

LA00,34,38

LA00 DMT DIAG
CZLAIB0

AH-E150B-MC

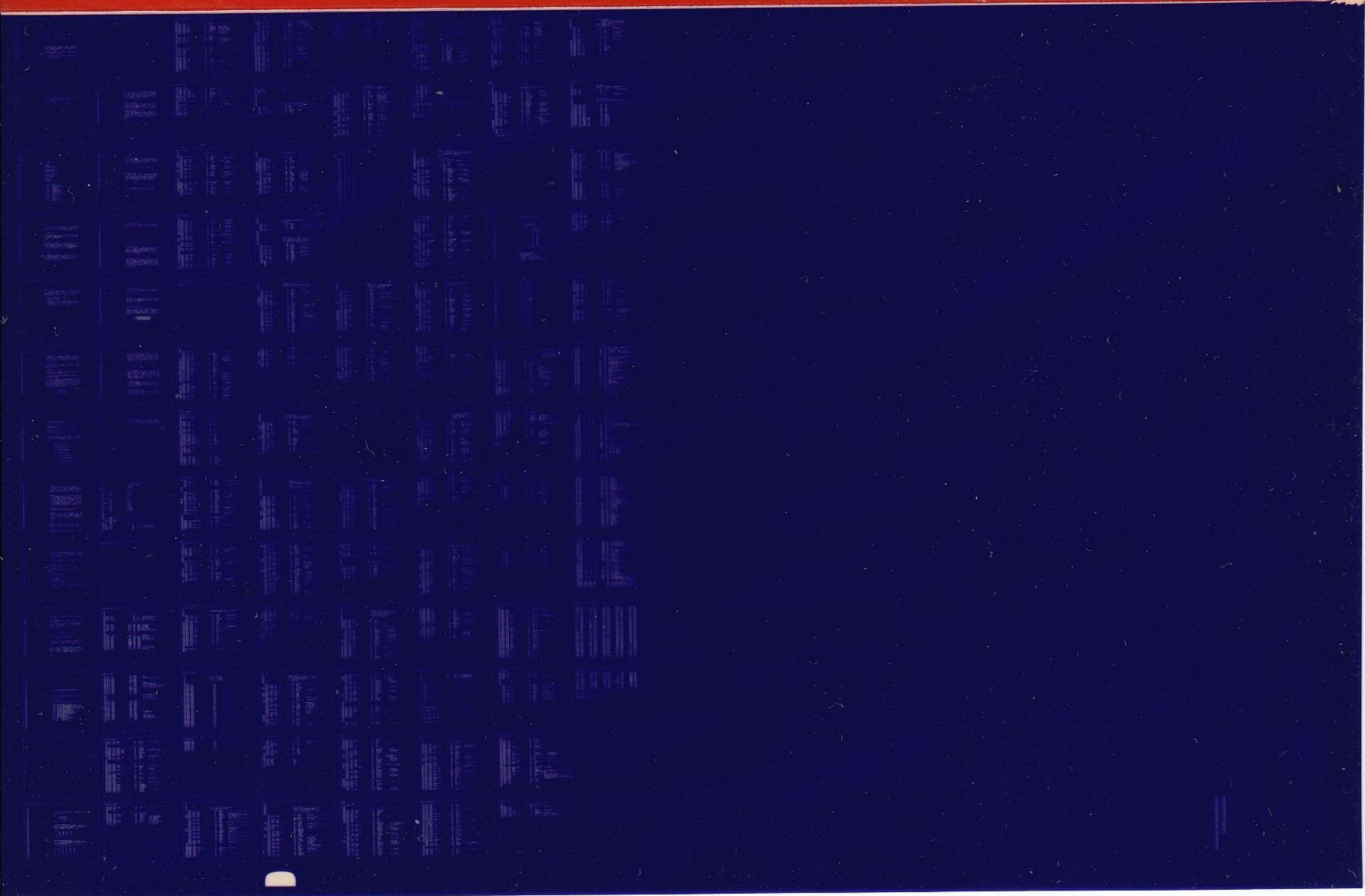
COPYRIGHT 78-79

FICHE 1 OF 1

SEP 1979

digital

MADE IN USA



200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200

IDENTIFICATION

PRODUCT CODE: AC-E149B-MC
PRODUCT NAME: CZLAIB0 LA00,LA34,LA38 DMT DIAG
DATE CREATED: 23 FEB 1979
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, 1979 BY DIGITAL EQUIPMENT CORPORATION

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900

EDIT HISTORY

REV B0 23-FEB-79 RALPH SCHAUBER

CHANGED QUIET SUBROUTINE TO INCLUDE TIMEOUT IF XON IS NOT
RECIEVED WITHIN TEN SECONDS. ALSO CHANGED SEND ROUTINE
TIMEOUT FROM TWENTY TO TEN SECONDS.

INCLUDED ERROR REPORT CALLS IN TIMEOUT ROUTINES TO IDENTI-
FY FAILING LINE NUMBERS.

LIFE TEST NOW RESETS THE PASS NO. TO 0 ON STARTUP.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300

TABLE OF CONTENTS

- 1.0 ABSTRACT
 - 1.1 FUNCTIONAL DESCRIPTION
 - 1.2 INTENDED USERS
- 2.0 REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 RELATED PROGRAMS
 - 2.3 TERMINAL CONFIGURATION
- 3.0 LOADING AND INITIALIZATION
 - 3.1 STARTING ADDRESSES
 - 3.2 MODIFICATIONS TO PROGRAM
 - 3.3 EXECUTION TIME
- 4.0 CONTROL AND TEST SELECTION
 - 4.1 SWITCH REGISTER CONTROL
 - 4.2 CONSOLE CONTROL
 - 4.2.1 COMMANDS
- 5.0 TEST GROUPS
 - 5.1 TERMINAL TESTS
 - 5.2 INTERVENTION TESTS
 - 5.3 EXERCISORS
- 6.0 TEST DESCRIPTIONS
 - 6.1 TEST00 DATA PATHS TEST
 - 6.2 TEST01 PRINTABLE CHARACTERS TEST
 - 6.3 TEST02 NONPRINTABLE CHARACTERS TEST
 - 6.4 TEST03 DOT MATRIX TEST
 - 6.5 TEST04 HORIZONTAL PITCH TEST
 - 6.6 TEST05 SPACE BACKSPACE TEST
 - 6.7 TEST06 SET MARGINS TEST
 - 6.8 TEST07 HORIZONTAL TABS TEST
 - 6.9 TEST10 MULTIPLE LINE FEED TEST
 - 6.10 TEST11 HORIZONTAL MOTION TEST
 - 6.11 TEST12 BUFFER OVERRUN TEST
 - 6.12 TEST13 VERTICAL PITCH TEST
 - 6.13 TEST14 BELL TEST
 - 6.14 TEST15 LIFE TEST
 - 6.15 TEST16 DYNAMIC EXERCISOR
 - 6.16 TEST17 INTERFACE SPEEDS TEST
 - 6.17 TEST20 KEYBOARD ECHO TEST
 - 6.18 TEST21 CHARACTER CODE ECHO TEST
 - 6.19 TEST22 PITCH SETUP TEST

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800

1.0 ABSTRACT

THIS PROGRAM IS A FUNCTIONAL TEST OF THE LA00,LA34,LA38 TERMINAL. IT CAN TEST UP TO 40 TERMINALS AT A TIME, INTERFACED THROUGH A DZ11-A/E ASYNCHRONOUS MULTIPLEXERS. THIS PROGRAM WAS DESIGNED TO TEST ALL OF THE FUNCTIONAL CHARACTERISTICS OF THE LA00,LA34,LA38 TERMINAL IN A DMT OR PMT ENVIRONMENT.

1.1 FUNCTIONAL DESCRIPTION

THIS PROGRAM CONSISTS OF A TEST SELECTION AND CONTROL SECTION, A CONSOLE TERMINAL DRIVER SECTION, MULTI UNIT DZ11 DRIVER SECTION, AND TWENTY TWO FUNCTIONAL TESTS. THE TESTS ARE OF THREE TYPES, PRINTER FUNCTION TESTS, MANUAL INTERVENTION TESTS, AND EXERCISORS.

1.2 INTENDED USERS

THIS PROGRAM WAS DESIGNED TO TEST EVERY FUNCTIONAL CHARACTERISTIC OF THE LA00,LA34,LA38 TERMINAL, AND AS SUCH WILL BE USED FOR DESIGN MATURITY TESTING. THE OPERATOR WILL HAVE THE OPTION OF RUNNING THE PROGRAM IN A NON-INTERVENTION MODE, THUS ALLOWING THE PROGRAM TO BE USED IN A PMT ENVIRONMENT. THE TESTS WERE NOT WRITTEN TO F.S. OR FA&T REQUIREMENTS, AND THE USE OF THIS PROGRAM IN THOSE AREAS IS NOT RECOMMENDED.

2.0 REQUIREMENTS

2.1 EQUIPMENT

THIS PROGRAM WILL REQUIRE A PDP-11 PROCESSOR, WITH 16K OF MEMORY. FOR EACH EIGHT TERMINALS TO BE TESTED A DZ11-A,E IS REQUIRED ALONG WITH ONE H317-E DISTRIBUTION PANNEL FOR EACH SIXTEEN TERMINALS UNDER TEST. A HARDWARE SWITCH REGISTER IS SUPPORTED, BUT IS NOT REQUIRED. IF PROGRAM CONTROL IS TO BE VIA CONSOLE TERMINAL THEN A TERMINAL AND INTERFACE AT THE STANDARD ADDRESS & VECTOR ARE REQUIRED.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600

2.2 RELATED PROGRAMS.

THIS PROGRAM WILL PERFORM CURSORARY TESTING OF THE DZ11 INTERFACE, AND SHOULD NOT BE CONSIDERED A VALID TEST OF ANYTHING OTHER THAN THE LA00,LA34,LA38 TERMINAL. PDP-11 PROCESSOR AND MEMORY DIAGNOSTIC PROGRAMS, ALONG WITH DZ11 DIAGNOSTIC PROGRAMS SHOULD BE RUN PERIODICLY TO INSURE CORRECT OPERATION OF THE SYSTEM.

OTHER LA00,LA34,LA38 DIAGNOSTIC PROGRAMS:
CILAJA-0 LA00,LA34,LA38 FA&T PROGRAM

2.3 TERMINAL CONFIGURATION

THIS PROGRAM REQUIRES THAT ALL TERMINALS TO BE TESTED BE SET UP FOR 300 BAUD, 1 STOP BIT, ODD PARITY, AND XON-XOFF ENABLED. ENTER SETUP MODE AND TYPE AN 8 TO GET A PRINTOUT OF THE CURRENT SWITCH SETTINGS. IF NOT CORRECT CHANGE THE SWITCHES THEN VERIFY AGAIN USING THE 8 KEY AGAIN. THESE SWITCHES ARE LOCATED ON THE PC BOARD DIRECTLY UNDER THE KEYBOARD ASSY.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200

3.0 LOADING PROCEDURE AND INITIALIZATION

LOAD THE LA00,LA34,LA38 DIAGNOSTIC PROGRAM TAPE FOLLOWING NORMAL PROCEDURES. IF A HARDWARE SWITCH REGISTER DOES NOT EXIST, THE PROGRAM WILL USE THE CONTENTS OF LOCATION 000176 AS THE VALUE OF THE SWITCHES. THEREFORE, BE SURE TO LOAD LOCATION 000176 WITH THE SWITCH VALUE BEFORE STARTING THE PROGRAM WHEN NOT USING SWITCHES.

3.1 STARTING ADDRESSES

THERE ARE TWO STARTING ADDRESSES FOR THIS PROGRAM. STARTING AT LOCATION 000200 WILL PUT THE TESTING UNDER SWITCH REGISTER CONTROL. STARTING AT LOCATION 000204 WILL PUT THE PROGRAM UNDER CONSOLE CONTROL.

3.2 MODIFICATIONS TO PROGRAM.

THERE ARE A NUMBER OF COMMON DATA STORAGE LOCATIONS WHICH MAY BE MODIFIED BY THE OPERATOR TO COMPENSATE FOR NON STANDARD CONFIGURATIONS, AND DIFFERENT CPU TYPES.

FOR DZ11'S NOT AT THE STANDARD ADDRESSES OR VECTORS THE LOCATIONS NAMED DZADDR AND DZVECT CAN BE CHANGED ACCORDINGLY PRIOR TO STARTING THE PROGRAM.

LOCATION LOOPC CONTAINS A TIME CONSTANT AND IS INITIALLY SET FOR A PDP-11/20 PROCESSOR. THIS TIME CONSTANT IS NOT CRITICAL, BUT LARGE VARIATIONS FROM THOSE LISTED IN THE TABLE WILL RESULT IN INEFFICIENT OPERATION. IT IS BETTER TO HAVE A LONGER TIME CONSTANT THAN ONE TOO SHORT BECAUSE THE ROUTINES THAT USE THE TIMEOUT FEATURE WILL ABORT THE TIMEOUT WHEN THE REQUIRED INPUT IS RECEIVED. THOSE TESTS THAT REQUIRE MANUAL INTERVENTION WILL NOT FUNCTION CORRECTLY IF THE TIMEOUT IS TOO FAST FOR OPERATOR RESPONSE TIMES.

THIS TABLE IS DUPLICATED IN THE LISTING.

LOOPC: 000314 ;TIME CONSTANT FOR 11/20
;SET TO 202 FOR 11/03
;SET TO 251 FOR 11/10
;SET TO 554 FOR 11/40
;SET TO 755 FOR 11/45, 11/60
;SET TO 1237 FOR 11/45, 11/70
;SET TO 2127 FOR 11/45 BIP, 11/55

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300

3.3 EXECUTION TIME

AT 300 BAUD AND EXCLUDING MANUAL INTERVENTION TESTS THIS PROGRAM SHOULD TAKE APPROXIMATELY 13 MIN.

4.0 CONTROL AND TEST SELECTION

THERE ARE TWO MEANS OF CONTROLLING THE EXECUTION OF THIS PROGRAM: VIA THE CONSOLE SWITCH REGISTER, OR VIA THE CONSOLE TERMINAL.

IF THE PROGRAM IS STARTED AT LOCATION 200 AND NO HARDWARE SWITCH REGISTER EXISTS THE PROGRAM WILL USE THE CONTENTS OF LOCATION 176 AS THE SWITCHES

4.1 SWITCH REGISTER CONTROL

THE VARIOUS SWITCHES AND THEIR FUNCTIONS ARE LISTED BELOW. SWITCHES MAY BE CHANGED AND SET AS DESIRED EXCEPT AS NOTED IN THE SPECIFIC SWITCH DESCRIPTIONS. REFER TO THE DETAILED SWITCH DESCRIPTIONS FOR FURTHER, MORE COMPLETE INFORMATION.

SWITCH NUMBER	DESCRIPTION
15	1(UP) = HALT ON ERROR 0(DOWN) = CONTINUE AFTER ERROR REPORT
14	1(UP) = LOOP ON TEST IF ERROR DETECTED 0(DOWN) = CONTINUE TESTING
13	1(UP) = INHIBIT ERROR REPORTS 0(DOWN) = PRINT ERROR REPORTS
12	1(UP) = RUN INDIVIDUAL TEST 0(DOWN) = RUN TESTS IN SEQUENCE
10	1(UP) = GET TEST NO. FROM SWS 4 : 0 0(DOWN) = USE DEFAULT TEST #0
9	1(UP) = PMT MODE MINIMUM MANUAL INTERVENTION 0(DOWN) = DMT MODE INTERVENTION REQUIRED
8	1(UP) = RUN 1 PASS OF TEST SEQUENCE THEN HALT 0(DOWN) = KEEP RUNNING TEST OR SEQUENCE
4-0	TEST NUMBER SELECTION

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300
5400
5500
5600
5700

4.1.1 SWITCH 15

PLACING SWITCH 15 DOWN WILL CAUSE THE PROGRAM TO CONTINUE ON ERRORS DURING ANY OF THE I/O TESTS. WITH SWITCH 15 UP, THE PROGRAM WILL HALT (AT ERRHLT) ON ANY ERROR DURING THE I/O TESTS WITH THE LOCATION OF THE ERROR IN RO. PRESSING CONTINUE WILL CAUSE THE PROGRAM TO CONTINUE IF SWITCH 12 IS DOWN (LOOP ON ERROR). WITH SWITCH 12 UP, PRESSING CONTINUE WILL CAUSE THE PROGRAM TO LOOP ON THE FAILING TEST.

4.1.2 SWITCH 14

PLACING SWITCH 14 UP WILL CAUSE THE PROGRAM TO 'LOOP ON TEST' IF AN ERROR IS DETECTED IN THAT TEST. ERROR REPORTS WILL BE TYPED UNLESS INHIBITED (SWITCH 13 UP). LOOPING WILL OCCUR AUTOMATICLY, WITHOUT OPERATOR INTERVENTION, AND WILL AND WILL CONTINUE UNTIL THE ERROR CEASES TO HAPPEN, OR THE SWITCH IS PLACED DOWN OR =0.

4.1.2 SWITCH 14

PLACING SWITCH 14 UP WILL CAUSE THE PROGRAM TO 'LOOP ON TEST' IF AN ERROR IS DETECTED IN THAT TEST. ERROR REPORTS WILL BE TYPED UNLESS INHIBITED (SWITCH 13 UP). LOOPING WILL OCCUR AUTOMATICLY, WITHOUT OPERATOR INTERVENTION, AND WILL CONTINUE UNTIL THE ERROR CEASES TO HAPPEN, OR THE SWITCH IS PLACED DOWN OR =0.

4.1.3 SWITCH 13

PLACING SWITCH 13 UP WILL INHIBIT THE PRINTING OF ALL ERROR REPORTS. CAN BE USED IN CONJUNCTION WITH SWITCH 14 TO LOOP IN ERRORS FOR TROUBLESHOOTING.

4.1.4 SWITCH 12

PLACING SWITCH 12 UP WILL CAUSE THE PROGRAM TO LOOP IN THE CURRENT, OR SELECTED TEST. IF SWITCH 8 IS UP THE TEST WILL HALT AT THE END OF THE TEST. PRESSING CONTINUE WILL CAUSE THE TEST TO BE STARTED OVER AGAIN. PLACING SWITCH 12 DOWN WILL CAUSE THE NEXT SEQUENTIAL TEST TO BE EXECUTED, UNLESS THE TEST IS AN INTERVENTION TEST AND PMT MODE IS SELECTED.

4.1.5 SWITCH 11

NOT USED.

4.1.6 SWITCH 10

PLACING SWITCH 10 UP WILL CAUSE THE PROGRAM TO USE THE CONTENTS OF SWITCHES 4 THRU 0 AS THE TEST NUMBER. IF SWITCH 12 IS UP THIS IS THE TEST THAT WILL BE RUN. IF SWITCH 12 IS DOWN THE SEQUENCE OF TESTS TO BE RUN WILL START WITH THIS TEST.

4.1.7 SWITCH 9

PUTTING SWITCH 9 UP AT THE START OF TESTING WILL INHIBIT MANUAL INTERVENTION TESTS, AND USE A FIXED SET OF PARAMETERS AS LISTED IN THE DESCRIPTION OF EACH TEST.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600

4.1.8 SWITCH 8

WITH SWITCH 8 DOWN THE PROGRAM WILL LOOP ON THE SELECTED TEST OR TEST SEQUENCE AS SELECTED BY SWITCH 12. PLACING SWITCH 8 UP WILL CAUSE THE PROGRAM TO HALT AT THE COMPLETION OF THE CURRENT TEST, OR TEST SEQUENCE. PRESSING CONTINUE WILL RESULT IN THE PROGRAM RESTARTING THE TEST OR SEQUENCE DEPENDING ON SWITCH 12.

4.1.9 SWITCHES 4 TO 0

SWITCHES 4 TO 0 ARE USED TO SELECT SPECIFIC TESTS WHEN UNDER SWITCH REGISTER CONTROL. TEST NUMBERS ARE ALWAYS IN OCTAL, FROM 00 TO 22.

4.2 KEYBOARD CONTROL

SWITCHES ON THE CONSOLE SWITCH REGISTER WILL HAVE NO EFFECT WHEN UNDER TERMINAL CONTROL EXCEPT FOR SWITCH 13.

THE PROGRAM WILL PRINT THE FOLLOWING : ENTER MODE D OR P : RESPOND BY TYPING EITHER A 'D' FOR DMT MODE, OR A 'P' FOR PMT MODE (NO MANUAL INTERVENTION).

THE PROGRAM WILL PRINT READY ON THE CONSOLE, THEN WAIT FOR COMMANDS FROM THE KEYBOARD.

THE FOLLOWING COMMANDS WILL BE RECOGNIZED :

R TO RUN A SELECTED TEST.
S TO SEQUENCE THRU TESTS.
L TO LOOP ON ERROR.
H TO HALT ON ERROR.
C TO CLEAR THE H & L COMMANDS
W TO SET THE 'WIDTH' CONTROL

THE PERIOD (.) IS A TERMINATOR USED IN CONJUNCTION WITH THE R AND S COMMANDS TO SPECIFY A SINGLE PASS. THAT IS TO STOP AFTER RUNNING A TEST, OR TO STOP AFTER RUNNING A SEQUENCE OF TESTS.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900

PAGE 10

TO ABORT OPERATIONS AND RETURN TO THE WAIT STATE AT ANY TIME TYPE A CTL-C. THE PROGRAM WILL RESPOND WITH READY AND WAIT FOR COMMAND INPUT.

ENTER ONE COMMAND PER LINE, FOLLOWED BY A RETURN. IF CONFLICTING COMMANDS ARE ENTERED THE LAST ENTRY WILL BE USED.

TO EXIT 'COMMAND MODE' TYPE AN ESCAPE. THE PROGRAM WILL TYPE READY AND BEGIN EXECUTION OF THE COMMANDS. COMMANDS CAN BE ENTERED AT ANY TIME, BUT NEW TESTS WILL NOT START UNTIL THE ESCAPE CHARACTER IS RECEIVED.

EXAMPLES OF COMMANDS :

R12 RUN TEST 12
R23. RUN TEST 23 THEN HALT
S. SEQUENCE ALL TESTS THEN HALT
S27 SEQUENCE ALL TESTS STARTING WITH TEST 27
W100 SET WIDTH TO 100 (OCTAL) COLUMNS
(204=132 COLM, 120=80 COLM)

IF A TEST IS SELECTED THAT IS AN OPERATOR INTERVENTION TEST, AND PMT MODE IS SELECTED THE FOLLOWING WILL BE TYPED: RUN INTERVENTION TEST ? ANSWER Y OR N. IF Y IS TYPED THE TEST WILL BE RUN. IF N IS TYPED A NEW TEST NUMBER WILL BE REQUESTED.

THE R,S,H,L,W, AND C MAY BE EITHER UPPER OR LOWER CASE, BUT THE TEST NUMBER MUST ALWAYS BE A 2 DIGIT OCTAL NUMBER. THE COMMAND, TEST NUMBER, AND TERMINATOR ARE ECHOED BY THE PROGRAM, THUS EACH CHARACTER WILL BE PRINTED TWICE IF THE TERMINAL IS IN HALF DUPLEX. IF AN ERROR IS DETECTED IN THE TEST SELECTION (ILLEGAL TEST NUMBER OR COMMAND CHARACTER) A QUESTION MARK IS PRINTED AND THE MESSAGE WILL BE REPEATED.

READY

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300
5400
5500
5600
5700

5.0 TEST GROUPS

5.1 TERMINAL TESTS

5.2 INTERVENTION TESTS

THE TESTS 17 THRU 24 REQUIRE MANUAL INTERVENTION. THESE TESTS ARE NOT RUN IN PMT MODE (SEE DESCRIPTION OF SW 9 4.1.7, AND CONSOLE CONTROL STARTUP 4.2).

5.3 EXERCISORS

TESTS 15 AND 16 ARE DESIGNED AS EXERCISORS, AND CAN BE RUN FOR EXTENDED PERIODS TO 'BURN IN' THE UNITS UNDER TEST.

5.4 TEST ASSIGNMENTS

TESTS LISTED AS DMT WILL NOT BE EXECUTED IN PMT MODE. SEE DESCRIPTION OF SWITCH 9 4.1.7 .

- TEST00 LA00,LA34,LA38 DATA PATHS TEST
- TEST01 LA00,LA34,LA38 ALL PRINTABLE CHARACTERS TEST
- TEST02 LA00,LA34,LA38 NON PRINTABLE CHARACTERS TEST
- TEST03 LA00,LA34,LA38 PRINthead DGT MATRIX TEST
- TEST04 LA00,LA34,LA38 HORIZONTAL PITCH TEST
- TEST05 LA00,LA34,LA38 SPACE-BACKSPACE TEST
- TEST06 LA00,LA34,LA38 SET MARGINS TEST
- TEST07 LA00,LA34,LA38 HORIZONTAL TABS TEST
- TEST10 LA00,LA34,LA38 MULTIPLE LINE FEED TEST
- TEST11 LA00,LA34,LA38 HORIZONTAL MOTION TEST
- TEST12 LA00,LA34,LA38 BUFFER OVERRUN TEST
- TEST13 LA00,LA34,LA38 VERTICAL PITCH TEST
- TEST14 LA00,LA34,LA38 BELL TEST
- TEST15 LA00,LA34,LA38 LIFE TEST
- TEST16 LA00,LA34,LA38 PRINTER DYNAMIC EXERCISOR
- TEST17 DMT INTERFACE SPEEDS TEST
- TEST20 DMT KEYBOARD ECHO TEST
- TEST21 DMT CHARACTER CODE ECHO TEST.
- TEST22 DMT PITCH SETUP TEST

6.0. TEST DESCRIPTION

5800

6.1 DATA PATHS TEST00

100
 200
 300
 400
 500
 600
 700
 800
 900
 1000
 1100
 1200
 1300
 1400
 1500
 1600
 1700
 1800
 1900
 2000
 2100
 2200
 2300
 2400
 2500
 2600
 2700
 2800
 2900
 3000
 3100
 3200
 3300
 3400
 3500
 3600
 3700
 3800
 3900
 4000
 4100
 4200
 4300
 4400
 4500
 4600
 4700
 4800
 4900
 5000
 5100
 5200
 5300
 5400
 5500
 5600
 5700

THIS TEST WILL PRINT FOUR LINES OF ALTERNATING *U*U PATTERN. IT IS A CONFIDENCE TEST OF THE INTERNAL DATA BUS, AND RECIEVER LOGIC.

EXAMPLE :

```
*U*U*U*U*U*U*U*U...
U*U*U*U*U*U*U*U...
*U*U*U*U*U*U*U*U...
U*U*U*U*U*U*U*U...
```

ESTIMATED TIME AT 300 BAUD 18 SECONDS.

6.2 ALL PRINTABLE CHARACTERS TEST01

THIS TEST WILL PRINT EACH OF THE PRINTABLE CHARACTERS IN GROUPS OF FOUR, SEPERATED BY TWO SPACES. THE GROUPS WILL BE PRINTED IN ORDER, AND THE NUMBER OF GROUPS PER LINE WILL BE DEPENDENT ON THE 'WIDTH' SET AT THE START OF THE DIAGNOSTIC. (DEFAULT 132 COLM)

EXAMPLE :

```
AAAA BBBB CCCC DDDD
EEEE FFFF GGGG HHHH
3333 4444 5555 6666
%%% @@@@ +??? ????
```

ESTIMATED TIME AT 300 BAUD 30 SEC

6.3 NON PRINTABLE CHARACTERS TEST02

THIS TEST CHECKS ALL NON-PRINTABLE CHARACTERS. IN THIS TEST ALL NON-PRINTABLE CHARACTER CODES ARE TRANSMITTED, FOLLOWED BY THE WORDS: 'NON-PRINTING CHARACTER TEST. THE NEXT LINE SHOULD BE BLANK. IF ANY CHARACTERS APPEAR ON THE NEXT LINE AN ERROR EXISTS.

THE FOLLOWING CODES ARE TRANSMITTED :

000	NUL	002	STX	006	ACK
020	DLE	021	DC1	022	DC2
023	DC3	024	DC4	025	NAK
026	SYN	027	ETB	030	CAN
031	EM	032	SUB	034	FS
035	GS	036	RS	037	US
177	DEL	021	DC1(XON)		

ESTIMATED TIME AT 300 BAUD 5 SECONDS

5800

6.4 DOT MATRIX TEST03

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100

THIS TEST WILL PRINT THE FIVE CHARACTERS ZH*#\$, THEN PRINT FOUR LINES OF DATA THAT WILL CREATE BLACK BOXES BY OVERPRINTING THE SAME FIVE CHARACTERS AS ABOVE. TEN BOXES WILL APPEAR ON EACH OF THE FOUR LINES AT DIFFERENT SPACINGS. THIS TEST WILL AMPLIFY ANY WEAK OR INTERMITTANT HEAD WIRE PROBLEMS. THE BOXES SHOULD APPEAR AN EVEN DARK BLACK, WITH NO DOTS MISSING OR LITE STREAKS.

ESTIMATED TIME AT 300 BAUD 10 SECONDS

6.5 HORIZONTAL PITCH TEST04

THIS TEST WILL PRINT FIVE GROUPS OF LINES AT EACH OF THE HORIZONTAL PITCH SETTINGS. EACH GROUP OF LINES WILL CONSIST OF FIRST A LINE STATING THE CURRENT PITCH SETTINGS, THEN A LINE OF THE CHARACTERS A THRU Z. THIS IS DONE FOR HORIZONTAL PITCH SETTINGS OF 10 CPI, 12 CPI, 13.2 CPI, AND 16.5 CPI. THE SETUP FOR THIS TEST IS DOWN LINE LOADED.

ESTIMATED TIME AT 300 BAUD 30 SECONDS

6.6 SPACE-BACKSPACE TEST05

A LINE OF ALTERNATING SLASHES AND SPACES IS PRINTED ACROSS THE PAGE. THE PROGRAM WILL THEN BACKSPACE THROUGH THE LINE AND OVERPRINT THE SLASHES WITH BACKSLASHES. TWO LINES ARE PRINTED FOR EACH PASS OF THE TEST. THE PATTERN PRODUCED IS A LINE OF ALTERNATING X'S AND SPACES. THE TWO SLASHES SHOULD CROSS EXACTLY IN THE MIDDLE CREATING THE X CHARACTER.

EXAMPLE : X

ESTIMATED TIME AT 300 BAUD 45 SECONDS

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300

6.7 SET MARGINS TEST06

THIS TEST WILL SET 4 PAIRS OF LEFT AND RIGHT MARGINS, THEN IT WILL PRINT A LINE OF '='S THAT SHOULD BE WITHIN THOSE MARGINS. ALSO A MESSAGE WILL BE SENT SPECIFYING AN ERROR IF IT'S NOT AT THE LEFT MARGIN. A REFERENCE LINE WILL BE PRINTED SHOWING THE MARGIN LIMITS BEING SET UP. ALL HORIZONTAL PITCH SETTINGS WILL BE TESTED.

EXAMPLE :

```
.....V.....V.....  
=====
```

ERROR IF NOT AT LH MARGIN

ESTIMATED TIME AT 300 BAUD 40 SECONDS

6.8 HORIZONTAL TABS TEST07

THIS TEST WILL PRINT A REFERENCE LINE COMPOSED OF A NUMBER OF PERIODS FOLLOWED BY A 'V'. THIS PATTERN IS REPEATED ACROSS THE PAGE. THE LOCATION OF EACH V WILL MARK THE LOCATION OF A TAB STOP SET BY THE PROGRAM. THREE LINES WILL THEN BE PRINTED UNDER THIS REFERENCE LINE, COMPOSED OF A HORIZONTAL TAB FOLLOWED BY AN I, REPEATED ACROSS THE PAGE. THE I'S SHOULD LINE UP DIRECTLY UNDER THE REFERENCE LINE V'S.

EXAMPLE :

```
.....V.....V.....V.....V....  
        I        I        I        I  
        I        I        I        I  
        I        I        I        I
```

THIS WILL BE REPEATED FOR A VARIETY OF DIFFERENT TAB SETTINGS. THE NUMBER OF TABS PER LINE WILL BE CONTROLLED BY THE 'WIDTH' SPECIFIED AT THE START OF THE DIAGNOSTIC.

ESTIMATED TIME AT 300 BAUD, 132 COL - 2 MIN

```

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300
5400
5500
5600

```

6.9 MULTIPLE LINE FEED TEST10

THIS TEST WILL PRINT A REFERENCE LINE OF DASHES THEN SKIP N LINES AND PRINT THE NO. OF LINES SKIPPED ALONG WITH SOME DASHES FOR VISUAL REFERENCE. EACH SKIP COUNT IS DONE TWICE FOR N = 1 TO 7. VERTICAL PITCH WILL BE 6 LINES PER INCH.

EXAMPLE :

```

-----
-----01
-----01
-----02
-----02
-----03
-----03

```

ESTIMATED TIME AT 300 BAUD 15 SECONDS

6.10 HORIZONTAL MOTION TEST11

THIS TEST WILL EXERCISE THE HEAD POSITIONING LOGIC BY PRINTING A LINE OF H'S AT RANDOM COLUMN LOCATIONS WITHIN THE LINE. THE HEAD WILL BE POSITIONED USING SPACES, BACK-SPACES, AND CARRIAGE RETURNS FOLLOWED BY SPACES. THE NUMBER OF COLUMNS PRINTED IS CONTROLLED BY THE 'WIDTH' AS SET AT THE START OF THE PROGRAM. ALL H'S SHOULD BE EVENLY SPACED, WITH NO OVERPRINTS.

ESTIMATED TIME AT 300 BAUD 4 MIN

6.11 BUFFER OVERRUN TEST12

THIS TEST WILL FORCE THE TERMINAL TO SEND AN XOFF CHAR (023) BY ISSUING A SERIES OF TIME CONSUMEING MOVEMENT COMMANDS, FOLLOWED BY ENOUGH CHARACTERS TO FILL THE BUFFER PAST IT'S 118 CHARACTER LIMIT. WHEN THE TERMINAL HAS EMPTIED THE BUFFER TO THE 10 CHARACTER LEVEL IT SHOULD TRANSMIT AN XON CHARACTER (021) ALLOWING THE HOST TO FINISH SENDING DATA. ANY TERMINAL THAT FAILS TO SEND THE XON WILL BE CONSIDERED TO BE 'DEAD', AND WILL BE Deselected OR SET INACTIVE.

ESTIMATED TIME AT 300 BAUD 10 SECONDS.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700

6.12 VERTICAL PITCH TEST13

THIS TEST WILL PRINT SIX LINES AT EACH OF THE VERTICAL PITCH SETTINGS: 2,3,4,6,8 AND 12 LINES PER INCH. THE LINE PRINTED WILL BE A MESSAGE THAT LISTS THE CURRENT CPI AND LPI SETTINGS. THE SETUP FOR THIS TEST IS DOWN LINE LOADED.

ESTIMATED TIME AT 300 BAUD 40 SECONDS

6.13 BELL TEST14

THIS TEST CHECKS THE PRINTER BELL TO INSURE THAT EIGHT BELLS ARE DISTINCTLY HEARD, EVEN WHEN SENT AT THE MAXIMUM TRANSFER RATE. THE PROGRAM SENDS 8 BELL CODES AT THE MAXIMUM RATE TO THE PRINTER THEN WAITS 2.5 SECONDS TO ALLOW THE OPERATOR TO HEAR THE BELLS.

ESTIMATED TIME 1 SECOND

6.14 LA00,LA34,LA38 LIFE TEST15

ORDINARILY THIS TEST SIMPLY PRINTS A LINE OF 'A'S.

WHEN THIS TEST IS LOOPED ON, IT PRINTS TWO LINES OF EACH PRINTABLE CHARACTER. WHEN ALL PRINTABLE CHARACTERS HAVE BEEN DONE, THEY WILL SIMPLY BE REPEATED. THE CURRENT PASS NUMBER IS PRINTED ON EACH LINE, WITH A 1 COLUMN OFFSET ON EACH NEW LINE. THE NUMBER OF CHARACTERS PER LINE WILL BE DETERMINED BY THE 'WIDTH' AS SELECTED AT PROGRAM STARTUP.

EXAMPLE :

```
01 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA..
A 01 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA..
BB 01 BBBBBBBBBBBBBBBBBBBBBBBBBBBB..
BBB 01 BBBBBBBBBBBBBBBBBBBBBBBBBBBB..
CCCC 01 CCCCCCCCCCCCCCCCCCCCCCCC..
CCCCC 01 CCCCCCCCCCCCCCCCCCCCCCCC..
```

ESTIMATED TIME 1 LINE 300 BAUD 5 SECONDS

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300
5400

6.15 LA00,LA34,LA38 DYNAMIC EXERCISOR TEST16

THIS TEST WILL PRINT 35 LINES OF MIXED FORMAT DATA. A PATTERN WILL BE CREATED WHICH IS COMPRISED OF THE UPPER AND LOWER CASE CHARACTER SET PLUS EIGHT OF THE SPECIAL SYMBOLS. THIS PATTERN WILL BE IN THE FORM OF A 10'' BY 6'' MATRIX, WHERE THE UPPER LEFT CORNER WILL HAVE THE GREATEST CHARACTER DENSITY AND THE LOWER RIGHT CORNER WILL HAVE THE LOWEST DENSITY. ALL POSSABLE COMBINATIONS OF HORIZONTAL AND VERTICAL PITCH WILL BE USED.

ESTIMATED TIME AT 300 BAUD 2.5 MIN.

INTERVENTION TESTS

NO TIME ESTIMATES GIVEN

6.16 INTERFACE BAUD RATES TEST17

THIS TEST WILL REQUEST THAT THE OPERATOR CHANGE THE SPEED ON ALL TERMINALS TO 110 BAUD. THE PROGRAM WILL THEN TRANSMIT A MESSAGE TO ALL TERMINALS AT THIS BAUD RATE, IDENTIFYING THE CURRENT SPEED, THEN BY USE OF THE ESCAPE SEQUENCE ESC [OC THE TERMINAL ID MESSAGE WILL BE SENT FROM EACH TERMINAL TO THE PDP-11 TO VERIFY CORRECT TRANSMISSION AND RECEPTION BY THE TERMINAL. THIS SAME PROCEDURE IS REPEATED FOR 300 BAUD. BECAUSE OF INTERVENTION NO TIME ESTIMATE IS GIVEN

6.17 KEYBOARD ECHO TEST20

THIS TEST WILL REQUIRE THE OPERATOR TO TYPE ALL THE PRINTING KEYS ON THE KEYBOARD. IF ANY KEYS ARE NOT SEEN BY THE HOST THEY WILL BE REQUESTED AGAIN, AND A THIRD TIME IF NECESSARY. INSTRUCTIONS WILL THEN BE TYPED TO PRESS THE TAB, RETURN, AND OTHER NON PRINTING KEYS. FIVE SECONDS IS ALLOWED PER KEY DELAY.

6.18 CHARACTER CODE ECHO TEST21

THIS TEST WILL PRINT THE OCTAL CODE OF ANY KEY PRESSED, ALONG WITH THE ASCII CHARACTER. WHERE THE CHARACTER IS A NON PRINTABLE CODE THE MNEMONIC OF THAT CODE WILL BE PRINTED. THE DELETE CHAR WILL BE ECHOED AS A MNEMONIC, THEN THE TEST WILL BE DONE.

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400

PAGE 18

6.19 PITCH SETUP TEST22

THIS TEST WILL REQUIRE THE OPERATOR TO CHANGE THE TERMINAL SETUP TO THAT REQUESTED. AFTER EACH SETUP CHANGE THE PDP-11 WILL SEND A LINE OF DATA THAT SHOULD CONFORM TO THE PARAMS SETUP. THE DATA SENT AFTER VERTICAL PITCH CHANGES WILL BE A NUMBER OF SHORT LINES THAT SHOULD SPAN 1 INCH VERTICALLY. THAT IS EIGHT LINES AFTER THE CHANGE TO 8 LPI ETC.

.ENDR


```

12300           000200           . =200
12400 000200   000137   001220   JMP     START
12500 000204   000137   001242   JMP     KSTART
    
```

```

12700          001000          .:=1000
12800          .EVEN
12900          :TOP OF STACK AREA
13000
13100          : APT PARAMETER BLOCK
13200
13300 001000    000000    APTHDR: .WORD    0            ;HIGH ORDER ADDRESS BITS
13400 001002    001014    .WORD   $MAIL        ;ADDRESS OF APT MAILBOX
13500 001004    000360    .WORD   240.         ;TIME FOR LONGEST TEST 4 MIN.
13600 001006    001440    .WORD   800.         ;TIME FOR QUICK PASS
13700 001010    000012    .WORD   10.          ;TIME FOR EACH ADDITIONAL DVC
13800 001012    000030    .WORD   $ETEND-$MAIL/2 ;LENGTH OF MAILBOX + ETABLE
13900
14000          :APT MAILBOX AREA
14100
14200 001014    000000    $MAIL:  .WORD   000000 ;MESSAGE TYPE CODE
14300 001016    000000    $FATAL: .WORD   000000 ;FATAL ERROR NO.
14400 001020    000000    $STNO:  .WORD   000000 ;TEST NUMBER
14500 001022    000000    $PASNO: .WORD   000000 ;PASS NUMBER
14600 001024    000000    $DEVCT: .WORD   000000 ;DEVICE COUNT
14700 001026    000000    $UNIT:  .WORD   000000 ;UNIT NO. UNDER TEST
14800 001030    000000    $MSGAD: .WORD   000000 ;MESSAGE ADDRESS (WORD BOUNDRY)
14900 001032    000000    $MSGL:  .WORD   000000 ;MESSAGE LENGTH (IN WORDS)
15000
15100          :APT ENVIRONMENT TABLE
15200
15300 001034     000          $ETABL: .BYTE    0            ;0= STAND ALONE, 1=AUTOMATIC MODE
15400 001035     000          .BYTE    0            ;CONTROL BITS
15500 001036    000000    $SWREG: .WORD   000000 ;APT SWITCH REGISTER
15600 001040    000000    .WORD   000000 ;USER SWITCHES
15700 001042    000000    $CPU:   .WORD   000000 ;CPU TYPE AND OPTIONS
15800 001044    000000    $MEMAD: .WORD   000000 ;MEM TYPE & HIGH ORDER BITS
15900 001046    000000    $MEMAR: .WORD   000000 ;MEMORY ADDRESS- HIGH
16000 001050    000000    $MEMA2: .WORD   000000
16100 001052    000000    $MEMR2: .WORD   000000
16200 001054    000000    $MEMA3: .WORD   000000
16300 001056    000000    $MEMR3: .WORD   000000
16400 001060    000000    $MEMA4: .WORD   000000
16500 001062    000000    $MEMR4: .WORD   000000
16600 001064    000000    $VECT1: .WORD   000000 ;VECTOR #1, AND PRIORITY
16700 001066    000000    $VECT2: .WORD   000000 ;VECTOR #2, AND PRIORITY
16800 001070    000000    $BASE:  .WORD   000000 ;BASE ADDRESS OF DEVICES
16900 001072    000000    $DEVM:  .WORD   000000 ;DEVICE MAP
17000 001074    000000    $ETENL: .WORD   000000 ;END: OF ETABLE
17100

```

17300
17400 001074 160010
17500 001076 000300
17600 001100 000000
17700 001102 000000
17800 001104 000000
17900 001106 000000
18000 001110 000000
18100 001112 000000
18200 001114 000000
18300 001116 000000
18400 001120 000001
18500 001122 177570
18600 001124 000000
18700 001126 000000
18800 001130 000000
18900 001132 000000
19000 001134 000000
19100 001136 000000
19200 001140 000000
19300 001142 000314
19400
19500
19600
19700
19800
19900
20000 001144 000000
20100 001146 000000
20200 001150 000000
20300 001152 000000
20400 001154 000000
20500 001156 000000
20600 001160 000000
20700 001162 000000
20800 001164 000000
20900 001166 000000
21000 001170 000000
21100 001172 000204
21200 001174 000000
21300 001176 000000
21400 001200 000000
21500 001202 000000
21600 001204 000000
21700 001206 000000
21800 001210 000000
21900 001212 000000
22000 001214 000000
22100 001216 000000
22200
22300
22400

.SBTTL COMMON DATA STORAGE
DZADDR: 160010 ;ADDRESS OF 1ST DZ11
DZVECT: 000300 ;ADDRESS OF 1ST DZ11 VECTOR
DXTMP: 000000 ;TEMP STORAGE FOR DZ XMIT INTERRUPT ROUTINE
MSGTYP: 000000
MSGADR: 000000
SENDTM: 000000
ERROR: 000000 ;ERROR SWITCH
SEQ: 000000 ;HOLDS TEST TABLE POINTER
TEST: 000000 ;POINTER TO CURRENT TEST
SO: 000000 ;THIS IS THE SIMULATED SWITCH REGISTER
SRCONT: 000001 ;THIS IS THE SWITCH REGISTER CONTROL SWITCH
SWR: 177570 ;POINTER TO SWITCH REG, OR SOFT SR
PASSNO: 000000 ;THIS IS THE PROGRAM PASS NUMBER
ANTMPO: 000000
ANTMP1: 000000
ANTMP2: 000000
TEMP: 000000
NOTYET: 000000
HOOK: 000000
LOOPC: 000314 ;TIME CONSTANT FOR 11/20
;SET TO 202 FOR 11/03
;SET TO 251 FOR 11/10
;SET TO 554 FOR 11/40
;SET TO 755 FOR 11/45, 11/60
;SET TO 1237 FOR 11/45, 11/70
;SET TO 2127 FOR 11/45 BIP, 11/55
LOOPI: 000000
LOOPD: 000000
TSTMP: 000000
NUMLIN: 000000
COM1: 000000
COM2: 000000
WORK: 000000
WORK1: 000000
WORK2: 000000
WORK3: 000000
CHARIN: 000000
WIDTH: 132 ;SET TO 120 FOR 80 COLM
MODE: 000000 ;DZ TRANSMIT MODE
PMODE: 000000 ;PMT MODE FLAG
RCTMP: 000000
DZNUM: 000000 ;NO. OF DZ'S ACTUALLY ON SYSTEM
ONLINE: 000000 ;LINE NO. UNDER TEST
PNTR: 000000 ;CONSOLE BUFFER POINTER
TMPTST: 000000 ;CONSOLE ROUTINE TEMP FLAGS
TSTTYP: 000000 ;TEST DESCRIPTION DATA
GO: 000000
UUT: 000000 ;# OF UNITS UNDER TEST
.SBTTL START POINT FOR PROGRAM


```
5600 001520 012760 000145 020634      MOV    #145,DZLINE(R0) ;INIT TO 300 BAUD ODD PARITY
5700 001526 062700 000002      ADD    #2,R0
5800 001532 062702 000050      ADD    #50,R2           ;NEW BUF=OLD BUF + 20.
5900 001536 005301              DEC    R1
6000 001540 001344              BNE    4$
6100
6200
6300
6400 001542 012702 032412      ;SETUP VECTORS FOR INTERRUPTS
INIT1: MOV    #DZRINT,R2
        MOV    #DZTINT,R3
6600 001546 012703 032330      MOV    DZVECT,R5
6700 001552 013705 001076      MOV    DZNUM,R4         ;FIRST VECTOR ADDRESS
6800 001556 013704 001202      MOV    R2,(R5)+         ;SETUP A COUNT FOR DZS
6900 001562 010225 000240      MOV    #240,(R5)+       ;SETUP RECIEVE INT VECTOR
7000 001570 010325              MOV    R3,(R5)+         ;AND ITS PRIORITY
7100 001572 012725 000240      MOV    #240,(R5)+       ;SETUP TRANSMIT VECTOR
7200 001576 062703 000012      ADD    #12,R3           ;AND ITS PRIORITY
7300 001602 062702 000012      ADD    #12,R2           ;SET POINTER TO NEXT INT SERVICE ROUTINE
7400 001606 005304              DEC    R4                ;SU NEXT RX INT SVC ROUTINE
7500 001610 001364              BNE    R4                ;NEXT LINE PLEASE
7600 001612 000240              NOP                     ;IF THERE IS ONE
```


13600					
13700	002046	005000		INIT6:	CLR R0
13800	002050	013701	001202		MOV DZNUM,R1
13900	002054	016002	031554	1\$:	MOV DZCSR(R0),R2
14000	002060	012737	002156	000004	MOV #5\$,TRAP4
14100	002066	005762	000006		TST 6(R2)
14200	002072	012737	002164	000004	MOV #6\$,TRAP4
14300	002100	012712	040140		MOV #40140,(R2)
14400	002104	012737	002172	000004	MOV #7\$,TRAP4
14500	002112	112762	000377	000004	MOV #377,4(R2)
14600	002120	012737	002200	000004	MOV #8\$,TRAP4
14700	002126	112762	000377	000005	MOV #377,5(R2)
14800	002134	062700	000002		MOV #2,R0
14900	002140	005301			DEC R1
15000	002142	001344			BNE 1\$
15100	002144	012737	000006	000004	MOV #6,TRAP4
15200	002152	000137	002220		JMP ISEQ
15300					
15400					
15500	002156	000000		5\$:	HALT
15600	002160	000137	002202		JMP 10\$;TRAPPED FROM 16XXX6 RING/CARRIER
15700					
15800	002164	000000		6\$:	HALT
15900	002166	000137	002202		JMP 10\$;TRAPPED FROM 16XXX0 CSR
16000					
16100	002172	000000		7\$:	HALT
16200	002174	000137	002202		JMP 10\$;TRAPPED FROM 16XXX4 TXMIT CTL
16300					
16400	002200	000000		8\$:	HALT
16500	002202	005737	001120	10\$:	TST SRCONT
16600	002206	001002			BNE 11\$
16700	002210	000137	001242		JMP KSTART
16800	002214	000137	001220	11\$:	JMP START
16900					
17000					


```

21100                               ;INITIAL TEST STARTUP SEQUENCE
21200
21300 002450 005737 001120         LSEQ:   TST      SRCONT       ;SWITCH CONTROL ?
21400 002454 001471                   BEQ      20$             ;NO-JUMP TO 20
21500 002456 004737 004036         JSR     PC,GETSWS      ;READ SWITCH REG.
21600 002462 032737 002000 001116 1$:   BIT     #BIT10,SO    ;TEST NO. IN SWS ?
21700 002470 001452                   BEQ     13$             ;NO- GOTO 13
21800 002472 004737 003302         JSR     PC,VALID      ;CHECK VALIDITY
21900 002476 005737 001214         TST     GO
22000 002502 001433                   BEQ     10$             ;NO GOOD GOTO 10
22100 002504 004737 003356         JSR     PC,GETTST     ;TEST ADDR & INFO
22200 002510 004737 003460         JSR     PC,MODCON    ;MODE CONFLICT ?
22300 002514 005737 001214         TST     GO
22400 002520 001402                   BEQ     3$             ;YES- GOTO 3
22500 002522 000137 002730         2$:   JMP     40$             ;OK- GO START TEST
22600 002526 032737 010000 001116 3$:   BIT     #BIT12,SO    ;SEQUENCE TESTS ?
22700 002534 001412                   BEQ     5$             ;YES GOTO 5
22800 002536 005037 001214         4$:   CLR     GO
22900 002542 012705 042552         SENDALL #MSG2,PC      ;ERROR MODE CONFLICT *****
          002542 012705 042552         MOV     #MSG2,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
          002546 005037 001174         CLR     MODE
          002552 004737 031706         JSR     PC,SEND      ;NOW SEND THE MESSAGE
23000 002556 000137 002730         JMP     40$
23100 002562 105237 001116         5$:   INCB   SO
23200 002566 000137 002462         JMP     1$             ;TRY NEXT TEST
23300
23400 002572 012705 042514         10$:  SENDALL #MSG1,PC   ;ERROR INVALID TEST NO. *****
          002572 012705 042514         MOV     #MSG1,R5    ;BUILD SEND CALL USING MESSAGE ADDRESS
          002576 005037 001174         CLR     MODE
          002602 004737 031706         JSR     PC,SEND      ;NOW SEND THE MESSAGE
23500 002606 005037 001214         CLR     GO
23600 002612 000137 002730         JMP     40$
23700
23800 002616 105037 001116         13$:  CLRB   SO           ;SU FOR TEST 0
23900 002622 004737 003356         JSR     PC,GETTST    ;TEST ADDR & INFO
24000 002626 012737 000001 001214 MOV     #1,GO
24100 002634 000137 002730         JMP     40$
24200
24300                               ; CONSOLE CONTROL SECTION
24400
24500 002640 004737 003356         20$:  JSR     PC,GETTST   ;GET TEST ADDR & INFO
24600 002644 004737 003460         JSR     PC,MODCON    ;MODE CONFLICT ?
24700 002650 005737 001214         TST     GO
24800 002654 001402                   BEQ     25$             ;YES- GOTO 25
24900 002656 000137 002730         21$:  JMP     40$             ;GO START TEST
25000 002662 004737 004104         25$:  JSR     PC,ANYWAY   ;RUN ANYWAY ?
25100 002666 042705 000240         BIC     #240,R5
25200 002672 122705 000131         CMPB   #'Y,R5
25300 002676 001411                   BEQ     27$             ;YES GOTO 27
25400 002700 012705 042327         SENDC  #MSGK1        ;SEND 'READY'
          002700 012705 042327         MOV     #MSGK1,R5    ;GET MESSAGE ADDRESS
          002704 004737 020310         JSR     PC,CSEND     ;SEND MESSAGE
25500 002710 012737 177777 001214 MOV     #-1,GO
25600 002716 000137 002730         JMP     40$             ;GO BACK TO WAIT
25700 002722 112737 000001 001214 27$:  MOVB   #1,GO
25800 002730 005737 001214         40$:  TST     GO
25900 002734 001405                   BEQ     43$

```

CZLAIBO LA00, LA34 DMT PROG
START POINT FOR PROGRAM

MACRO M1110 26-FEB-79 14:37 PAGE 28-1 ^{F 3}

SEQ 0031

26000	002736	100002	
26100	002740	000137	002336
26200	002744	000137	002754
26300	002750	000137	002450
26400			

	BPL	42\$
41\$:	JMP	WSEQ
42\$:	JMP	RSEQ
43\$:	JMP	LSEQ

:WAIT FOR NEW COMMANDS
:START TESTING
:GET NEW TEST DATA FROM SWS


```

31800 003260 000000
31900 003262 000137 002450
32000
32100 003266 105037 001116
32200 003272 004737 003356
32300 003276 000137 002754
32400
32500
32600 003302 005037 001214
32700 003306 105737 001116
32800 003312 002407
32900 003314 123727 001116 000022
33000 003322 003003
33100 003324 012737 000001 001214
33200 003332 000207
33300
33400 003334 105037 001214
33500 003340 005737 001212
33600 003344 100403
33700 003346 012737 000001 001214
33800 003354 000207
33900
34000 003356 005037 001160
34100 003362 005037 001112
34200 003366 113737 001116 001160
34300 003374 006337 001160
34400 003400 063737 001160 001112
34500 003406 006337 001112
34600 003412 063737 001160 001112
34700 003420 062737 004424 001112
34800 003426 017737 175460 001114
34900 003434 062737 000002 001112
35000 003442 017737 175444 001212
35100 003450 062737 000002 001112
35200 003456 000207
35300
35400 003460 112737 000001 001214
35500 003466 005737 001176
35600 003472 001405
35700 003474 105737 001212
35800 003500 100002
35900 003502 005037 001214
36000 003506 000207
36100
36200
36300
36400 003510 005037 001134
36500 003514 113737 001116 001134
36600 003522 012705 020600
36700 003526 004737 033730
36800 003532 113737 020604 035106
36900 003540 113737 020605 035107
37000 003546 017737 175340 001134
37100 003554 012705 020600
37200 003560 004737 033730
37300 003564 113737 020603 035075
37400 003572 113737 020604 035076

      HALT
17$:  JMP      LSEQ      ;END OF PASS
      ;GET NEW TEST NO. ETC.

19$:  CLRB     SO
      JSR     PC,GETTST  ;SET TEST 0

22$:  JMP     RSEQ      ;START TEST.....

VALID: CLR     GO
      TSTB   SO
      BLT    4$
      CMPB   SO,#22
      BGT    4$
      MOV    #1,GO
4$:   RTS     PC

REAL:  CLRB   GO
      TST    TSTTYP
      BMI    1$
      MOV    #1,GO
1$:   RTS     PC

GETTST: CLR    WORK
      CLR    SEQ
      MOVB   SO,WORK
      ASL    WORK
      ADD    WORK,SEQ
      ASL    SEQ
      ADD    WORK,SEQ
      ADD    #TSTTBL,SEQ
      MOV    @SEQ,TEST
      ADD    #2,SEQ
      MOV    @SEQ,TSTTYP
      ADD    #2,SEQ      ;POINT TO PASS NO.
      RTS    PC

MODCON: MOVB   #1,GO
      TST    PMODE
      BEQ    2$
      TSTB   TSTTYP
      BPL    2$
      CLR    GO
2$:   RTS     PC

;END OF TEST PASS ROUTINE

EOPT: CLR    TEMP
      MOVB   SO,TEMP      ;CONVERT TEST NO TO ASCII
      MOV    #EBUF,R5
      JSR    PC,BIOCT
      MOVB   EBUF+4,MSG03+23. ;PUT IN MSG03
      MOVB   EBUF+5,MSG03+24.
      MOV    @SEQ,TEMP      ;CONVERT PASS NO.
      MOV    #EBUF,R5
      JSR    PC,BIOCT
      MOVB   EBUF+3,MSG03+14. ;PUT IN MSG03
      MOVB   EBUF+4,MSG03+15.

```

```

37500 003600 113737 020605 035077
37600 003606
      003606 012705 035057
      003612 005037 001174
      003616 004737 031706
37700 003622
      003622 012705 035057
      003626 004737 020310
37800 003632 000207
37900
38000
38100
38200
38300 003634
      003634 012705 035040
      003640 005037 001174
      003644 004737 031706
38400 003650
      003650 012705 035040
      003654 004737 020310
38500 003660 013737 001124 001134
38600 003666 012705 020600
38700 003672 004737 033730
38800 003676 105037 020606
38900 003702
      003702 012705 020600
      003706 005037 001174
      003712 004737 031706
39000 003716
      003716 012705 020600
      003722 004737 020310
39100 003726
      003726 012705 037370
      003732 005037 001174
      003736 004737 031706
39200 003742
      003742 012705 037370
      003746 004737 020310
39300 003752 000207
39400
39500
39600
39700
39800
39900 003754 012737 004002 000004
40000 003762 012737 000340 000006
40100 003770 017737 175126 001116
40200 003776 000240
40300 004000 000407
40400 004002 012737 000176 001122
40500 004010 017737 175106 001116
40600 004016 000002
40700 004020 012737 000006 000004
40800 004026 012737 000000 000006
40900 004034 000207
41000
41100
    
```

```

MOV B EBUF+5,MSG03+16.
SENDALL #MSG03 ;REPORT END OF TEST PASS
MOV #MSG03,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
SENDC #MSG03
MOV #MSG03,R5 ;GET MESSAGE ADDRESS
JSR PC,CSEND ;SEND MESSAGE
RTS PC
    
```

```

:.....:
:END OF PASS SUBROUTINE
EOP: SENDALL #MSG01
      MOV #MSG01,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR MODE
      JSR PC,SEND ;NOW SEND THE MESSAGE
      SENDC #MSG01
      MOV #MSG01,R5 ;GET MESSAGE ADDRESS
      JSR PC,CSEND ;SEND MESSAGE
      MOV PASSNO,TEMP ;CONVERT PASS NO TO ASCII
      MOV #EBUF,R5
      JSR PC,BIOCT
      CLR B EBUF+6 ;PRINT PASS NO.
      SENDALL #EBUF
      MOV #EBUF,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR MODE
      JSR PC,SEND ;NOW SEND THE MESSAGE
      SENDC #EBUF
      MOV #EBUF,R5 ;GET MESSAGE ADDRESS
      JSR PC,CSEND ;SEND MESSAGE
      SENDALL #MSG75 ;SEND CRLF
      MOV #MSG75,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR MODE
      JSR PC,SEND ;NOW SEND THE MESSAGE
      SENDC #MSG75
      MOV #MSG75,R5 ;GET MESSAGE ADDRESS
      JSR PC,CSEND ;SEND MESSAGE
      RTS PC ;RETURN
    
```

```

:.....:
: TEST FOR HARDWARE SWITCH REGISTER
: SWR = 176 IF NONE ON SYSTEM
SWRTST: MOV #2$,TRAP4
        MOV #PRI7,TRAP4+2
1$: MOV @SWR,S0
      NOP
BR 3$
2$: MOV #176,SWR ;TRAPPED TO 4 SET UP FOR
      MOV @SWR,S0 ;SOFTWARE SWITCH REG.
      RTI
3$: MOV #6,TRAP4 ;RESET TRAP CATCHER
      MOV #0,TRAP4+2
      RTS PC
    
```

41200
41300
41400
41500 004036 023727 001122 000176
41600 004044 001001
41700 004046 000000
41800 004050 017737 175046 001116
41900 004056 032737 001000 001116
42000 004064 001404
42100 004066 052737 100000 001176
42200 004074 000402
42300 004076 005037 001176
42400 004102 000207
42500
42600
42700
42800
42900 004104 012737 004146 000060
43000 004112 005005
43100 004114
004114 012705 042426
004120 004737 020310
43200 004124
004124 012705 023420
004130 004737 033676
43300 004134 105705
43400 004136 001002
43500 004140 112705 000116
43600 004144 000207
43700
43800 004146 113705 177562
43900 004152 012737 017240 000060
44000 004160 105737 177564
44100 004164 100375
44200 004166 110537 177566
44300 004172
004172 012705 037370
004176 004737 020310
44400 004202 005037 001146
44500 004206 012737 000101 177560
44600 004214 000002
44700

.....
: ROUTINE TO GET SWITCHES

GETSWS: CMP SWR,#000176 ;REAL SWS ?
BNE 3\$;YES SKIP HALT
HALT ;ALLOW OPERATOR TO CHANGE 176
3\$: MOV @SWR,SO ;READ SWS TO WORK COPY
BIT #BIT9,SO ;PMT MODE ?
BEQ 1\$;NO
BIS #BIT15,PMODE ;YES- SET THE FLAG
BR 2\$
1\$: CLR PMODE
2\$: RTS PC

.....
: ROUTINE TO HANDLE MODE CONFLICTS

ANYWAY: MOV #3\$,@#60 ;SET INTERRUPT TO 3\$
CLR R5
SENDC #MSGK5 ;RUN ANYWAY ? MSG
MOV #MSGK5,R5 ;GET MESSAGE ADDRESS
JSR PC,CSEND ;SEND MESSAGE
STALL #10000
MOV #10000,R5 ;SETUP STALL TIME CONSTANT
JSR PC,MSTALL
TSTB R5
BNE 2\$
1\$: MOVB #'N,R5 ;ASSUME NO OF NO ANS
2\$: RTS PC
3\$: MOVB @#177562,R5 ;GET ANS
MOV #TTYIN,@#60 ;RESTORE TTY INTR HANDLER
4\$: TSTB @#177564
BPL 4\$;ECHO THE CHAR
MOVB R5,@#177566
SENDC #MSG75
MOV #MSG75,R5 ;GET MESSAGE ADDRESS
JSR PC,CSEND ;SEND MESSAGE
CLR LOPO ;ABORT THE TIMEOUT
MOV #101,@#177560 ;ENABLE CONSOLE
RTI

5800	004562	000000	000000	
5900	004564	012260	TEST20	;DMT KEYBOARD ECHO TEST
6000	004566	000220	000220	
6100	004570	000000	000000	
6200	004572	014412	TEST21	;DMT CHARACTER CODE ECHO TEST
6300	004574	000221	000221	
6400	004576	000000	000000	
6500	004600	015214	TEST22	;DMT PITCH SETUP TEST
6600	004602	000222	000222	
6700	004604	000000	000000	
6800	004606	177777	177777	;END OF TABLE FLAG
6900	004610	000000	000000	
7000				

15900
 16000
 16100
 16200 005374 036157
 16300 005376 011320
 16400 005400 000000
 16500 005402 036326
 16600 005404 012720
 16700 005406 000000
 16800
 16900
 17000
 17100
 17200
 17300
 17400
 17500
 17600
 17700
 17800
 17900
 18000
 18100 005410
 005410 012705 036751
 005414 005037 001174
 005420 004737 031706
 18200 005424
 005424 012705 037005
 005430 112737 000004 001174
 005436 112737 000020 001175
 005444 004737 031706
 18300 005450 000207
 18400
 18500

T03TBL: MSG32 ;110 , ODD PARITY , 7 BIT
 11320
 000000
 T03TB2: MSG35 ;300 BAUD , ODD PARITY , 7 BIT
 12720
 000000 ;END OF TABLE

.....
 : THIS IS THE TEST OF DATA PATHS WITHIN THE LA00
 : THE *U*U PATTERN IS ALTERNATING 0 AND ONE BITS

TEST00: SENDALL #MSG42 ;ANNOUNCE TEST
 MOV #MSG42,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
 CLR MODE
 JSR PC,SEND ;NOW SEND THE MESSAGE
 SENDR #MSG43,#4
 MOV #MSG43,R5
 MOVB #4,MODE
 MOVB #20,MODE+1
 JSR PC,SEND
 4\$: RTS PC

```

18700
18800
18900
19000
19100
19200
19300 005452
       005452 012705 037445
       005456 005037 001174
       005462 004737 031706
19400 005466 013737 001172 001160
19500 005474 005037 010070
19600 005500 162737 000006 001160
19700 005506 003403
19800 005510 005237 010070
19900 005514 000771
20000 005516 012737 000041 001164
20100 005524 013737 010070 001160
20200 005532 123727 001164 000177
20300 005540 002050
20400 005542 005737 001160
20500 005546 003433
20600 005550
       005550 013705 001164
       005554 012737 000004 001174
       005562 112737 000020 001175
       005570 004737 032310
20700 005574
       005574 012705 000040
       005600 012737 000002 001174
       005606 112737 000020 001175
       005614 004737 032310
20800 005620 004737 034250
20900 005624 105237 001164
21000 005630 005337 001160
21100 005634 000736
21200 005636
       005636 012705 037370
       005642 005037 001174
       005646 004737 031706
21300 005652 013737 010070 001160
21400 005660 000724
21500 005662
       005662 012705 037373
       005666 005037 001174
       005672 004737 031706
21600 005676 000207
21700
21800

```

```

.....
:PRINTABLE CHARACTERS TEST
:THIS TEST PRINTS FOUR OF EACH PRINTABLE CHARACTER.
:ASCII CODES 041 THRU 176.
.....
TEST01: SENDALL #MSG81 ;SEND TEST ID
          MOV #MSG81,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
          CLR MODE
          JSR PC,SEND ;NOW SEND THE MESSAGE
          MOV WIDTH,WORK
          CLR COUNT
1$: SUB #6,WORK ;WORK = WIDTH / 6
   BLE 2$
   INC COUNT
   BR 1$
2$: MOV #41,WORK2 ;INIT ASCII CODES
   MOV COUNT,WORK
3$: CMPB WORK2,#177 ;DO WHILE CHAR < 177
   BGE 8$
4$: TST WORK ;DO WHILE WORK > 0
   BLE 6$
5$: SENDC2 WORK2,#4 ;SEND CHAR 4 TIMES
   MOV WORK2,R5 ;GET CHAR TO R5
   MOV #4,MODE ;GET REPEAT COUNT
   MOVB #20,MODE+1 ;SET REPEAT MODE
   JSR PC,CHROUT ;CALL CHAR OUTPUT ROUTINE
   SENDC2 #40,#2 ;SEND 2 SPACES
   MOV #40,R5 ;GET CHAR TO R5
   MOV #2,MODE ;GET REPEAT COUNT
   MOVB #20,MODE+1 ;SET REPEAT MODE
   JSR PC,CHROUT ;CALL CHAR OUTPUT ROUTINE
   JSR PC,QUIET
   INCB WORK2 ;NEXT ASCII CODE
   DEC WORK
   BR 3$
6$: SENDALL #MSG75 ;CRLF
   MOV #MSG75,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
   CLR MODE
   JSR PC,SEND ;NOW SEND THE MESSAGE
   MOV COUNT,WORK ;RESTORE WIDTH/6
   BR 3$
8$: SENDALL #MSG77 ;SKIP 3 LINES
   MOV #MSG77,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
   CLR MODE
   JSR PC,SEND ;NOW SEND THE MESSAGE
   RTS PC

```

```

22000
22100
22200
22300
22400
22500
22600
22700 005700
      005700 012705 037104
      005704 005037 001174
      005710 004737 031706
22800 005714
      005714 012705 037214
      005720 005037 001174
      005724 004737 031706
22900 005730 000207
23000
23100
23200
23300
23400
23500
23600
23700
23800
23900
24000
24100
24200
24300
24400 005732
      005732 012705 037511
      005736 005037 001174
      005742 004737 031706
24500 005746
      005746 012705 037373
      005752 005037 001174
      005756 004737 031706
24600 005762
      005762 012705 037536
      005766 005037 001174
      005772 004737 031706
24700 005776
      005776 012705 037550
      006002 112737 000004 001174
      006010 112737 000020 001175
      006016 004737 031706
24800 006022
      006022 012705 037373
      006026 005037 001174
      006032 004737 031706
24900 006036 000207
25000
25100
    
```

```

:.....:
:THIS IS THE NON PRINTING CHARACTER TEST.
:ALL NON-PRINTING CHARS ARE SENT TO THE LA00. THE RESULT SHOULD BE
:A BLANK LINE.
:.....:
    
```

```

TEST02: SENDALL #MSG44
        MOV      #MSG44,R5      ;BUILD SEND CALL USING MESSAGE ADDRESS
        CLR      MODE
        JSR      PC,SEND        ;NOW SEND THE MESSAGE
        SENDALL #MSG45
        MOV      #MSG45,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
        CLR      MODE
        JSR      PC,SEND        ;NOW SEND THE MESSAGE
        RTS      PC
    
```

```

:.....:
:DOT MATRIX TEST
:THIS TEST PRINTS FIVE CHARACTERS 'Z*#$' IN LINE.
:A LINE OF Z'S AND SPACES IS PRINTED, THEN THIS
:LINE IS OVERPRINTED WITH A LINE OF H'S AND SPACES,
:*'S AND SPACES, AND #'S AND SPACES. THIS OVERPRINT
:CREATES TEN BOXES THAT SHOULD BE ALL BLACK.
:.....:
    
```

```

TEST03: SENDALL #MSG83      ;SEND TEST ID
        MOV      #MSG83,R5   ;BUILD SEND CALL USING MESSAGE ADDRESS
        CLR      MODE
        JSR      PC,SEND     ;NOW SEND THE MESSAGE
        SENDALL #MSG77      ;SKIP 3 LINES
        MOV      #MSG77,R5   ;BUILD SEND CALL USING MESSAGE ADDRESS
        CLR      MODE
        JSR      PC,SEND     ;NOW SEND THE MESSAGE
        SENDALL #MSG84      ;SEND THE CHARS
        MOV      #MSG84,R5   ;BUILD SEND CALL USING MESSAGE ADDRESS
        CLR      MODE
        JSR      PC,SEND     ;NOW SEND THE MESSAGE
2$: SENDR #MSG85,#4        ;MAKE 4 LINES OF 10 BOXES
   MOV #MSG85,R5
   MOVB #4,MODE
   MOVB #20,MODE+1
   JSR PC,SEND
5$: SENDALL #MSG77      ;SKIP 3 LINES
   MOV #MSG77,R5       ;BUILD SEND CALL USING MESSAGE ADDRESS
   CLR MODE
   JSR PC,SEND         ;NOW SEND THE MESSAGE
   RTS PC
    
```



```

13900 013034 012703 014140      MOV      #KEYTBL,R3      ;POINT TO TABLE
14000 013040 042737 010000 014302    BIC      #BIT12,FLAG21
14100 013046 005037 001162      CLR      WORK1
14200 013052 020327 014300      12$:    CMP      R3,#KEYEND    ;DONE YET ?
14300 013056 001431      BEQ      13$           ;YES ..GO TO 13$
14400 013060 005723      TST      (R3)+        ;CHECK CHAR IN FLAG(BIT 15)
14500 013062 100773      BMI      12$
14600 013064 005037 001160      CLR      WORK
14700 013070 052737 004000 014302    BIS      #BIT11,FLAG21 ;SET LEFTOVER KEY FLAG
14800 013076 005303      DEC      R3
14900 013100 114337 041307    MOVVB   -(R3),MSG162   ;PUT CHAR IN MSG
15000 013104      SENDI   #MSG162,ONLINE ;AND TYPE IT OUT
                          ;MESSAGE ADDRESS TO R5
                          MOV      #MSG162,R5
013104 012705 041307      MOVVB   #10,MODE+1    ;SET SINGLE LINE MODE
013110 112737 000010 001175    MOVVB   ONLINE,MODE   ;SELECTED LINE NO.
013116 113737 001204 001174    JSR     PC,SEND
013124 004737 031706      JSR     PC,QUIET
15100 013130 004737 034250      ADD     #2,R3         ;GET NEXT TABLE ENTRY
15200 013134 062703 000002      BR      12$          ;KEEP SCANNING FOR LEFTOVERS
15300 013140 000744      BIT     #BIT11,FLAG21 ;ANY LEFTOVERS ?
15400 013142 032737 004000 014302    BEQ     15$          ;NO GO CLEAN THE TABLE ETC.
15500 013150 001465      SENDI   #MSG143,ONLINE ;NOT SEEN MSG.....
15600 013152      MOV     #MSG143,R5    ;MESSAGE ADDRESS TO R5
                          MOVVB   #10,MODE+1    ;SET SINGLE LINE MODE
                          MOVVB   ONLINE,MODE   ;SELECTED LINE NO.
013152 012705 040730      JSR     PC,SEND
013156 112737 000010 001175    MOV     #KEYEND-2,R3
013164 113737 001204 001174    BIC     #BIT15,(R3)  ;RESET SPACE IN FLAG
013172 004737 031706      INC     FLAG21       ;OPERATOR GETS THREE TRIES
15700 013176 012703 014276      MOV     FLAG21,WORK
15800 013202 042713 100000      BIC     #-8.,WORK
15900 013206 005237 014302      CMP     WORK,#3
16000 013212 013737 014302 001160    BGT     14$         ;3 STRIKES YOU'RE OUT !!!
16100 013220 042737 177770 001160    SENDI   #MSG144,ONLINE ;TRY AGAIN MSG.....
16200 013226 023727 001160 000003    MOV     #MSG144,R5    ;MESSAGE ADDRESS TO R5
16300 013234 003026      MOVVB   #10,MODE+1    ;SET SINGLE LINE MODE
16400 013236      MOVVB   ONLINE,MODE   ;SELECTED LINE NO.
013236 012705 040764      JSR     PC,SEND
013242 112737 000010 001175    SENDI   #MSG145,ONLINE ;HIT SPACE LAST MSG.....
013250 113737 001204 001174    MOV     #MSG145,R5    ;MESSAGE ADDRESS TO R5
013256 004737 031706      MOVVB   #10,MODE+1    ;SET SINGLE LINE MODE
16500 013262      MOVVB   ONLINE,MODE   ;SELECTED LINE NO.
013262 012705 040632      JSR     PC,SEND
013266 112737 000010 001175    SENDI   #MSG143,-(SP) ;NEVER RECD ERROR MSG....
013274 113737 001204 001174    MOV     #MSG143,-(SP)
013302 004737 031706      JSR     PC,ERRORT
16600 013306 000137 012406      JMP     3$
16700
16800 013312 012746 040730      .....:.....:.....:.....:.....:.....:.....
16900 013316 004737 020352      14$:    MOV     #MSG143,-(SP) ;NEVER RECD ERROR MSG....
17000 013322 000000      JSR     PC,ERRORT
17100 013324 012703 014140      HALT
17200 013330 042723 100000      15$:    MOV     #KEYTBL,R3    ;CLEAN THE TABLE FLAGS
17300 013334 020327 014300      16$:    BIC     #BIT15,(R3)+
17400 013340 103773      CMP     R3,#KEYEND
17500 013342 005037 014302      BLO     16$
17600 013346      CLR     FLAG21
013346 012705 037373      SENDI   #MSG77,ONLINE  ;SKIP 3 LINES
013352 112737 000010 001175    MOV     #MSG77,R5    ;MESSAGE ADDRESS TO R5
013360 113737 001204 001174    MOVVB   #10,MODE+1    ;SET SINGLE LINE MODE
                          MOVVB   ONLINE,MODE   ;SELECTED LINE NO.

```

52

SEQ 0065

```

17700 013366 004737 031706 JSR PC,SEND
17800 013372 000240 NOP
17900
18000
18100
18200 013374 012737 014304 001166 17$: MOV #CTL TBL,WORK3
18300 013402 012703 014312 MOV #SHITBL-2,R3
18400 013406 012704 014352 MOV #CODTBL,R4
18500 013412 012737 013740 001140 18$: MOV #22$,HOOK
18600 013420 005777 165542 TST @WORK3 ;END OF CTL TBL ?
18700 013424 001004 BNE 19$
18800 013426 005237 001204 INC ONLINE ;SELECT NEXT LINE
18900 013432 000137 012264 JMP 1$ ;YES EXIT TEST
19000 013436 062703 000002 19$: ADD #2,R3
19100 013442 005713 TST (R3) ;END OF SHITBL ?
19200 013444 001004 BNE 20$
19300 013446 062737 000002 001166 ADD #2,WORK3
19400 013454 000761 014302 20$: BR 18$
19500 013456 042737 000200 BIC #BIT7,FLAG21 ;CLEAR DONE FLAG
19600 013464 SENDI #MSG150,ONLINE ;SEND INSTRUNTIONS
013464 012705 041055 MOV #MSG150,R5 ;MESSAGE ADDRESS TO R5
013470 112737 000010 001175 MOV #10,MODE+1 ;SET SINGLE LINE MODE
013476 113737 001204 001174 MOV #ONLINE,MODE ;SELECTED LINE NO.
013504 004737 031706 JSR PC,SEND
19700 013510 SENDI @WORK3,ONLINE ;SEND INSTRUCTION #2
013510 017705 165452 MOV @WORK3,R5 ;MESSAGE ADDRESS TO R5
013514 112737 000010 001175 MOV #10,MODE+1 ;SET SINGLE LINE MODE
013522 113737 001204 001174 MOV #ONLINE,MODE ;SELECTED LINE NO.
013530 004737 031706 JSR PC,SEND
19800 013534 SENDI (R3),ONLINE
013534 011305 MOV (R3),R5 ;MESSAGE ADDRESS TO R5
013536 112737 000010 001175 MOV #10,MODE+1 ;SET SINGLE LINE MODE
013544 113737 001204 001174 MOV #ONLINE,MODE ;SELECTED LINE NO.
013552 004737 031706 JSR PC,SEND
19900 013556 004737 034250 JSR PC,QUIET
20000 013562 STALL #5000. ;WAIT 5 SECONDS
013562 012705 011610 MOV #5000.,R5 ;SETUP STALL TIME CONSTANT
013566 004737 033676 JSR PC,MSTALL
20100 013572 105737 014302 TSTB FLAG21 ;DONE FLAG SET ?
20200 013576 100445 BMI 21$ ;YES BRANCH
20300 013600 SENDI #MSG146,ONLINE ;ERROR DIDN'T RECV CHAR
013600 012705 041001 MOV #MSG146,R5 ;MESSAGE ADDRESS TO R5
013604 112737 000010 001175 MOV #10,MODE+1 ;SET SINGLE LINE MODE
013612 113737 001204 001174 MOV #ONLINE,MODE ;SELECTED LINE NO.
013620 004737 031706 JSR PC,SEND
20400 013624 162703 000002 SUB #2,R3
20500 013630 SENDI (R3)+,ONLINE
013630 012305 MOV (R3)+,R5 ;MESSAGE ADDRESS TO R5
013632 112737 000010 001175 MOV #10,MODE+1 ;SET SINGLE LINE MODE
013640 113737 001204 001174 MOV #ONLINE,MODE ;SELECTED LINE NO.
013646 004737 031706 JSR PC,SEND
20600 013652 SENDI #MSG143,ONLINE
013652 012705 040730 MOV #MSG143,R5 ;MESSAGE ADDRESS TO R5
013656 112737 000010 001175 MOV #10,MODE+1 ;SET SINGLE LINE MODE
013664 113737 001204 001174 MOV #ONLINE,MODE ;SELECTED LINE NO.
013672 004737 031706 JSR PC,SEND

```

```

20700 013676 005237 001110
20800 013702 005737 001116
20900 013706 100253
21000 013710 000000
21100 013712 000240
21200 013714 123724 001170
21300 013720 001646
21400 013722 004737 013772
21500 013726 005304
21600 013730 162703 000002
21700 013734 000137 013436
21800
21900
22000
22100
22200 013740 000240
22300 013742 052737 000200 014302
22400 013750 042705 177600
22500 013754 010537 001170
22600 013760 005037 001146
22700 013764 004737 033446
22800 013770 000207
22900
23000
23100
23200
23300 013772 032737 020000 001116
23400 014000 001056
23500 014002 013737 001170 001134
23600 014010 012705 020600
23700 014014 004737 033730
23800 014020 113737 020603 041132
23900 014026 113737 020604 041133
24000 014034 113737 020605 041134
24100 014042
      014042 012705 041001
      014046 112737 000010 001175
      014054 113737 001204 001174
      014062 004737 031706
24200 014066
      014066 012705 041014
      014072 112737 000010 001175
      014100 113737 001204 001174
      014106 004737 031706
24300 014112
      014112 012705 041132
      014116 112737 000010 001175
      014124 113737 001204 001174
      014132 004737 031706
24400 014136 000207
24500
24600
24700 014140 000054 000055 000056
      014146 000057 000060 000061
      014154 000062 000063 000064
      014162 000073 000047
24800 014166 000065 000066 000067

```

```

INC      ERROR
TST      SO      ;HALT ON ERROR ?
BPL      19$
HALT
21$:     NOP      ;IF BIT 15 SET
      CMPB     CHARIN,(R4)+ ;CHECK FOR CORRECT CODE
      BEQ      19$
      JSR      PC,T21E    ;CALL ERROR ROUTINE
      DEC      R4
      SUB      #2,R3
      JMP      19$

;: CODE CHECKER ROUTINE
22$:     NOP      ;GET CHAR FROM FIFO
      BIS      #BIT7,FLAG21 ;SET DONE FLAG
      BIC      #177600,R5    ;CLEAR PARITY BIT
      MOV      R5,CHARIN
23$:     CLR      LOOP0     ;TURN OFF TIMER
      JSR      PC,KBOUT
      RTS      PC

;:BAD CHAR CODE ROUTINE
T21E:    BIT      #BIT13,SO  ;CHECK SW 13
      BNE      26$
      MOV      CHARIN,TEMP    ;SET UP CONVERTER
      MOV      #EBUF,R5
      JSR      PC,BIOCT      ;CONVERT TO ASCII
      MOV      EBUF+3,MSG149
      MOV      EBUF+4,MSG149+1
      MOV      EBUF+5,MSG149+2
      SENDI    #MSG146,ONLINE
      MOV      #MSG146,R5     ;MESSAGE ADDRESS TO R5
      MOV      #10,MODE+1    ;SET SINGLE LINE MODE
      MOV      ONLINE,MODE   ;SELECTED LINE NO.
      JSR      PC,SEND
      SENDI    #MSG148,ONLINE
      MOV      #MSG148,R5     ;MESSAGE ADDRESS TO R5
      MOV      #10,MODE+1    ;SET SINGLE LINE MODE
      MOV      ONLINE,MODE   ;SELECTED LINE NO.
      JSR      PC,SEND
      SENDI    #MSG149,ONLINE
      MOV      #MSG149,R5     ;MESSAGE ADDRESS TO R5
      MOV      #10,MODE+1    ;SET SINGLE LINE MODE
      MOV      ONLINE,MODE   ;SELECTED LINE NO.
      JSR      PC,SEND
26$:     RTS      PC

KEYTBL: .WORD 54,55,56,57,60,61,62,63,64,73,47

      .WORD 65,66,67,70,71,75,133,134,135

```

014174	000070	000071	000075		
014202	000133	000134	000135		
24900 014210	000140	000141	000142	.WORD	140,141,142,143,144,145,146,147
014216	000143	000144	000145		
014224	000146	000147			
25000 014230	000150	000151	000152	.WORD	150,151,152,153,154,155,156,157
014236	000153	000154	000155		
014244	000156	000157			
25100 014250	000160	000161	000162	.WORD	160,161,162,163,164,165,166,167
014256	000163	000164	000165		
014264	000166	000167			
25200 014270	000170	000171	000172	.WORD	170,171,172,40
014276	000040				
25300 014300	000000			KEYEND: .WORD	0
25400 014302	000000			FLAG21: .WORD	0
25500					
25600 014304	041446	041110	041212	CTLTBL: .WORD	MSG170,MSG156,MSG157,0
014312	000000				
25700					
25800 014314	041136	041066	041255	SHITBL: .WORD	MSG151,MSG152,MSG158,MSG159,MSG154,MSG166
014322	041263	041167	041402		
25900 014330	041416	041431	000000	.WORD	MSG167,MSG168,000000,MSG169,MSG153,000000
014336	041442	041163	000000		
26000 014344	041442	041163	000000	.WORD	MSG169,MSG153,000000
26100					
26200 014352	101	102	011	CODTBL: .BYTE	101,102,011,015,020,010,012,177,104,044,104,064,0
014355	015	020	010		
014360	012	177	104		
014363	044	104	064		
014366	000				
26300				.EVEN	
26400					
26500 014370	005037	001146		END22: CLR	LOOP0
26600 014374	005037	001204		CLR	ONLINE
26700 014400	005037	014302		CLR	FLAG21
26800 014404	005037	001140		CLR	HOOK
26900 014410	000207			RTS	PC
27000					

```

27200
27300
27400
27500
27600
27700
27800
27900
28000 014412 005037 001204
28100 014416 005037 014302
28200 014422 013700 001204
28300 014426 006300
28400 014430 023737 001204 001152
28500 014436 103402
28600 014440 000137 015146
28700 014444 105760 020634
28800 014450 100003
28900 014452 005237 001204
29000 014456 000761
29100
29200 014460
      014460 012705 035524
      014464 112737 000010 001175
      014472 113737 001204 001174
      014500 004737 031706
29300 014504 012737 015164 001140
29400 014512 004737 034250
29500 014516
      014516 012705 010000
      014522 004737 033676
29600 014526 032737 000004 014302
29700 014534 001013
29800 014536
      014536 012705 042366
      014542 112737 000010 001175
      014550 113737 001204 001174
      014556 004737 031706
29900 014562 000753
30000
30100 014564 005037 014302
30200 014570 123727 001170 000177
30300 014576 001557
30400 014600 005037 001160
30500 014604 113737 001170 001160
30600 014612 113737 001160 001134
30700 014620 105037 001135
30800 014624 012705 016324
30900 014630 004737 033730
31000 014634 113737 016327 041132
31100 014642 113737 016330 041133
31200 014650 113737 016331 041134
31300 014656
      014656 012705 041132
      014662 112737 000010 001175
      014670 113737 001204 001174
      014676 004737 031706
31400 014702

```

```

.....
: CHARACTER CODE ECHO TEST 21
: THIS TEST WILL ECHO THE OCTAL CODE OF THE CHARACTER
: RECIEVED, ALONG WITH THE CHARACTER IF IT IS PRINTABLE.
: IF NONPRINTABLE THE MNEMONIC WILL BE RETURNED.
: TYPE A DELETE TO EXIT THIS TEST.
.....

TEST21: CLR ONLINE          ;SU FOR LINE 0
        CLR FLAG21
1$:     MOV     ONLINE,R0
        ASL     R0           ;MAKE WORD OFFSET TO TABLES
        CMP     ONLINE,NUMLIN ;DONE YET ?
        BLO     4$
        JMP     20$
4$:     TSTB    DZLINE(R0)   ;IS LINE SELECTED ?
        BPL     2$           ;YES- GO TEST LINE
        INC     ONLINE       ;NO- TRY NEXT LINE
        BR      1$

2$:     SENDI   #MSG18,ONLINE ;SEND TEST ID MSG
        MOV     #MSG18,R5    ;MESSAGE ADDRESS TO R5
        MOVB    #10,MODE+1   ;SET SINGLE LINE MODE
        MOVB    ONLINE,MODE  ;SELECTED LINE NO.
        JSR     PC,SEND
        MOV     #30$,HOOK
3$:     JSR     PC,QUIET      ;WAIT FOR PRINTING TO FINISH
        STALL   #10000       ;THEN WAIT 10 SECONDS
        MOV     #10000,R5    ;SETUP STALL TIME CONSTANT
        JSR     PC,MSTALL
        BIT     #BIT2,FLAG21 ;CHAR RECVD FLAG SET ?
        BNE     5$          ;YES GOTO 5
        SENDI   #MSGK3,ONLINE ;NO- PROMPT OPERATOR
        MOV     #MSGK3,R5    ;MESSAGE ADDRESS TO R5
        MOVB    #10,MODE+1   ;SET SINGLE LINE MODE
        MOVB    ONLINE,MODE  ;SELECTED LINE NO.
        JSR     PC,SEND
        BR      3$

5$:     CLR     FLAG21
        CMPB    CHARIN,#177  ;DELETE CHAR ?
        BEQ     10$          ;YES JUMP TO 10
        CLR     WORK
        MOVB    CHARIN,WORK  ;SAVE CHAR
        MOVB    WORK,TEMP    ;SU TO CONVERT TO OCTAL/ASCII
        CLRB   TEMP+1
        MOV     #T30BUF,R5
        JSR     PC,BIOCT     ;CONVERT & STORE AT T30BUF
        MOVB    T30BUF+3,MSG149
        MOVB    T30BUF+4,MSG149+1
        MOVB    T30BUF+5,MSG149+2
        SENDI   #MSG149,ONLINE ;SEND OCTAL DATA
        MOV     #MSG149,R5   ;MESSAGE ADDRESS TO R5
        MOVB    #10,MODE+1   ;SET SINGLE LINE MODE
        MOVB    ONLINE,MODE  ;SELECTED LINE NO.
        JSR     PC,SEND
        SENDI   #MSG115,ONLINE ;AND AN '='

```


35400
 35500
 35600
 35700
 35800
 35900
 36000
 36100
 36200 015214 005037 001204
 36300 015220 012701 015622
 36400 015224 005037 001160
 36500 015230 012737 015602 001140
 36600 015236 013700 001204
 36700 015242 006300
 36800 015244 023737 001204 001152
 36900 015252 001550
 37000 015254 105760 020634
 37100 015260 100003
 37200 015262 005237 001204
 37300 015266 000763
 37400 015270
 015270 012705 042136
 015274 112737 000010 001175
 015302 113737 001204 001174
 015310 004737 031706
 37500 015314 005037 001164
 37600 015320 023727 001164 000012
 37700 015326 002403
 37800 015330 005237 001204
 37900 015334 000740
 38000 015336
 015336 012705 041617
 015342 112737 000010 001175
 015350 113737 001204 001174
 015356 004737 031706
 38100 015362 013701 001164
 38200 015366 006301
 38300 015370
 015370 016105 015622
 015374 112737 000010 001175
 015402 113737 001204 001174
 015410 004737 031706
 38400 015414
 015414 012705 041676
 015420 112737 000010 001175
 015426 113737 001204 001174
 015434 004737 031706
 38500 015440 012737 177777 001136
 38600 015446
 015446 012705 035230
 015452 004737 033676
 38700 015456 005737 001136
 38800 015462 001420
 38900 015464 105761 024554
 39000 015470 100410
 39100 015472 012746 036441
 39200 015476 004737 020352

```

.....
: PITCH SETUP TEST
: THIS TEST WILL REQUIRE THE OPERATOR TO ENTER
: SETUP MODE, AND CHANGE THE MODE TO THAT SPECIFIED.
: A LINE OR LINES OF DATA WILL BE PRINTED AND
: SHOULD BE AT THE NEW PITCH.
.....
TEST22: CLR ONLINE ;START ON LINE 0
MOV #TABL24,R1
CLR WORK
MOV #11$,HOOK ;SET INTR CATCHER
1$: MOV ONLINE,R0
ASL R0
CMP ONLINE,NUMLIN ;DONE ALL LINES ?
BEQ 10$ ;YES JUMP
TSTB DZLINE(R0) ;ACTIVE LINE ?
BPL 2$ ;YES- START TESTS
INC ONLINE ;NO- TRY NEXT LINE
BR 1$
2$: SENDI #MSG320,ONLINE ;SEND TEST ID
MOV #MSG320,R5 ;MESSAGE ADDRESS TO R5
MOV #10,MODE+1 ;SET SINGLE LINE MODE
MOV ONLINE,MODE ;SELECTED LINE NO.
JSR PC,SEND
CLR WORK2 ;SUBTEST 0 OF 9
3$: CMP WORK2,#10. ;DONE 10 YET?
BLT 4$ ;NO KEEP TESTING
INC ONLINE ;YES GET NEXT LINE
BR 1$
4$: SENDI #MSG303,ONLINE ;SEND INSTRUCTIONS
MOV #MSG303,R5 ;MESSAGE ADDRESS TO R5
MOV #10,MODE+1 ;SET SINGLE LINE MODE
MOV ONLINE,MODE ;SELECTED LINE NO.
JSR PC,SEND
MOV WORK2,R1 ;POINT TO MSG TABLE
ASL R1
SENDI TABL24(R1),ONLINE ;SEND MSG FOR SUBTEST
MOV TABL24(R1),R5 ;MESSAGE ADDRESS TO R5
MOV #10,MODE+1 ;SET SINGLE LINE MODE
MOV ONLINE,MODE ;SELECTED LINE NO.
JSR PC,SEND
5$: SENDI #MSG304,ONLINE ;MESSAGE ADDRESS TO R5
MOV #MSG304,R5 ;SET SINGLE LINE MODE
MOV #10,MODE+1 ;SELECTED LINE NO.
MOV ONLINE,MODE ;SELECTED LINE NO.
JSR PC,SEND
MOV #-1,NOTYET ;GETS CLEARED BY XON MSG
5$: STALL #15000. ;ALLOW SETUP TIME 15 SEC
MOV #15000.,R5 ;SETUP STALL TIME CONSTANT
JSR PC,MSTALL
TST NOTYET ;SEEN XON YET ?
BEQ 7$ ;YES CONTINUE
TSTB STOP(R1) ;LINE SELECTED ?
BMI 6$ ;YES WAIT MORE TIME
MOV #MSG40,-(SP) ;REPORT ERROR
JSR PC,ERRORT

```

```

39300 015502 000000          HALT              ;IF SW 15 SET
39400 015504 005237 001204   INC      ONLINE   ;TRY NEXT LINE
39500 015510 000652          BR       1$
39600 015512 000240          6$:  NOP
39700 015514 105062 024554   CLRB     STOP(R2)
39800 015520 000137 015446   JMP     5$
39900 015524 000240          7$:  NOP
40000 015526 005761 015650   TST     TAB24B(R1)
40100 015532 001414          BEQ     9$                ;YES    JUMP
40200 015534          SENDI  TAB24B(R1),ONLINE ;SEND THE MSG
      015534 016105 015650   MOV     TAB24B(R1),R5  ;MESSAGE ADDRESS TO R5
      015540 112737 000010 001175   MOVB   #10,MODE+1    ;SET SINGLE LINE MODE
      015546 113737 001204 001174   MOVB   ONLINE,MODE   ;SELECTED LINE NO.
      015554 004737 031706   JSR    PC,SEND
40300 015560 004737 034250   JSR    PC,QUIET
40400 015564 005237 001164          9$:  INC     WORK2       ;SU NEXT SUBTEST
40500 015570 000137 015320   JMP     3$
40600 015574 005037 001140          10$: CLR     HOOK         ;RELEASE INTR CATCHER
40700 015600 000207          RTS     PC            ;EXIT.....
40800
40900
41000 015602 122705 000021          11$: CMPB   #21,R5       ;XON ?
41100 015606 001004          BNE    12$
41200 015610 005037 001136   CLR    NOTYET       ;CLEAR IN XON
41300 015614 005037 001146   CLR    LOOP0        ;ABORT TIMEOUT
41400 015620 000207          12$: RTS     PC
41500
41600
41700 015622 041755 041767 042001   TABL24: .WORD  MSG309,MSG310,MSG311,MSG312,MSG317,MSG314
      015630 042013 042075 042037
41800 015636 042025 041743 042063   .WORD  MSG313,MSG308,MSG316,MSG315,000000
      015644 042051 000000
41900
42000 015650 040056 040056 040056   TAB24B: .WORD  MSG107,MSG107,MSG107,MSG107,MSG321
      015656 040056 042164
42100 015662 042224 042244 042264   .WORD  MSG322,MSG323,MSG324,MSG325,MSG326,000000
      015670 042274 042304 000000

```

```

42300
42400
42500
42600
42700
42800
42900
43000
43100
43200
43300
43400 015676
      015676 012705 041527
      015702 005037 001174
      015706 004737 031706
43500 015712 005037 001124
43600 015716 032737 010000 001116
43700 015724 001021
43800 015726
      015726 012705 000101
      015732 013737 001172 001174
      015740 112737 000020 001175
      015746 004737 032310
43900 015752
      015752 012705 037373
      015756 005037 001174
      015762 004737 031706
44000 015766 000207
44100
44200 015770 013737 001172 001160
44300 015776 162737 000005 001160
44400 016004 113737 001160 007512
44500 016012 012737 000041 001160
44600 016020 112737 000002 007514
44700 016026 005037 001162
44800 016032 004737 016270
44900 016036 032737 010000 001116
45000 016044 001742
45100 016046 013737 001162 007516
45200 016054 001412
45300 016056
      016056 013705 001160
      016062 013737 007516 001174
      016070 112737 000020 001175
      016076 004737 032310
45400 016102
      016102 012705 041612
      016106 005037 001174
      016112 004737 031706
45500 016116 005003
45600 016120 113737 007512 007516
45700 016126 163737 001162 007516
45800 016134 001412
45900 016136
      016136 013705 001160
      016142 013737 007516 001174
      016150 112737 000020 001175
    
```

```

.....
LIFE TEST #15
.....
THIS TEST WILL PRINT A CONTINUOUS PATTERN
OF ALL PRINTABLE CHARACTERS. EACH CHARACTER
WILL BE PRINTED ON 2 FULL LINES,
WITH THE PASS COUNT INBEDDED IN THE LINES.
THIS PATTERN WILL PRECESS 1 CHAR POSITION
EACH LINE PRINTED.
LOOPING IS CONTROLLED BY SWITCH #12.
.....

TEST15: SENDALL #MSG270      ;SEND TEST ID
MOV #MSG270,R5           ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND             ;NOW SEND THE MESSAGE
CLR PASSNO              ; START WITH PASS 0
BIT #BIT12,S0           ;IF LOOPING GO TO SECTION 4
BNE 3$
SENDC2 #'A,WIDTH        ;PRINT A FULL LINE OF A'S
MOV #'A,R5              ;GET CHAR TO R5
MOV WIDTH,MODE          ;GET REPEAT COUNT
MOVB #20,MODE+1         ;SET REPEAT MODE
JSR PC,CHR0UT           ;CALL CHAR OUTPUT ROUTINE
2$: SENDALL #MSG77      ;SKIP 3 LINES
MOV #MSG77,R5           ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND             ;NOW SEND THE MESSAGE
RTS PC                  ;EXIT.....

3$: MOV WIDTH,WORK      ;GET WIDTH
SUB #5,WORK             ;PRECESS LIMIT
MOVB WORK,W1            ;SAVE IN W1
MOV #41,WORK            ;PRINTING CHAR CODE
MOVB #2,W2              ;SU 2 LINES PER CHAR
CLR WORK1               ;CURRENT PRECESS COUNT
JSR PC,GETPN           ;CONVERT PASSNO TO ASCII
4$: BIT #BIT12,S0      ;DO WHILE BIT 12 = 1
BEQ 2$
MOV WORK1,W3            ;GET PRECESS COUNT
BEQ 6$
SENDC2 WORK,W3          ;PRINT THE CHARACTER
MOV WORK,R5             ;GET CHAR TO R5
MOV W3,MODE             ;GET REPEAT COUNT
MOVB #20,MODE+1        ;SET REPEAT MODE
JSR PC,CHR0UT           ;CALL CHAR OUTPUT ROUTINE
6$: SENDALL #MSG271     ;PRINT THE PASS COUNT
MOV #MSG271,R5         ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND            ;NOW SEND THE MESSAGE
CLR R3
MOVB W1,W3              ;CHAR COUNT = WIDTH - 5 - PRECESS CNT
SUB WORK1,W3
BEQ 8$
SENDC2 WORK,W3         ;PRINT CHARS TO END
MOV WORK,R5            ;GET CHAR TO R5
MOV W3,MODE            ;GET REPEAT COUNT
MOVB #20,MODE+1       ;SET REPEAT MODE
    
```

TESTS
 46000 016156 004737 032310
 016162
 016162 012705 037370
 016166 005037 001174
 016172 004737 031706
 46100 016176 005237 001162
 46200 016202 123737 001162 007512
 46300 016210 103402
 46400 016212 005037 001162
 46500 016216 105337 007514
 46600 016222 001020
 46700 016224 112737 000002 007514
 46800 016232 005237 001160
 46900 016236 123727 001160 000177
 47000 016244 001007
 47100 016246 012737 000041 001160
 47200 016254 005237 001124
 47300 016260 004737 016270
 47400 016264 000137 016036
 47500
 47600
 47700 016270 013737 001124 001134
 47800 016276 012705 016324
 47900 016302 004737 033730
 48000 016306 113737 016330 041613
 48100 016314 113737 016331 041614
 48200 016322 000207
 48300
 48400
 48500
 48600
 48700 016324

```

8$:  JSR  PC,CHROUT      ;CALL CHAR OUTPUT ROUTINE
      SENDALL #MSG75     ;SEND CRLF
      MOV  #MSG75,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR  MODE
      JSR  PC,SEND       ;NOW SEND THE MESSAGE
      INC  WORK1         ;NEW PRECESS COUNT
      CMPB WORK1,W1     ;RESET TO 0 IF MAX
      BLO  9$
      CLR  WORK1
9$:  DECB  W2             ;2 LINE DONE YET?
      BNE  10$
      MOVB #2,W2         ;YES RESET LINE COUNT
      INC  WORK          ;GET NEXT CHAR CODE
      CMPB WORK,#177    ;UNLESS ALL DONE
      BNE  10$
      MOV  #41,WORK      ;THEN RESET CHAR CODE AND
      INC  PASSNO        ;INC PASS COUNT
      JSR  PC,GETPN     ;REFORMAT MSG
10$:  JMP  4$           ;GO CHECK SW 11

GETPN: MOV  PASSNO,TEMP
      MOV  #T30BUF,R5
      JSR  PC,BIOC       ;CONVERT TO ASCII
      MOVB T30BUF+4,MSG271+1
      MOVB T30BUF+5,MSG271+2
      RTS  PC
  
```

T30BUF: .B:KW 6

100
200
300
400
500
600
700
800 016340
016340 012705 041547
016344 005037 001174
016350 004737 031706
900 016354 112703 000011
1000 016360 123727 001172 000120
1100 016366 101002
1200 016370 112703 000007
1300 016374
016374 012705 037373
016400 005037 001174
016404 004737 031706
1400 016410
016410 012705 040037
016414 005037 001174
016420 004737 031706
1500 016424 005037 007512
1600 016430 023727 007512 000005
1700 016436 003402
1800 016440 000137 017002
1900 016444 013700 007512
2000 016450 006300
2100 016452 016001 012154
2200 016456
016456 010105
016460 005037 001174
016464 004737 031706
2300 016470 016037 017154 007514
2400 016476 005737 007514
2500 016502 001002
2600 016504 000137 016772
2700 016510 005037 007516
2800 016514 004737 034250
2900 016520
016520 012705 037370
016524 005037 001174
016530 004737 031706
3000 016534 023703 007516
3100 016540 003402
3200 016542 000137 016762
3300 016546 013700 007516
3400 016552 006300
3500 016554 016001 017120
3600 016560
016560 010105
016562 005037 001174
016566 004737 031706
3700 016572 004737 034250
3800 016576 000240
3900 016600 000240

```

: PRINTER EXERCISOR
: THIS TEST WILL PRINT A 10 BY 6 INCH MATRIX OF CHARACTERS
: UTILIZING ALL POSSABLE COMBINATIONS OF PITCH SETTINGS.
:
TEST16: SENDALL #MSG280 ;SEND TEST ID
MOV #MSG280,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
MOVB #9,R3 ;IF 80 COL MAKE 6X8 MATRIX
CMPB WIDTH,#120
BHI 7$
MOVB #7,R3
7$: SENDALL #MSG77 ;SKIP 3 LINES
MOV #MSG77,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
SENDALL #MSG104 ;SET H-PITCH TO 16.5
MOV #MSG104,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
CLR W1 ;DO 6 V PITCH GROUPS
1$: CMP W1,#5 ;IF W1 > 5 GOTO 50
BLE 2$
JMP 50$
2$: MOV W1,R0 ;GET V GROUP NO.
ASL R0
MOV TABLVF(R0),R1 ;POINT TO V PITCH SETUP
SENDALL R1 ;SETUP V PITCH
MOV R1,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
MOV TBL31E(R0),W2 ;GET LINE COUNT FOR THIS PITCH
3$: TST W2 ;IF ALL LINES DONE GOTO 40
BNE 4$
JMP 40$
4$: CLR W3 ;DO 10 H PITCH GROUPS PER LINE
JSR PC,QUIET
SENDALL #MSG75 ;SEND A CRLF
MOV #MSG75,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
5$: CMP W3,R3 ;IF 10 DONE GOTO 30
BLE 6$
JMP 30$
6$: MOV W3,R0 ;POINT TO H PITCH SETUP
ASL R0
MOV TBL31C(R0),R1 ;ADDRESS IN R1
SENDALL R1 ;SETUP H PITCH
MOV R1,R5 ;BUILD SEND CALL USING MESSAGE ADDRESS
CLR MODE
JSR PC,SEND ;NOW SEND THE MESSAGE
JSR PC,QUIET
NOP
NOP

```

4000	016602	013700	007512	MOV	W1,R0	;GET ADDRESS OF CHARACTER
4100	016606	006300		ASL	R0	
4200	016610	010037	001166	MOV	R0,WORK3	
4300	016614	006337	001166	ASL	WORK3	
4400	016620	006337	001166	ASL	WORK3	
4500	016624	063700	001166	ADD	WORK3,R0	;R0= W1*10.
4600	016630	063700	007516	ADD	W3,R0	;R0= V ROW + COLM OFFSET
4700	016634	116037	017024	MOVB	TBL31A(R0),WORK	;PUT CHAR IN WORK
4800	016642	013737	007516	MOV	W3,WORK1	;GET FORMAT SELECTOR
4900	016650	006337	001162	ASL	WORK1	
5000	016654	062737	017214	ADD	#TBL31G,WORK1	
5100	016662	017700	162274	MOV	@WORK1,R0	;R0 HAS SELECTOR
5200	016666	016001	017144	MOV	TBL31D(R0),R1	;R1 HAS OUTPUT MSG ADDRESS
5300	016672	013737	007516	MOV	W3,WORK1	
5400	016700	006337	001162	ASL	WORK1	;GET PRINT REPEAT COUNT
5500	016704	062737	017170	ADD	#TBL31F,WORK1	
5600	016712	017737	162244	MOV	@WORK1,WORK2	;WORK2 HAS REPEAT COUNT
5700	016720	113711	001160	MOVB	WORK,(R1)	;PUT CHAR IN OUTPUT MSG
5800	016724			SENDER	R1,WORK2	;PRINT H GROUP OF CHARS
	016724	010105		MOV	R1,R5	
	016726	113737	001164	MOVB	WORK2,MODE	
	016734	112737	000020	MOVB	#20,MODE+1	
	016742	004737	031706	JSR	PC,SEND	
5900	016746	004737	034250	JSR	PC,QUIET	
6000	016752	005237	007516	INC	W3	;NEXT H GROUP
6100	016756	000137	016534	JMP	5\$	
6200	016762	005337	007514	DEC	W2	;ADJUST LINE COUNT -1
6300	016766	000137	016476	JMP	3\$;DO NEXT LINE
6400	016772	005237	007512	INC	W1	;NEXT V GROUP
6500	016776	000137	016430	JMP	1\$;DO NEXT V GROUP
6600	017002	004737	033062	JSR	PC,RESETO	;RESET THE TERMINALS
6700	017006			SENDALL	#MSG77	
	017006	012705	037373	MOV	#MSG77,R5	;BUILD SEND CALL USING MESSAGE ADDRESS
	017012	005037	001174	CLR	MODE	
	017016	004737	031706	JSR	PC,SEND	;NOW SEND THE MESSAGE
6800	017022	000207		RTS	PC	;ALL DONE...BYE
6900						
7000						
7100						
7200						


```

100
200
300
400
500
600 017240 105737 177560
700 017244 100402
800 017246 000137 017376
900 017252 113777 177562 161726
1000 017260 142777 000200 161720
1100 017266 122777 000033 161712
1200 017274 001002
1300 017276 000137 020150
1400 017302 105737 177564
1500 017306 100375
1600 017310 117737 161672 177566
1700 017316 122777 000003 161662
1800 017324 001010
1900 017326
    017326 012705 042327
    017332 004737 020310
2000 017336 012716 002336
2100 017342 000137 017376
2200 017346 122777 000015 161632
2300 017354 001006
2400 017356
    017356 012705 037370
    017362 004737 020310
2500 017366 000137 017406
2600 017372 005237 001206
2700 017376 012737 000101 177560
2800 017404 000002
2900 017406 012737 020332 001206
3000 017414 127727 161566 000071
3100 017422 002403
3200 017424 142777 000040 161554
3300 017432 127727 161550 000015
3400 017440 001005
3500 017442 012737 020332 001206
3600 017450 000137 017376
3700 017454 122777 000110 161524
3800 017462 001012
3900 017464 052737 100000 001116
4000 017472 052737 100000 001210
4100 017500 005237 001206
4200 017504 000137 017414
4300 017510 122777 000114 161470
4400 017516 001012
4500 017520 052737 040000 001116
4600 017526 052737 040000 001210
4700 017534 005237 001206
4800 017540 000137 017414
4900 017544 122777 000103 161434
5000 017552 001012
5100 017554 042737 140400 001116
5200 017562 042737 140400 001210
5300 017570 005237 001206
    
```

```

.SBTTL  CONSOLE DRIVER ROUTINES
: CONSOLE REC'V INTERRUPT HANDLER
TTYIN:  TSTB    @#177560          ;READY ?
        BMI     1$
        JMP     211$              ;FALSE INTERRUPT
1$:     MOVB    @#177562,@PNTR    ;READ CHAR INTO BUFFER
        BICB    #200,@PNTR       ;STRIP PARITY BIT
        CMPB    #33,@PNTR        ;DECODE INPUT IF ESCAPE
        BNE     111$
        JMP     18$
111$:   TSTB    @#177564          ;ECHO THE CHAR
        BPL     111$
        MOVB    @PNTR,@#177566
        CMPB    #03,@PNTR        ;CTL-C ?
        BNE     113$
        SENDC   #MSGK1            ;SEND READY
        MOV     #MSGK1,R5         ;GET MESSAGE ADDRESS
        JSR     PC,CSEND          ;SEND MESSAGE
        MOV     #WSEQ,(SP)        ;RETURN TO WAIT STATE
        JMP     211$
113$:   CMPB    #15,@PNTR         ;DECODE INPUT IF CR
        BNE     2$
        SENDC   #MSG75            ;ECHO CRLF
        MOV     #MSG75,R5         ;GET MESSAGE ADDRESS
        JSR     PC,CSEND          ;SEND MESSAGE
        JMP     3$
2$:     INC     PNTR              ;GET NEXT BUFFER SPACE
211$:   MOV     #101,@#177560     ;TURN CONSOLE ON AGAIN
        RTI
3$:     MOV     #TKBUF,PNTR
4$:     CMPB    @PNTR,#71
        BLT     5$
        BICB    #40,@PNTR        ;RESET LC BIT IF ALFA
5$:     CMPB    @PNTR,#15         ;STOP DECODE IF CR
        BNE     6$
        MOV     #TKBUF,PNTR      ;RESET BUFFER POINTER FIRST
        JMP     211$
6$:     CMPB    #'H,@PNTR        ;HALT COMMAND?
        BNE     7$
        BIS     #BIT15,SO         ;YES- SET BIT 15
        BIS     #BIT15,TMPTST
        INC     PNTR
        JMP     4$
7$:     CMPB    #'L,@PNTR        ;LOOP COMMAND ?
        BNE     8$
        BIS     #BIT14,SO         ;YES- SET BIT 14
        BIS     #BIT14,TMPTST
        INC     PNTR
        JMP     4$
8$:     CMPB    #'C,@PNTR        ;CLEAR COMMAND ?
        BNE     9$
        BIC     #140400,SO        ;RESET THE BITS
        BIC     #140400,TMPTST
        INC     PNTR
    
```



```

10900 020120 000137 020146
11000 020124 142777 000370 161054      17$:  JMP      16$
      020132 006300                BICB    #370,@PNTR ;STRIP AWAY ASCII BITS
11100 020134 006300                ASL     RO
11200 020136 006300                ASL     RO
11300 020140 157700 161042      ASL     RO ;MAKE ROOM FOR NEW DIGIT
11400 020144 000745                BISB    @PNTR,RO ;ADD NEW LSD
11500 020146 000207                BR      15$ ;GET NEXT CHAR
11600 020150 012705 035657      16$:  RTS      PC ;EXIT OCTAL IN RO
11700 020154 004737 020310      18$:  SENDC   #MSG22 ;ECHO $ AND CRLF
      020160 013737 001210 001116      MOV     #MSG22,R5 ;GET MESSAGE ADDRESS
      020166 062706 000002      JSR     PC,CSEND ;SEND MESSAGE
11900 020172 012746 002450      MOV     TMPTST,SO ;PUT TEST NO IN SO
12000 020176 012737 020332 001206      ADD     #2,SP ;FIX RETURN PC
      020204 012705 042327      MOV     #LSEQ,-(SP) ;TO TEST SEQUENCER
      020210 004737 020310      MOV     #TKBUF,PNTR ;RESTORE BUFFER POINTER
12100 020214 012737 000101 177560      SENDC   #MSGK1 ;SEND 'READY'
12200 020222 012705 000100      MOV     #MSGK1,R5 ;GET MESSAGE ADDRESS
12300 020226 004737 033676      JSR     PC,CSEND ;SEND MESSAGE
12400 020232 000002      MOV     #101,@#177560 ;ENABLE CONSOLE
12500 020234 105700      STALL   #100 ;SETUP STALL TIME CONSTANT
      020236 000002      MOV     #100,R5
12600 020238 000002      JSR     PC,MSTALL ;TO TEST SEQUENCER
12700 020242 001006      RTI
12800 020244 042737 002000 001210      NUMCHK: TSTB   RO ;TEST NO. ENTERED ?
12900 020246 105037 001210      BNE     3$
13000 020248 000207      CLRB   TMPTST
13100 020250 120027 000022      1$:  BIC     #BIT10,TMPTST ;NO SELECT
13200 020252 052737 002000 001210      2$:  RTS     PC ;BYE
13300 020254 110037 001210      3$:  CMPB   RO,#22 ;TOO BIG ?
13400 020256 000766      BGT     4$ ;YES
13500 020258 000766      BIS     #BIT10,TMPTST ;OK SELECT TEST
13600 020260 000766      MOVB   RO,TMPTST ;SAVE TEST NO.
13700 020262 000766      BR      2$
13800 020264 012705 042366      4$:  SENDC   #MSGK3 ;? ? ? ?
      020266 004737 020310      MOV     #MSGK3,R5 ;GET MESSAGE ADDRESS
13900 020268 000756      JSR     PC,CSEND ;SEND MESSAGE
14000 020270 000756      BR      1$

```

.....
; CONSOLE TRANSMIT ROUTINE

```

14600 020310 105715      CSEND: TSTB   (R5) ; NULL?
14700 020312 001406      BEQ     2$ ;YES- ALL DONE
14800 020314 105737 177564      1$:  TSTB   @#177564 ;WAIT FOR READY BIT
      020320 100375      BPL     1$
14900 020322 112537 177566      MOVB   (R5)+,@#177566 ;SEND CHARACTER
15000 020324 000770      BR      CSEND
15100 020326 000207      2$:  RTS     PC
15200 020330 000207
15300
15400 020332      TKBUF: .BLKW 10 ;CONSOLE INPUT BUFFER AREA
15500
15600

```

```

15800
15900
16000
16100
16200
16300
16400
16500
16600 020352 032737 020000 001116
16700 020360 001073
16800 020362 013737 001204 001134
16900 020370 012705 020600
17000 020374 004737 033730
17100 020400 113737 020604 020626
17200 020406 113737 020605 020627
17300 020414 013737 001212 001134
17400 020422 042737 177700 001134
17500 020430 012705 020600
17600 020434 004737 033730
17700 020440 113737 020604 020615
17800 020446 113737 020605 020616
17900 020454
      020454 012705 020610
      020460 112737 000010 001175
      020466 113737 001204 001174
      020474 004737 031706
18000 020500 010346
18100 020502 016603 000004
18200 020506
      020506 010305
      020510 112737 000010 001175
      020516 113737 001204 001174
      020524 004737 031706
18300 020530
      020530 012705 020610
      020534 004737 020310
18400 020540
      020540 010305
      020542 004737 020310
18500 020546 012603
18600 020550 011666 000002
18700 020554 062706 000002
18800 020560 005237 001110
18900 020564 005737 001116
19000 020570 100402
19100 020572 062716 000002
19200 020576 000207
19300
19400
19500 020600 000000 000000 000000
      020606 000000
19600 020610 124 105 123
      020613 124 040 060
      020616 060 054 040
      020621 114 111 116
      020624 105 040 060
      020627 060 040 015
  
```

```

.SBTTL ERROR HANDLER
:.....:
:ERROR: THIS ROUTINE WILL HANDLE THE PRINTING OF
:      ERROR MESSAGES, UPDATE ERROR COUNTS, AND
:      CHECK ON SWITCH 13.
:.....:

ERROR: BIT      #BIT13,SO      ;INHIBIT PRINT ?
      BNE      1$              ;YES JUMP
      MOV      ONLINE,TEMP     ;CONVERT LINE NO. TO ASCII
      MOV      #EBUF,R5
      JSR      PC,BIOCT        ;CALL CONVERTER
      MOV      EBUF+4,MSGE+14. ;FORMAT ERROR MSG
      MOV      EBUF+5,MSGE+15.
      MOV      TSTTYP,TEMP     ;GET TEST NO.
      BIC      #177700,TEMP
      MOV      #EBUF,R5        ;CONVERT IT TO ASCII
      JSR      PC,BIOCT
      MOV      EBUF+4,MSGE+5    ;FORMAT ERROR MSG
      MOV      EBUF+5,MSGE+6
      SENDI    #MSGE,ONLINE     ;TEST AND LINE NO'S
      MOV      #MSGE,R5        ;MESSAGE ADDRESS TO R5
      MOV      #10,MODE+1      ;SET SINGLE LINE MODE
      MOV      ONLINE,MODE     ;SELECTED LINE NO.
      JSR      PC,SEND
      MOV      R3,-(SP)         ;SAVE R3
      MOV      4(SP),R3        ;GET MSG ADDRESS FROM STACK
      SENDI    R3,ONLINE       ;SEND ERROR MSG
      MOV      R3,R5           ;MESSAGE ADDRESS TO R5
      MOV      #10,MODE+1      ;SET SINGLE LINE MODE
      MOV      ONLINE,MODE     ;SELECTED LINE NO.
      JSR      PC,SEND
      SENDC    #MSGE           ;SAME THING TO CONSOLE
      MOV      #MSGE,R5        ;GET MESSAGE ADDRESS
      JSR      PC,CSEND        ;SEND MESSAGE
      SENDC    R3
      MOV      R3,R5           ;GET MESSAGE ADDRESS
      JSR      PC,CSEND        ;SEND MESSAGE
      MOV      (SP)+,R3         ;RESTORE R3
1$:   MOV      (SP),2(SP)       ;ERASE ADDR FROM STACK
      ADD      #2,SP           ;ADJUST STACK POINTER
      INC      ERROR           ;FLAG THE ERROR
      TST      SO              ;HALT ON ERROR SET ?
      BMI      2$
      ADD      #2,(SP)         ;JUMP OVER ERROR HALT
2$:   RTS      PC              ;RETURN

EBUF:  .WORD   0,0,0,0         ;BUFFER AREA
MSGE:  .ASCIZ  /TEST 00, LINE 00 /<15><12> ;STD MSG HEADER
  
```

19700 020632 012 000
19800

.EVEN

100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200

020634

020754

```
.SBTTL  DZ11 DRIVER ROUTINES
:THSES ROUTINES WILL HANDLE FROM 1 TO 8 DZ11'S
:JOHN COMEAU INVENTED THESE WONDERFULL ROUTINES

:NOW A BUNCH OF TABLES

:HERE IS A ONE WORD PER LINE TABLE. IT HOLDS LINE PARAMETERS

:THE PROGRAM IS RESPONSIBLE FOR SETTING IT UP.
:THE DZ11 ROUTINES SIMPLY READ IT.

:BIT 7 IN EACH BYTE, IS THE INACTIVE BIT. IF SET, THE LINE
:WILL BE IGNORED BY THE DRIVER ROUTINES

:BITS 3-0 HOLD THE LINES BAUD RATE INFO
          BITS 3-0/BAUD
          0000      50
          0001      75
          0010     110
          0011     134.5
          0100     150
          0101     300
          0110     600
          0111     1200
          1000     1800
          1001     2000
          1010     2400
          1011     3600
          1100     4800
          1101     7200
          1110     9600
          1111     RESERVED
:BIT 6 SELECTS THE TYPE OF PARITY, 0= EVEN 1=ODD
:BIT 5 IT THE PARITY ENABLING BIT, 0 IF NO PARITY, 1 IF PARITY
DZLINE: .BLKW  DZCON*8.          ; NO. OF DZ'S TIMES 8 LINES PER DZ=# WORDS

:HERE ARE THE DZ11 COMMAND BUFFER AREAS
:THERE IS ONE FOR EACH LINE.
:EACH OF 20 WORDS LONG
:THE COMMAND FORMAT IS AS FOLLOWS.
:1ST WORD IS THE ADDRESS OF THE MESSAGE BEING TYPED
:THE 2ND WORD. IF 0, STANDARD MEGSSAGE
:IF HIGH BYTE IS 10, LOW BYTE HOLDS LINE NO TO SEND TO
:IF HIGH BYTE IS 20 LOW BYTE HOLDS REPEAT COUNT
:IF HIGH BYTE IS 30 LOW BYTE HOLDS SPECIAL TERMINATOR.
DZCOMB: .BLKW  DZCON*8.*20.    ; 8 LINES PER DZ TIMES 20. WORDS PER LINE TI
```

```

5400                                     ;TABLE OF FLAGS FOR ACTIVE LINES
5500 024054 ACTIVE: .BLKW DZCON*8.
5600
5700
5800                                     ;HERE IS THE TABLE OF CURRENT REPEAT COUNTS.
5900 024174 CURREP: .BLKW DZCON*8.
6000
6100                                     ;HERE IF THE TABLE OF CURRENT TERMINATORS
6200 024314 CURTER: .BLKW DZCON*8.
6300
6400                                     ;HERE IS THE LINE REPLY TABLE
6500 024434 REPTBL: .BLKW DZCON*8.
6600
6700                                     ;HERE IS A TABLE OF SWITCH WORDS SET TO CLEAR TCR REG
6800 024554 STOP: .BLKW DZCON*8.
6900
7000                                     ;HERE IS THE TABLE OF CURRENT TEXT ADDRESSES
7100 024674 CURADD: .BLKW DZCON*8.
7200
7300                                     ;HERE ARE THE PRINTING COMMAND BUFFER POINTERS
7400 025014 COMCNT: .BLKW DZCON*8.
7500 025134 COMIN: .BLKW DZCON*8.
7600 025254 COMOUT: .BLKW DZCON*8.
7700 025374 COMEND: .BLKW DZCON*8.
7800
7900 025514 TCRBIT: .BLKW DZCON*8. ;LINE1=1, LINE2=2, LINE3=4, LINE4=10
8000
8100                                     ;CHAR COUNT
8200 025634 KBCNT: .BLKW DZCON*8.
8300
8400                                     ;END OF BUFFER TABLE
8500 025754 KBBUFE: .BLKW DZCON*8.
8600
8700                                     ;BEGIN OF BUFFER TABLE
8800 026074 KBBUFB: .BLKW DZCON*8.
8900
9000                                     ;BUFFER PUT IN POINTER
9100 026214 KBBUFI: .BLKW DZCON*8.
9200
9300                                     ;BUFFER TAKE OUT POINTER
9400 026334 KBBUFO: .BLKW DZCON*8.
9500
9600
9700                                     ;HERE IF THE KEYBOARD BUFFER AREA
9800 026454 KBBUF: .BLKW DZCON*8.*20. ;8 WORDS TIMES 8 LINES TIMES # OF DZS
9900
10000
10100                                     ;DZ11 STATUS REG ADDRESS TABLE
10200 031554 DZCSR: .BLKW DZCON ;ONE CSR PER DZ11(REALLY!)
10300
10400                                     ;DZ11 RECIEVE ERROR BIT TABLE
10500 031566 RECERR: .BLKW DZCON*8.

```

```

10700
10800
10900
11000
11100
11200
11300
11400
11500
11600
11700
11800
11900
12000
12100
12200
12300 031706 010046
12400 031710 010146
12500 031712 010246
12600 031714 010537 001104
12700 031720 122737 000010 001175
12800 031726 001014
12900 031730 105037 001175
13000 031734 013700 001174
13100 031740 006300
13200 031742 005037 001174
13300 031746 012737 000001 001106
13400 031754 000137 031770
13500 031760 013737 001152 001106
13600 031766 005000
13700 031770 105760 020634
13800 031774 100534
13900 031776 026027 025014 000010
14000 032004 002463
14100 032006 005760 024054
14200 032012 100017
14300 032014
      032014 013705 000144
      032020 004737 033676
14400 032024 005760 024554
14500 032030 100006
14600 032032 105260 024054
14700 032036 126027 024054 000144
14800
14900 032044 103002
15000 032046 000137 031770
15100 032052 052760 000200 020634
15200 032060 005337 001216
15300 032064 005060 024554
15400 032070 005060 024054
15500 032074 012705 036404
15600 032100 004737 020310
15700 032104 010037 001134
15800 032110 006237 001134
15900 032114 012705 020600
16000
16100 032120 004737 033730

```

```

.....
DZ SEND ROUTINE
CALLING SEQUENCES
      JSR      PC,SEND      ;CALL
      R5      ;THIS IS THE MESSAGE ADDRESS
      MODE    ;THIS SPECIFIES THE TYPE OF MESSAGE AS FOLLOWS...
      MODE    HIGH BYTE    LOW BYTE
      0       0           SEND TO ALL ACTIVE DZ LINES
      10      SELECT     ;SEND TO SELECTED LINE
                        ;USE LOW BYTE AS LINE NO.
      20      REPEAT     SEND TO ALL ACTIVE LINES
                        USE LOW BYTE AS THE MESSAGE REPEAT COUNT
      30      TERMIN    SEND TO ALL ACTIVE LINES
                        USE LOW BYTE AS MESSAGE TERMINATOR
.....

SEND:  MOV      R0,-(SP)      ;SAVE R0
      MOV      R1,-(SP)      ;AND R1
      MOV      R2,-(SP)      ;AND R2
      MOV      R5,MSGADR
      CMPB     #10,MODE+1    ;IS THIS MESSAGE MEANT FOR ONLY 1 TERMINAL?
      BNE     2$             ;NO.
      CLRB    MODE+1        ;YES
      MOV     MODE,R0        ;GET LINE #
      ASL     R0             ;MAKE WORD OFFSET
      CLR     MODE          ;NO SPECIAL STUFF FOR INDIVIDUAL LINES
      MOV     #1,SENDTM     ;COUNT = 1 LINE ONLY
      JMP     SEND1         ;DO DO IT
2$:    MOV     NUMLIN,SENDTM ; A COUNT OF LINES SO WE KNOW WHEN WE ARE THROUGH
      CLR     R0            ;START WITH THE 1ST LINE
SEND1: TSTB   DZLINE(R0)    ;IS THE LINE INACTIVE?
      BMI     7$           ;IF SO, DONT TRY TO SEND IT ANYTHING.
      CMP     COMCNT(R0),#8. ;ALREADY FULL?
      BLT     4$           ;IF ROOM IS THERE, PUT STUFF IN.
      TST     ACTIVE(R0)   ;IS THE LINE ACTIVE ?
      BPL     2$          ;NO- DESELECT THE LINE
      STALL   100.         ;WAIT A SHORT TIME THEN RETRY
      MOV     100.,R5      ;SETUP STALL TIME CONSTANT
      JSR     PC,MSTALL
      TST     STOP(R0)     ;IS LINE WAITING FOR XON
      BPL     1$           ;NO-
      INCB   ACTIVE(R0)    ;COUNT THIS PASS THRU
      CMPB   ACTIVE(R0),#100. ;CHECK FOR EXCESSIVE DELAY
                        ;ALLOW 10 SECONDS MAX.
                        ;TOO LONG- ABORT WAIT
1$:    BHS    2$
      JMP     SEND1
2$:    BIS    #BIT7,DZLINE(R0) ;DESELECT THE LINE
      DEC    UUT           ;ONE LESS UNIT TO TEST
      CLR    STOP(R0)
      CLR    ACTIVE(R0)
      MOV    #MSG39,R5
      JSR    PC,CSEND      ; REPORT NO XON
      MOV    R0,TEMP
      ASR    TEMP
      MOV    #EBUF,R5
      JSR    PC,BIOCT

```

```

16200 032124 113737 020604 020626      MOVB   FBUF+4,MSGE+14.
16300 032132 113737 020605 020627      MOVB   EBUF+5,MSGE+15.
16400 032140 012705 020620                MOV    #MSGE+10,R5
16500 032144 004737 020310                JSR    PC,SEND
16600 032150 000137 032266                JMP    7$
16700 032154 013770 001104 025134      4$:   MOV    MSGADR,@COMIN(R0);PUT MESSAGE ADDRESS INTO THE COMMAND BUFFER
16800 032162 105060 024054                CLRB   ACTIVE(R0);ERASE ANY DELAY COUNT
16900 032166 062760 000002 025134      ADD    #2,COMIN(R0);BUMP POINTER
17000 032174 013770 001174 025134      MOV    MODE,@COMIN(R0);PUT PRINTING MODE INTO THE BUFFER ALSO
17100 032202 062760 000002 025134      ADD    #2,COMIN(R0);BUMP POINTER
17200 032210 026060 025374 025134      CMP    COMEND(R0),COMIN(R0);IN POINTER AT END OF COMMAND BUFFER?
17300 032216 101003                BHI    6$;IF NOT.
17400 032220 162760 000050 025134      SUB    #50,COMIN(R0);YES, AT END. RESET IT TO THE BEGINING
17500 032226 005260 025014      6$:   INC    COMCNT(R0);ADD 1 TO COUNT OF COMMANDS IN THERE
17600 032232 005760 024554                TST    STOP(R0);IS THE LINE WAITING FOR XON?
17700 032236 100413                BMI    7$;YES. DONT SET TCR BIT
17800 032240 010001                MOV    R0,R1
17900 032242 006201                ASR    R1
18000 032244 006201                ASR    R1
18100 032246 006201                ASR    R1
18200 032250 042701 177761                BIC    #177761,R1
18300 032254 016101 031554                MOV    DZCSR(R1),R1;GET CSR ADDRESS
18400 032260 156061 025514 000004      7$:   BISB   TCRBIT(R0),4(R1);SET THE LINES TCR BIT
18500 032266 062700 000002                ADD    #2,R0;NEXT LINE #
18600 032272 005337 001106                DEC    SENDTM;DONE ALL OF THEM?
18700 032276 001234                BNE    SEND1;NO, GO DO ANOTHER
18800 032300 012602                MOV    (SP)+,R2;NOW ALL WE HAVE TO DO IS
18900 032302 012601                MOV    (SP)+,R1;RESTORE REGS WE
19000 032304 012600                MOV    (SP)+,R0;SAVED UPON ENTRY
19100 032306 000207                RTS    PC;RETURN
19200
19300
19400
19500
19600 032310 162705 000040      CHROUT: SUB    #40,R5;CHARACTER TABLE STARTS AT 40
19700 032314 006305                ASL    R5;MAKE WORD OFFSET
19800 032316 062705 042604                ADD    #PCTABL,R5;ADD PRINT CHAR TABLE ADDRESS
19900 032322 004737 031706                JSR    PC,SEND;SEND MESSAGE WORD
20000
20100 032326 000207                RTS    PC
20200

```

; SINGLE CHARACTER OUTPUT ROUTINE ALL TERMINALS

20400
20500
20600
20700
20800
20900
21000
21100
21200
21300

032330 010046
032332 012700 000000
032336 000137 032474
000002
032342 010046
032344 012700 000002
032350 000137 032474
000004
032354 010046
032356 012700 000004
032362 000137 032474
000006
032366 010046
032370 012700 000006
032374 000137 032474
000010
032400 010046
032402 012700 000010
032406 000137 032474
000012

:HERE ARE THE TRANSMIT INTERRUPT ROUTINES
DZTINT:

X=0
.REPT DZCON
MOV R0,-(SP) ;SAVE R0
MOV #X,R0 ;PUT DZ # IN R0
JMP TXINT ;GO TO MAIN ROUTINE
X=X+2
.ENDR
MOV R0,-(SP) ;SAVE R0
MOV #X,R0 ;PUT DZ # IN R0
JMP TXINT ;GO TO MAIN ROUTINE
Y=X+2
MOV R0,-(SP) ;SAVE R0
MOV #X,R0 ;PUT DZ # IN R0
JMP TXINT ;GO TO MAIN ROUTINE
X=X+2
MOV R0,-(SP) ;SAVE R0
MOV #X,R0 ;PUT DZ # IN R0
JMP TXINT ;GO TO MAIN ROUTINE
X=X+2
MOV R0,-(SP) ;SAVE R0
MOV #X,R0 ;PUT DZ # IN R0
JMP TXINT ;GO TO MAIN ROUTINE
X=X+2
MOV R0,-(SP) ;SAVE R0
MOV #X,R0 ;PUT DZ # IN R0
JMP TXINT ;GO TO MAIN ROUTINE
X=X+2

21400
21500

```
21700
21800 032412
21900          000000
22000          000005
22100
22200
22300
22400
22500
    032412 010046
    032414 012700 000000
    032420 000137 033164
          000002
    032424 010046
    032426 012700 000002
    032432 000137 033164
          000004
    032436 010046
    032440 012700 000004
    032444 000137 033164
          000006
    032450 010046
    032452 012700 000006
    032456 000137 033164
          000010
    032462 010046
    032464 012700 000010
    032470 000137 033164
          000012
22600
22700
```

```

;HERE ARE THE RECIEVE INTERRUPT ROUTINES
DZRINT:
X=0
.REPT DZCON
      MOV    R0,-(SP)      ;SAVE R0
      MOV    #X,R0        ;PUT DZ # IN R0
      JMP    RCINT        ;GO TO MAIN ROUTINE
X=X+2
.ENDR
      MOV    R0,-(SP)      ;SAVE R0
      MOV    #X,R0        ;PUT DZ # IN R0
      JMP    RCINT        ;GO TO MAIN ROUTINE
X=X+2
      MOV    R0,-(SP)      ;SAVE R0
      MOV    #X,R0        ;PUT DZ # IN R0
      JMP    RCINT        ;GO TO MAIN ROUTINE
X=X+2
      MOV    R0,-(SP)      ;SAVE R0
      MOV    #X,R0        ;PUT DZ # IN R0
      JMP    RCINT        ;GO TO MAIN ROUTINE
X=X+2
      MOV    R0,-(SP)      ;SAVE R0
      MOV    #X,R0        ;PUT DZ # IN R0
      JMP    RCINT        ;GO TO MAIN ROUTINE
X=X+2
      MOV    R0,-(SP)      ;SAVE R0
      MOV    #X,R0        ;PUT DZ # IN R0
      JMP    RCINT        ;GO TO MAIN ROUTINE
X=X+2
```

```

22900
23000
23100 032474 010146
23200 032476 010246
23300 032500 000240
23400 032502 016001 031554
23500 032506 006300
23600 032510 006300
23700 032512 011137 001100
23800 032516 013737 001101 001100
23900 032524 042737 177770 001100
24000 032532 063700 001100
24100 032536 006300
24200 032540 005760 024554
24300 032544 100005
24400 032546 146061 025514 000004
24500 032554 000137 033052
24600 032560 052760 100000 024054
24700 032566 005760 024674
24800 032572 001012
24900 032574 005760 025014
25000 032600 001051
25100 032602 146061 025514 000004
25200 032610 005060 024054
25300 032614 000137 033052
25400 032620 117037 024674 001100
25500 032626 005260 024674
25600 032632 123760 001100 024314
25700 032640 001101
25800 032642 005360 024174
25900 032646 003071
26000 032650 005060 024674
26100 032654 062760 000004 025254
26200 032662 026060 025254 025374
26300 032670 103403
26400 032672 162760 000050 025254
26500 032700 005360 025014
26600 032704 001007
26700 032706 146061 025514 000004
26800 032714 005060 024054
26900 032720 000137 033052
27000 032724 017060 025254 024674
27100 032732 005060 024174
27200 032736 005060 024314
27300 032742 016002 025254
27400 032746 062702 000002
27500 032752 011237 001100
27600 032756 001416
27700 032760 122737 000020 001101
27800 032766 001412
27900 032770 122737 000030 001101
28000 032776 001401
28100 033000 000000
28200 033002 113760 001100 024314
28300 033010 000137 032560
28400 033014 105037 001101
28500 033020 013760 001100 024174

```

```

:HERE IS THE MAIN TRANSMIT INTERRUPT ROUTINE
TXINT: MOV R1,-(SP) ;SAVE ALL OF
MOV R2,-(SP) ;REGS WE INTEND TO USE
NOP
MOV DZCSR(R0),R1 ;DZ11 CSR ADDRESS
ASL R0
ASL R0
MOV (R1),DXTMP ;GET LINE #
MOVB DXTMP+1,DXTMP ;MOVE INTO LOW BYTE
BIC #177770,DXTMP ;CLEAR ALL BITS EXCEPT LINE # BITS
ADD DXTMP,R0 ;BIG LINE # IF DZ# PLUS LINE #
ASL R0 ;(DZ# *8 + LINE NO.)*2 FOR OFFSET
TST STOP(R0)
BPL 1$
BICB TCRBIT(R0),4(R1)
JMP 9$
1$: BIS #BIT15,ACTIVE(R0) ;SET LINE ACTIVE FLAG
TST CURADD(R0)
BNE 2$
TST COMCNT(R0)
BNE 4$
BICB TCRBIT(R0),4(R1)
CLR ACTIVE(R0) ;CLEAR THE LINES ACTIVE FLAG
JMP 9$
2$: MOVB @CURADD(R0),DXTMP
INC CURADD(R0) ;POINT AT THE NEXT NEXT CHAR
CMPB DXTMP,CURTER(R0) ;IS IT THE TERMINATOR?
BNE 8$ ;NO. GO XMIT IT.
DEC CURREP(R0)
BGT 7$
CLR CURADD(R0)
ADD #4,COMOUT(R0)
CMP COMOUT(R0),COMEND(R0)
BLO 3$
SUB #50,COMOUT(R0)
3$: DEC COMCNT(R0)
BNE 4$
BICB TCRBIT(R0),4(R1)
CLR ACTIVE(R0)
JMP 9$
4$: MOV @COMOUT(R0),CURADD(R0)
CLR CURREP(R0)
CLR CURTER(R0)
MOV COMOUT(R0),R2 ;GET ADDR OF ADDR
ADD #2,R2
MOV (R2),DXTMP
BEQ 6$
CMPB #20,DXTMP+1
BEQ 6$
CMPB #30,DXTMP+1
BEQ 5$
5$: HALT ;*****
MOV DXTMP,CURTER(R0)
1$
6$: CLRB DXTMP+1
MOV DXTMP,CURREP(R0)

```

```

28600 033026 000137 032560
28700 033032 017060 025254 024674
28800 033040 000137 032560
28900 033044 113761 001100 000006
29000 033052 012602
29100 033054 012601
29200 033056 012600
29300 033060 000002
29400
29500
29600
29700
29800
29900 033062
      033062 012705 040032
      033066 005037 001174
      033072 004737 031706
30000 033076
      033076 012705 040113
      033102 005037 001174
      033106 004737 031706
30100 033112
      033112 012705 007500
      033116 005037 001174
      033122 004737 031706
30200 033126
      033126 012705 037304
      033132 005037 001174
      033136 004737 031706
30300 033142
      033142 012705 035176
      033146 005037 001174
      033152 004737 031706
30400 033156 004737 034250
30500 033162 000207
30600

```

```

      JMP      1$
7$:    MOV     @COMOUT(R0),CURADD(R0)
      JMP     1$
8$:    MOVSB  DXTMP,6(R1)      ;PUT CHAR INTO XMIT BUFFER
9$:    MOV     (SP)+,R2        ;RESTORE THE
      MOV     (SP)+,R1        ;REGISTERS THAT WE
      MOV     (SP)+,R0        ;DESTROYED
      RTI

; THIS ROUTINE IS USED TO RESET ALL TERMINALS

RESETO: SENDALL #MSG103      ;SET 6 LPI.
      MOV     #MSG103,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR     MODE
      JSR     PC,SEND        ;NOW SEND THE MESSAGE
      SENDALL #MSG108      ;SET 10 CPI.
      MOV     #MSG108,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR     MODE
      JSR     PC,SEND        ;NOW SEND THE MESSAGE
      SENDALL #T12FIX      ;RESET MARGINS
      MOV     #T12FIX,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR     MODE
      JSR     PC,SEND        ;NOW SEND THE MESSAGE
      SENDALL #MSG61       ;RESET ALL TABS
      MOV     #MSG61,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR     MODE
      JSR     PC,SEND        ;NOW SEND THE MESSAGE
      SENDALL #MSG10       ;SET TABS EVERY 8
      MOV     #MSG10,R5     ;BUILD SEND CALL USING MESSAGE ADDRESS
      CLR     MODE
      JSR     PC,SEND        ;NOW SEND THE MESSAGE
      JSR     PC,QUIET
      RTS     PC

```

```

30800
30900
31000 033154 010546
31100 033166 010246
31200 033170 010146
31300 033172 016001 031554
31400 033176 016105 000002
31500 033202 100401
31600 033204 000000
31700 033206 010537 001200
31800 033212 113737 001201 001200
31900 033220 042737 177770 001200
32000 033226 006300
32100 033230 006300
32200 033232 063700 001200
32300 033236 006300
32400 033240 050560 031566
32500 033244 042760 107777 031566
32600 033252 042705 177400
32700 033256 032760 000200 020634
32800 033264 001405
32900 033266 042760 000200 020634
33000 033274 005237 001216
33100 033300 122705 000023
33200 033304 001010
33300 033306 052760 100200 024554
33400 033314 146061 025514 000004
33500 033322 000137 033422
33600 033326 122705 000021
33700 033332 001013
33800
33900 033334 156061 025514 000004
34000 033342 042760 100000 024554
34100 033350 052760 000001 024554
34200 033356 000137 033422
34300 033362 010570 026214
34400 033366 062760 000002 026214
34500 033374 026060 026214 025754
34600 033402 001003
34700 033404 016060 026074 026214
34800 033412 105260 025634
34900 033416 001001
35000 033420 000000
35100 033422 005737 001140
35200 033426 001402
35300 033430 004777 145504
35400 033434 012601
35500 033436 012602
35600 033440 012605
35700 033442 012600
35800 033444 000002
35900
36000
36100
36200

```

```

:HERE IS THE MAIN RECIEVE INTERRUPT ROUTINE
RCINT: MOV R5,-(SP)
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV DZCSR(R0),R1
      MOV 2(R1),R5
      BMI 1$
      HALT ;INVALID DATA FROM DZ ? ? ? ? ? ?
1$:   MOV R5,RCTMP
      MOVB RCTMP+1,RCTMP
      BIC #177770,RCTMP
      ASL R0
      ASL R0
      ADD RCTMP,R0
      ASL R0
      BIS R5,RECERR(R0) ;COPY ERROR BITS
      BIC #107777,RECERR(R0) ;DATA VALID,LINE NO.,DATA
      BIC #177400,R5 ;CLEAR ERROR BITS
      BIT #BIT7,DZLINE(R0) ;IS UNIT SELECTED ?
      BEQ 6$
      BIC #BIT7,DZLINE(R0) ;SELECT THE LINE
      INC UUT ;ADD TO UNIT COUNT
6$:   CMPB #23,R5
      BNE 7$
      BIS #100200,STOP(R0);SET STOP FLAG & XOFF FLAGS
      BICB TCRBIT(R0),4(R1) ;DISABLE TX INTR
7$:   JMP RCRTN
      CMPB #21,R5
      BNE KBN
9$:   BISB TCRBIT(R0),4(R1) ;ENABLE TX INTR
      BIC #BIT15,STOP(R0) ;CLEAR STOP FLAG
      BIS #BIT0,STCP(R0) ;SET XON FLAG
8$:   JMP RCRTN
KBN:  MOV R5,@KBBUFI(R0) ;STICK IT IN THERE
      ADD #2,KBBUFI(R0) ;GIVE THE POINTER A LITTLE PUSH TO THE NEXT EMPTY PL
      CMP KBBUFI(R0),KBBUFE(R0) ;IS THAT THE END?
      BNE 1$ ;IF NOT.
      MOV KBBUFB(R0),KBBUFI(R0);YES IT WAS AT THE END. RESET IT
1$:   INCB KBcnt(R0) ;TALLY UP ONE MORE ENTRY
      BNE RCRTN ;AND GO RETURN IF WE HAVE LESS THAN 377 OF THEM
      HALT ;400 ENRTYS IS TOO MANY. LET THIS HALT SERVE AS WARN
RCRTN: TST HOOK ;DOES ANOTHER ROUTINE WANT TO SEE CHARS IMMEDIATLY?
      BEQ 2$ ;NO. GO RETURN
      JSR PC,@HOOK ;YES. GO OFF TO SOME MYSTERIOUS PLACE
2$:   MOV (SP)+,R1
      MOV (SP)+,R2
      MOV (SP)+,R5
      MOV (SP)+,R0 ;FROM INTERRUPT CATCHER
      RTI

```

36400
36500
36600
36700 033446 105760 025634
36800 033452 001003
36900 033454 012705 177777
37000 033460 000420
37100 033462 005360 025634
37200 033466 017005 026334
37300 033472 042705 000400
37400 033476 062760 000002 026334
37500 033504 022760 025754 026334
37600 033512 001003
37700 033514 016060 026074 026334
37800 033522 000207
37900

```
:THIS IS THE TAKE STUFF OUT OF THE KBFO BUFFER ROUTINE  
:CALL USING A "JSR PC"  
:IT RETURNS WITH R5 = THE KBRST ENTRY  
KBOUT: TSTB KBCNT(R0) ;ANYTHING THERE?  
        BNE 1$ ;I HOPE SO  
        MOV #-1,R5  
        BR 2$  
1$: DEC KBCNT(R0) ;REDUCE COUNT OF # ENTRYS IN THERE  
     MOV @KBBUFO(R0),R5 ;GET KBRST ENTRY  
     BIC #400,R5  
     ADD #2,KBBUFO(R0) ;BUMP POINTER TO NEXT ENTRY  
     CMP #KBBUFE,KBBUFO(R0);REACHED THE END OF THE BUFFER SPACE?  
     BNE 2$ ;IF NOT, JUST RETURN  
     MOV KBBUFB(R0),KBBUFO(R0);YES, REACHED END. RESET POINTER TO THE BEGININ  
2$: RTS PC ;RETURN
```



```

41700
41800
41900
42000
42100
42200
42300 033676 010537 001146
42400 033702 013737 001142 001144
42500 033710 000240
42600 033712 005337 001144
42700 033716 001374
42800 033720 005337 001146
42900 033724 003366
43000 033726 000207
43100
43200
43300
43400
43500
43600
43700
43800
43900
44000 033730 113765 001134 000005
44100 033736 006037 001134
44200 033742 113765 001135 000002
44300 033750 006037 001134
44400 033754 006037 001134
44500 033760 113765 001134 000004
44600 033766 006037 001134
44700 033772 113765 001135 000001
44800 034000 006037 001134
44900 034004 006037 001134
45000 034010 113765 001134 000003
45100 034016 006037 001134
45200 034022 113715 001135
45300 034026 142715 000376
45400 034032 142765 000370 000001
45500 034040 142765 000370 000002
45600 034046 142765 000370 000003
45700 034054 142765 000370 000004
45800 034062 142765 000370 000005
45900 034070 152715 000060
46000 034074 152765 000060 000001
46100 034102 152765 000060 000002
46200 034110 152765 000060 000003
46300 034116 152765 000060 000004
46400 034124 152765 000060 000005
46500 034132 000207
46600
    
```

```

:STALL ROUTINE
:CALL WITH JSR,PC
:THE LOCATION FOLLOWING THE CALL SHOULD CONTAIN
:THE AMOUNT OF MILLISECONDS TO HANG IN A NULL LOOP
:RETURN IS TO THE LOCATION +4 OF THE CALL
MSTALL: MOV     R5,LOOP0      ;GET # OF MILLISECONDS
1$:     MOV     LOOPC,LOOPI   ;SETUP CONSTANTFOR CORRECT STALLING TIME
2$:     NOP
        DEC     LOOPI
        BNE    2$
        DEC     LOOP0        ;ONE MILLISECOND DOWN
        BGT    1$           ;SOME MORE TO GO
        RTS     PC          ;RETURN
    
```

```

:BINARY TO ASCII CONVERT SUBROUTINE.
:CALL USING A 'JSR PC'
:DERIVES ASCII CHARACTERS REPRESENTING THE CONTENTS
:OF LOCATION 'TEMP', AND PUTS THEM INTO THE 6 BYTES POINTED TO
:BY R5
:THIS IS A STOLEN ROUTINE. IT IS ROTTENLY WRITEN
    
```

```

BIOCT:  MOVB    TEMP,5(R5)
        ROR     TEMP
        MOVB    TEMP+1,2(R5)
        ROR     TEMP
        ROR     TEMP
        MOVB    TEMP,4(R5)
        ROR     TEMP
        MOVB    TEMP+1,1(R5)
        ROR     TEMP
        ROR     TEMP
        MOVB    TEMP,3(R5)
        ROR     TEMP
        MOVB    TEMP+1,(R5)
        BICB    #376,(R5)
        BICB    #370,1(R5)
        BICB    #370,2(R5)
        BICB    #370,3(R5)
        BICB    #370,4(R5)
        BICB    #370,5(R5)
        BISB    #60,(R5)
        BISB    #60,1(R5)
        BISB    #60,2(R5)
        BISB    #60,3(R5)
        BISB    #60,4(R5)
        BISB    #60,5(R5)
        RTS     PC          ;YEAH
    
```

```

46800
46900
47000
47100
47200 034134 010346
47300 034136 010446
47400 034140 012704 034236
47500 034144 112725 000260
47600 034150 100005
47700 034152 005137 001134
47800 034156 112763 000235 177777
47900 034164 112713 000257
48000 034170 105213
48100 034172 161437 001134
48200 034176 100374
48300 034200 005203
48400 034202 062437 001134
48500 034206 005714
48600 034210 001365
48700 034212 062737 000260 001134
48800 034220 113713 001134
48900 034224 012637 001134
49000 034230 012604
49100 034232 012603
49200 034234 000206
49300
49400
49500 034236 023420
49600 034240 001750
49700 034242 000144
49800 034244 000012
49900 034246 000000
50000
50100
50200
50300
50400 034250 010046
50500 034252 010146
50600 034254 010546
50700 034256 013700 001152
50800 034262 006300
50900 034264 005001
51000 034266 020100
51100 034270 001466
51200 034272 005761 024054
51300 034276 100403
51400 034300 062701 000002
51500 034304 000770
51600 034306 005761 024554 4$:
51700 034312 100047
51800 034314 105261 024054
51900 034320 126127 024054 000144
52000 034326 002441
52100 034330 052761 000200 020634
52200 034336 005337 001216
52300 034342 005061 024554
52400 034346 005061 024054

```

```

: BINARY TO DECIMAL CONVERT ROUTINE
: CALL WITH A JSR SP
: WROTE THIS MYSELF. ITS WONDERFULL.
BIDEC: MOV R3, -(SP) ; SAVE R3
MOV R4, -(SP) ; ALSO R4 WHICH WE WILL USE
MOV #BIDECC, R4 ; POINT R4 AT SOME CONSTANTS
MOVB #260, (R5)+ ; MAKE THE FIRTS DIGIT OF THE NUMBER 0
BPL 1$ ; IS THE # POSITIVE?
COM TEMP ; NO. MAKE IT SO
MOVB #235, -1(R3) ; AND CHANGE THAT 1ST DIGIT TO A '-'
1$: MOVB #257, (R3) ; INIT A DIGIT
2$: INCB (R3) ; ADD 1 TO THE DIGIT
SUB (R4), TEMP ; KEEP SUBTRACTING CONSTANT TILL IT GOES NEGATIVE
BPL 2$ ; IF WE ARE STILL POSITIVE, DO IT AGAIN
INC R3 ; NO WE WENT NEGATIVE. POINT AT THE NEXT DIGIT
ADD (R4)+, TEMP ; ADD BACK THE CONSTANT, AND GO ON TO THE NEXT CONSTA
TST (R4) ; DONE THE 1ST 5 DIGITS YET?
BNE 1$ ; IF NOT, GO BACK AND DO ANOTHER
ADD #260, TEMP ; YES. ONE REMAINS TO BE DONE
MOVB TEMP, (R3) ; SET THE LAST DIGIT NOW.
MOV (SP)+, TEMP ; RESTORE EVERYTHING
MOV (SP)+, R4 ; THAT WE USED TO
MOV (SP)+, R3 ; ITS ORIGINAL VALUE
RTS SP ; AND RETURN

```

```

: CONSTANTS
BIDECC: 10000.
1000.
100.
10.
0.

```

: WAIT FOR MESSAGE TO FINISH PRINTING

```

QUIET: MOV R0, -(SP)
MOV R1, -(SP)
MOV R5, -(SP)
MOV NUMLIN, R0 ; GET NO OF LINES
ASL R0
1$: CLR R1
2$: CMP R1, R0 ; IF DONE GO TO 5
BEQ 5$
TST ACTIVE(R1) ; STILL WORKING ?
BMI 4$ ; STILL SET -BRANCH
3$: ADD #2, R1 ; TEST NEXT LINE
BR 2$
TST STOP(R1) ; WAITING FOR XON ?
BPL 7$ ; NO
INCB ACTIVE(R1) ; COUNT OF PASSES
CMPB ACTIVE(R1), #100. ; ALLOW 10 SECONDS
BLT 7$
BIS #BIT7, DZLINE(R1) ; DESELECT THE LINE
DEC UUT
CLR STOP(R1)
CLR ACTIVE(R1)

```

52500	034352	012705	036404		MOV	#MSG39,R5	
52600	034356	004737	020310		JSR	PC,CSEND	; REPORT NO XON
52700	034362	010137	001134		MOV	R1,TEMP	
52800	034366	006237	001134		ASR	TEMP	
52900	034372	012705	020600		MOV	#EBUF,R5	
53000							
53100	034376	004737	033730		JSR	PC,BIOCT	
53200	034402	113737	020604	020626	MOVB	EBUF+4,MSGE+14.	
53300	034410	113737	020605	020627	MOVB	EBUF+5,MSGE+15.	
53400	034416	012705	020620		MOV	#MSGE+10,R5	
53500	034422	004737	020310		JSR	PC,CSEND	
53600	034426	000137	034264		JMP	1\$	
53700	034432				7\$: STALL	#100.	; DELAY A WHILE
	034432	012705	000144		MOV	#100.,R5	; SETUP STALL TIME CONSTANT
	034436	004737	033676		JSR	PC,MSTALL	
53800	034442	000137	034264		JMP	1\$	
53900	034446	012605			5\$: MOV	(SP)+,R5	
54000	034450	012601			MOV	(SP)+,R1	
54100	034452	012600			MOV	(SP)+,R0	
54200	034454	000207			RTS	PC	
54300							

```

54500                                     ;THIS IS THE REPLY SUBROUTINE
54600                                     ;CALL WITH A JSR PC
54700                                     ;IT WILL WAIT .5 SECONDS FOR REPLY FROM ON ALL LINES
54800                                     ;IF IT SEES A REPLY, THE WORD FOR THE LINE IN THE REPTBL IS SET
54900 034456 010046 REPLY: MOV      RO,-(SP)
55000 034460 010146      MOV      R1,-(SP)          ;SAVE REGS WE USE
55100 034462 013700 001152      MOV      NUMLIN,R0
55200 034466 006300      ASL      RO
55300 034470 162700 000002 1$:  SUB      #2,R0
55400 034474 100404      BMI     2$
55500 034476 005060 024434      CLR     REPTBL(R0)
55600 034502 000137 034470      JMP     1$
55700 034506 012737 034540 001140 2$:  MOV      #9$,HOOK          ;PUT CLAWS INTO INPUT ROUTINE
55800 034514 012705 000764      STALL  #500.             ;WAIT .5 SECONDS
                                ;SETUP STALL TIME CONSTANT
                                MOV      #500.,R5
                                JSR     PC,MSTALL
55900 034524 012737 000000 001140  MOV      #0,HOOK          ;TAKE HOOK OUT OF INPUT ROUTINE
56000 034532 012601      MOV      (SP)+,R1
56100 034534 012600      MOV      (SP)+,R0
56200 034536 000207      RTS     PC              ;RETURN
56300
56400 034540 020527 000023 9$:  CMP      R5,#23          ;IS THE CHAR XOF?
56500 034544 001007      BNE     3$              ;NO.
56600 034546 022705 000021      CMP      #21,R5         ;IS THE CHAR XON?
56700 034552 001407      BEQ     4$              ;YES. LET RECIEVE ROUTINE HANDLE IT
56800 034554 112760 000001 024435  MOVB    #1,REPTBL+1(R0) ;YES. SET HIGH BYTE
56900 034562 000403      BR      4$
57000 034564 112760 000001 024434 3$:  MOVB    #1,REPTBL(R0)  ;SET LOW BYTE INDICATING NOT XON OR XOF
57100 034572 000207      4$:  RTS     PC
57200
57300
57400
57500
57600 034574 012737 000001 001136 ;THIS IS THE SUBROUTINE THAT WAITS FOR A CARRIAGE RETURN
57700 034602 012737 034624 001140 AWAIT: MOV      #1,NOTYET    ;SET NO CR YET SWITCH
57800 034610 005737 001136      MOV      #2$,HOOK      ;PUT HOOK INTO RECIEVE ROUTINE SO WE CAN TEST
57900 034614 001375      1$:  TST     NOTYET        ;SEEN A CARRIAGE RETURN YET?
58000 034616 005037 001140      BNE     1$              ;NO, KEEP LOOKING
58100 034622 000207      CLR     HOOK
58200      RTS     PC              ;RETURN
58300 034624 042705 177600 2$:  BIC     #177600,R5     ;REMOVE JUNK FROM DATA BITS
58400 034630 122705 000015      CMPB   #15,R5         ;CARRIAGE RETURN ?
58500 034634 001002      BNE     3$              ;NO.
58600 034636 005037 001136      CLR     NOTYET        ;YES. MAKE THE SWITCH REFLECT IT
58700 034642 004737 033446 3$:  JSR     PC,KBOUT      ;REMOVE CHAR FROM BUFFER
58800 034646 000207      RTS     PC
58900
59000
59100

```


9700	037775	040	070	040
9800	040001	114	120	111
9900	040006	033	133	064
10000	040013	033	133	063
10100	040020	033	133	062
10200	040025	033	133	064
10300	040032	033	133	061
10400	040037	033	133	063
10500	040044	033	133	065
10600	040051	033	133	062
10700	040056	012	015	101
10800	040113	033	133	061
10900	040120	110	117	122
11000	040153	126	105	122
11100	040204	123	105	124
11200	040231	033	133	060
11300	040244	033	133	060
11400	040257	075	000	
11500	040261	105	122	122
11600	040315	033	133	066
11700	040322	040	062	040
11800	040326	120	122	111
11900	040355	115	125	114
12000	040412	055	055	055
12100	040430	055	055	055
12200	040442	123	105	124
12300	040512	120	122	105
12400	040542	012	015	104
12500	040601	054	040	102
12600	040632	120	122	105
12700	040666	120	122	105
12800	040730	040	072	040
12900	040764	124	122	131
13000	041001	012	015	105
13100	041014	111	116	126
13200	041042	124	105	123
13300	041055	120	122	105
13400	041066	040	122	110
13500	041110	123	105	124
13600	041132	077	077	077
13700	041136	040	114	110
13800	041163	064	012	015
13900	041167	040	103	124
14000	041200	040	105	123
14100	041212	122	105	123
14200	041255	124	101	102
14300	041263	122	105	124
14400	041274	060	061	062
14500	041307	077	012	015
14600	041313	111	116	126
14700	041344	115	111	101
14800	041374	123	120	101
14900	041402	102	101	103
15000	041416	114	111	116
15100	041431	104	105	114
15200	041442	104	012	015
15300	041446	040	000	

MSG97: .ASCII / 8 /
MSG98: .ASCII /LPI /
MSG99: .BYTE 33,133,64,167,0
MSG100: .BYTE 33,133,63,167,0
MSG101: .BYTE 33,133,62,167,0
MSG102: .BYTE 33,133,64,172,0
MSG103: .BYTE 33,133,61,172,0
MSG104: .BYTE 33,133,63,172,0
MSG105: .BYTE 33,133,65,172,0
MSG106: .BYTE 33,133,62,172,0
MSG107: .ASCII <12><15>/ABCDEF GHIJKLMNOPQRSTUVWXYZ/
MSG108: .BYTE 33,133,61,167,0
MSG109: .ASCII /HORIZONTAL PITCH TEST 04/<12><15>
MSG110: .ASCII /VERTICAL PITCH TEST 13/<12><15>
MSG111: .ASCII /SET MARGINS TEST 06/<15>
MSG113: .BYTE 33,133,60,60,61,73,61,63,62,163,0
MSG114: .BYTE 33,133,60,60,60,73,60,60,60,163,0
MSG115: .ASCII /=/
MSG116: .ASCII /ERROR IF NOT AT LH MARGIN/<12><15>
MSG117: .BYTE 33,133,66,172,0
MSG118: .ASCII / 2 /
MSG120: .ASCII /PRINTER BELL TEST 14/<12><15>
MSG123: .ASCII /MULTIPLE LINE FEED TEST 10/<12><15>
MSG124: .ASCII /-----/<15>
MSG125: .ASCII /-----00/<15>
MSG140: .ASCII /SET CAPS LOCK OFF, SHIFT LOCK OFF, THEN /
.ASCII /PRESS ALL PRINTING KEYS./
.ASCII <12><15>/DON'T PRESS ESC, TAB, RETURN/
.ASCII / , BS, OR FUNCTION KEYS./<12><15>
MSG145: .ASCII /PRESS THE SPACE BAR LAST./<12><15>
MSG142: .ASCII /PRESS THE SPACE BAR IF FINISHED/<12><15>
MSG143: .ASCII / : KEYS WERE NOT RECIEVED/<12><15>
MSG144: .ASCII /TRY AGAIN, /
MSG146: .ASCII <12><15>/ERROR * /
MSG148: .ASCII /INVALID CODE RECVD : /
MSG147: .ASCII /TEST #21/<12><15>
MSG150: .ASCII /PRESS /<12><15>
MSG152: .ASCII / RH SHIFT AND B/<12><15>
MSG156: .ASCII /SET SHIFT LOCK, /
MSG149: .ASCII /???/
MSG151: .ASCII / LH SHIFT AND 'A/<12><15>
MSG153: .ASCII /4/<12><15>
MSG154: .ASCII / CTL-P/<12><15>
MSG155: .ASCII / ESCAPE/<12><15>
MSG157: .ASCII /RESET SHIFT LOCK, SET CAPS LOCK, /
MSG158: .ASCII /TAB/<12><15>
MSG159: .ASCII /RETURN/<12><15>
MSG160: .ASCII /0123456789/
MSG162: .ASCII /?/<12><15>
MSG163: .ASCII /INVALID SEQUENCE RECVD/<12><15>
MSG164: .ASCII /MIAN KEYBOARD TEST 20/<12><15>
MSG165: .ASCII /SPACE/
MSG166: .ASCII /BACKSPACE/<12><15>
MSG167: .ASCII /LINEFEED/<12><15>
MSG168: .ASCII /DELETE/<12><15>
MSG169: .ASCII /D/<12><15>
MSG170: .ASCII / /

```

15400 041450 033 133 062
15500 041460 033 133 062
15600 041471 033 133 065
15700 041502 033 133 067
15800 041514 033 133 061
15900 041527 114 111 106
16000 041547 114 101 060
16100 041610 077 000
16200 041612 040 060 060
16300 041617 012 015 105
16400 041676 105 130 111
16500 041743 126 075 106
16600 041755 110 075 104
16700 041767 110 075 103
16800 042001 110 075 102
16900 042013 110 075 101
17000 042025 126 075 101
17100 042037 126 075 102
17200 042051 126 075 104
17300 042063 126 075 105
17400 042075 126 075 103
17500 042107 116 117 040
17600 042136 120 111 124
17700 042164 133 055 055
17800 042174 133 055 055
17900 042204 133 055 055
18000 042214 133 055 055
18100 042224 133 055 055
18200 042234 133 055 055
18300 042244 133 055 055
18400 042254 133 055 055
18500 042264 133 055 055
18600 042274 133 055 055
18700 042304 133 055 055
18800 042314 133 055 055
18900 042327 012 015 122
19000 042341 105 116 124
19100 042366 077 040 077
19200 042400 012 015 105
19300 042426 122 125 116
19400 042460 007 012 015
19500 042514 012 015 105
19600 042552 012 015 105
19700
19800 042604 000040 000041 000042
19900 042644 000060 000061 000062
20000 042704 000100 000101 000102
20100 042732 000113 000114 000115
20200 042762 000127 000130 000131
20300 043012 000143 000144 000145
20400 043042 000157 000160 000161
20500 043066 000171 000172 000173
20600 001220

```

```

MSG180: .BYTE 33,133,62,73,62,66,163,0
MSG181: .BYTE 33,133,62,66,73,65,60,163,0
MSG182: .BYTE 33,133,65,62,73,67,66,163,0
MSG183: .BYTE 33,133,67,70,73,61,60,62,163,0
MSG184: .BYTE 33,133,61,60,60,73,61,62,64,163,0
MSG270: .ASCIZ /LIFE TEST #15/<12><15>
MSG280: .ASCIZ /LA00 DYNAMIC EXERCISOR TEST 16/<12><15>
MSG281: .ASCIZ /?/
MSG271: .ASCIZ / 00 /
MSG303: .ASCIZ <12><15>/ENTER SETUP MODE, THEN TYPE THE FOLLOWING : /
MSG304: .ASCIZ /EXIT SETUP MODE, AND TYPE A CTL-Q/<12><15>
MSG305: .ASCIZ /V=F(CR)/<12><15>
MSG309: .ASCIZ /H=D(CR)/<12><15>
MSG310: .ASCIZ /H=C(CR)/<12><15>
MSG311: .ASCIZ /H=B(CR)/<12><15>
MSG312: .ASCIZ /H=A(CR)/<12><15>
MSG313: .ASCIZ /V=A(CR)/<12><15>
MSG314: .ASCIZ /V=B(CR)/<12><15>
MSG315: .ASCIZ /V=D(CR)/<12><15>
MSG316: .ASCIZ /V=E(CR)/<12><15>
MSG317: .ASCIZ /V=C(CR)/<12><15>
MSG318: .ASCIZ /NO RESPONSE RECIEVED/<12><15>
MSG320: .ASCIZ /PITCH SETUP TEST 22/<12><15>
MSG321: .ASCII /[-----/<12><15>
          .ASCII /[-----/<12><15>
          .ASCII /[-----/<12><15>
          .ASCII /[-----/<12><15>
MSG322: .ASCII /[-----/<12><15>
          .ASCII /[-----/<12><15>
MSG323: .ASCII /[-----/<12><15>
          .ASCII /[-----/<12><15>
MSG324: .ASCII /[-----/<12><15>
MSG325: .ASCII /[-----/<12><15>
MSG326: .ASCII /[-----/<12><15>
          .ASCIZ /[-----/<12><15><12><12>
MSGK1: .ASCIZ <12><15>/READY/<12><15>
MSGK2: .ASCIZ /ENTER MODE D OR P :/
MSGK3: .ASCIZ /? ? ? ?/<12><15>
MSGK4: .ASCIZ <12><15>/ENTER COMMAND(S) /<12><15>
MSGK5: .ASCIZ /RUN INTERVENTION TEST ?/<12><15>
MSGK6: .ASCIZ <007><12><15>/NO TERMINALS SELECTED/<007><12><15>
MSGS1: .ASCIZ <12><15>/ERROR * INVALID TEST NO. /<12><15><07>
MSGS2: .ASCIZ <12><15>/ERROR * PMT CONFLICT/<12><15><07>
.EVEN
PCTABL: .WORD 40,41,42,43,44,45,46,47,50,51,52,53,54,55,56,57
         .WORD 60,61,62,63,64,65,66,67,70,71,72,73,74,75,76,77
         .WORD 100,101,102,103,104,105,106,107,110,111,112
         .WORD 113,114,115,116,117,120,121,122,123,124,125,126
         .WORD 127,130,131,132,133,134,135,136,137,140,141,142
         .WORD 143,144,145,146,147,150,151,152,153,154,155,156
         .WORD 157,160,161,162,163,164,165,166,167,170
         .WORD 171,172,173,174,175,176
.END START

```

SYMBOL TABLE	
ACTDVC	000041
ACTIVE	024054
ANTMPC	001126
ANTMP1	001130
ANTMP2	001132
ANVENT	033524
ANYWAY	004104
APTHDR	001000
AWAIT	034574
BIDEC	034134
BIDECC	034236
BIOCT	033730
BIT0	= 000001
BIT1	= 000002
BIT10	= 002000
BIT11	= 004000
BIT12	= 010000
BIT13	= 020000
BIT14	= 040000
BIT15	= 100000
BIT2	= 000004
BIT3	= 000010
BIT4	= 000020
BIT5	= 000040
BIT6	= 000100
BIT7	= 000200
BIT8	= 000400
BIT9	= 001000
CHARIN	001170
CHROUT	032310
CODTBL	014352
COLTBL	010756
COMCNT	025014
COMEND	025374
COMIN	025134
COMOUT	025254
COM1	001154
COM2	001156
COUNT	010070
CSEND	020310
CTLTBL	014304
CURADD	024674
CURREP	024174
CURTER	024314
DLRVEC=	000060
DLTVEC=	000064
DXTMP	001100
DZADDR	001074
DZCOMB	020754
DZCON =	000005
DZCSR	031554
DZLINE	020634
DZNUM	001202
DZRINT	032412
DZTINT	032330
DZVECT	001076
EBUF	020600
END22	014370
EOP	003634
EOPT	003510
ERROR	001110
ERRORT	020352
FLAG21	014302
GETPN	016270
GETSWS	004036
GETTST	003356
GO	001214
HOOK	001140
INIT	001342
INIT1	001542
INIT2	001614
INIT3	001674
INIT4	001726
INIT5	001760
INIT6	002046
ISEQ	002220
KBBUF	026454
KBBUFB	026074
KBBUFE	025754
KBBUFI	026214
KBBUFO	026334
KBCNT	025634
KBN	033362
KBOUT	033446
KEYEND	014300
KEYTBL	014140
KSTART	001242
LOOPC	001142
LOOPI	001144
LOOPO	001146
LSEQ	002450
MODCON	003460
MODE	001174
MSGADR	001104
MSGE	020610
MSGK1	042327
MSGK2	042341
MSGK3	042366
MSGK4	042400
MSGK5	042426
MSGK6	042460
MSG1	042514
MSG2	042552
MSGTYP	001102
MSG00	034650
MSG01	035040
MSG03	035057
MSG04	035113
MSG05	035155
MSG06	035162
MSG08	035172
MSG09	035174
MSG10	035176
MSG100	040013
MSG101	040020
MSG102	040025
MSG103	040032
MSG104	040037
MSG105	040044
MSG106	040051
MSG107	040056
MSG108	040113
MSG109	040120
MSG110	040153
MSG111	040204
MSG113	040231
MSG114	040244
MSG115	040257
MSG116	040261
MSG117	040315
MSG118	040322
MSG12	035264
MSG120	040326
MSG123	040355
MSG124	040412
MSG125	040430
MSG13	035305
MSG14	035327
MSG140	040442
MSG142	040666
MSG143	040730
MSG144	040764
MSG145	040632
MSG146	041001
MSG147	041042
MSG148	041014
MSG149	041132
MSG15	035415
MSG150	041055
MSG151	041136
MSG152	041066
MSG153	041163
MSG154	041167
MSG155	041200
MSG156	041110
MSG157	041212
MSG158	041255
MSG159	041263
MSG16	035455
MSG160	041274
MSG162	041307
MSG163	041313
MSG164	041344
MSG165	041374
MSG166	041402
MSG167	041416
MSG168	041431
MSG169	041442
MSG17	035503
MSG170	041446
MSG18	035524
MSG180	041450
MSG181	041460
MSG182	041471
MSG183	041502
MSG184	041514
MSG19	035554
MSG20	035602
MSG21	035630
MSG22	035657
MSG25	035663
MSG26	035706
MSG27	035762
MSG270	041527
MSG271	041612
MSG28	036007
MSG280	041547
MSG281	041610
MSG29	036043
MSG30	036052
MSG303	041617
MSG304	041676
MSG308	041743
MSG309	041755
MSG31	036074
MSG310	041767
MSG311	042001
MSG312	042013
MSG313	042025
MSG314	042037
MSG315	042051
MSG316	042063
MSG317	042075
MSG318	042107
MSG32	036157
MSG320	042136
MSG321	042164
MSG322	042224
MSG323	042244
MSG324	042264
MSG325	042274
MSG326	042304
MSG33	036163
MSG35	036326
MSG36	036332
MSG37	036342
MSG38	036373
MSG39	036404
MSG40	036441
MSG41	036477
MSG42	036751
MSG43	037005
MSG44	037004
MSG45	037214
MSG47	037242
MSG46	037242
MSG60	037245
MSG61	037304
MSG62	037307
MSG63	037311
MSG64	037314
MSG65	037316
MSG66	037320
MSG67	037322
MSG68	037324
MSG70	037330
MSG71	037333
MSG72	037337
MSG73	037344
MSG75	037370
MSG77	037373
MSG78	037400
MSG79	037435
MSG80	037440
MSG81	037445
MSG82	037507
MSG83	037511
MSG84	037536
MSG85	037550
MSG88	037677
MSG89	037727
MSG90	037734
MSG91	037741
MSG92	037746
MSG93	037753
MSG94	037761
MSG95	037765
MSG96	037771
MSG97	037775
MSG98	040001
MSG99	040006
MSTALL	033676
NOTYET	001136
NUMCHK	020234
NUMLIN	001152
ONLINE	001204
PASSNO	001124
PCTABL	042604
PFAIL	000024
PMODE	001176
PNTR	001206
PRI0 =	000000
PRI4 =	000200
PRI7 =	000340
QUIET	034250
RCINT	033164
RCRTN	033422
RCTMP	001200
REAL	003334
RECERR	031566
REPLY	034456
REPTBL	024434
RESETO	033062
RSEQ	002754
SCAN	004216
SEND	031706

SYMBOL TABLE

SENDTM	001106	TBL12B	007466	TEST12	011164	T12FIX	007500	SETABL	001034
SEND1	031770	TBL12C	007473	TEST13	011630	T17A	012134	SETEND	001074
SEQ	001112	TBL31A	017024	TEST14	012170	T17B	012136	\$FATAL	001016
SEQMS	002342	TBL31C	017120	TEST15	015676	T21E	013772	\$MAIL	001014
SEQ8	002316	TBL31D	017144	TEST16	016340	T30BUF	016324	\$MEMAD	001044
SHITBL	014314	TBL31E	017154	TEST17	004612	UUT	001216	\$MEMAR	001046
SO	001116	TBL31F	017170	TEST20	012260	VALID	003302	\$MEMA2	001050
SRCONT	001120	TBL31G	017214	TEST21	014412	WIDTH	001172	\$MEMA3	001054
START	001220	TCRBIT	025514	TEST22	015214	WORK	001160	\$MEMA4	001060
STOP	024554	TEMP	001134	TKBUF	020332	WORK1	001162	\$MEMR2	001052
SWR	001122	TEST	001114	TMPTST	001210	WORK2	001164	\$MEMR3	001056
SWRTST	003754	TEST00	005410	TRAP4	000004	WORK3	001166	\$MEMR4	001062
TAB	010066	TEST01	005452	TSTMP	001150	WSEQ	002336	\$MSGAD	001030
TABLH	006372	TEST02	005700	TSTTBL	004424	W1	007512	\$MSGL	001032
TABLHF	006402	TEST03	005732	TSTTYP	001212	W2	007514	\$PASNG	001022
TABLV	012140	TEST04	006040	TTYIN	017240	W3	007516	\$SWREG	001036
TABLVF	012154	TEST05	006412	TXINT	032474	X	= 000012	\$TSTNO	001020
TABL13	010050	TEST06	006610	T03TBL	005374	\$BASE	001070	\$UNIT	001026
TABL24	015622	TEST07	007520	T03TB2	005402	\$CPU	001042	\$VECT1	001064
TAB24B	015650	TEST10	010072	T11A	006366	\$DEVCT	001024	\$VECT2	001066
TBL12A	007454	TEST11	010464	T11B	006370	\$DEVM	001072		

. ABS. 043102 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 21488 WORDS (84 PAGES)
DYNAMIC MEMORY: 20620 WORDS (79 PAGES)
ELAPSED TIME: 00:02:12
CZLAIB.BIN,CZLAIB.SEQ=CZLAIB.MAC