVEC,SAVE          ;SAVE THE VECTOR
1501 006324 062737 000002 016150 00000          ADD  #2,SAVE          ;
1502 006332 013777          MOV  SAVE,0TXVEC          ;PUT ADDR+2 INTO ADDR
1503 006340 012777 000004 007602 00100          MOV  #IOT,MSAVE          ;PUT TRAP INTO ADDR+2
1504 006346 012737          MOV  #200,DELAYT          ;WAIT FOR 200 MS.
1505 006354 113777 006646 007560 00200          MOV  MSAVE,0DVCTXB          ;PUT CHAR IN BUF REG
1506 006362 012777          MOV  #100,0DVCTXS          ;ENABLE TX INTERRUPT
1507 006370 005237 020670          INC  ENDS          ;ADD 1 TO INTR PENDING COUNT
1508 006374 004737          JSR  PC,LINMON
1509 006400 032737 000400 001364 00400          BIT  #LDONE,PCFLAG          ;END OF DVC LIST ?
1510 006406 001741          BEQ  10          ;NO DO THIS LINE
1511 006410 042737 000400 001364 00500          BIC  #LDONE,PCFLAG
1512 006416 010237 006650          MOV  R2,MSAVE+2          ;SAVE STACK2 POINTER
1513 006422 104006          DELAYR ENDS
1514 006424 005737 020670          TST  ENDS          ;ALL PENDING INTERRUPTS SHOULD
1515 006430 001004          BNE  30          ;BE COUNTED DOWN BY TXTRAP,
1516 006432          1201
1517 (4) 006432 012604          MOV  (SP)+,R4
1518 006434 013702 006646          MOV  MSAVE,R2
1519 006440 000207          201  RTS  PC          ;EXIT..
1520 006442 010437 006652          301  MOV  R4,MSAVE+4          ;SAVE STACK 3 LIMIT
1521 006446 012704 020574          MOV  #STACK3,R4          ;RESET STACK3 POINTER
1522 006452 012702 020434          MOV  #STACK2,R2          ;RESET STACK2 POINTER
1523 006456 021224          401  CMP  (R2),(R4)+          ;VECTOR MATCH ?
1524 006462 001404          BEQ  50          ;YES - BRANCH
1525 006466 020437 006652          CMP  R4,MSAVE+4          ;STACK END ?
1526 006470 000772          BEQ  60          ;YES - BRANCH
1527          BR  40          ;COMPARE NEXT VECT,

```

```

1527 006472 005012          09300  501  CLR  (R2)          ;ERASE VECT OUT
1528 006474 005044          09350          CLR  -(R4)          ;ERASE VECT IN
1529 006476 062702 000002          09400  601  ADD  #2,R2          ;MOVE STACK POINTER
1530 006502 020237 006650          09450          CMP  R2,MSAVE+2          ;END OF OUT STACK ?
1531 006506 001403          BEQ  70          ;YES - GO GET ODD VECTOR
1532 006510 012704 020574          09500          MOV  #STACK3,R4          ;RESET STACK3 POINTER
1533 006514 000760          BR  40          ;KEEP SORTING
1534 006516 012702 020434          701  MOV  #STACK2,R2          ;RESET STACK2 POINTER
1535 006522 005712          801  TST  (R2)          ;CHECK FOR NON ZERO
1536 006524 001003          BNE  90
1537 006526 062702 000002          901  ADD  #2,R2
1538 006532 000773          BR  80
1539 006534 012737 016174 006652          901  MOV  #LIN00+4,MSAVE+4          ;GET VECT FROM TABLE
1540 006542 027712 000104          1001  CMP  #MSAVE+4,(R2)          ;MATCH ?
1541 006546 001404          BEQ  110          ;YES THIS LINE IS N.G.
1542 006550 062737 000010 006652 10050          ADD  #10,MSAVE+4          ;MOVE POINTER TO NEXT
1543 006556 000771          BR  100
1544 006560 062737 000002 006652 10150          1101  ADD  #2,MSAVE+4,ONLIN          ;GET LINE NUMBER
1545 006566 017737 000060 016160 10200          MOV  #MSAVE+4,ONLIN
1546 006574 105037 016160          CLR  ONLIN
1547 006600 000337 016160          SWAB ONLIN          ;ERASE JUNK BITS
1548 006604 004737 002432          JSR  PC,MTW          ;MOVE TABLE TO WORK AREAS
1549 006610 012737 020055 002034 10400          MOV  #ER7,TSCPTR          ;POINT TO ERROR MESSAGE
1550 006616 112737 000377 002032 10450          MOVR #377,CFLAGS          ;ERRNO.
1551 006624 052737 100000 002032 10500          BIS  #MERR,CFLAGS          ;SET ERROR FLAG
1552          10551          ;*****
1553          10600          ; ERROR 377
1554          10650          ;*****
1555 006632 004737 005124          JSR  PC,ERROR
1556 006636 042737 100377 002032 10750          BIC  #MERR,CFLAGS          ;ERASE ERRNO DATA
1557 006644 000072          BR  120          ;CLEAN HOUSE & EXIT
1558 006646 000000 000000 000000 10850          MSAVE! ,WORD 0,0,0
1559          10855
1560          10856
1561          10857
1562          10860          ;*****
1563          10861          ; SECHO  SINGLE LINE ECHO ROUTINE
1564          10862          ; ENTER WITH CHAR IN R2
1565          10863          ; TRANSMITS TO DVC VIA I/O DRIVER WORK AREA
1566          10864          ;*****
1567          10865
1568          10866
1569          SECHO!
1570 006654          MOV  TXVEC,SAVE
1571 006662 062737 000002 016150          ADD  #2,SAVE
1572 006670 012777 007510 007246          MOV  #STRAP,0TXVEC
1573 006676 012777 000200 007244          MOV  #PR14,MSAVE
1574 006704 012737 000144 007564          MOV  #100,DELAYT
1575 006712 110277 007224          MOV  R2,0DVCTXB
1576 006716 012777 000100 007214          MOV  #100,0DVCTXS
1577 006724 005237 020670          INC  ENDS
1578 006730 104006          10877  DELAYR ENDS
1579 006732 005737 020670          TST  ENDS
1580 (9) 006736 001413          BEQ  500020
1580 006740 012737 020055 002034          MOV  #ER7,TSCPTR
1581 006746 052737 100376 002032          BIS  #376,0MERR,CFLAGS

```

```

1582 006754 004737 005124 JSR PC,ERROR
1583 006760 042737 100377 002032 BIC #MERRN,CFLAGS
1584 006766 500020:
1585 006766 013777 016150 007150 MOV SAVE,0TXVEC
1586 006774 005077 007150 CLR @SAVE
1587 007000 013737 016160 016150 MOV ONLIN,SAVE
1588 007006 500000:
(3) 007006 500010:
(2) 007006 000207 RTS PC
1589 10900
1590 10950
1591 11000 ;*****
1592 11050 ;MTYPE TERMINAL OUTPUT ROUTINE = LINE TABLE VERSION
1593 11100 ; ENTER WITH ADDR OF MSG IN R0
1594 11150
1595 11200 ;*****
1596 007010 112002 MTYPE: MOVB (R0)+,R2 ;GET CHAR TO PRINT
1597 007012 001403 BEQ 1$ ;EXIT IF NULL CHAR
1598 007014 004737 006270 JSR PC,MECHO
1599 007020 000773 BR MTYPE ;GET NEXT CHAR
1600 007022 000207 11400 1$: RTS PC ;EXIT...
1601 11450
1602 11462
1603 11464
1604 11466 ;*****
1605 11468 ; READIO THIS ROUTINE MONITORS AN I/O READ OPERATION
1606 11470 ;*****
1607 11472
1608 11474
1609 007024 READIO:
1610 007024 042737 004000 001364 BIC #DATAIN,PCFLAG
1611 007032 010237 007564 MOV R2,DELAYT
1612 007036 012777 000101 007064 MOV #101,@DLADR
1613 007044 104006 11480 DELAYR
1614 11481 ; IF NO CHAR RECVD WITHIN (R2) MS SET ERROR FLAG
1615 007046 032737 004000 001364 BIT #DATAIN,PCFLAG
(9) 007054 001003 BNE 500020
1616 007056 052765 100000 000004 BIS #MERR,MFLAGS(R5)
1617 007064 500020:
1618 11487 ; IF ON LINE=0 CLEAR I/O MODE FLAGS
1619 007064 105737 016160 TSTB ONLIN
(9) 007070 001004 BNE 500030
1620 007072 042737 000003 001364 BIC #FLAG1,#FLAG2,PCFLAG
1621 007100 000402 BR 500040
(3) 007102 500030:
1622 007102 005077 007022 CLR @DLADR
1623 007106 500040:
1624 007106 500000:
(3) 007106 500010:
(2) 007106 000207 RTS PC
1625 11500
1626 11502
1627 11504
1628 11506 ;*****
1629 11508 ; TYPES TERMINAL OUTPUT ROUTINE SINGLE LINE
1630 11510 ;*****

```

```

1631 11512
1632 11514
1633 007110 112002 11516 TYPES: MOVB (R0)+,R2 ;GET CHAR IO PRINT
1634 007112 001403 11518 BEQ 1$ ;EXIT IF NULL
1635 007114 004737 006654 11520 JSR PC,SECHO ;SEND THE MESSAGE
1636 007120 000773 11522 BR TYPES
1637 007122 000207 11524 1$: RTS PC ;EXIT
1638 11526
1639 11528
1640 11530 ;*****
1641 11532 ; READS THIS ROUTINE SETS UP DVC RECVR VECTOR AREAS
1642 11533 ; IF THE CURRENT LINE IS NOT LINE=00
1643 11534 ;*****
1644 11536
1645 11538
1646 007124 READS:
1647 007124 105737 016160 TSTB ONLIN
(9) 007130 001003 BNE 500020
1648 007132 004737 005600 JSR PC,SETIO
1649 007136 000410 BR 500030
(3) 007140 500020:
1650 007140 010277 006766 MOV R2,@DLVEC
1651 007144 013702 016132 MOV DLVEC,R2
1652 007150 062702 000002 ADD #2,R2
1653 007154 012712 000200 MOV #PRI4,(R2)
1654 007160 500030:
1655 007160 500000:
(3) 007160 500010:
(2) 007160 000207 RTS PC
1656 11554
1657 11556

```

```

1659          11595          ,SBTTL TRAP ROUTINES
1660          11600          ;*****
1661          11650          ;INTRAP: USED BY TABLE BUILD TO GET ADDRESS THAT A LINE
1662          11700          ; INTERRUPTS TO AN STORE IT IN = DLVEC,
1663          11750          ; TRANSMIT INTERRUPT USED, DLV HAS NO MAINT MODE,
1664          11800          ;*****
1665          007162 005077 006752 11875 INTRAP: CLR 0DVCTXS ;DISABLE THE INTERRUPTS.
1666          007166 012637 016132          MOV (SP)+,DLVEC
1667          007172 062706 000002 11900 ADD #2,SP ;SP+2 ADJUST STACK POINTER
1668          007176 162737 000010 11950 SUB #10,DLVEC ;ADJUST TO RCVR INTR ADDR
1669          007204 005037 007564 12000 CLR DELAYT ;RESET TIMER
1670          007210 000002          12050 RTI ;GO BACK TO BUILD ROUTINE
1671          12100
1672          12150          ;*****
1673          12200          ;PRTLTB THIS ROUTINE TYPES THE LINE TABLE ON THE CONSOLE
1674          12250          ; DEVICE, DROPPED FLAGS ARE DECODED AND THE
1675          12300          ; APPROPRIATE INFORMATION IS PRINTED FOR EACH LINE,
1676          12350          ;*****
1677          12400
1678          007212          12450 PRTLTB:
          (2) 007212 013746 016100          MOV TEMP,=(SP)
1679          007216 013746 016102          MOV TEMP+2,=(SP)
1680          007222 012702 016170          12650 MOV #LIN0,R2 ;POINTER TO ;START OF TABLE
1681          007226 012700 017676          12700 MOV #HEADR2,R0
1682          007232 104000          12750 TYPE ;PRINT HEADER
1683          007234 005712          12800 100 TST (R2) ;LINE PRESENT?
1684          007236 100406          12850 BMI 20 ;YES = BRANCH
1685          007240 062702 000010          12900 ADD #10,R2 ;MOVE POINTER TO NEXT ENTRY
1686          007244 021227 177777          12950 600 CMP (R2),#-1 ;END OF TABLE?
1687          007250 001452          13000 BEQ 100
1688          007252 000770          13050 BR 10 ;YES = BRANCH
1689          007254 012237 016100          13100 200 MOV (R2)+,TEMP ;SAVE FLAG WORD
1690          007260 012246          MOV (R2)+,=(SP)
1691          007262 012746 000004          MOV #4,=(SP)
1692          007266 012746 017743          MOV #DLAD,=(SP)
1693          007272 004737 007566          13300 JSR PC,02ASC ;CONVERT ADDRESS TO ASCII
1694          007276 012246          MOV (R2)+,=(SP)
1695          007300 012746 000003          13400 MOV #3,=(SP)
1696          007304 012746 017752          MOV #DLV,=(SP)
1697          007310 004737 007566          13500 JSR PC,02ASC ;CONVERT LINE NO,
1698          007314 012237 016102          13550 MOV (R2)+,TEMP+2
1699          007320 000337 016102          13600 SWAP TEMP+2
1700          007324 013746 016102          MOV TEMP+2,=(SP)
1701          007330 012746 000002          MOV #2,=(SP)
1702          007334 012746 017733          MOV #LIN,=(SP)
1703          007340 004737 007566          13800 JSR PC,02ASC
1704          007344 012700 017733          13850 MOV #LIN,R0 ;TYPE FORMATTED LINE
1705          007350 104000          13900 TYPE
1706          007352 105737 016100          13950 300 TSTB TEMP ;SELECTED?
1707          007356 001403          14000 BEQ 40 ;NO = BRANCH
1708          007360 012700 020117          14050 MOV #81,R0 ;SEND STAR
1709          007364 000402          14100 BR 50
1710          007366 012700 020105          14150 400 MOV #DR,R0 ;SEND DROPPED MSG
1711          007372 104000          14200 500 TYPE
1712          007374 000723          14250 BR 60
1713          007376          14300 1000
    
```

```

          (2) 007376 012637 016102          MOV (SP)+,TEMP+2
1714          007402 012637 016100          MOV (SP)+,TEMP
1715          007406 012700 017051          14400 MOV #L3,R0
1716          007412 104000          14450 TYPE
1717          007414 000002          14500 RTI
1718          14550
    
```



```

1720 14600 ;*****
1721 14650 ;TXTRAP THIS ROUTINE CATCHES THE INTERRUPTS FROM
1722 14700 ; DL11'S IN USE BY THE MECHO ROUTINE .
1723 14750 ;*****
1727 14950
1728 007416 162716 000010 15000 TXTRAP1 SUB #10,(SP) ;SUB 10 FROM UPDATED PC ON STACK
1729 007422 011614 15020 MOV (SP),(R4) ;PUT BASE VECTOR INTO STACK
1730 007424 012746 016174 15040 MOV #LIN00+4,-(SP) ;GET POINTER TO LINE TABLE VECTORS
1731 007430 027614 000000 15060 181 CMP 00(SP),(R4) ;COMPARE TABLE TO STACK
1732 007434 001403 15080 BEQ 20 ;SAME = BRANCH
1733 007436 062716 000010 15100 ADD #10,(SP) ;POINT TO NEXT TABLE ENTRY
1734 007442 000772 15120 BR 18 ;KEEP LOOKING FOR A MATCH
1735 007444 162716 000002 15130 201 SUB #2,(SP) ;ADDR OF OLADR NOW ON STACK
1736 007450 017637 000000 016100 15140 MOV 00(SP),TEMP ;GET OLADR FROM TABLE
1737 007456 062737 000004 016100 15160 ADD #4,TEMP ;POINT TO DVCTIX REGISTER
1738 007464 005077 006410 15180 CLR 0TEMP ;DISABLE INTERRUPTS
1739 007470 062706 000006 15200 ADD #6,SP ;SET STACK POINTER TO DRIVER PC
1740 007474 005337 020670 15220 DEC ENDS ;DECREMENT INTERRUPT PENDING COUNT
1741 007500 003002 15240 BGT 30 ;ABORT TIMEOUT IF ALL ACCOUNTED FOR
1742 007502 005037 007564 15260 CLR DELAYT ;RETURN TO I/O DRIVER
1743 007506 000002 15280 301 RTI
1744 15300
1745 15500
1746 15501
1747 15502
1748 15503
1749 15504 ;*****
1750 15505 ; STRAP SINGLE LINE TRANSMIT INTERRUPT CATCHER
1751 15506 ; USED IN CONJUNCTION WITH MECHO ROUTINE.
1752 15507 ;*****
1753 15509
1754
1755 007510
1756 007510 005077 006424 STRAP1 CLR 0DVCTIX
1757 007514 005337 020670 DEC ENDS
1758 007520 005037 007564 CLR DELAYT
1759 007524 000002 15515 RTI
1760 007526
1761 (3) 007526 5000001
1762 (2) 007526 5000101
1763 15517 RTS PC
1764 15518
1765 15519
1766 15550
1767 15750 ;*****
1768 15800 ;DELAYM DELAYS FOR X MILLI SECONDS, X STORED IN = DELAYT
1769 15850 ;
1770 15900 ;*****
1771 15950
1772 15955
1773 007530 005737 007564 DELAYM1 TST DELAYT
1774 007534 001411 15960 BEQ 30
1775 007536 010346 15960 MOV #3,-(SP)
1776 007540 013703 007562 16050 181 MOV ,MEXIT
1777 007544 005303 16100 201 TIMER,R3 ;1MS LOOP TIME
1778 007546 001376 16150 BNE 20

```

```

1779 007550 005337 007564 16200 DEC DELAYT
1780 007554 003371 16250 BGT 18
1781 007556 012003 16300 MOV (SP)+,R3
1782 007560 000002 16350 301 RTI ;MEXIT
1783 16400
1784 007562 000554 16450 TIMER1 ,WORD 554 ;SET FOR 11/35 = 11/40
1785 16500 ;SET TO 202 IF 11/03
1786 16550 ; 251 11/05 = 11/10
1787 16600 ; 314 11/15 = 11/20
1788 16650 ; 2127 11/45 FIFOLAN
1789 16700 ; 1237 11/45 = 11/70
1790 16750 ; 755 11/45 *05
1791 007564 000000 16800 DELAYT: ,WORD 0 ;DELAY TIME BUFFER

```

```

1793          16900          ,SBTTL CONVERSION ROUTINES
1794          16950
1795          17000
1796          17050 ;O2ASCI OCTAL TO ASCII CONVERSION ROUTINE - ENTER WITH
1797          17100 ; NUMBER TO BE CONVERTED ON THE STACK, FOLLOWED
1798          17150 ; BY THE NUMBER OF DIGITS TO CONVERT, FOLLOWED
1799          17200 ; BY THE STORAGE ADDRESS FOR THE ASCII STRING.
1800          17250 ;*****
1801 007566 016637 000006 016150 17300 O2ASCI! MOV 6(SP),SAVE ;GET WORK COPY OF NUMBER
1802 007574 013746 016150 17350 MOV SAVE,=(SP)
1803 007600 066666 000006 000004 17400 ADD 6(SP),4(SP) ;ADD COUNT TO POINTER
1804 007606 005366 000004 17450 DEC 4(SP) ;DEC FOR END ADDR
1805 007612 042716 177770 17500 28! BIC #177770,(SP) ;MASK OUT ALL BUT 3 BITS
1806 007616 052716 000006 17550 BIS #60,(SP) ;MAKE CHAR ASCII
1807 007622 111676 000004 17600 MOV#B (SP),04(SP) ;PUT ASCII CHAR IN BUFFER
1808 007626 005366 000004 17650 DEC 4(SP) ;INC POINTER
1809 007632 005366 000006 17700 DEC 6(SP) ;DEC DIGIT COUNT
1810 007636 001411 17750 BEQ 18 ;BRANCH IF DONE
1811 007640 006266 000010 17800 ASR 10(SP) ;
1812 007644 006266 000010 17850 ASR 10(SP) ;
1813 007650 006266 000010 17900 ASR 10(SP) ;GET NEXT DIGIT
1814 007654 016616 000010 17950 MOV 10(SP),(SP)
1815 007660 000754 18000 BR 28 ;DO NEXT CHAR FOR CONVERSION
1816 007662 016666 000002 000010 18050 18! MOV 2(SP),10(SP) ;PUT RETURN PC AT TOP OF JUNK
1817 007670 062706 000010 18100 ADD #10,5P ;POINT TO RETURN PC
1818 007674 000207 18150 RTS PC ;EXIT,,,
1819          18200
1820          18250 ;*****
1821          18300 ;A2BIN CONVERTS INPUT ASCII TO BINARY NUMBER
1822          18350 ; ENTER WITH ADDR OF ASCII STRING ON STACK
1823          18400 ; FOLLOWED BY 4 DIGITS TO CONVERT
1824          18450 ; FOLLOWED BY ADDR OF WORD FOR ANSWER,
1825          18500 ;*****
1826          18700
1830 007676          010046          A2BIN!
1831 007676 010046          MOV R0,=(SP)
1832 007700 005037 010004          CLR A2SAV
1833 007704 016600 000010          MOV 10(SP),R0
1834 007710          500028!
1835 007710 142710 000370          BIC#B #370,(R0)
1836 007714 005366 000006          DEC 6(SP)
1837 007720 152037 010004          BIS#B (R0)+,A2SAV
1838 007724 005766 000006          TST 6(SP)
1839 007730 001407          BEQ 500035
1840 007732 006337 010004          ASL A2SAV
1841 007736 006337 010004          ASL A2SAV
1842 007742 006337 010004          ASL A2SAV
1843 007746 000766          BR 500028
1844 007750          500035!
1845 007754 016600 000004          MOV 4(SP),R0
1846 007758 013710 010004          MOV A2SAV,(R0)
1847 007762 011637 010004          MOV (SP),A2SAV
1848 007766 016600 000002          MOV 2(SP),R0
1849 007770 062706 000010          ADD #10,5P
1850 007774 010016          MOV R0,(SP)
1851 007778 013700 010004          MOV A2SAV,R0
    
```

```

1848 010002          500005!
1849 (3) 010002          500018!
1850 (2) 010002 000207          RTS PC
1851          19600
1852 010004 000000          A2SAV! ,WORD 0 ;STORAGE AREA
1853          19650
1854          19750
1855          19850
1856          19950
1857          20050 ;*****
1858          20150 ;BIN2DA BINARY TO DECIMAL, ASCII CONVERSION ROUTINE
1859          20250 ; ENTER WITH NUMBER TO CONVERT ON THE STACK,
1860          20350 ; FOLLOWED BY THE ADDRESS OF THE ASCII BUFFER,
1861          20450 ; 5 DIGITS WILL BE CONVERTED
1862          20500 ;*****
1863          20650
1864          20750 ;BIN2DA! MOV #TABDA,R0 ;INITIALIZE TABLE POINTER
1865          20850 MOV#B #5,DIGITS
1866          20900 CLR CNTDA
1867          20950 18! CMP (R0),4(SP)
1868          21050 BGT 28
1869          21150 SUB (R0),4(SP)
1870          21250 INCR CNTDA
1871          21350 BR 18
1872          21450 28! BISR #60,CNTDA
1873          21550 TSTR FLAGDA
1874          21650 BNE 48
1875          21750 CMP#B CNTDA,#*0
1876          21850 BNE 38
1877          21950 MOV#B #177,CNTDA
1878          22050 BR 48
1879          22150 38! COMB FLAGDA
1880          22250 48! MOV#B CNTDA,02(SP)
1881          22350 INC 2(SP)
1882          22400 ADD #2,R0
1883          22450 CLR#B CNTDA
1884          22500 DECR DIGITS
1885          22550 BNE 18
1886          22600 MOV (SP),4(SP)
1887          22650 ADD #4,5P
1888          22700 RTS PC
1889          22750
1890          22800 TABDA! ,WORD 10000,,1000,,100,,10,,1
1891          22850 CNTDA! ,BYTE 0
1892          22900 FLAGDA! ,BYTE 0
1893          22950 DIGITS! ,BYTE 0,0
    
```

```

1898          00250          ,SBTTL LA36 OPTION TESTS
1899          00300          ;* * * * *
1900          00350          ;
1901          00400          ;TEST0 SECONDARY CHARACTER SET OPTION
1902          00450          ; NO MANUAL INTERVENTION REQUIRED
1903          00500          ;* * * * *
1904          00550          ;
1905 010164 012705 010352 00600 TEST0: MOV #T0BLK,R5 ;SET UP POINTER TO MODULE BLOCK
1906 010170 012700 010366 00650 MOV #T0,R0 ;SU TEST ID
1907 010174 004737 007010 JSR PC,MYTYPE
1908 010200 012700 010424 00750 T01: MOV #PRI,R0 ;SU PRIMARY MSG
1909 010204 004737 007010 JSR PC,MYTYPE
1910 010210 004737 010310 JSR PC,CHARS ;SEND ALL CHARACTERS
1911 010214 012700 017730 MOV #L1,R0
1912 010220 004737 007010 JSR PC,MYTYPE
1913 010224 012700 010434 01000 MOV #SEC,R0 ;SU SECONDARY MSG,
1914 010230 004737 007010 JSR PC,MYTYPE
1915 010234 012702 000016 01100 MOV #SO,R2 ;SEND SO - SELECT APL SET
1916 010240 004737 006270 JSR PC,MECHO
1917 010244 004737 010310 JSR PC,CHARS ;SEND ALL CHARS AGAIN
1918 010250 012700 017730 MOV #L1,R0
1919 010254 004737 007010 JSR PC,MYTYPE
1920 010260 012702 000017 01350 MOV #SI,R2 ;SEND SI=SELECT ASCII
1921 010264 004737 006270 JSR PC,MECHO
1922          01450          ;
1923 010270 052765 020000 000004 01500 BIS #TDONE,MFLAGS(R5) ;SET DONE AND ATTENTION FLAGS
1924 010276 012702 000012 01550 MOV #I2,R2 ;SU FOR LF
1925 010302 004737 006270 JSR PC,MECHO
1926 010306 000207 01650 RTS PC
1927          01700          ;*****
1928          01750          ; SUBROUTINE TO FILL OUTPUT LINE WITH ALL CHARACTERS
1929          01800          ;
1930 010310 013701 016146 01850 CHARS: MOV WIDTH,R1 ;SAVE WIDTH
1931 010314 012702 000040 01900 MOV #40,R2 ;SAVE START CHAR
1932 010320 162701 000007 01950 SUB #7,R1 ;ADJUST WIDTH FOR PRI/SEC MSG
1933 010324 010324 006270 02000 25: JSR PC,MECHO
1934 010330 005202 020500 INC R2 ;NEXT CHAR
1935 010332 020237 010364 02100 CMP R2,PUB ;LAST CHAR?
1936 010336 001403 02150 BEQ 36 ;YES = EXIT
1937 010340 005301 02200 DEC R1 ;END OF PAPER?
1938 010342 001401 02250 BEQ 36 ;YES = EX11
1939 010344 000767 02300 BR 26 ;SEND NEXT
1940 010346 000207 02350 38: RTS PC
1941 010350 000006 02400 ,WORD 6 ;ITERATION COUNT
1942 010352 000000 02450 T0BLK: ,WORD 0 ;CTLCNT
1943 010354 000000 02500 ,WORD 0 ;PASS COUNT
1944 010356 000000 02550 ,WORD 0 ;STATUS FLAGS
1945 010360 000000 02600 ,WORD 0 ;POINTER
1946 010362 010200 02650 ,WORD T01 ;RETURN PC
1947 010364 000177 02700 RUB: ,WORD 177
1948          02750          ,NLIST BEX
1949 010366 042524 052123 030040 02800 T0: ,ASCIZ *TEST 0 APL/ASCII CHAR SET=<15><12><12>
1950 010424 051501 044503 026511 02850 PRI: ,ASCIZ /ASCII=/
1951 010434 050101 026514 026455 02900 SEC: ,ASCIZ /APL=---/
1952          02950          ,EVEN

```

```

1954          03050          ;
1955          03100          ,LIST BEX
1956          03150          ;* * * * *
1957          03200          ;
1958          03250          ;TEST1 SELECTIVE ADDRESSING OPTION
1959          03300          ; OPERATOR MUST COMPARE TYPEOUT AND SWITCHES ON THE M7737
1960          03350          ; TO VERIFY CORRECT OPERATION,
1961          03400          ; IF A GROUP OR UNIT SELECT CODE OF LESS THAN 20(8)
1962          03450          ; IS USED MODIFY LOCATION GSEL ACCORDINGLY.
1963          03500          ;
1964          03550          ;* * * * *
1965          03600          ;
1966 010444          03700 TEST1: MOV #T0BLK,R5 ;SET UP POINTER TO MODULE BLOCK
1967 010444 012705 010722 03750 MOV #T1,R0
1968 010450 012700 010742 03750 JSR PC,MYTYPE
1969 010454 004737 007010 JSR PC,MYTYPE
1970 010460          03800 T11: ;DESELECT ALL TERMINALS, THEN TRY TO
1971          03900 ;PRINT ERROR MESSAGES,...SHOULD NOT PRINT
1972          03950 ;TRANSMIT A BAD SELECT SEQUENCE, THEN TRY TO
1973          04000 ;PRINT ERROR MESSAGES,...SHOULD NOT PRINT
1974          04050 ;SELECT ALL TERMINALS, PRINT GP MESSAGE.
1975          04100 ;
1976 010460 012765 010566 000010 MOV #T13,RPC(R5)
1977 010466 013701 010734 MOV GSEL,R1
1978 010472 012737 011176 010740 MOV #TABL1,T1TEMP+2
1979 010500 012737 000001 010736 MOV #1,T1TEMP
1980 (5) 010506 000402 500038: BR 500028
1981 (4) 010510 005237 010736 500025: INC T1TEMP
1982 (5) 010514 010514 010736 500010: CMP T1TEMP,#8.
1983 (7) 010522 003014 010736 500048: BGT 500048
1984 010524 012700 017730 MOV #L1,R0
1985 010530 004737 007010 JSR PC,MYTYPE
1986 010534 017700 000200 MOV #T1TEMP+2,R0
1987 010540 004737 007010 JSR PC,MYTYPE
1988 010544 062737 000002 010740 ADD #2,T1TEMP+2
1989 (3) 010554 000756 500045: BR 500038
1990          04650          ; TRANSMIT SELECT CODES TO ALL TERMINALS
1991          04700          ; FOLLOWED BY ASCII EQUIV OF CODE.
1992          04750          ;
1993 010554 012765 010460 000010 T12: MOV #T11,RPC(R5)
1994 (4) 010554 013701 010734 MOV GSEL,R1
1995 010562 013701 010734 04850 T13: ;OUTPUT ALL CODES AND ACCII EQUIVELANTS
1996 010566 010566 04850 500055: ;
1997 010566 020127 000200 CMP R1,#200
1998 (5) 010572 001420 500066: BEQ 500066
1999 010574 012700 020421 MOV #SCODE,R0
2000 010600 110100 000003 MOV R1,3(R0)
2001 010604 004737 007010 JSR PC,MYTYPE
2002 010610 112702 000002 MOV #STX,R2
2003 010614 004737 006270 JSR PC,MECHO
2004          05250          ; NOW CONVERT SELECT CODE TO ASCII FOR OUTPUT
2005 010620 004737 010662 JSR PC,CON
2006 010624 004737 007010 JSR PC,MYTYPE

```

```

2001 010630 005201          INC      R1
2002 010632 000755          BR       500065;
(3) 010634                    ; TURN ALL TERMINALS ON AND EXIT TEST
2003                    05500
2004 010634                    05550 T16;
(4) 010634 012765 010460 000010 MOV      #T11,RPC(R5)
2005 010642 052765 020000 000004 BIC     #TDONE,MFLAGS(R5)
2006 010650 012700 020411 MOV     #ALLON,R0
2007 010654 004737 007010 JSR     PC,MYTYPE
2008 010660                    500005;
(3) 010660                    500015;
(2) 010660 000207          RTS      PC
2009                    05750
2010                    05800 CON;
2011 010662                    ; THIS ROUTINE CONVERTS THE SELECT CODE
2012 010662 005037 010736 CLR     T1TEMP
2013 010666 110137 010736 MOV     R1,T1TEMP
2014 010672 013746 010736 MOV     T1TEMP,=(SP)
2015 010676 012746 000003 MOV     #3,=(SP)
2016 010702 012746 011520 MOV     #OCTALC,=(SP)
2017 010706 004737 007566 JSR     PC,O2ASC
2018 010712 012700 011520 MOV     #OCTALC,R0
2019 010716                    500005;
(3) 010716                    500015;
(2) 010716 000207          RTS      PC
2020                    06300
2021 010720 000002          06350 T01BLK;
2022 010722 000000          06400      ,WORD 2          ;ITERATION COUNT
2023 010724 000000          06450      ,WORD 0          ;CTLCNT
2024 010726 000000          06500      ,WORD 0          ;PASS COUNT
2025 010730 000000          06550      ,WORD 0          ;STATUS FLAGS
2026 010732 010460          06600      ,WORD 0          ;POINTER
2027 010734 000020          06650 GSEL;   ,WORD 20          ;RETURN PC
2028 010736 000000 000000 06700 T1TEMP; ,WORD 0,0          ;START OF SELECT CODES
2029                    06750
2030 010742 005015 052012 051505 06800 ;NLST BEX
2031 011005 105 051122 051117 06850 T11   .ASCIZ <15><12><12>/TEST 1 SELECTIVE ADDRESSING/<15><12><12>
2032 011045 116 020117 042523 06900 E9;   .ASCIZ /ERROR = THIS SHOULD NOT PRINT */
2033 011100 042523 042514 052103 06950 E12;  .ASCIZ /NO SELECT CHARACTER SENT/<15><12>
2034 011137 101 046114 052040 07000 GP;   .ASCIZ /SELECT CHARACTERS RECOGNIZED =/
2035                    07050 E10;  .ASCIZ /ALL TERMINALS SHOULD BE OFF/<15><12>
2036 011176 020416 011005 011137 07100 .EVEN
2037                    07150 TABL1; ,WORD ALLOFF,E9,E10,NSELC,E9,E12,ALLON,GP
2038                    07200
2039                    07250
2040                    07300
2041                    07350 ;*****
2042                    07400 ;GETANS THIS ROUTINE SETS UP AND READS THE ANSWERBACK
2043                    07450 ; MESSAGE FROM THE TERMINAL UNDER TEST.
2044                    07500 ;*****
2045                    07550
2046                    07600
2047                    07650
2048 011216
2049 011216 010337 011304 GETANS;
2050 011222 012702 013144 MOV     R3,26
MOV     #T220,R2
  
```

```

2051 011226 004737 007124 JSR     PC,READS
2052 011232 012702 000005 MOV     #ENO,R2
2053 011236 004737 006654 JSR     PC,SECHO
2054 011242                    00000 15;
(4) 011242 013702 011304 MOV     #28,R2
2055 011246 004737 007024 JSR     PC,READIO
2056 011252 032765 100000 000004 BIT     #MERR,MFLAGS(R5)
(9) 011260 001405 BEQ     500025
2057 011262 042765 100000 000004 BIC     #MERR,MFLAGS(R5)
2058 011270 105011 CLRB   (R1)
2059 011272 000403 BR      500035
(3) 011274                    500028;
2060 011274 105237 013212 INCR   T2CNT1
2061 011300 000760 00350 BR      18
2062 011302                    500036;
2063 011302                    500008;
(3) 011302                    500018;
(2) 011302 000207          RTS      PC
2064 011304 000000          00500 25;
2065                    00550      ,WORD 0
2066                    00600
2067                    00650 ;*****
2068                    00700 ;TYPANS THIS ROUTINE PRINTS THE ANSWERBACK MESSAGE
2069                    00750 ; IN OCIAL FORMATT , AND ASCII FORMATT.
2070                    00800 ;*****
2071                    00850
2072                    00900
2073 011306 TYPANS;
2074                    00900
2075 011306 012700 020421 MOV     #SCODE,R0
2076 011312 004737 007110 JSR     PC,TYPES
2077 011316 012702 000002 MOV     #SIX,R2
2078 011322 004737 006654 JSR     PC,SECHO
2079 011326 012700 011472 MOV     #ANSHDR,R0
2080 011332 004737 007110 JSR     PC,TYPES
2081 011336 013746 013212 MOV     T2CNT1,=(SP)
2082 011342                    09337 16;
(4) 011342 005046 CLR     =(SP)
2083 011344 112116 MOV     (R1)+,(SP)
2084 011346 012746 000003 MOV     #3,=(SP)
2085 011352 012746 011520 MOV     #OCTALC,=(SP)
2086 011356 004737 007566 JSR     PC,O2ASC
2087 011362 012700 011520 MOV     #OCTALC,R0
2088 011366 004737 007110 JSR     PC,TYPES
2089 011372 105337 013212 DECB   T2CNT1
2090 011376 105737 013212 TSTB  T2CNT1
(9) 011402 003402 BLE     500028
2091 011404 000756 BR      18
2092 011406 000426 BR      500038
(3) 011410                    500025;
2093 011410 012700 017730 MOV     #L1,R0
2094 011414 004737 007110 JSR     PC,TYPES
2095 011420 012700 000023 MOV     #19,R0
2096 011424                    500048;
2097 011424 012702 000040 MOV     #40,R2
2098 011430 004737 006654 JSR     PC,SECHO
  
```

2099	011434	005300							DEC	R0
2100	011436	005700							TST	R0
(5)	011440	001401							BEQ	500050
2101	011442	000770							BR	500040
(3)	011444							500050:		
2102	011444	012700	013220						MOV	#T2BUF,R0
2103	011450	004737	007110						JSR	PC, TYPES
2104	011454	012700	017730						MOV	#L1,R0
2105	011460	004737	007110						JSR	PC, TYPES
2106	011464							500030:		
2107	011464	012637	013212						MOV	(SP)+, T2CNT1
2108	011470							500000:		
(3)	011470							500010:		
(2)	011470	000207							RTS	PC
2109	011472	005015	047101	053523	10100			ANSHDR:	ASCIZ	<15><12>/ANSWERBACK RECVD = /
2110	011520	030060	027460	000	10150			OCTALC:	ASCIZ	*000/*
2111	011520				10200					,EVEN
2112					10250					

2114					10350				.LIST	BEX
2115					10400				;	***
2116					10450				;	*****
2117					10500				;	TEST2
2118					10550				;	AUTO ANSWER BACK OPTION
2119					10600				;	SINGLE LINE TESTS REQUIRE MANUAL INTERVENTION
2120					10650				;	*****
2121					10700				;	*****
2122					10750				.ENABL	LSB
2123	011526								TEST2:	
2124	011526	012700	013326		10850				MOV	#T2,R0
2125	011532	012701	013220		10900				MOV	#T2BUF,R1
2126	011536	012705	013176		10950				MOV	#T2BLK,R5
2127	011542	004737	007010						JSR	PC, MTYPE
2128	011546				11050			T211	;	IF THE LINE UNDER TEST HASN'T BEEN SIZED
2129					11100				;	FOR THE ANSWERBACK OPTION DO SO NOW.
2130					11150				;	
2131	011546	005737	016134						;	
(9)	011552	001065							TSTB	DLOTH
2132	011554	012701	013220						BNE	500020
2133					11300				MOV	#T2BUF,R1
2134					11350				;	CHECK DLOTH ENTRY OF LINE TABLE FOR CURRENT
2135					11400				;	LINE, IF LOBYTE IS 0 NO SIZE HAS BEEN DONE.
2136					11450				;	IF = 200 LINE SIZED BUT NO ANSWER RECVD.
2137	011560	013737	010734	013214					;	
2138	011566	013737	016160	013216					MOV	GSEL, T2TEMP
(7)	011574	006337	013216						MOV	ONLN, T2TEMP+2
(7)	011600	006337	013216						ASL	T2TEMP+2
(7)	011604	006337	013216						ASL	T2TEMP+2
2139	011610	0062737	016176	013216					ASL	T2TEMP+2
2140	011616	112777	000200	001372					ADD	#LINE0+6, T2TEMP+2
2141	011624							500030:	MOV	#200, #T2TEMP+2
2142	011624	023727	013214	000200					CMP	T2TEMP, #200
(5)	011632	001435							BEQ	500040
2143					11800				;	SEND EACH POSSIBLE SELECT CODE TO THE
2144					11850				;	TERMINAL, THEN REQUEST AN ANSWERBACK.
2145					11900				;	IF AN ANSWER IS REVIEWED STORE THE SELECT
2146					11950				;	CODE IN DLOTH ENTRY OF THE LINE TABLE.
2147					12000				;	OTHERWISE SET DLOTH TO 200.
2148	011634	113737	013214	020424					MOV	T2TEMP, SCODE+3
2149	011642	012700	020421						MOV	#SCODE, R0
2150	011646	004737	007110						JSR	PC, TYPES
2151	011652	012703	000310						MOV	#200, R3
2152	011656	105037	013212						CLRB	T2CNT1
2153	011662	004737	011216						JSR	PC, GETANS
2154	011666	105737	013212						TSTB	T2CNT1
(9)	011672	001412							BEQ	500050
2155	011674	113777	013214	001314					MOV	T2TEMP, #T2TEMP+2
2156	011702	113737	013214	016134					MOV	T2TEMP, DLOTH
2157	011710	012737	000200	013214					MOV	#200, T2TEMP
2158	011716	000402							BR	500060
(3)	011720							500050:		
2159	011720	105237	013214						INCB	T2TEMP
2160	011724							500060:		
2161	011724	000737							BR	500030
(3)	011726							500040:		

```

2162 011726          500020:
2163          12600
2164 011726          12650 T22: ;IF THE LINE HAS BEEN SIZED, BUT NO
2165          12700 ;SELECT CODE HAS BEEN MAPPED NOTIFY THE
2166          12750 ;OPERATOR .
2167          12800
2168 011726 123727 016134 000200
(9) 011734 001015
2169          12900
2170          12950 ; ERROR #0 NO ANSWERBACK FROM TERMINAL
2171          13000 ;*****
2172 011736 012765 012530 000010
2173 011744 052765 100000 000004
2174 011752 105065 000004
2175 011756 012765 013356 000006
2176 011764 000207 13250
2177 011766 000463
(3) 011770
2178 011770 012701 013220          500070:
2179          13400 MOV #T2BUF,R1
2180          13450 ;GET THE SELECT CODE FROM THE LINE TABLE &
2181          13500 ;REQUEST AN ANSWERBACK.
2182 011774 113737 016134 020424
2183 012002 012700 020421
2184 012006 004737 007110
2185 012012 012703 000310
2186 012016 105037 013212
2187 012022 004737 011216
2188          13850
2189          13900 ;CHECK FOR ANY RESPONSE FROM TERMINAL
2190          14000 ;
2191          14050 TSTB T2CNT1
(9) 012032 001015          500110:
2192          14100 ; ERROR #1 NO ANSWERBACK RECEIVED.
2193          14100 ;*****
2194 012034 012765 012530 000010
2195 012042 052765 100000 000004
2196 012050 112765 000001 000004
2197 012056 012765 013356 000006
2198 012064 000424
(3) 012066
2199 012066 012765 012140 000010          500110:
2200          14450 MOV #T23,PC(R5)
2201          14500 ;TEST LENGTH OF ANSWERBACK SHOULD BE 20 MAX.
2202 012074 123727 013212 000024
(9) 012102 003411
2203          14600
2204          14650 ; ERROR #2 ANSWERBACK OVER 20 CHARS LONG.
2205          14700 ;*****
2206 012104 052765 100000 000004
2207 012112 112765 000002 000004
2208 012120 012765 013404 000006
2209 012126
2210 012126 012701 013220          500130:
2211 012132 004737 011306          500120:
2212 012136
    
```

```

2213 012136          500100:
2214 012136 000207          15150
2215 012140          15200 T23: RTS PC
2216          15250 ;SAVE COPIES OF THE ANSWERBACK AND IT'S LENGTH
2217          15300 ;THEN READ ANSWERBACKS 10 TIMES MORE.
2218          15350 ;VERIFY THEY ARE ALL THE SAME.
2219 012140 012765 012206 000010
2220 012146 113737 013212 013213
2221 012154 012701 013220
2222 012160 012700 020574
2223 012164
2224          15650          500140:
2225 012164 105737 013212
(5) 012170 001404
2226 012172 112120
2227 012174 105337 013212
2228 012200 000771
(3) 012202
2229 012202 105037 013210          500150:
2230          15950 CLRAB T2SAV1
2231 012206          16000 T23A: ;RESET INPUT BUFFER POINTER AND ZERO COUNTER
2232          16050
2233 012206 012701 013220
2234 012212 105037 013212
2235          16200
2236          16250 ;SEND SELECT SEQUENCE TO TERMINAL
2237 012216 012700 020421
2238 012222 012703 000310
2239 012226 004737 011216
2240 012232 105237 013210
2241          16500
2242          16550 ;IF NO ANSWER NOTIFY OPERATOR
2243          16600 ;IF OLD ANSWER DIFFERENT FROM NEW ANSWER
2244          16650 ;NOTIFY OPERATOR.
2245 012236 105737 013212
(9) 012242 001012
2246          16750
2247          16800 ; ERROR #30 NO ANSWERBACK DURING TEN READ LOOP
2248          16850 ;*****
2249 012244 052765 100000 000004
2250 012252 012765 013356 000006
2251 012260 112765 000030 000004
2252 012266 000500
(3) 012270
2253 012270 105011          500160:
2254          17150 CLRAB (R1)
2255 012272 123737 013212 013213
(9) 012300 001416
2256 012302 012701 013220
2257 012306 004737 007110
2258          17350
2259          17400 ;ERROR #31 INCONSISTANT ANSWERBACKS
2260          17450 ;*****
2261 012312 052765 100000 000004
2262 012320 112765 000031 000004
2263 012326 012765 013356 000006
    
```

```

2264 012334 000455          BR      50021$
(3) 012336
2265 012336 012701 013220    50020$1 MOV    #T2BUF,R1
2266 012342 012700 020574    MOV    #STACK3,R0
2267          17000          J      ;COMPARE MESSAGES FOR SAME DATA
2268          17850          J
2269 012346 005037 013214    CLR    T2TEMP
2270 012352 113737 013212 013214  MOVB   T2CNT1,T2TEMP
2271 012360 005037 013216    CLR    T2TEMP+2
2272 012364 012737 000001 013216  MOV    #1,T2TEMP+2
(5) 012372 000402          BR      50022$
(4) 012374          50023$1 INC    T2TEMP+2
(7) 012374 005237 013216    50022$1 CMP    T2TEMP+2,T2TEMP
(5) 012400 023737 013216 013214  BGT    50024$
(7) 012400 003024    CMPB   (R0)+,(R1)+
2273 012410 122021    BEQ    50025$
(9) 012412 001421          J      ;ERROR #32 INCONSISTANT ANSWERBACKS
2274          18150          J
2275          18200          ;*****
2276          18250          J
2277 012414 052765 100000 000004    BIS    #MERR,MFLAGS(R5)
2278 012422 112765 000032 000004    MOVB   #32,MFLAGS(R5)
2279 012430 012765 013575 000006    MOV    #E21,POINT(R5)
2280 012436 012701 013220    MOV    #T2BUF,R1
2281 012442 113737 013213 013212  MOVB   T2CNT2,T2CNT1
2282 012450 013737 013214 013216  MOV    T2TEMP,T2TEMP+2
2283 012456          50025$1 BR      50023$
2284 012456 000746          50024$1 J      ;ECHO ANSWER TO TERMINAL IN ASCII AND
(3) 012460          18700          ;OCTAL FORMATS,
2285          18750          MOV    #T2BUF,R1
2286          J      ;PC,TYPANS
2287 012460 012701 013220    50021$1 JSR    PC,TYPANS
2288 012464 004737 011306    50017$1 JSR    PC,TYPANS
2289 012470          50021$1 JSR    PC,TYPANS
2290 012470          50017$1 JSR    PC,TYPANS
2291 012470 032765 100000 000004    BIT    #MERR,MFLAGS(R5)
(9) 012476 001401    BEQ    50026$
2292 012500 000207          RTS    PC
2293 012502          50026$1 RTS    PC
2294          19150          J
2295 012502 123727 013210 000010  J      ;CHECK FOR TEN ITERATIONS
(9) 012510 001005    CMPB   T2SAV1,#10
2296 012512 012755 012530 000010  BNE    50027$
2297 012520 000207    MOV    #T24,RPC(R5)
2298 012522 000402    RTS    PC
(3) 012524          50027$1 BR      50030$
2299 012524 000137 012206 19400    JMP    T23A
2300 012530          50030$1 JMP    T23A
2301          19500          J
2302 012530          T24:  J      ;RESTORE POINTERS & TEST THE BROADCAST (BEL)
2303          19550          ;WON'T ACTIVATE THE AUTOANSWER,
2304          19600          J
2305 012530 012701 013220    MOV    #T2BUF,R1
2306 012534 105037 013212    CLRB   T2CNT1
2307 012540 012702 000002    MOV    #SIX,R2
  
```

```

2308 012544 004737 006654          JSR    PC,SECHO
2309 012550 012700 020421    MOV    #SCODE,R0
2310 012554 113737 000007 020424  MOVB   #7,SCODE+3
2311 012562 004737 007110    MOVB   #7,SCODE+3
2312 012566 012703 000310    JSR    PC,TYPES
2313 012572 004737 011216    MOV    #200,R3
2314 012576 105737 013212    JSR    PC,GETANS
(9) 012602 001415    TSTB   T2CNT1
2315          BEQ    50031$
2316          20050          J      ;ERROR #4 RECVD ANSWERBACK FROM BROADCAST
2317          20100          ;*****
2318          20150          J
2319 012604 052765 100000 000004    BIS    #MERR,MFLAGS(R5)
2320 012612 112765 000004 000004  MOVB   #4,MFLAGS(R5)
2321 012620 012765 013632 000006    MOV    #E22,POINT(R5)
2322 012626 012765 012636 000010  MOV    #T25,RPC(R5)
2323 012636 000207    RTS    PC
2324 012636          20400          50031$1 RTS    PC
2325          20500          T25:  J      ;IF IN MULTI LINE MODE SETUP NEXT LINE POINTERS
2326          20550          ;IF SINGLE LINE MODE TEST KEYBOARD STUFF,
2327          20600          J
2328 012636 032737 000040 001364  BIT    #MULTI,PCFLAG
(9) 012644 001424    BEQ    50032$
2329 012646 004737 002110    JSR    PC,LINMON
2330 012652 012765 012734 000010  MOV    #T25A,RPC(R5)
2331 012660 032737 000400 001364  BIT    #LDONE,PCFLAG
(9) 012666 001406    BEQ    50033$
2332 012670 042737 000400 001364  BIC    #LDONE,PCFLAG
2333 012676 052765 020000 000004  BIS    #TDONE,MFLAGS(R5)
2334 012704          50033$1 J
2335 012712 000207    MOV    #T21,RPC(R5)
2336 012714 000512    RTS    PC
(3) 012716          21000          BR      50034$
2337 012716 113737 016134 020424  MOVB   DLOTH,SCODE+3
2338 012724 012700 020421    MOV    #SCODE,R0
2339 012730 004737 007110    JSR    PC,TYPES
2340 012734          21100          T25A:  J
(4) 012734 012734 012765 013034 000010  MOV    #T26,RPC(R5)
2341 012742 012701 013220    MOV    #T2BUF,R1
2342 012746 012702 000002    MOV    #BTX,R2
2343 012752 004737 006654    JSR    PC,SECHO
2344          21300          ;SET UP TO TEST HERE-IS KEY SINGLE LINE ONLY
2345          21350          J
2346 012756 012700 013252    MOV    #HI,R0
2347 012762 004737 007110    JSR    PC,TYPES
2348 012766 105037 013212    CLRB   T2CNT1
2349 012772 012703 007640    MOV    #4000,R3
2350          21600          J      ;READ ANSWERBACK
2351 012776 004737 011216    JSR    PC,GETANS
2352 013002 105737 013212    TSTB   T2CNT1
(9) 013006 001012    BNE    50035$
2353          21750          J      ;ERROR #5 NO ANSWERBACK FROM HERE-IS KEY
2354          21800          ;*****
2355          21850          J
2356 013010 052765 100000 000004    BIS    #MERR,MFLAGS(R5)
2357 013016 112765 000005 000004  MOVB   #5,MFLAGS(R5)
  
```

```

2358 013024 012765 013475 000006          MOV    #E17,POINT(R5)
2359 013032 000207          RTS    PC
2360 013034          500350:
2361          22150
2362 013034          T26:   ;TEST CTL-E FUNCTION
2363          22200
2364          22250
2365          22300
2366          22350
2366 013034 012765 011726 000010          ;
2367 013042 012701 013220          MOV    #T22,RPC(R5)
2368 013046 012700 013305          MOV    #T2BUF,R1
2369 013052 012703 007640          MOV    #CE,R0
2370 013056 105037 013212          MOV    #4000,R3
2371 013062 004737 007110          CLR   T2CNT1
2372 013066 004737 011216          JSR   PC,TYPES
2373 013072 105737 013212          JSR   PC,GETANS
2374 013076 001013          TSTB  T2CNT1
2375          BNE   500360
2376          ; ERROR #6 NO ANSWERBACK FROM CTL-E KEY
2377          ;*****
2378          ;
2379 013100 052765 100000 000004          BIS   #MERR,MFLAGS(R5)
2380 013106 112765 000006 000004          MOV   #6,MFLAGS(R5)
2381 013114 012765 013540 000006          MOV   #E10,POINT(R5)
2382 013122 000207          RTS   PC
2383 013124 000406          BR    500370
2384 013126          500360:
2385 013126 052765 020000 000004          BIS   #TDONE,MFLAGS(R5)
2386 013134 012765 011726 000010          MOV   #T22,RPC(R5)
2387          500370:
2388 013142          500340:
2389 013142 000207          RTS   PC
2390          ;*****
2391          ;THIS ROUTINE IS THE KEYBOARD INTERRUPT HANDLER
2392          ;FOR TESTS #1 AND #2
2393          ;*****
2394 013144 117721 002766          T220: MOV   #0VCRXB,(R1)+ ;STORE CHAR IN POINTER
2395 013150 052737 004000 001364          BIS   #DATAIN,PCFLAG ;SET DATA-IN FLAG
2396 013156 012777 000101 002744          MOV   #101,DLADR ;REENABLE THE RECVR
2397 013164 005037 007564          CLR   DELAYT ;ABORT THE TIMEOUT
2398 013170 000002          RTI
2399          23950
2400          500000:
2401 013172 000207          500010:
2402          RTS   PC
2403          ;*****
2404          ;DSABL LSB
2405          ;
2406 013174 000 003          2410:   .BYTE 0,3 ;ITERATION COUNTS
2407 013176 000000          T02BLK1 .WORD 0 ;CTLCNT
2408 013200 000000          .WORD 0 ;PASS COUNT
2409 013202 000000          .WORD 0 ;STATUS FLAGS
2410 013204 000000          .WORD 0 ;POINTER
2411 013206 011548          .WORD T21 ;RETURN PC
2412 013210 000000          T2SAV1: .WORD 0
2413 013212 000          T2CNT1: .BYTE 0
2414 013213 000          T2CNT2: .BYTE 0
    
```

```

2410 013214 000000 000000          24600 T2TEMP1 .WORD 0,0
2411 013220 000000 000000 000000          24605 T2BUF1 .WORD 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ; BUFFER FOR ANSWERBACK
2412 013226 000000 000000 000000          013234 000000 000000 000000
2413 013242 000000 000000 000000          013250 000000
2414          24650
2415 013252 005015 042504 051120          24700 .NLIST BEX
2416 013305 015 052012 050131          24750 HI:   .ASCIZ <15><12>/DEPRESS HERE IS -- KEY/<15><12>
2417          24800 CE:   .ASCIZ <15><12>/TYPE CONTL-E/<15><12>
2418          24850
2419 013326 005015 052012 051505          T21   .ASCIZ <15><12><12>/TEST 2 AUTO ANSWER/<15><12>
2420 013356 047516 040440 051516          E14:  .ASCIZ /NO ANSWERBACK RECVD/<15><12>
2421 013404 047101 041123 041501          E15:  .ASCIZ /ANSBACK MSG OVER 20 CHARS/<15><12>
2422 013440 047101 041123 041501          E16:  .ASCIZ /ANSBACK MSG CONTAINED NULL/<15><12>
2423 013475 110 051105 020105          E17:  .ASCIZ /HERE IS KEY DIDN'T TXMIT ANSBACK/<15><12>
2424 013540 052103 026514 020105          E18:  .ASCIZ /CTL-E DIDN'T TXMIT ANSBACK/<15><12>
2425 013575 101 051516 042527          E21:  .ASCIZ /ANSBACKS DIDN'T COMPARE/<15><12>
2426 013632 047101 041123 041501          E22:  .ASCIZ /ANSBACK RECVD FROM BROADCAST SELECT/<15><12>
2427          25250 .EVEN
2428          25300
2429          25350 .LIST BEX
    
```



```

2428 25450
2429 25500
2430 25550
2431 25600
2432 25650
2433 25700
2434 25750
2435 25800
2436 25850
2437 013700 012700 014376 25900
2438 013704 004737 007010
2439 013710 012705 014342
2440 013714 032737 000040 001364 26000
2441 013722 001424 26050
2442 013724 012701 000102 26100
2443 013730 113702 016167 26150
2444 013734 004737 006270 26200
2445 013740 012702 000006 26300
2446 013744 004737 006270 26400
2447 013750 005301 26450
2448 013752 001372 26500
2449 013754 012700 014427
2450 013760 004737 007010
2451 013764 052765 020000 000004 26600
2452 26650
2453 013772 000207 26700
2454 26750
2455 26800
2456 26850
2457 013774 012765 013774 000010 26900
2458 014002 012737 014010 014356 26950
2459 014010 012737 014302 014354 27000
2460 014010 012700 014443 27050
2461 014022 004737 007110 27100
2462 014026 012700 014046 27150
2463 014032 112037 014564 27200
2464 014036 112037 014565 27250
2465 014042 111037 014566 27300
2466 014046 012700 014557 27350
2467 014052 004737 007110 27400
2468 014056 013700 014356 27450
2469 014062 112037 014564 27500
2470 014066 112037 014565 27550
2471 014072 112037 014566 27600
2472 014076 010037 014356 27650
2473 014102 012702 014124 27700
2474 014106 004737 007124 27750
2475 014112 012702 035230 27800
2476 014116 004737 007024 27850
2477 014122 000406 27900
2478 014124 005037 007564 27950
2479 014130 052737 004000 001364 28000
2480 014136 000002 28050
2481 014140 012700 017730 28100
2482 014144 004737 007110 28150
2483 014150 042737 004000 001364 28200

```

```

2484 014156 042765 100000 000004 28250
2485 014164 117737 000164 014360 28300
2486 014172 113702 016167 28350
2487 014176 004737 006654 28400
2488 014202 012702 000006 28450
2489 014206 004737 006654 28500
2490 014212 005337 014360 28550
2491 014216 001371 28600
2492 014220 012700 014427 28650
2493 014224 004737 007110 28700
2494 014230 012702 000012 28750
2495 014234 004737 006654 28800
2496 014240 117737 000110 014360 28850
2497 014246 113702 016167 28900
2498 014252 004737 006654 28950
2499 014256 012702 000006 29000
2500 014262 004737 006654 29050
2501 014266 005337 014360 29100
2502 014272 001371 29150
2503 014274 012700 014427 29200
2504 014300 004737 007110 29250
2505 014304 012702 000012 29300
2506 014310 004737 006654 29350
2507 29400
2508 014314 005237 014354 29450
2509 014320 023727 014356 014651 29500
2510 014326 001247 29550
2511 014330 052765 020000 000004 29600
2512 29650
2513 014336 000207 29700
2514 29750
2515 014340 000 003 29800
2516 014342 000000 29850
2517 014344 000000 29900
2518 014346 000000 29950
2519 014350 000000 30000
2520 014352 013724 30050
2521 30100
2522 014354 000000 30150
2523 014356 000000 30200
2524 014360 000000 30250
2525 30300
2526 30350
2527 30400
2528 014362 022 025 030 30450
2529 014370 000 063 102 30500
2530 014376 005015 052012 051505 30550
2531 014427 075 036475 036475 30600
2532 014443 120 042522 051523 30650
2533 014473 101 052106 051105 30700
2534 014526 054524 042520 042040 30750
2535 014557 055 042523 020124 30800
2536 30850
2537 014610 027063 065 30900
2538 014613 040 032040 30950
2539 014616 027065 065 31000

```

2540	014621	040	033040	31050		,ASCII / 6/
2541	014624	020040	067	31100		,ASCII / 7/
2542	014627	040	034040	31150		,ASCII / 8/
2543	014632	027070	065	31200		,ASCII / 8,5/
2544	014635	040	030401	31250	HDR5B1	,ASCII / 11/
2545	014640	030440	062	31300		,ASCII / 12/
2546	014643	040	032001	31350		,ASCII / 14/
2547	014646	020040	063	31400	HDR5A1	,ASCII / 3/
2548	014651			31450	HDR5E1	
2549		014652		31500		,EVEN
2550				31550		,LIST BEX

2552				31650		,DSABL L5B
2553				31700		
2554				31750		*****
2555				31800		;
2556				31850	TEST4	HORIZONTAL TAB OPTION
2557				31900		IF USING OTHER THAN 132 COL PAPER CHANGE LOC "WIDTH"
2558				31950		TO APPROPRIATE VALUE, SEE *N COMMAND
2559				32000		;
2560				32050		*****
2561				32100		;
2562	014652	012705	015246	32150	TEST41	MOV #T04BLK,R5 ;SET UP POINTER TO MODULE BLOCK
2563	014656	012700	015320	32200		MOV #T4,R0
2564				32250		; PRINT TEST HEADER
2565	014662	004737	007010		JSR	PC,MTYPE
2566	014666	012737	015270	015264	T411	MOV #TABL4,T4SAV2
2567	014674	012765	014702	000010		MOV #T42,RPC(R5)
2568	014702	012702	000033		T421	MOV #ESC,R2
2569	014706	013737	010146	015262		MOV WIDTH,T4SAV1
2570				32550		; SEND ESC-2 TO RESET ALL TABS.
2571	014714	004737	006270		JSR	PC,MECHO
2572	014720	012702	000062		MOV	#2,R2
2573	014724	004737	006270		JSR	PC,MECHO
2574	014730	117737	0000330	015316	MOV#	#T4SAV2,IAB ;GET TAB COUNT FROM IABL4
2575	014736	005237	015264		INC	T4SAV2
2576	014742	105077	0000316		CLRB	#T4SAV2 ;INITIALIZE COUNT TO ZIP
2577	014746	013701	015316		MOV	IAB,R1
2578	014752	012700	017730		MOV	#11,R0
2579				33000		; SEND CR/LF
2580	014756	004737	007010		JSR	PC,MTYPE
2581	014762	163737	015316	015262	351	SUB TAB,T4SAV1 ;SU TAB COUNT PER LINE
2582	014770	002434		33150	BLT	60 ;FINISHED THIS LINE - BRANCH
2583				33200		;
2584	014772	005301		33250	481	DEC R1 ;TYPE (TAB-1) PERIODS
2585	014774	001405		33300	BEQ	58 ;AS A FORMATT FOR
2586	014776	012702	000056		MOV	#*,R2 ;COMPARISON
2587	015002	004737	006270		JSR	PC,MECHO ;PRINT PERIOD
2588	015006	000771		33450	BR	65
2589	015010	012702	000033		MOV	#ESC,R2 ;SET TAB
2590				33550		; SEND ESC-1 TO SET A TAB
2591	015014	004737	006270		JSR	PC,MECHO
2592	015020	012702	000061		MOV	#1,R2
2593	015024	004737	006270		JSR	PC,MECHO
2594				33750		; SEND A BACKSPACE
2595	015030	012702	000010		MOV	#10,R2 ;
2596	015034	004737	006270		JSR	PC,MECHO
2597				33900		; PRINT A V FOR REFERENCE
2598	015040	012702	000126		MOV	#V,R2
2599	015044	004737	006270		JSR	PC,MECHO
2600	015050	105277	000210		INCB	#T4SAV2 ;INCB TAB COUNT
2601	015054	013701	015316		MOV	IAB,R1 ;GET TAB POS AGAIN
2602	015060	000740		34150	BR	38 ;FORMAT NEXT SECTION
2603				34200		;LINE SHOULD LOOK LIKE THIS:V.....V.....V..ETC
2604				34250		
2605	015062	012737	000003	015266	681	MOV #3,COUNT ;DO 3 LINES OF TABS
2606	015070	117737	000170	015316	781	MOV# #T4SAV2,IAB ;GET TAB COUNT
2607	015076	001440		34400	BEQ	118 ;#0? = BRANCH OUT

2608	015100	005237	015264		34450	INC	T4SAV2	
2609	015104	012700	017730		34500	MOV	#L1,R0	
2610					34550			; SEND A CR/LF
2611	015110	004737	007010			JSR	PC,MTYPE	
2612	015114	012702	000011		34650	061	MOV	#11,R2
2613					34700			; SEND A HORIZ=TAB
2614	015120	004737	006270			JSR	PC,MECHO	
2615	015124	117737	000134	015260	34800	MOV	#T4SAV2,T4SAV	;GET FILL COUNT TABS/2
2616	015132	012702	000006		34850	961	MOV	#ACK,R2
2617					34900			; SEND FILL CHARACTERS
2618	015136	004737	006270			JSR	PC,MECHO	
2619	015142	005337	015260		35000	DEC	T4SAV	
2620	015146	001371			35050	BNE	98	
2621	015150	012702	000130		35100	MOV	#*X,R2	
2622					35150			; PRINT AN X UNDER EACH V
2623	015154	004737	006270			JSR	PC,MECHO	
2624	015160	005337	015316		35250	DEC	TAB	;DEC TAB COUNT
2625	015164	001353			35300	BNE	88	;MORE TABS = BRANCH
2626	015166	005337	015264		35350	DEC	T4SAV2	;FIX POINTER
2627	015172	005337	015266		35400	1061	DEC	COUNT
2628	015176	001334			35450	BNE	78	;NOT DONE = BRANCH
2629	015200	012700	017051		35500	1161	MOV	#L3,R0
2630	015204	004737	007010		35550	JSR	PC,MTYPE	
2631	015210	062737	000002	015264	35600	ADD	#2,T4SAV2	;GET NEXT TABLE ENTRY
2632	015216	023727	015264	015315	35650	CMP	T4SAV2,*TAB-1	;END OF TABLE?
2633	015224	001226			35700	BNE	T42	;NO = DO NEXT SET
2634					35750			
2635	015226	052765	020000	000004	35950	BIS	#TDONE,MFLAGS(R5)	;SET ATTENTION AND DONE FLAGS
2636	015234	012765	014666	000010	36000	MOV	#T41,RPC(R5)	
2637	015242	000207			36050	RTS	PC	
2638					36100			
2639	015244	000	004		36150			;ITERATION COUNTS
2640	015246	000000			36200	T04BLK1	,WORD 0	;CICLNT
2641	015250	000000			36250		,WORD 0	;PASS COUNT
2642	015252	000000			36300		,WORD 0	;STATUS FLAGS
2643	015254	000000			36350		,WORD 0	;POINTER
2644	015256	014702			36400		,WORD T42	;RETURN PC
2645					36450			
2646	015260	000000			36500	T4SAV1	,WORD 0	;STORAGE
2647	015262	000000			36550	T4SAV1	,WORD 0	
2648	015264	000000			36600	T4SAV2	,WORD 0	
2649					36650			
2650					36700			
2651	015266	000002			36750	COUNT1	,WORD 2	
2652	015270	004	000	002	36800	TABL4i	,BYTE 4,,0,2	;TAB, TAB COUNT, FILL COUNT
2653	015273	010	000	004	36850		,BYTE 8,,0,4	
2654	015276	011	000	005	36900		,BYTE 9,,0,5	;TABLE FOR TEST 4
2655	015301	020	000	010	36950		,BYTE 16,,0,8	
2656	015304	022	000	012	37000		,BYTE 18,,0,10	
2657	015307	040	000	021	37050		,BYTE 32,,0,17	
2658	015312	100	000	041	37100		,BYTE 64,,0,33,,0	
	015315	000						
2659	015316	000000			37150	TAB1	,WORD 0	
2660	015320	005015	052012	051505	37200	T41	,ASCIZ <15><12><12>/TEST 4 HORIZONTAL TAB/<15><12>	
	015326	020124	020064	047510				
	015334	044522	047532	052116				

2661	015342	046101	052040	041101	37250	,EVEN		
	015350	005015	000					
	015354							

```

2663          37350
2664          37400 ;*****
2665          37450 ;
2666          37500 ;TEST5 VERTICAL TAB OPTION
2667          37550 ; SINGLE LINE TEST REQUIRES OPERATOR INTERVENTION
2668          37600 ;
2669          37650 ;*****
2670          37700 ;ENABL LSB
2671          37750
2672 015354 012700 016040 37800 TEST5: MOV #T5,R0 ;SU TEST HEADER
2673 015360 012705 016014 37850 MOV #T05BLK,R5 ;SET UP POINTER TO MODULE BLOCK
2674 015364 004737 007010 JSR PC,MYTPE
2675 015370 032737 000040 001364 37950 BIT #MULTI,PCFLAG ;MULTI LINE MODE?
2676 015376 001046 38000 BNE 48 ;YES = BRANCH OVER INTERVENTION
2677 015400 012700 014635 38050 MOV #HDR50,R0 ;SET UP INSTRUCTIONS
2678 015404 112037 014564 38100 MOVB (R0)+,HDR4+5
2679 015410 112037 014565 38150 MOVB (R0)+,HDR4+6
2680 015414 112037 014566 38200 MOVB (R0)+,HDR4+7
2681 38250
2682 015420 012700 014443 38250 ; TYPE INSTRUCTIONS
2683 015424 004737 007110 T51: MOV #HDR3,R0
2684 015430 012700 014557 38400 JSR PC,TYPES
2685 015436 004737 007110 MOV #HDR4,R0
2686 015440 012702 015500 JSR PC,TYPES
2687 015444 004737 007124 38500 MOV #3,R2 ;SU FOR INTERRUPT TO 3#
2688 015450 012702 015230 38550 JSR PC,READS ;INITIALIZE VECTOR AREA
2689 015454 004737 007024 38600 MOV #15000,R2 ;ALLOW 15 SEC.
2690 015460 032737 004000 001364 38650 JSR PC,READ10
2691 015466 001770 38700 BIT #DATAIN,PCFLAG
2692 015470 042737 004000 001364 38750 BEO 28
2693 015476 000406 38800 BIC #DATAIN,PCFLAG
2694 38850 BR 48
2695 38900 ;***** THIS SECTION HANDLES RECVH INTERRUPTS*****
2696 015500 005037 007564 39000 ;
2697 015504 052737 004000 001364 39050 30: CLR DELAYT ;ABORT THE TIMEOUT
2698 015512 000002 39100 BIS #DATAIN,PCFLAG ; FLAG RECEIVED CHAR.
2699 39150 RTI
2700 39200 ;*****
2701 39250
2702 015514 012737 000002 015266 39300 40: MOV #2,COUNT
2703 015522 012765 015514 000010 39350 MOV #40,RPC(R5) ;SET RETURN TO 4#
2704 015530 012737 000001 016032 39400 MOV #1,LINES
2705 015536 005037 016036 39450 CLR TABS
2706 015542 012737 000014 016034 39500 MOV #12,MAX
2707 015550 012702 000033 39550 MOV #ESC,R2 ;RESET ALL TABS
2708 39600 ; ESC=4 RESETS THE TABS,
2709 015554 004737 006270 39650 JSR PC,MECHO
2710 015560 012702 000064 39700 MOV #4,R2
2711 015564 004737 006270 39750 JSR PC,MECHO
2712 015570 013701 016032 39800 50: MOV #LINES,R1 ;GET LINE COUNT
2713 015574 012702 000012 39850 60: MOV #12,R2
2714 39900 ; SEND LINE FEED,
2715 015600 004737 006270 39900 JSR PC,MECHO
2716 015604 005301 40000 DEC R1
2717 015606 001372 40050 BNE 68
2718 015610 012702 000033 40100 MOV #ESC,R2 ;SET TAB
    
```

```

2719 015614 004737 006270 JSR PC,MECHO
2720 015620 012702 000063 40200 MOV #3,R2
2721 015624 004737 006270 40250 ; ESC=3 SETS A TAB LOCATION,
2722 015624 004737 006270 JSR PC,MECHO
2723 015630 012700 014427 40350 MOV #DAS,R0
2724 40400
2725 015634 004737 007010 JSR PC,MYTPE
2726 015640 005237 016032 40500 INC LINES
2727 015644 023737 016032 016034 40550 CMP LINES,MAX
2728 015652 001346 40600 BNE 58 ;11 TABS YET?
2729 015654 012737 000001 016032 40650 70: MOV #1,LINES ;NO = BRANCH
2730 015662 012737 000001 016030 40700 MOV #1,TS5AV1 ;RESET LINE COUNT
2731 015670 012702 000013 40750 80: MOV #13,R2 ;FILL COUNT
2732 40800 ; SEND A VERT-TAB COMMAND,
2733 015674 004737 006270 JSR PC,MECHO
2734 015700 012702 000006 40900 90: MOV #ACK,R2
2735 015704 004737 006270 40950 ; SEND A FILL CHARACTER,
2736 015704 004737 006270 JSR PC,MECHO
2737 015710 005337 016030 41050 DEC TS5AV1
2738 015714 001371 41100 BNE 98
2739 41150 ;
2740 41200 ; CONVERT NO. OF LINES FOR OUTPUT MSG.
2741 41250 ;
2742 015716 013746 016032 41300 MOV LINES,=(SP)
2743 015722 012746 016071 41400 MOV #T52,=(SP)
2744 015726 004737 010006 41450 JSR PC,BIN2PA
2745 015732 012700 016071 41500 MOV #T52,R0
2746 015736 004737 007010 41550 JSR PC,MYTPE
2747 015742 012700 014427 41600 MOV #DAS,R0 ;SU LINE OF DASHES
2748 015746 004737 007010 JSR PC,MYTPE
2749 015752 005237 016032 41700 INC LINES ;NEW LINE COUNT
2750 015756 013737 016032 016030 41750 MOV LINES,TS5AV1 ;FILL COUNT = LINES
2751 015764 023737 016032 016034 41800 CMP LINES,MAX ;11 TABS DONE?
2752 015772 001336 41850 BNE 88 ;NO = CONTINUE
2753 015774 005337 015266 41900 DEC COUNT ;DU 2 PAGES TOTAL
2754 016000 001325 41950 BNE 78 ;RE=DO PAGE
2755 016002 052765 020000 000004 42000 BIS #TDONE,MFLAGS(R5) ;SET ATTENTION & DONE FLAGS
2756 42050
2757 016010 000207 42100 RTS PC
2758 42150
2759 42200 ;*****
2760 016012 000 002 42250 ;BYTE 0,2 ;ITERATION COUNTS
2761 016014 000000 42300 T05BLK: .WORD 0 ;CTL CNT
2762 016016 000000 42350 .WORD 0 ;PASS COUNT
2763 016020 000000 42400 .WORD 0 ;STATUS FLAGS
2764 016022 000000 42450 .WORD 0 ;POINTER
2765 016024 015420 42500 .WORD T51 ;RETURN PC
2766 42550
2767 016026 000000 42600 TS5AV1: .WORD 0
2768 016030 000000 42650 TS5AV11: .WORD 0
2769 42700
2770 016032 000000 42750 LINES: .WORD 0
2771 016034 000000 42800 MAX: .WORD 0
2772 016036 000000 42850 TABS: .WORD 0
2773 016040 005015 052012 051505 42900 T51: .ASCII <15><12><12>/TEST 5 VERTICAL TAB/<15><12>
    
```

	016054	052122	041511	046101			
	016062	052040	041101	005015			
	016070	000					
2774	016071	060	030060	030060	42950	T521	,ASCIZ /00000/
	016076	000					
2775		016100			43000		,EVEN
2776					43050		,DBABL LSB

2778					43150		,SBTIL STORAGE & CONSTANTS
2779					43200		;*****
2780					43250		; PROGRAM STORAGE, CONSTANTS, AND VARIABLES
2781					43300		
2782					43350		,EVEN
2783	016100	000000	000000	000000	43400	TEMP:	,WORD 0,0,0,0,0 ;TEMPORARY WORK AREA
	016106	000000	000000	000000			
2784	016114	000004			43450	INBUF:	,BLKW 4 ;INPUT BUFFER
2785	016124	177570			43500	SWR:	177570 ;SWIICH REGISTER POINTER
2786					43550		;MAY BE CHANGED TO 176
2787					43600		;***** I/O DRIVER WORK AREA *****
2788					43650		
2789	016126	000000			43700	DLFLAG:	,WORD 0 ;LINE FLAG WORD
2790	016130	000000			43750	DLADRI:	,WORD 0 ;LINE ADDR WORD
2791	016132	000000			43800	DLVEC:	,WORD 0 ;LINE VECTOR WORD
2792	016134	000000			43850	DL0TH:	,WORD 0 ;LINE "OTHER" WORD
2793	016136	000000			43900	DVCRXB:	,WORD 0 ;RECEIVER DATA BUFFER
2794	016140	000000			43950	DVCTXS:	,WORD 0 ;TRANSMJ STATUS REGISTER
2795	016142	000000			44000	DVCTXB:	,WORD 0 ;TRANSMIT DATA BUFFER
2796	016144	000000			44050	TXVEC:	,WORD 0 ;TRANSMIT INTERRUPT VECTOR
2797					44100		
2798					44150		;***** GENERAL USE *****
2799	016146	000204			44200	WIDTH:	,WORD 132
2800	016150	000000			44250	SAVE:	,WORD 0
2801	016152	000000			44300	NEXT:	,WORD 0 ;NEXT TEST NO.
2802	016154	000000			44350	INTEST:	,WORD 0 ;CURRENT TEST
2803	016156	000000			44400	TESTAD:	,WORD 0 ;CURRENT TEST PC
2804	016160	000000			44450	ONLNI:	,WORD 0 ;CURRENT LINE UNDER TEST
2805	016162	000000			44500	NXTLIN:	,WORD 0 ;NEXT LINE TO TEST
2806	016164	016114			44550	PTR:	INBUF ;INPUT BUFFER POINTER
2807	016166	177			44600	DEL:	,BYTE 177
2808	016167	014			44650	FF:	,BYTE 14
2809					44750		,EVEN
2810					44800		

Address	Hex	Hex	Hex	Hex	Hex	Label	Text
2812	016170	000000	177560	000000	44900	LIN001	WORD 0,177560,0,0
2813	016176	000000			44950		
2814	016200	000000	175610	000000	45150	LIN011	WORD 0,175610,0,400
2815	016206	000400			45000		
2816	016210	000000	175620	000000	45200	LIN021	WORD 0,175620,0,1000
2817	016216	001000			45050		
2818	016220	000000	175630	000000	45250	LIN031	WORD 0,175630,0,1400
2819	016226	001400			45100		
2820	016230	000000	175640	000000	45300	LIN041	WORD 0,175640,0,2000
2821	016236	002000			45150		
2822	016240	000000	176500	000000	45350	LIN051	WORD 0,176500,0,2400
2823	016246	002400			45200		
2824	016250	000000	176510	000000	45400	LIN061	WORD 0,176510,0,3000
2825	016256	003000			45250		
2826	016260	000000	176520	000000	45450	LIN071	WORD 0,176520,0,3400
2827	016266	003400			45300		
2828	016270	000000	176530	000000	45500	LIN081	WORD 0,176530,0,4000
2829	016276	004000			45350		
2830	016300	000000	176550	000000	45550	LIN111	WORD 0,176550,0,4400
2831	016306	004400			45400		
2832	016310	000000	176560	000000	45600	LIN121	WORD 0,176560,0,5000
2833	016316	005000			45450		
2834	016320	000000	176570	000000	45650	LIN131	WORD 0,176570,0,5400
2835	016326	005400			45500		
2836	016330	000000	175700	000000	45700	LIN141	WORD 0,175700,0,6000
2837	016336	006000			45550		
2838	016340	000000	175710	000000	45750	LIN151	WORD 0,175710,0,6400
2839	016346	006400			45600		
2840	016350	000000	175720	000000	45800	LIN161	WORD 0,175720,0,7000
2841	016356	007000			45650		
2842	016360	000000	175730	000000	45850	LIN171	WORD 0,175730,0,7400
2843	016366	007400			45700		
2844	016370	000000	175740	000000	45900	LIN201	WORD 0,175740,0,10000
2845	016376	010000			45750		
2846	016400	000000	176540	000000	45950	LIN211	WORD 0,176540,0,10400
2847	016406	010400			45800		
2848	016410	000000	176550	000000	46000	LIN221	WORD 0,176550,0,11000
2849	016416	011000			45850		
2850	016420	000000	176560	000000	46050	LIN231	WORD 0,176560,0,11400
2851	016426	011400			45900		
2852	016430	000000	176570	000000	46150	LIN241	WORD 0,176570,0,12000
2853	016436	012000			46000		
2854	016440	000000	176600	000000	46200	LIN251	WORD 0,176600,0,12400
2855	016446	012400			46050		
2856	016450	000000	176610	000000	46250	LIN261	WORD 0,176610,0,13000
2857	016456	013000			46100		
2858	016460	000000	176575	000000	46300	LIN271	WORD 0,176575,0,13400
2859	016466	013400			46150		
2860	016470	000000	176576	000000	46350	LIN301	WORD 0,176576,0,14000
2861	016476	014000			46200		
2862	016500	000000	176577	000000	46400	LIN311	WORD 0,176577,0,14400
2863							

2843	016506	014400	176000	000000	46450	LIN321	WORD 0,176000,0,15000	BINARY LINE NO.
2844	016510	000000	176010	000000	46500	LIN331	WORD 0,176010,0,15400	
2845	016516	015000			46350			
2846	016520	000000	176020	000000	46550	LIN341	WORD 0,176020,0,16000	
2847	016526	015400			46400			
2848	016530	000000	176030	000000	46600	LIN351	WORD 0,176030,0,16400	
2849	016536	016000			46450			
2850	016540	000000	176040	000000	46650	LIN361	WORD 0,176040,0,17000	
2851	016546	016400			46500			
2852	016550	000000	176050	000000	46800	LIN371	WORD 0,176050,0,17400	
2853	016556	017000			46650			
2854	016560	000000	176060	000000	47000	LIN401	WORD 0,176060,0,20000	
2855	016566	017400			46800			
2856	016570	000000	176070	000000	47050	LIN411	WORD 0,176070,0,20400	
2857	016576	020000			46850			
2858	016600	000000	176080	000000	47200	LIN421	WORD 0,176080,0,21000	
2859	016606	020400			47000			
2860	016610	000000	176090	000000	47250	LIN431	WORD 0,176090,0,21400	
2861	016616	021000			47050			
2862	016620	000000	176100	000000	47500	LIN441	WORD 0,176100,0,22000	
2863	016626	021400			47300			
2864	016630	000000	176110	000000	47550	LIN451	WORD 0,176110,0,22400	
2865	016636	022000			47350			
2866	016640	000000	176120	000000	47600	LIN461	WORD 0,176120,0,23000	
2867	016646	022400			47400			
2868	016650	000000	176130	000000	47800	LIN471	WORD 0,176130,0,23400	
2869	016656	023000			47600			
2870	016660	000000	176140	000000	48000	LIN481	WORD 0,176140,0,23800	
2871	016666	023400			47800			
2872	016670	000000	176150	000000	48200	LIN491	WORD 0,176150,0,24400	
2873	016676	024000			48000			
2874	016700	000000	176160	000000	48400	LIN501	WORD 0,176160,0,24800	
2875	016706	024400			48200			
2876	016710	000000	176170	000000	48600	LIN511	WORD 0,176170,0,25400	
2877	016716	025000			48400			
2878	016720	000000	176180	000000	48800	LIN521	WORD 0,176180,0,26000	
2879	016726	025400			48600			
2880	016730	000000	176190	000000	49000	LIN531	WORD 0,176190,0,26600	
2881	016736	026000			48800			
2882	016740	000000	176200	000000	49200	LIN541	WORD 0,176200,0,27200	
2883	016746	026400			49000			
2884	016750	000000	176210	000000	49400	LIN551	WORD 0,176210,0,27800	
2885	016756	027000			49200			
2886	016760	000000	176220	000000	49600	LIN561	WORD 0,176220,0,28400	
2887	016766	027400			49400			
2888	016770	177777			47550	TABEND1	WORD -1	
2889					47600			

```

2868          47700  J * * * * *
2869          47750  ; SYSTEM MESSAGES
2870          47800  ,NLIST BEX
2871          47850
2872 016772 005015 055103 040514 47900 PROGI01 ,ASCII <15><12>/CZLAF0 LA36 OPTIONS TESTS/
2873 017030 005015 042522 052123 47950 ,ASCII <15><12>/RESTART AT 1372/
2874 017051 015 005012 000012 48000 L31 ,ASCIIZ <15><12><12><12>
2875 017056 047503 046515 047101 48050 HEADR11 ,ASCIIZ /COMMAND SUMMARY 1/<15><12><12>
2876 017103 123 020040 020040 48100 CONSUM1 ,ASCII /S SINGLE LINE MODE/<15><12><12><1>
2877 017135 115 020040 020040 48150 ,ASCII /M MULTI-LINE MODE/<15><12><12><1>
2878 017166 020121 020040 020040 48200 ,ASCII /O SEQUENCE TESTS/<15><12><12><1>
2879 017216 047122 020040 020040 48250 ,ASCII /RN RUN TEST "N" /<15><12><12><1>
2880 017245 104 020116 020040 48300 ,ASCII /DN DROP LINE "N" /<15><12><12><1>
2881 017324 101 020116 020040 48350 ,ASCII /AN ADD LINE "N" /<15><12><12><1>
2882 017324 020124 020040 020040 48400 ,ASCII /T TYPE LINE TABLE /<15><12><12><1>
2883 017356 047127 020040 020040 48450 ,ASCII /WN CHANGE "WIDTH" TO N/<15><12><12><1>
2884 017413 114 020040 020040 48500 ,ASCII /L LOOP ON ERROR /<15><12><12><1>
2885 017443 110 020040 020040 48550 ,ASCII /H HALT ON ERROR /<15><12><12><1>
2886 017473 103 020040 020040 48600 ,ASCII /C CLEAR ; RESETS H & L COMMANDS/<15><12><12><1>
2887 017542 020116 020040 020040 48650 ,ASCII /N INHIBIT REPORTS /<15><12><12><1>
2888 017574 020120 020040 020040 48700 ,ASCII /P PRINT ERROR REPORTS /<15><12><12><1>
2889 017632 051505 020103 020040 48750 ,ASCIIZ /ESC TO EXECUTE COMMAND STRING/<15><12><12><12>
2890 017676 005015 046012 047111 48800 HEADR21 ,ASCIIZ <15><12><12><12>/LINE# ADDR VECTOR SEL/
2891 017730 005015 000 000 48850 L11 ,ASCIIZ <15><12>
2892 017733 000 020060 020040 48900 L11 ,ASCII /00 17/
2893 017743 000 030060 020060 48950 DLAD1 ,ASCII /0000 /
2894 017752 030060 020060 020040 49000 DLV1 ,ASCII /000 /
2895 017761 052 051105 047522 49050 ER01 ,ASCIIZ /*ERROR 000 TEST 00 LINE 00/<15><12><12><7>
2896 020017 040 020040 037477 49100 ER11 ,ASCIIZ / 7777/<15><12><12><7>
2897 020032 026455 046055 047111 49150 ER21 ,ASCIIZ /---LINE INVALID/<15><12><12><7>
2898 020055 116 020117 047111 49200 ER71 ,ASCIIZ /NO INTERRUPT ON TXMIT/<15><12><12><7>
2899 020105 104 047522 050120 49250 DR1 ,ASCIIZ /DROPPED/<15><12><12><7>
2900 020117 052 005015 000 49300 S11 ,ASCIIZ /*/<15><12><12><7>
2901 020123 015 051012 040505 49350 RD11 ,ASCIIZ <15><12>/READY /<15><12><12><7>
2902 020136 005015 040520 051523 49400 EOPM1 ,ASCIIZ <15><12>/PASS 00000 TEST 00/<15><12><12><7>
2903 020166 005015 025052 020052 49450 EOTM1 ,ASCIIZ <15><12>/*** END OF TEST 00/<15><12><12><7>
2904 49500
2905 020215 015 050012 043103 49550 SW1 ,ASCIIZ <15><12>/PCFLAG 1 000000 /<15><12><12><7>
2906 020242 054105 042503 051523 49600 DR01 ,ASCII /EXCESSIVE ERRORS,,LINE/
2907 020270 030060 042040 047522 49650 DR11 ,ASCIIZ /NO DROPPED/<15><12><12><7>
2908 020306 047516 046040 047111 49700 E191 ,ASCIIZ /NO LINES AVAILABLE FOR TEST/<15><12><12><7>
2909 020345 114 047111 020105 49750 E201 ,ASCIIZ /LINE RE-SELECTED/<15><12><12><7>
2910 020370 047503 051516 046117 49800 CTLM1 ,ASCIIZ /CONSOLE CONTROL7/
2911 49850
2912 020411 004 003401 000002 49900 ALLO1 ,ASCIIZ <4><1><7><2> /SELECT ALL ESCAPE SEQUENCE
2913 020416 000404 000 49950 ALLOFF1 ,ASCIIZ <4><1> /DESELECT ALL SEQUENCE
2914 020421 004 001401 000000 50000 SCODE1 ,ASCIIZ <4><1><3><000> /SELECT UNIQUE SEQUENCE
2915 020426 000404 001003 000 50050 NSELCT ,ASCIIZ <4><1><3><2> /BAD SELECT SEQUENCE
2916 50100
2917 020434 000060 50150 STACK21 ,BLKW 40,
2918 020574 000030 50200 STACK31 ,BLKW 30,
2919 020670 000000 50250 ENDS1 ,WORD 0
2920 001102 50300 ,END START
    
```

```

ABO = 000020 CTLBLK 001364 ERRVEC= 000004 ICNT = 177776 LIN44 016630
ACK = 000006 CTLC = 000003 ER0 017761 INBUF 016114 LIN45 016640
ADDC = 000004 CTLCNT= 000000 ER1 020017 INHR = 000040 LIN46 016650
ALLOFF 020416 CTLG = 000007 ER2 020032 INHRRPT= 020000 LIN47 016660
ALLO 020411 CTLGX 003064 ER7 020055 INTEST 016154 LIN48 016670
ANSHDR 011472 CTLH = 000010 EXC = 000033 INTTRAP 007162 LIN51 016700
ATTN = 000200 CTLK = 000013 ETX = 000003 IOTVEC= 000020 LIN52 016710
A2BIN 007676 CTLK = 000014 ETYPE 006240 ISP = 005726 LIN53 016720
A2SAV 010004 CTLM = 020370 E10 011137 ISP2 = 022626 LIN54 016730
A3 014046 CTLN = 000016 E12 011045 ITRAP = 104004 LIN55 016740
BIN2DA 010006 CTLP = 000020 E14 013356 LDONE = 000400 LIN56 016750
BIT0 = 000001 DAS = 014427 E15 013404 LF = 000012 LIN57 016760
BIT00 = 000001 DATA 004776 E16 013440 LIN 017733 LOOPC = 000100
BIT01 = 000002 DATAIN= 004000 E17 013475 LINENO 001370 LOUPOE= 040000
BIT02 = 000004 DATA2 005000 E18 013540 LINES 010032 LOOP1 001454
BIT03 = 000010 DDISP = 177570 E19 020306 LINESE 002204 LOOP2 001466
BIT04 = 000020 DECODE 004230 E20 020345 LINMON 002110 L1 017730
BIT05 = 000040 DECSAV 004774 E21 013575 LIN00 016170 L3 017051
BIT06 = 000100 DECTBL 004656 E22 013632 LIN01 016200 MACHCK 000004
BIT07 = 000200 DEL 016166 E29 011005 LIN02 016210 MAJOR = 003000
BIT08 = 000400 DELAYM 007530 FF 016167 LIN03 016220 MAX 016030
BIT09 = 001000 DELAYP= 104006 FILL3 014362 LIN04 016230 MECHU 006274
BIT1 = 000002 DELAYT 007564 FLAGDA 010161 LIN05 016240 MERM = 100000
BIT10 = 002000 DIGITS 010162 FLAG1 = 000001 LIN06 016250 MERRN = 100377
BIT11 = 004000 DEAD 017743 FLAG2 = 000002 LIN07 016260 MFLAGS= 000004
BIT12 = 010000 DLADR 016130 GETSRS 011216 LIN08 016270 MSAVE 006646
BIT13 = 020000 DLFLAG 016126 GETSRC 003350 LIN11 016300 NTW 002432
BIT14 = 040000 DLOTH 016134 GETSWS 003130 LIN12 016310 NTW1 002452
BIT15 = 100000 DLP = 100000 GNL 002556 LIN13 016320 MTYPE 007010
BIT2 = 000004 DLV = 017752 G01 002600 LIN14 016330 MULTI = 000040
BIT3 = 000010 DLVEC 016132 G02 002604 LIN15 016340 NEWADD= 001000
BIT4 = 000020 DP 020105 G03 002612 LIN16 016350 NEWTST= 002000
BIT5 = 000040 DPROP = 000010 G04 002644 LIN17 016360 NEXT 016152
BIT6 = 000100 DR0 020242 G05 002650 LIN20 016370 NOOP = 000240
BIT7 = 000200 DR1 020270 GP 011100 LIN21 016400 NOP = 000240
BIT8 = 000400 DSWR = 177570 GSEL 010734 LIN22 016410 NRE0 = 000340
BIT9 = 001000 DTEND 004774 GVL 002332 LIN23 016420 /SELC 000206
BPTVEC= 000014 DVCRXB 016136 G1A 002340 LIN24 016430 OXLDIN 016162
BUILD 005040 DVCTXB 016142 G1B 002344 LIN25 016440 OCTALC 011520
CATCH 006126 DVCTXS 016140 G1C 002356 LIN26 016450 ONLIN 016160
CE 013305 ECHO 005044 G1D 002370 LIN27 016460 Q2ASC 007566
CFIAGS 002032 EMTABL 006230 HALTC = 000200 LIN30 016470 PASCNI= 000002
CHARS 010310 EMTROS 006162 HALTOE= 100000 LIN31 016500 PCFLAG 001364
CHKW 005006 EMTVEC= 000030 HDR3 014443 LIN32 016510 PIR0 = 177776
CMDERR 000062 ENDS 020670 HDR4 014557 LIN33 016520 PIR0VE= 000240
CNTD 010100 ENQ = 000005 HDR5 014610 LIN34 016530 POINT = 000006
CONSUM 017103 EOL = 004000 HDRSA 014646 LIN35 016540 PRI 010424
CON 010662 EOP = 020000 HDRSB 014635 LIN36 016550 PRINT = 000020
CONSON 003332 EOPM 020136 HDRSE 014651 LIN37 016560 PRINT1= 010000
COUNT 015266 EOT = 040000 HDRS5 014657 LIN40 016570 PRI0 = 000000
CR = 000015 EOTM 020166 HEADR1 017056 LIN41 016600 PRI4 = 000200
CSLF = 000200 ERROR 005124 HEADR2 017676 LIN42 016610 PRI7 = 000340
C81 004000 ERRSAV 005434 HT = 000011 LIN43 016620 PROGI0 016772
    
```

PRTLTB 007212	SWR 016124	TKV 000060	T4SAV1 015262	\$ASK0 000300
PRTTBL 104002	SWRTST 003006	TKVEC 000060	T4SAV2 015264	\$ASK1 000110
PR0 000000	SWTEST 000172	TPB 177566	T41 014666	\$ASK10 000110
PR1 000040	SW0 000001	TPS 177564	T42 014702	\$ASK11 000110
PR2 000100	SW00 000001	TPVEC 000064	T5 010040	\$ASK12 000110
PR3 000140	SW01 000002	TKAPVE 000034	T5SAV 016026	\$ASK2 000110
PR4 000200	SW02 000004	TRTVEC 000014	T5SAV1 016030	\$ASK3 000210
PR5 000240	SW03 000010	TSCCNT 002036	T5 015420	\$ASK4 000110
PR6 000300	SW04 000020	ISCPTR 002034	T52 010071	\$ASK5 000110
PR7 000340	SW05 000040	TSBTL 002040	UPDATE 002660	\$ASK6 000110
PS 177776	SW06 000100	TSCTIL 001446	WIDTH 010146	\$ASK7 000110
PSW 177776	SW07 000200	TSIMON 000000	\$BGNLE 177777	\$SAVLE 177777
PTR 016164	SW08 000400	TXTRAP 007416	\$ERFLG 000400	\$SSK0 000023
PWRVEC 000024	SW09 001000	TXVEC 010144	\$FAND 000310	\$SVPC 000204
RDSAV 003776	SW1 000002	TXPANS 011306	\$FBAD 000401	\$SWR 100000
RDY 020123	SW10 002000	TYPE 104000	\$FBLA 000170	\$TAGLE 177777
READIO 007024	SW11 004000	TPES 007110	\$FCAS 000150	\$TAGNU 000040
READKB 003466	SW12 010000	T0 010306	\$FDEC 000220	\$TEMP 000300
READS 007124	SW13 020000	T00BLK 010352	\$FDO 000340	\$TN 000001
READY 000200	SW14 040000	T01 010200	\$FFAL 000405	\$TSK0 000034
REPORT 005436	SW15 100000	T01BLK 010722	\$FGOD 000400	\$TSK1 000037
RESTR 001372	SW2 000004	T02HLK 013176	\$FIF 000110	\$TSK10 000023
RESEVC 000010	SW3 000010	T03BLK 014342	\$FINC 000210	\$TSK11 000024
RPC 000010	SW4 000020	T04BLK 015246	\$FLOO 000200	\$TSK2 000024
RUB 010364	SW5 000040	T05BLK 016014	\$FNAM 000160	\$TSK3 000023
R6 0000006	SW6 000100	T1 010742	\$FNO 000403	\$TSK4 000025
R7 0000007	SW7 000200	T1TEMP 010736	\$FAOR 000320	\$TSK5 000022
SAVE 016150	SW8 000400	T11 010460	\$FRTN 000300	\$TSK6 000017
SCODE 020421	SW9 001000	T12 010554	\$FSEL 000100	\$TSK7 000021
SEC 010434	S1 020117	T13 010566	\$VTHE 000330	\$BARGC 000000
SECHO 006654	TAB 015316	T16 010634	\$FTRU 000404	\$BYTE 000402
SEL 000200	TABDA 010146	T2 013326	\$FUNT 000130	\$CASC 000000
SELERR 005100	TABEND 016770	T2BUF 013220	\$FVHI 000120	\$DST 000000
SEQ 000100	TABL1 011176	T2CNT1 013212	\$FYES 000402	\$ELOC 000000
SETIO 005600	TABL4 015270	T2CNT2 013213	\$HD 000003	\$ERFL 000000
SI 000017	TABS 016036	T2SAV1 013210	\$IFLEV 177777	\$FLAG 000001
SO 000016	TBITVE 000014	T2TEMP 013214	\$ISK0 000001	\$FNUM 000000
SOH 000001	IDONE 020000	T21 011546	\$ISK1 000001	\$LOC 013076
SSWR 000176	TEMP 016100	T22 011726	\$ISK10 000001	\$RECG 177777
STACK 001100	TEMPF 005002	T220 013144	\$ISK11 000001	\$RETCU 000000
STACK2 020434	TEMPT 005004	T23 012140	\$ISK2 000001	\$SRIN1 000000
STACK3 020574	TESTAD 016156	T23A 012206	\$ISK3 000001	\$SRIN2 000001
START 001102	TESTNO 001366	T24 012530	\$ISK4 000001	\$SSMC 000000
START2 001172	TEST0 010164	T25 012636	\$ISK5 000001	\$SK 000000
START3 001230	TEST1 010444	T25A 012734	\$ISK6 000001	\$TGS1 000000
STKMT 177774	TEST2 011526	T26 013034	\$ISK7 000001	\$TGS2 000000
STRAP 007510	TEST3 013700	T3 014376	\$LOCTA 177777	\$TGS 000000
STX 000002	TEST4 014652	T3SAV 014354	\$LSTCN 177777	\$TGS2 000000
SUTEST 002056	TEST5 015354	T3SAV1 014356	\$LSTIN 000000	\$TGS 000000
SW 020215	TIMER 007562	T3SAV2 014360	\$LSTST 177777	\$SSTAG 000000
SWCTL 000020	TKB 177562	T4 015320	\$LSTIA 000000	\$ = 020672
SWLINE 000174	TKS 177560	T4SAV 015200	\$NESTL 177777	

, ABS, 020672 000

ERRORS DETECTED: 0

CZLAFAP, CZLAFAP, LST=SYSMAC, SML/ML, SPMAC, SML/ML, CZLAFAP, P11
RUN=TIME: 120 101 .6 SECONDS
RUN=TIME RATIO: 00633/222=272,2
CORE USED: 19K (37 PAGES)