VT61/VT71T

MULTI LINE LOOPBACK TEST

MD-11-DZVTL-A

EP-DZVTL-A-DL-A

FICHE 1 OF 1

APR 1977

MADE IN USA

B01

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZVTL-A-D

PRODUCT NAME: VT61/T &VT71/T LOOPBACK TEST

PROGRAM DATE: APRIL 1977

MAINTAINER: DIAGNOSTICS

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR FIELD SERVICE PERSONEL. IT PLACES THE VT61/T IN LOOPBACK MODE AND TRANSMITS A 256 RANDOM CHARACTER PATTERN VIA DH-11, AND COMPARES LOOPBACKED DATA FOR ERRORS.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

A. PDP-11 SYSTEM WITH AT LEAST 4K OF CORE.
B. TTY
C. DH-11
D. VT61/T OR LOOPBACK CONNECTOR FOR DH-11
E. VT71/T WITH MRV11-VA CARD INSTALLED
F. KW11/L OR KW11/P.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE
THE ABS LOADER IS USED TO LOAD THE PROGRAM AND OVERLAYS.

4.0 OPERATING PROCEDURES.

A.
1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

B. CONSOLE DIALOGUE PARAMETER INPUT

1. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
   A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
   B. TYPEIN ACTUAL BUS ADDRESS

2. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
   A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS
   B. TYPEIN ACTUAL VECTOR ADDRESS

3. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
   NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.
   A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
   B. TYPEIN ACTUAL VALUE

4. THE PROGRAM WILL ASK FOR THE LINE SELECTION
   A. TYPE IN VALUES (IN OCTAL) OF LINES DESIRED
      SEPARATED BY COMMAS

(U) TO SELECT LINES 17,16,8,2,0
ENTER 17,16,8,2,0 <CR> THE ORDER

DOES NOT MATTER

5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
   A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
   B. ENTER ACTUAL VALUE

6. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
   A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
      AND TYPE A CAR. RETURN.

NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING NEW VALUES, THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT RESTARTS OF THE PROGRAM.

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1.	<CR>	IF NO CHANGES ARE TO BE MADE
2.	6 DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3.	↑U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR
SW14=1 SINGLE PASS AND REQUEST NEW PARAMS
SW13=1 INHIBIT ERROR TYPEOUTS
SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
SW09=1 DATA SELECT
  0=USE 256 RANDOM CHARACTER PATTERN
  1=USE DATA INPUT FROM OPERATOR
SW07=1 DO NOT TEST RECEIVED DATA
SW04=1 TYPE OUT LINE BEING TESTED
SW03=1 REPORT FIRST DATA COMPARE ERROR ONLY
  =0 REPORT ALL DATA COMPARE ERRORS
SW02=1 TYPE TRANSMITTED AND RECEIVED DATA

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A
REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE.
TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN
↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 40 TO 177)
SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).

F. TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING
CHARACTERS ON THE CONSOLE TTY.

QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)
THEN TYPE EITHER:
*WXXXXXX TO PRINTOUT THE 8 WORDS
AT LOC XXXXXX.

*BXXXXXY TO PRINTOUT THE 16 BYTES
AFTER LOC XXXXXX.

*C TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.

↑C = EXIT LOOPBACK MODE (E.E.SEND ↑C TO VT61/T)
AND RESTART PROGRAM AT LOC 200.

9.0 PARAMETERS FOR THE DH11
PARAM#1 IS LOADED INTO THE LINE PARAMETER REGISTER, (LPR)
BITS 0,1 CHARACTER LENGTH, DEFAULT= 8 BITS (11)
BIT 2 STOP BITS, DEFAULT= 1 STOP BITS (0)
BIT 4 PARITY ENABLED (1), DEFAULT= (1)
BIT 5 ODD PARITY (1), DEFAULT= (0)
BITS 6-9 RECEIVER SPEED, DEFAULT= 9600 BAUD (1101)
BITS 10-13 TRANSMIT SPEED, DEFAULT= 9600 BAUD (1101)
BIT 14 HALF DUPLEX (1), DEFAULT= FULL DUPLEX (0)

****DEFAULT PARAM#1=33523**********

```
                                ;BASIC DEFINITIONS
                                ;**************
                                ;INITIAL ADDRESS OF THE STACK POINTER
001070                          STACK=  1070

                                ;****************
                                .EQUIV  EMT,HLT                 ;BASIC DEFINITION OF ERROR CALL


                                ;REGISTER DEFINITION
000000                          R0=     %0                      ;GENERAL REGISTER
000001                          R1=     %1                      ;GENERAL REGISTER
000002                          R2=     %2                      ;GENERAL REGISTER
000003                          R3=     %3                      ;GENERAL REGISTER
000004                          R4=     %4                      ;GENERAL REGISTER
000005                          R5=     %5                      ;GENERAL REGISTER
000006                          R6=     %6                      ;GENERAL REGISTER
000007                          R7=     %7                      ;GENERAL REGISTER
000000                          MODE=   %0
000006                          R6=SP
000007                          R7=PC


                                ;SWITCH DEFINITION
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0
000000                          PRTY0=  0
000040                          PRTY1=  40
000100                          PRTY2=  100
000140                          PRTY3=  140
000200                          PRTY4=  200
000240                          PRTY5=  240
```

```
                                PRTY6=  300
                                PRTY7=  340

                                ;MISCELLANEOUS BIT ASSIGNMENT
                                BIT15=  100000
                                BIT14=  40000
                                BIT13=  20000
                                BIT12=  10000
                                BIT11=  4000
                                BIT10=  2000
                                BIT09=  1000
                                BIT08=  400
                                BIT07=  200
                                BIT06=  100
                                BIT05=  40
                                BIT04=  20
                                BIT03=  10
                                BIT02=  4
                                BIT01=  2
                                BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

000004                          RCSR=R4
000020                          LOOP=20
000200                          NODAT=200
000000                          XCSR=0
000002                          NRCR=2
000004                          LPR=4
000006                          CAR=6
000010                          BCR=10
000012                          BAR=12
000016                          SSR=16
000100                          RIE=BIT6
040000                          HALF.DUPLEX=BIT14
040000                          ESC=BIT14
004000                          MC=BIT11
010000                          SIE=BIT12
040000                          SI=BIT14
020000                          TIE=BIT13
100000                          TI=BIT15
002000                          NEM=BIT10
040000                          DO=BIT14
020000                          FE=BIT13
010000                          PE=BIT12
000000                          STAT=R0
000010                          FSTERR=BIT3
100000                          XFLG=100000             ;XMIT COMPLETE FLAG
```

```
                    040000                  .       RFLG=40000                      ;RCV COMPLETE FLAG
                    020000                          DSFLG=20000                     ;DATA SET STATUS CHANGE FLAG
                    020000                          BIT13=20000                     ;INHIBIT PRINTOUTS
                                            ;VECTOR ADDRESSES
                    000004                  ERRVEC= 4
                    000010                  RESVEC= 10
                    000014                  TBITVEC=14
                    000014                  TRTVEC= 14
                    000014                  BPTVEC= 14
                    000020                  IOTVEC= 20
                    000024                  PWRVEC= 24
                    000030                  EMTVEC= 30
                    000034                  TRAPVEC=34
                                            .EQUIV  R4,CSR
                                            .EQUIV  R4,RCSR

                    000000                          .=0
                                            ;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
                                            ;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS
                    000024                          .=24
000024  006750                                      $PWRDN
000026  000340                                      340
000030  005600                                      $HLT
000032  000340                                      340
000034  006710                                      $TRAP
000036  000340                                      340

                    000100                          .=100
000100  006456                                      TIMER
000102  000340                                      340

                    000174                          .=174
000174  000000                  DISPREG:0
000176  000000                  SWREG:  0

                    000200                          .=200

000200  000137  004466                              JMP     @#BEGIN         ;JUMP TO STARTING ADDRESS OF PROGRAM
000204  000137  004632                              JMP     @#RSTART        ;RESTART AT 204. DO THE RESTART.
```

```
                                ;*******************************************************************************
            001100                      .=1100

                                ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;NOTE1: NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.

                                ;CALL:
                                ;1) USING A TRAP INSTRUCTION
                                ;       TYPE    ,MESADR         ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;OR
                                ;       TYPE
                                ;       MESADR

001100  010046                  $TYPE:  MOV     R0,-(SP)        ;SAVE R0
001102  010146                          MOV     R1,-(SP)        ;SAVE R1
001104  017600  000004                  MOV     @4(SP),R0       ;GET ADDRESS OF ASCIZ STRING
001110  062766  000002  000004          ADD     #2,4(SP)        ;ADJUST RETURN PC
001116  012701  000101          7$:     MOV     #101,R1         ;SET CHARACTER COUNT
001122  005301                  1$:     DEC     R1
001124  001432                          BEQ     6$
001126  112046                          MOVB    (R0)+,-(SP)     ;PUSH CHARACTER TO BE TYPED ONTO STACK
001130  001004                          BNE     2$              ;BR IF IT ISN'T THE TERMINATOR
001132  005726                          TST     (SP)+           ;IF TERMINATOR POP IT OFF THE STACK
001134  012601                          MOV     (SP)+,R1        ;RESTORE R1
001136  012600                          MOV     (SP)+,R0        ;RESTORE R0
001140  000002                          RTI                     ;RETURN
001142  004737  001174          2$:     JSR     PC,5$           ;GO TYPE THIS CHARACTER
001146  122726  000012          3$:     CMPB    #12,(SP)+       ;CHECK IF THE CHAR. TYPED WAS A  LINE FEED
001152  001363                          BNE     1$              ;GO GET NEXT CHAR. IF NOT LINE FEED
001154  013746  001574                  MOV     NULL,-(SP)      ;GET # OF FILLER CHARS. NEEDED
                                                                ;AND THE NULL CHAR.
001160  105366  000001          4$:     DECB    1(SP)           ;DOES A NULL NEED TO BE TYPED?
001164  002770                          BLT     3$              ;BR IF NO--GO POP THE NULL OFF OF STACK
001166  004737  001174                  JSR     PC,5$           ;GO TYPE A NULL
001172  000772                          BR      4$              ;LOOP
001174  105777  000614          5$:     TSTB    @TPS            ;WAIT UNTIL PRINTER IS READY
001200  100375                          BPL     5$
001202  116677  000002  000606          MOVB    2(SP),@TPB      ;LOAD CHAR TO BE TYPED INTO DATA REG.
001210  000207                          RTS     PC
001212  104400  010266          6$:     TYPE,CRLF               ;TYPE <CR><LF>
001216  000737                          BR      7$              ;NEXT LINE
```

```
                                        ;*****************************************************************
                                        ;       DEBUG DUMP ROUTINE
                                        ;*****************************************************************
001220  013746  001760          DMPHLT: MOV     FLAG,-(SP)
001224  042737  000070  001760          BIC     #70,FLAG        ;INIT FLAGS
001232  104400  010262                  TYPE    ,ASTRK          ;TYPE *
001236  104402                          GETCHR
001240  012637  001342                  MOV     (SP)+,6$
001244  104400  001342                  TYPE    ,6$
001250  042737  000040  001544          BIC     #BIT5,$CHAR     ;STRIP LOWER CASE OFF
001256  122737  000127  001544          CMPB    #127,$CHAR      ;W? FOR WORD
001264  001004                          BNE     1$
001266  052737  000010  001760          BIS     #BIT3,FLAG      ;SET FLAG BIT
001274  000430                          BR      3$
001276  122737  000102  001544  1$:     CMPB    #102,$CHAR      ;B? FOR BYTE
001304  001004                          BNE     2$
001306  052737  000020  001760          BIS     #BIT4,FLAG
001314  000420                          BR      3$
001316  122737  000103  001544  2$:     CMPB    #103,$CHAR      ;C? FOR CONTINUE
001324  001014                          BNE     3$
001326  012637  001760                  MOV     (SP)+,FLAG
001332  052737  000040  001760          BIS     #BIT5,FLAG
001340  000413                          BR      DUMP
001342  000000                  6$:     000000
001344  104400  010256          4$:     TYPE    ,QUES
001350  104400  010266                  TYPE    ,CRLF
001354  000721                          BR      DMPHLT
001356  005037  001546          3$:     CLR     WORK
001362  005726                          TST     (SP)+
001364  104406  001546                  ACCEPT  ,WORK
001370  012700  001524          DUMP:   MOV     #DMPLST,R0      ;INIT DUMP LIST
001374  062710  000020                  ADD     #20,    (R0)    ;BUMP ADDRESS
001400  032737  000040  001760          BIT     #BIT5,FLAG
001406  001005                          BNE     L1
001410  013737  001546  001524          MOV     WORK,DMPLST
001416  001001                          BNE     L1
001420  022020                          CMP     (R0)+,  (R0)+   ;SKIP 1ST TWO ENTRIES

001422  012001                  L1:     MOV     (R0)+,  R1      ;GET ADDR OF DATA FROM LIST
001424  001675                          BEQ     DMPHLT          ;BR IF END OF LIST
001426  104400  010266                  TYPE    ,CRLF
001432  010146                          MOV     R1,     -(SP)   ;PUSH ADDR ON STACK
001434  004037  006514                  JSR     R0,$B2016       ;PRINT OUT ADDRESS

001440  032737  000010  001760          BIT     #BIT3,FLAG
001446  001014                          BNE     L3
001450  012702  000020                  MOV     #20,    R2      ;SET WORD COUNTER = 8
001454  005046                  L2:     CLR     -(SP)
001456  112116                          MOVB    (R1)+,(SP)
001460  104400  007500                  TYPE    ,MSG00
001464  004037  006502                  JSR     R0,$B20CT
001470     003                          .BYTE   3
001471     001                          .BYTE   1
001472  005302                          DEC     R2              ;DECREMENT WORD COUNTER
001474  001367                          BNE     L2              ;BR IF NOT = 0
001476  000751                          BR      L1              ;GET NEXT ENTRY
```

```
451
452  001500  012702  000010          L3:     MOV     #10,R2
453  001504  012146                  1$:     MOV     (R1)+,-(SP)
454  001506  104400  007500                  TYPE    ,MSG00
455  001512  004037  006514                  JSR     R0,$82016
456  001516  005302                          DEC     R2              ;DECREMENT THE WORD COUNT
457  001520  001371                          BNE     1$
458  001522  000737                          BR      L1              ;GET NEXT ENTRY
459  001524  000000                  DMPLST: 0                       ;RESERVED FOR SW. REG
460  001526  000000                          0                       ;END OF TABLE FOR SW. REG
461  001530  000001                  .RX:    .BLKW   1
462  001532  000001                  .TX:    .BLKW   1
463  001534  000000                          0
464  001536  000000                          0
465  001540  000000                          0
```

```
                                ;*****************************************************************
                                ;       CONSTANTS AND WORKING STORAGE
                                ;*****************************************************************
                                        $WORK=WORK
001546                  $LPADR: 0
001542  000000         $CHAR:  0
001544  000000         WORK:   0
001546  000000         WORK1:  0
001550  000000         WORK2:  0
001552  000000         WORK3:  0
001554  000000         WORK4:  0
001556  000000         WORK5:  0
001560  000000         WORK6:  0
001562  000000         LINE:   .WORD 0
001564  000000         LINMSK: .WORD 0
001566  000000         FLAGS:  0
001570  000000         STYME:  .WORD 10
001572  000010         NULL:   .WORD   6               ;CONTAINS NULL CHARACTER FOR FILLS
001574  000006         $PASS:  0
001576  000000         $ICNT:  .BYTE   0
001600     000         $TIMES: .BYTE   10.
001601     012         $HLTAD: 0
001602  000000         $ERTTL: 0
001604  000000         .BAUD:  .WORD   0               ;TIME DELAY FOR 0 BAUD
001606  000000                 .WORD   80.             ;TIME DELAY FOR 50 BAUD
001610  000120                 .WORD   55.             ;TIME DELAY FOR 75 BAUD
001612  000067                 .WORD   39.             ;TIME DELAY FOR 110 BAUD
001614  000047                 .WORD   31.             ;TIME DELAY FOR 134.5 BAUD
001616  000037                 .WORD   28.             ;TIME DELAY FOR 150 BAUD
001620  000034                 .WORD   23.             ;TIME DELAY FOR 200 BAUD
001622  000027                 .WORD   17.             ;TIME DELAY FOR 300 BAUD
001624  000021                 .WORD   9.              ;TIME DELAY FOR 600 BAUD
001626  000011                 .WORD   6.              ;TIME DELAY FOR 1200 BAUD
001630  000006                 .WORD   5.              ;TIME DELAY FOR 1800 BAUD
001632  000005                 .WORD   3.              ;TIME DELAY FOR 2400 BAUD
001634  000003                 .WORD   3.              ;TIME DELAY FOR 4800 BAUD
001636  000003                 .WORD   3.              ;TIME DELAY FOR 9600 BAUD
001640  000003
001642  022333         L:      22333
001644  123456         H:      123456
001646  000000  000000 COUNT:  .WORD   0,0
001652  000000         QTPIE:  000000
001654  000000         SAVR0:  0
001656  000000         SAVR1:  0
001660  000000         SAVR2:  0
001662  000000         SAVR3:  0
001664  000000         SAVR4:  0
001666  000000         SAVR5:  0
001670  000000         DELAY:  0
001672  000000         BACK:   0
001674  000000         STOP:   0
001676  000000         $ERROR: 0
001700  000000         TEMP1:  0
001702  000000         TEMP2:  0
001704  000000         BAKTMP: 0
001706  000000         SNCFLG: 0
001710  000000         TRNFLG: 0
```

```
                                FILL:   .BYTE   177
                                .EVEN
                                VISR:
                                ISR:    .ASCIZ  /DH /           ;ISR NAME
                                BA:     160020                  ;BUS ADDRESS
                                VA:     300                     ;VECTOR ADDRESS
                                PRIOR:  240                     ;PRIORITY
                                PARAM1: 0                       ;PARAM #1
                                PARAM2: 033523                  ;PARAM #2
                                PARAM3: 177777                  ;PARAM #3
                                IRDA:   .WORD   0               ;INITIAL READ DATA ADDRESS
                                IXDA:   .WORD   0               ;INITIAL XMIT DATA ADDRESS
                                SETTLE: .WORD   0               ;LINE SETTLE DELAY FLAG
                                IRCC:   .WORD   0               ;
                                B2016:  .WORD   0               ;ADDR OF BIN TO OCT TYPE ROUTINE
                                TIME:   .WORD   0               ;TIMER
                                MODEA:  .WORD   0               ;
                                        .WORD   START           ;ADDR OF START OF PROGRAM
                                TX.TERM:
                                        .BYTE   000             ;TRANSMITTER TERMINATING CHAR.
                                RX.TERM:
                                        .BYTE   001             ;RECEIVER TERMINATING CHAR.
                                ESC.TERM:
                                        .BYTE   115             ;ESCAPE SEQ TERMINATOR
                                FLAG:   .WORD   0
                                SWR:    177570
                                DISPLAY:177570
                                SXCSR:  0                       ;SAVED XMIT CSR
                                SRCSR:  0                       ;SAVED RCV CSR
                                ERCSR:  0                       ;RCV CSR SAVED ON ERROR
                                ERDBR:  0                       ;RCV DATA REG SAVED ON ERROR
                                DSSTAT: 0                       ;RCV CSR SAVED ON DS CHANGE
                                XCC:    0                       ;XMIT CHAR COUNT
                                RCC:    0                       ;RCV CHAR COUNT
                                RDA:    0                       ;RCV DATA ADDR.
                                XDA:    0                       ;XMIT DATA ADDR.
                                TKS:    177560
                                TKB:    177562
                                TPS:    177564
                                TPB:    177566
                                ;DATA PATTERN ADDRESS TABLE AND PATTERNS
                                DAT:    .WORD   DP1             ;ADDRESS OF VARIABLE DATA BUFFER
                                        .WORD   VDB             ;ADDRESS OF MESSAGE 1

                                ;RECEIVER DATA BUFFER STARTS HERE..
                                IBUF:   .BLKW   400
                                VDB:
                                        .BLKW   100             ;VARIABLE DATA BUFFER

                                DP1:    .BLKW   400
                                LPB:    .BYTE   002,033,117,115,000
                                        .EVEN
```

```
(1)                                     ;*****************************************************************
(1)                                     ;       TTY INTERUPT ROUTINE
(1)                                     ;*****************************************************************
004232  017701  175554          TTYINT: MOV     @TKB,R1         ;CLEAR TTY BUFFER
004236  042777  000100  175544          BIC     #100,@TKS       ;RESET INT. ENABLE
004244  042701  177700                  BIC     #177700,R1      ;STRIP JUNK
004250  022701  000007                  CMP     #7,R1
004254  001013                          BNE     2$
004256  022737  000176  001762          CMP     #SWREG,SWR
004264  001003                          BNE     1$
004266  052737  000001  001760          BIS     #BIT0,FLAG
004274  052777  000100  175506  1$:     BIS     #100,@TKS
004302  000002                          RTI
004304  022701  000077          2$:     CMP     #'?,    R1      ;IS IT ?
004310  001017                          BNE     3$              ;BR IF NO
004312  012700  001524                  MOV     #DMPLST,R0      ;SETUP DUMP LIST
004316  012710  002024                  MOV     #IBUF,  (R0)    ;TO PRINTOUT INPUT BUFFER
004322  017737  175406  001530          MOV     @IRDA,.RX       ;IF SWITCH REG. =0 PRINT RX BUFFER.
004330  017737  175402  001532          MOV     @IXDA,.TX       ;" " " "           PRINT TX BUFFER
004336  052777  000100  175444          BIS     #100,@TKS
004344  000137  001422                  JMP     L1              ;AND GO PRINT IT
004350  122701  000003          3$:     CMPB    #003,R1         ;WAS IT ^C
004354  001347                          BNE     1$
004356  104410  000140          .CTLC:  STPS,PRTY3              ;SET PSW TO 140
004362  005037  001564                  CLR     LINE            ;INIT LINE #
004366  042737  000100  001760          BIC     #BIT6,FLAG      ;INIT ALL-LINES-DONE FLAG
004374  042700  100000          1$:     BIC     #XFLG,STAT      ;INIT XMIT DONE FLAG
004400  005227  000000          4$:     INC     #0
004404  001375                          BNE     4$
004406  004737  012560                  JSR     PC,NEEXT        ;GET LINE #
004412  032737  000100  001760          BIT     #BIT6,FLAG      ;DONE ALL LINES?
004420  001021                          BNE     3$              ;EXIT IF TRUE
004422  012705  000001                  MOV     #1,R5           ;SET UP TO
004426  012764  177777  000010          MOV     #-1,BCR(R4)     ;XMIT ONE CHAR
004434  012764  011040  000006          MOV     #CTLC,CAR(R4)   ;WHERE IT IS
004442  013764  001704  000012          MOV     BARTMP,BAR(R4)  ;SET BAR BIT
004450  052714  020000                  BIS     #TIE,@RCSR      ;SET INT-ENABLE
004454  032700  100000          2$:     BIT     #XFLG,STAT      ;SENT YET
004460  001775                          BEQ     2$              ;NO WAIT
004462  000744                          BR      1$              ;NEXT
004464  005726                  3$:     TST     (SP)+           ;ADJUST AND EXIT
```

```
                                        ;*****************************************************************
                                        ;       START OF PROGRAM
                                        ;*****************************************************************
004466                                  BEGIN:
004466  012706  001070                        MOV     #STACK,SP       ;SETUP THE STACK POINTER
004472  104410  000340                        STPS,PRTY7              ;SET PSW TO LEVEL 7
004476  104414                               SUSWR
004500  005037  001564                        CLR     LINE            ;ALWAYS START AT LINE 0
004504  005037  001576                        CLR     $PASS           ;CLEAR THE PASS COUNT
004510  012737  004466  001542                MOV     #BEGIN,$LPADR   ;INITILIZE THE LOOP ADDRESS FOR SCOPE
004516  005037  001570                        CLR     FLAGS           ;RESET FLAGS
004522  042737  177773  001760                BIC     #^C<BIT2>,FLAG  ;CLEAR ALL BUT TITLE BITS
004530  000005                               RESET
                                        ;*****************************************************************
                                        ;       ISR PARAM INPUT ROUTINE
                                        ;*****************************************************************
004532  012700  001714                GETIT:  MOV     #VISR,  R0      ;PRESET ISR ADDR IN R0
004536  032737  000004  001760                BIT     #BIT2,FLAG      ;FIRST TIME THRU
004544  001005                               BNE     GETBA           ;NO
004546  104400  007502                        TYPE    ,MSG01          ;TYPE ID
004552  052737  000004  001760                BIS     #BIT2,FLAG      ;SET TITLE FLAG
004560  010004                        GETBA:  MOV     R0,     R4      ;SAVE POINTER
004562  022020                               CMP     (R0)+,  (R0)+   ;INCREMENT ISR POINTER
004564  104400  007571                        TYPE    ,MSG03          ;<15><12> BUS ADDRESS=
004570  004737  005550                        JSR     PC,GETANY       ;GET THE BUS ADDR

004574  104400  007610                GETVA:  TYPE    ,MSG04          ;<15><12> VECTOR ADDRESS=
004600  004737  005550                        JSR     PC,GETANY       ;GET THE VECTOR ADDR.

004604  104400  007632                GETPRI: TYPE    ,MSG05          ;<15><12> PRIORITY=
004610  004737  005550                        JSR     PC,GETANY       ;GET THE PRIORITY

004614  004737  011704                        JSR     PC,GETPRM       ;GET LINE #'S TO TEST

004620  104400  007670                        TYPE    ,MSG07          ;<15><12> PARAMS #2=
004624  004737  005550                        JSR     PC,GETANY       ;GET PARAM
004630  000410                               BR      SWPRNT          ;CONTINUE




004632  104400  010253                RSTART: TYPE    ,MFILL
004636  000005                               RESET
004640  105227  000000                        INCB    #0              ;DELAY HERE FOR AWILE
004644  001375                               BNE     .-4
004646  005037  001564                        CLR     LINE
004652  022737  000176  001762        SWPRNT: CMP     #SWREG,SWR
004660  001007                               BNE     XSWPNT
004662  052737  000002  001760                BIS     #BIT1,FLAG
004670  104400  007724                        TYPE    ,MSG09
004674  104416                               SETSWI
004676  000417                               BR      REST
004700  104400  007724                XSWPNT: TYPE    ,MSG09          ;<15><12> SET SWITCHES
```

```
                                ;*****************************************************************
                                ;       SET SWITCH OPTIONS
                                ;*****************************************************************

004734  000000                  SWRSET: HALT            ;HALT FOR SWITCH SETUP

004736  012737  004744  001542  REST:   MOV     #RESTRT,$LPADR  ;SETUP LOOP
004744  017701  175012          RESTRT: MOV     @SWR,R1
004750  000301                          SWAB    R1
004752  042701  177775                  BIC     #177775, R1     ;STRIP JUNK
004756  016137  002020  001736          MOV     DAT(1), IXDA    ;SETUP INIT DATA ADDR FROM TABLE
004764  005701                          TST     R1              ;VARIABLE DATA SPECIFIED?
004766  001404                          BEQ     SUXCC           ;BR IF NO
                                ;*****************************************************************
                                ;       GET VARIABLE DATA
                                ;*****************************************************************
004770  104400  007746                  TYPE    ,MSG10          ;(15)(12) ENTER DATA (15)(12)
004774  104404                          GETSTR                  ;GET DATA
004776  003024                          VDB                     ;AND PUT IT HERE
005000  012737  002024  001734  SUXCC:  MOV     #IBUF,  IRDA    ;SETUP READ BUFFER ADDR
005006  004737  007230                  JSR     PC,SETBUF       ;GET BYTE COUNTS
005012  004737  006064                  JSR     PC,$BAUD        ;CALCULATE TIME DELAY

005016                          SWRNXT:
005016  012737  004232  000060          MOV     #TTYINT,@#60    ;SETUP TTY VECTOR
005024  012737  000340  000062          MOV     #340,   @#62    ;
005032  012777  000100  174750          MOV     #100,   @TKS    ;AND ENABLE INTERRUPTS
005040  012702  002024                  MOV     #IBUF,  R2
005044  005022                  CLRIB:  CLR     (R2)+           ;CLEAR INPUT BUFFER
005046  022702  003024                  CMP     #VDB,R2
005052  001374                          BNE     CLRIB
```

```
                                        ;****************************
                                        ;       SETUP TIMER        *
                                        ;****************************

005054  012737  000060  006500  SUTIME: MOV     #60,    MSECS   ;PRESET COUNTER
005062  012737  006456  000100          MOV     #TIMER, 100     ;SETUP LINE CLOCK VECTOR
005070  012737  000340  000102          MOV     #340,   102     ;AND PRIORITY
005076  012737  005122  000004          MOV     #NOLC,  4       ;SETUP BUS ERROR VECTOR
005104  012737  000340  000006          MOV     #340,   6       ;SET PRIORITY
005112  052737  000100  177546          BIS     #100,   177546  ;ENABLE LINE CLOCK
005120  000441                          BR      NORTC

                                        ;BUS ERROR RETURNS HERE IF NO LINE CLOCK

005122  012737  006456  000104  NOLC:   MOV     #TIMER, 104     ;SETUP RTC VECTOR
005130  012737  000340  000106          MOV     #340,   106     ;AND PRIORITY
005136  012737  005172  000004          MOV     #1$,    4       ;SETUP BUS ERROR VECTOR
005144  012737  000340  000006          MOV     #340,   6       ;SET PRIORITY
005152  012737  003100  172542          MOV     #1600.,  172542 ;SET COUNTER BUFFER.
005160  012737  000111  172540          MOV     #111,   172540 ;ENABLE REAL TIME TIME CLOCK
005166  000240                          NOP                     ;WAIT
005170  000415                          BR      NORTC           ;CONTINUE.
005172  012737  005216  000100  1$:     MOV     #2$,100         ;SET UP FOR LSI
005200  104410  000000                  STPS,PRTY0              ;LOWER PSW TO 0
005204  005227  000000          3$:     INC     #0
005210  001375                          BNE     3$
005212  104400  010055                  TYPE,MSG19
005216  012737  006456  000100  2$:     MOV     #TIMER,100

005224                          NORTC:
005224  012737  000006  000004          MOV     #6,@#4          ;SET TRAP VECTOR
005232  005037  000006                  CLR     @#6             ;SET BUS ERROR VECTOR
005236  012706  001070                  MOV     #STACK, SP      ;SETUP STACK
005242  104410  000000                  STPS,PRTY0
005246  012737  006514  001744          MOV     #$B2016,B2016   ;SETUP BIN TO OCT ADDR


                                        ;****************************************
                                        ;       GOTO THE MODULE AND RUN         *
                                        ;****************************************

005254  004737  011042          VIGO:   JSR     PC,START        ;GO TO ISR
```

```
                                        ;*****************************************************************
                                        ;       END OF PASS ROUTINE
                                        ;*****************************************************************
005260  005746                  EOPPY:  TST     -(SP)           ;SAVE RETURN
005262  005737  001576                  TST     $PASS           ;FIRST PASS THRU
005266  001406                          BEQ     1$              ;YESS NO ITERATIONS
005270  105237  001600                  INCB    $ICNT           ;INCREMENT ITERATION COUNT
005274  123737  001601  001600          CMPB    $TIMES,$ICNT    ;DONE ENOUGH YET
005302  002102                          BGE     8$              ;NO DO SOME MORE
005304  112737  000001  001600  1$:     MOVB    #1,$ICNT        ;INIT ITERATION COUNTER
005312  004737  012560                  JSR     PC,NEEXT        ;GO CALCULATE BAR REG
005316  032737  000100  001760          BIT     #BIT6,FLAG      ;THRU ALL LINES YET
005324  001453                          BEQ     7$              ;NO
005326  042737  040100  001760          BIC     #BIT6+ESC,FLAG  ;INIT FLAG AND TYPE END PASS
005334  104412                          KBDIN
005336  005737  001576                  TST     $PASS           ;FIRST PASS
005342  001442                          BEQ     5$
005344  032777  010000  174410          BIT     #SW12,@SWR      ;INHIBIT TYPEOUTS?
005352  001036                          BNE     5$              ;BR IF YES
005354  104400  007765                  TYPE    ,MSG11          ;(15)(12) END OF PASS
005360  013746  001576                  MOV     $PASS,  -(SP)
005364  004037  006514                  JSR     R0,$B2016       ;PRINTOUT PASS COUNT
005370  104400  010271                  TYPE    ,TUCRLF         ;COUPLE FORMS
005374  032777  000004  174360          BIT     #SW02,@SWR      ;TYPE OUT XMIT +RCV DATA
005402  001416                          BEQ     6$              ;NO BRANCH
005404  104400  010156                  TYPE    ,MSG26          ;TRANSMITTED DATA=
005410  013737  001736  005420          MOV     IXDA,2$         ;SET POINTER TO TXBUF
005416  104400                          TYPE                    ;TYPE TXBUFFER
005420  000000                  2$:     0
005422  104400  010204          3$:     TYPE    ,MSG27          ;RECEIVED DATA=
005426  013737  001734  005436          MOV     IRDA,4$         ;SET POINTER TO RXBUF
005434  104400                          TYPE                    ;TYPE RXBUFFER
005436  000000                  4$:     0
005440  032777  040000  174314  6$:     BIT     #BIT14,@SWR     ;ONE PASS ONLY
005446  001025                          BNE     9$
005450  005237  001576          5$:     INC     $PASS           ;UP PASS COUNT
005454  005737  001576          7$:     TST     $PASS
005460  001413                          BEQ     8$
005462  032777  000020  174272          BIT     #LOOP,@SWR      ;TYPE UNIT UNDER TEST
005470  001407                          BEQ     8$              ;NOP
005472  104400  011014                  TYPE    ,TESTXX
005476  113746  001564                  MOVB    LINE,-(SP)
005502  004037  006502                  JSR     R0,$B2OCT       ;TYPE UNIT #
005506     002                          .BYTE   2
005507     001                          .BYTE   1
005510  016600  000002          8$:     MOV     2(SP),R0        ;GET RETURN ADDRESS
005514  104410  000000                  STPS,PRTY0
005520  000110                          JMP     (R0)            ;GO BACK TO MODULE.
005522  012706  001070          9$:     MOV     #STACK,SP       ;RESET THE STACK POINTER.
005526  013700  000042                  MOV     @#42,   R0      ;GET MONITOR ADDRESS
005532  001404                          BEQ     $DOAGN          ;BR IF NONE
005534  004710                          JSR     PC,(R0)         ;GO TO MONITOR
005536  000240                          NOP                     ;SAVE ROOM FOR
005540  000240                          NOP                     ;ACT-11
005542  000240                          NOP
005544  000137  004356          $DOAGN: JMP     .CTLC           ;RESTART TEST
```

```
                                ;*****************************************************************
                                ;       SUBROUTINE TO INPUT OCTAL WORD FROM OPERATOR
                                ;*****************************************************************
005550  011046                  GETANY: MOV     (R0),   -(SP)   ;PUT WORD ON STACK
005552  004037  006514                  JSR     R0,$B2016       ;AND TYPE IT
005556  104400  010256                  TYPE    ,QUES
005562  011037  001546                  MOV     (R0),WORK       ;PRESET FOR DEFAULT (CR)
005566  104406  001546          ANYMOR: ACCEPT  ,WORK           ;OCTAL READIN
005572  013720  001546                  MOV     WORK,   (R0)+   ;MOVE IT TO ISR
005576  000207                          RTS     PC              ;SUB/ROUTINE EXIT

                                ;*****************************************************************
                                ;       ERROR HLT HANDLER
                                ;*****************************************************************
005600                          $HLT:
005600  000240                          NOP
005602  000240                          NOP
005604  005237  001604          $HLOT:  INC     $ERTTL          ;INCREMENT ERROR COUNTER
005610  001775                          BEQ     $HLOT           ;MAKE SURE ITS NOT ZERO
005612  011637  001602                  MOV     (SP)    ,$HLTAD ;SAVE ADDRESS OF HLT
005616  162737  000002  001602          SUB     #2,     $HLTAD  ;AND BACK IT UP
005624  010146                          MOV     R1,     -(SP)   ;SAVE R1

005626  032777  020000  174126          BIT     #BIT13, @SWR    ;INHIBIT ERR TYPEOUTS?
005634  001075                          BNE     TRX             ;BR IF YES

005636  104400  010266                  TYPE    ,CRLF
005642  117701  173734                  MOVB    @$HLTAD,R1      ;EXTRACT HLT CODE
005646  006301                          ASL     R1              ;AND ALIGN IT
005650  016137  006050  005660          MOV     EMTAB(R1),.+10  ;GET HEADER ADDRESS
005656  104400  010275                  TYPE    ,EMO            ;AND PRINT HEADER
005662  113746  001564                  MOVB    LINE,-(SP)      ;PUT LINE # ON STACK
005666  004037  006502                  JSR     R0,$B2OCT       ;TYPE IT
005672     002                          .BYTE   2
005673     001                          .BYTE   1
005674  104400  010004                  TYPE    ,MSG12          ; < AT LOC >
005700  013746  001602                  MOV     $HLTAD,-(SP)            ;GET HLT ADDRESS
005704  004037  006514                  JSR     R0,$B2016       ;AND PRINT IT
005710  005701                          TST     R1              ;HLT CODE = 0?
005712  001446                          BEQ     TRX             ;BR IF YES

005714  022701  000006                  CMP     #6,R1           ;IS IT HLT+3?
005720  001023                          BNE     1$              ;BR IF NO
005722  005702                          TST     R2              ;PRINTOUT BAD DATA?
005724  001406                          BEQ     2$              ;BR IF NO
005726  104400  010124                  TYPE    ,MSG23          ;< BAD DATA= >
005732  110246                          MOVB    R2,-(SP)        ;GET DATA
005734  004037  006502                  JSR     R0,$B2OCT       ;AND PRINT IT
005740     003                          .BYTE   3
005741     001                          .BYTE   1
005742  005703                  2$:     TST     R3              ;PRINT OUT GOOD DATA?
005744  001410                          BEQ     3$              ;BR IF NO
005746  104400  010140                  TYPE    ,MSG24          ;< GOOD DATA= >
```

```
                                                MOVB    R3,-(SP)        ;GET DATA
                                                JSR     R0,$B2OCT       ;AND PRINT IT
                                                .BYTE   3
                                                .BYTE   1
                                                TYPE    ,CRLF
                                        3$:     BR      TRX

                                        1$:     TST     R2              ;PRINTOUT RCV CSR?
                                                BEQ     TR3             ;BR IF NO
                                                TYPE    ,MSG13          ; < RCV CSR=>
                                                MOV     R2,     -(SP)   ;GET DATA
                                                JSR     R0,$B2016       ;AND PRINT IT

                                        TR3:    TST     R3              ;PRINTOUT XMIT CSR?
                                                BEQ     TRX             ;BR IF NO
                                                TYPE    ,MSG14          ; < XMIT CSR=>
                                                MOV     R3,     -(SP)   ;GET DATA
                                                JSR     R0,$B2016       ;AND PRINT IT
                                                TYPE    ,CRLF

                                        TRX:    BIT     #BIT15, @SWR    ;HALT ON ERROR?
                                                BEQ     HLTX            ;BR IF NO
                                                HALT

                                        HLTX:   KBDIN
                                                MOV     (SP)+,  R1      ;RESTORE R1
                                                RTI                     ;AND RETURN TO PROGRAM

                                        EMTAB:  EM0
                                                EM1
                                                EM2
                                                EM7
                                                EM6
                                                EM10


                                        ;***************************************************************
                                        ;       CALCULATE TIMER DELAY
                                        ;***************************************************************

                                        $BAUD:  MOV     PARAM2,R2               ;GET PARAM2
                                                BIC     #176077,R2              ;STRIP OFF ALL BUT SPEED
                                                ROL     R2
                                                ROL     R2
                                                ROL     R2
                                                SWAB    R2
                                                MOV     .BAUD(R2),$TYME         ;SET TIMER DELAY
                                                RTS     PC                      ;EXIT
```

```
                                        ;*****************************************************************
                                        ;       READ A CHAR. ROUTINE
                                        ;*****************************************************************

                                        ;       CALL=   GETCHR  ;INPUT A CHAR FROM TTY
                                        ;                               ;RETURNS HERE WITH CHAR ON STACK
006114  011646                  $READC: MOV     (SP),-(SP)      ;PUSH THE PC
006116  016666  000004  000002          MOV     4(SP),2(SP)     ;SAVE THE PS
006124  105777  173660                  TSTB    @TKS            ;IS RECEIVE DONE
006130  100375                          BPL     .-4             ;LOOP IF NO
006132  017737  173654  001544          MOV     @TKB    ,$CHAR  ;SAVE THE CHAR.
006140  042737  177600  001544          BIC     #177600,$CHAR   ;&STRIP JUNK
006146  013766  001544  000004          MOV     $CHAR,4(SP)     ;PUT CHAR ON STACK
006154  000002                          RTI                     ;EXIT
                                        ;*****************************************************************
                                        ;       READ A STRING ROUTINE
                                        ;*****************************************************************

                                        ;       CALL=   GETSTR          ;INPUT A STRING OF CHARS FROM TTY
                                        ;               ADDR    ;TO THIS ADDRESS
                                        ;                               ;TERMINATE INPUT WITH LINE FEED
006156  011602                  $READS: MOV     (SP),   R2      ;SETUP ADDRESS OF INPUT BUFFER
006160  012201                          MOV     (R2)+,  R1      ;INCREMENT RETURN ADDRESS
006162  010216                          MOV     R2,     (SP)    ;AND PUT BACK ON STACK
                                        ;
006164  104402                  GETIC:  GETCHR                  ;GET A CHAR
006166  104400  001544                  TYPE    ,$CHAR
006172  122726  000136                  CMPB    #136,   (SP)+   ;IS IT BINARY DELIMITER
006176  001011                          BNE     GOTIC           ;BR IF NO
                                        ;
006200  104406  001546          OCT:    ACCEPT  ,$WORK          ;GET OCT. CHAR
006204  113721  001546                  MOVB    $WORK,(R1)+     ;STORE OCT. CHAR.
006210  122737  000136  001544          CMPB    #136,$CHAR      ;TERMINATOR=BIN. DELIMITER?
006216  001370                          BNE     OCT             ;BR IF NO
006220  000761                          BR      GETIC           ;
006222                          GOTIC:
006222  113721  001544                  MOVB    $CHAR,  (R1)+   ;STORE CHAR. IN BUFFER
006226  022737  000015  001544          CMP     #15,    $CHAR   ;IS IT END OF INPUT (CAR. RETURN)
006234  001353                          BNE     GETIC           ;BR IF NO
006236  005737  001570                  TST     FLAGS
006242  100401                          BMI     1$
006244  105041                          CLRB    -(R1)
006246  113711  001544          1$:     MOVB    $CHAR,(R1)
006252  104400  010266                  TYPE    ,CRLF           ;TYPE A LINE FEED
006256  000002                          RTI
```

```
                                ;*************************************************
                                ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
                                ;CALL:
                                ;       ACCEPT  .ADDR           ;PUT OCTAL NUMBER IN ADDR
                                ;
006260                          $ACCEPT:
006260  010046                          MOV     R0,-(SP)        ;SAVE R0
006262  010146                          MOV     R1,-(SP)        ;SAVE R1
006264  010246                          MOV     R2,-(SP)        ;SAVE R2
006266  010346                          MOV     R3,-(SP)        ;SAVE R3
006270  016600  000010                  MOV     10(SP),R0               ;GET ADDRESS OF WHERE TO PUT NUMBER
006274  005001                  1$:     CLR     R1              ;CLEAR PARTIAL NUMBER
006276  012702  000006                  MOV     #6,R2           ;MAX. # OF DIGITS ALLOWED
006302  104402                  2$:     GETCHR                  ;GET ONE CHARACTER
006304  112603                          MOVB    (SP)+,R3        ;AND PUT IT IN R3
006306  110337  006454                  MOVB    R3,6$
006312  104400  006454                  TYPE    ,6$             ;ECHO THE CHARACTER
006316  022703  000025                  CMP     #25,R3
006322  001451                          BEQ     8$
006324  022703  000015                  CMP     #15,R3          ;WAS THIS CHARACTER A "CR"?
006330  001427                          BEQ     5$              ;BR IF YES
006332  022703  000040                  CMP     #40,R3          ;WAS "SPACE" HIT?
006336  001433                          BEQ     7$              ;BR IF YES
006340  022703  000136                  CMP     #136,R3         ;WAS " ↑ " HIT?
006344  001430                          BEQ     7$              ;BR IF YES
006346  032703  000110                  BIT     #110,R3         ;INSURE THE CHARACTER IS
006352  001011                          BNE     4$              ;A DIGIT BETWEEN 0 AND 7.
                                ;
006354  005302                          DEC     R2              ;CHECK NUMBER OF CHARACTERS
006356  002407                          BLT     4$              ;BR IF TO MANY
006360  006301                          ASL     R1              ;POSITION PARTIAL NUMBER
006362  006301                          ASL     R1              ;FOR THIS DIGIT
006364  006301                          ASL     R1
006366  042703  177770                  BIC     #^C(7),R3       ;GET RID OF THE ASCII JUNK
006372  050301                          BIS     R3,R1           ;COMBINE THIS DIGIT WITH PARTIAL
006374  000742                          BR      2$              ;GO GET ANOTHER DIGIT
006376  104400  010256          4$:     TYPE    ,QUES           ;TYPE "?"
006402  104400  010266                  TYPE    ,CRLF           ;TYPE CARRAGE RETURN AND LINE FEED.
006406  000732                          BR      1$              ;GO START OVER
006410  104400  010267          5$:     TYPE    ,LF             ;FOLLOW "CR" WITH A "LF"
006414  022702  000006                  CMP     #6,R2           ;WERE ANY DIGITS INPUT
006420  001002                          BNE     .+6             ;BR IF YES
006422  013001                          MOV     @(R0)+,R1       ;USE OLD DATA
006424  005740                          TST     -(R0)           ;BACKUP R0--
006426  010130                  7$:     MOV     R1,@(R0)+       ;PASS THE NUMBER TO THE USER
006430  010066  000010                  MOV     R0,10(SP)               ;SET FOR RETURN
006434  012603                          MOV     (SP)+,R3        ;RESTORE R3
006436  012602                          MOV     (SP)+,R2        ;RESTORE R2
006440  012601                          MOV     (SP)+,R1        ;RESTORE R1
006442  012600                          MOV     (SP)+,R0        ;RESTORE R0
006444  000002                          RTI
006446  104400  010246          8$:     TYPE,   CTLU
006452  000710                          BR      1$
006454  000     000             6$:     .BYTE   0,0             ;STORAGE FOR ASCII CHAR, AND TERMINATOR
```

```
                                ;*****************************************************************
                                ;       CLOCK INTERRUPT ROUTINE
                                ;*****************************************************************
006456  005337  006500        TIMER:  DEC     MSECS           ;COUNT 60 CYCLES
006462  001005                        BNE     TIMEX           ;BR IF NOT 60
006464  012737  000060  006500        MOV     #60,    MSECS   ;RESTORE COUNT
006472  005237  001746                INC     TIME            ;INCREMENT SECONDS
006476                          TIMEX:
006476  000002                        RTI                     ;RETURN FROM INTERRUPT
006500  000000                  MSECS:  0
                                ;*********************************************************
                                ;BINARY TO OCTAL (ASCII) AND TYPE
                                ;$B20CT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;CALL:
                                ;       MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                                ;       JSR     R0,$B20CT               ;CALL FOR TYPEOUT
                                ;       .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;       .BYTE   M                       ;M=1 OR 0
                                ;                                               ;1=TYPE LEADING ZEROS
                                ;                                               ;0=SUPPRESS LEADING ZEROS
                                ;
                                ;$B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B5OCT OR $B2016
                                ;CALL:
                                ;       MOV     NUM,-(SP)
                                ;       JSR     R0,$B201
                                ;
                                ;$B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;CALL:
                                ;       MOV     NUM,-(SP)
                                ;       JSR     R0,$B2016
                                ;
006502  112037  006707        $B20CT: MOVB    (R0)+,$OMODE+1          ;PICKUP THE NUMBER OF DIGITS TO TYPE
006506  112037  006705                MOVB    (R0)+,$OFILL            ;GET THE ZERO FILL SWITCH
006512  000406                        BR      $B201
006514  112737  000001  006705  $B2016: MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
006522  112737  000006  006707        MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
006530  112737  000005  006704  $B201:  MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
006536  010346                        MOV     R3,-(SP)                ;SAVE R3
006540  010446                        MOV     R4,-(SP)                ;SAVE R4
006542  010546                        MOV     R5,-(SP)                ;SAVE R5
006544  113704  006707                MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
006550  005404                        NEG     R4
006552  062704  000006                ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
006556  110437  006706                MOVB    R4,$OMODE               ;SAVE IT FOR USE
006562  113704  006705                MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
006566  016605  000010                MOV     10(SP),R5               ;PICKUP THE INPUT NUMBER
006572  005003                        CLR     R3                      ;CLEAR THE OUTPUT WORD
006574  006105                  1$:   ROL     R5                      ;ROTATE MSB INTO "C"
006576  000404                        BR      3$                      ;GO DO MSB
006600  006105                  2$:   ROL     R5                      ;FORM THIS DIGIT
006602  006105                        ROL     R5
006604  006105                        ROL     R5
006606  010503                        MOV     R5,R3
006610  006103                  3$:   ROL     R3                      ;GET LSB OF THIS DIGIT
006612  105337  006706                DECB    $OMODE                  ;TYPE THIS DIGIT?
006616  100016                        BPL     7$                      ;BR IF NO
```

```
1027  006620  042703  177770            BIC     #177770,R3          ;GET RID OF JUNK
1028  006624  001002                    BNE     4$                  ;TEST FOR 0
1029  006626  005704                    TST     R4                  ;SUPPRESS THIS 0?
1030  006630  001403                    BEQ     5$                  ;BR IF YES
1031  006632  005204              4$:   INC     R4                  ;DON'T SUPPRESS ANYMORE 0'S
1032  006634  052703  000060          BIS     #'0,R3              ;MAKE THIS DIGIT ASCII
1033  006640  052703  000040    5$:   BIS     #' ,R3              ;MAKE ASCII IF NOT ALREADY
1034  006644  110337  006702          MOVB    R3,8$               ;SAVE FOR TYPING
1035  006650  104400  006702          TYPE    ,8$                 ;GO TYPE THIS DIGIT
1036  006654  105337  006704    7$:   DECB    $OCNT               ;COUNT BY 1
1037  006660  003347                    BGT     2$                  ;BR IF MORE TO DO
1038  006662  002402                    BLT     6$                  ;BR IF DONE
1039  006664  005204                    INC     R4                  ;INSURE LAST DIGIT ISN'T A BLANK
1040  006666  000744                    BR      2$                  ;GO DO THE LAST DIGIT
1041  006670  012605              6$:   MOV     (SP)+,R5            ;RESTORE R5
1042  006672  012604                    MOV     (SP)+,R4            ;RESTORE R4
1043  006674  012603                    MOV     (SP)+,R3            ;RESTORE R3
1044  006676  012616                    MOV     (SP)+,(SP)          ;SET THE STACK FOR RETURNING
1045  006700  000200                    RTS     R0                  ;RETURN
1046  006702     000              8$:   .BYTE   0                   ;STORAGE FOR ASCII DIGIT
1047  006703     000                    .BYTE   0                   ;TERMINATOR FOR TYPE ROUTINE
1048  006704     000              $OCNT: .BYTE   0                   ;OCTAL DIGIT COUNTER
1049  006705     000              $OFILL:.BYTE   0                   ;ZERO FILL SWITCH
1050  006706  000000              $OMODE:0                           ;NUMBER OF DIGITS TO TYPE
1051
1052
1053                              ;*****************************************************
1054                              ;TRAP HANDLER
1055  006710  010046              $TRAP:  MOV     R0,-(SP)           ;SAVE R0
1056  006712  016600  000002             MOV     2(SP),R0           ;GET TRAP ADDRESS
1057  006716  005740                     TST     -(R0)              ;BACKUP BY 2
1058  006720  111000                     MOVB    (R0),R0            ;GET RIGHT BYTE OF TRAP
1059  006722  016000  006730             MOV     $TRPAD(R0),R0      ;INDEX TO TABLE
1060  006726  000200                     RTS     R0                 ;GO TO ROUTINE
1061
1062                              ;TRAP TABLE
1063                              ;       ROUTINE
1064                              ;       -------
1065
1066  006730  001100              $TRPAD: $TYPE
1067          104400                              TYPE=TRAP+0
1068  006732  006114                     $READC
1069          104402                              GETCHR=TRAP+2
1070  006734  006156                     $READS
1071          104404                              GETSTR=TRAP+4
1072  006736  006260                     $ACCEPT
1073          104406                              ACCEPT=TRAP+6
1074  006740  007120                     .STPS
1075          104410                              STPS=TRAP+10
1076  006742  007010                     .KBDIN
1077          104412                              KBDIN=TRAP+12
1078  006744  007142                     .SUSWR
1079          104414                              SUSWR=TRAP+14
1080  006746  007026                     .SETSWI
1081          104416                              SETSWI=TRAP+16
1082
```

```
1083
1084                                    ;*****************************************************************
1085                                    ;
1086                                    ;POWER DOWN ROUTINE.
1087                                    ;SINCE INIT IS ISSUED IN A PWR DN/UP SEQUENCE.
1088                                    ;PROGRAM MUST BE RESTARTED AGAIN.
1089                                    ;
1090
1091  006750  012737  006762  000024  SPWRDN: MOV     #SPWRUP,@#24
1092  006756  000000                         HALT
1093  006760  000776                         BR      .-2
1094
1095                                    ;*****************************************************************
1096                                    ;
1097                                    ;POWER UP ROUTINE.
1098                                    ;MESSAGE "POWER HAS FAILED..." WILL BE PRINTED OUT.
1099                                    ;PROGRAM WILL BE RESTARTED.
1100                                    ;
1101
1102  006762  012737  006750  000024  SPWRUP: MOV     #SPWRDN,@#24
1103  006770  012706  001070                 MOV     #STACK,SP
1104  006774  104400  010103                 TYPE    ,MSG20  ;<15><12> POWER FAILED..
1105  007000  104410  000000                 STPS,PRTY0
1106  007004  000137  000200                 JMP     @#200
1107
1108  007010  032737  000001  001760  .KBDIN: BIT     #BIT0,FLAG      ;TEST ^G FLAG
1109  007016  001437                         BEQ     TOOUT           ;NO, EXIT
1110  007020  042777  000100  172762         BIC     #100,@TKS       ;CLEAR TTY IE
1111  007026  104400  010227          .SETSWI:TYPE    ,SWR?           ;TYPE SWR=
1112  007032  017746  172724                 MOV     @SWR,-(SP)      ;SET UP OCTAL TYPEOUT
1113  007036  004037  006514                 JSR     R0,$$2016       ;DO IT
1114  007042  104400  010237                 TYPE    ,NEQ            ;TYPE NEW=
1115  007046  017737  172710  001546         MOV     @SWR,WORK       ;SET UP FOR CR DEFAULT
1116  007054  104406  001546                 ACCEPT  ,WORK           ;GET VALUE
1117  007060  013777  001546  172674         MOV     WORK,@SWR       ;REPLACE IT
1118  007066  032737  000002  001760         BIT     #BIT1,FLAG      ;SEE HOW WE GOT HERE
1119  007074  001005                         BNE     1$              ;WRONG WAY?
1120  007076  005777  172710                 TST     @TKB            ;CLEAR BUFFER
1121  007102  052777  000100  172700         BIS     #100,@TKS       ;RESET TTY IE
1122  007110  042737  000003  001760  1$:    BIC     #BIT0+BIT1,FLAG ;CLEAR FLAG BITS
1123  007116  000002                 TOOUT:  RTI                     ;EXIT
1124
1125  007120  042766  000340  000002  .STPS:  BIC     #PRTY7,2(SP)    ;CLEAR OUT PRIORITY BITS
1126  007126  057666  000000  000002         BIS     @(SP),2(SP)     ;SET NEW PRIORITY
1127  007134  062716  000002                 ADD     #2,(SP)         ;SETUP EXIT
1128  007140  000002                         RTI                     ;EXIT
1129
1130  007142  013746  000006          .SWR:  MOV     6,-(SP)         ;SAVE 6 ON STACK
1131  007146  013746  000004                 MOV     4,-(SP)         ;SAVE 4 ON STACK
1132  007152  012737  000  000004            MOV     #1$,4           ;SETUP TIMEOUT
1133  007160  012737  000  000006            MOV     #340,6          ;SET PRIORITY
1134  007166  022777  177777  172566         CMP     #-1,@SWR        ;TEST FOR 177570
1135  007174  001  002                      BEQ     2$              ;NOT ALL 1'S
1136  007176  000  007                      BR      3$              ;IT'S THERE - EXIT
1137  007200  005  026          1$:        CMP     (SP)+,(SP)+     ;ADJUST STACK AFTER TRAP
1138  007202  012737  000176  001762  2$:    MOV     #SAREG,SWR      ;REPLACE HARDWARE REGISTERS
```

```
1139  007210  012737  000174  001764          MOV     #DISPREG,DISPLAY  ;WITH SOFTWARE REGISTERS
1140  007216  012637  000004          3$:      MOV     (SP)+,4           ;RESTORE 4
1141  007222  012637  000006                   MOV     (SP)+,6           ;RESTORE 6
1142  007226  000002                           RTI                       ;EXIT
1143
1144
1145
1146                                  ;*****************************************************************
1147                                  ;       RANDOM CHARACTER BUFFER GENERATOR
1148                                  ;*****************************************************************
1149
1150  007230  010546          SETBUF: MOV     R5,-(SP)          ;SAVE R5
1151  007232  010346                  MOV     R3,-(SP)          ;SAVE R3
1152  007234  010046                  MOV     R0,-(SP)          ;SAVE R0
1153  007236  005701                  TST     R1                ;VARIABLE DATA?
1154  007240  001105                  BNE     10$               ;YES,NOT RANDOM PATTERN
1155  007242  012703  003224          MOV     #DP1,R3           ;SETUP BUFFER ADDR
1156  007246  012705  000400          MOV     #256.,R5                  ;SETUP BUFF COUNT
1157  007252  010537  001650          MOV     R5,COUNT+2
1158  007256  005205                  INC     R5                ;ADJUST COUNTER
1159  007260  004737  007322  1$:     JSR     PC,3$             ;GENERATE RANDOM CHAR
1160  007264  120027  000040          CMPB    R0,#40  ;LOW BYTE OVER 40
1161  007270  002773                  BLT     1$                ;NO TRY AGAIN
1162  007272  120027  000177          CMPB    R0,#177 ;LOW BYTE UNDER 177
1163  007276  003370                  BGT     1$                ;NO TRY AGAIN
1164  007300  005305                  DEC     R5                ;DEC COUNT
1165  007302  001402                  BEQ     2$                ;ALL DONE
1166  007304  110023                  MOVB    R0,(R3)+          ;MOVE CHAR TO BUFF
1167  007306  000764                  BR      1$                ;GET NEXT
1168  007310  105013          2$:     CLRB    (R3)              ;SET TERMINATOR TO ZERO
1169  007312  012600                  MOV     (SP)+,R0          ;RESTORE R0
1170  007314  012603                  MOV     (SP)+,R3          ;RESTORE R3
1171  007316  012605                  MOV     (SP)+,R5          ;RESTORE R5
1172  007320  000207                  RTS     PC
1173
1174  007322  010146          3$:     MOV     R1,-(SP)                  ;SAVE R1
1175  007324  010346                  MOV     R3,-(SP)          ;SAVE R3
1176  007326  010246                  MOV     R2,-(SP)          ;SAVE R2
1177  007330  013700  001642          MOV     L,R0
1178  007334  013701  001644          MOV     H,R1
1179  007340  012703  000010          MOV     #10,R3
1180  007344  005002                  CLR     R2
1181  007346  006302          4$:     ASL     R2
1182  007350  000241                  CLC
1183  007352  006000                  ROR     R0
1184  007354  006001                  ROR     R1
1185  007356  103412                  BCS     5$
1186  007360  042702  000001          BIC     #1,R2
1187  007364  042700  010000          BIC     #10000,R0
1188  007370  032700  002000          BIT     #2000,R0
1189  007374  001414                  BEQ     6$
1190  007376  052700  010000          BIS     #10000,R0
1191  007402  000411                  BR      6$
1192  007404  052702  000001  5$:     BIS     #1,R2
1193  007410  052700  010000          BIS     #10000,R0
1194  007414  032700  002000          BIT     #2000,R0
```

```
1195  007420  001402                        BEQ     6$
1196  007422  042700  010000                BIC     #10000,R0
1197  007426  005303                6$:     DEC     R3
1198  007430  001346                        BNE     4$
1199  007432  010037  001642                MOV     R0,L
1200  007436  010137  001644                MOV     R1,H
1201  007442  010200                        MOV     R2,R0
1202  007444  012602                        MOV     (SP)+,R2        ;RESTORE R2
1203  007446  012603                        MOV     (SP)+,R3        ;RESTORE R3
1204  007450  012601                        MOV     (SP)+,R1        ;RESTORE R1
1205  007452  000207                        RTS     PC              ;EXIT
1206
1207  007454  005005                10$:    CLR     R5
1208  007456  012703  003024                MOV     #VDB,R3         ;SET BUFFER ADDRESS
1209  007462  105723                11$:    TSTB    (R3)+           ;CHECK FOR TERM CHAR
1210  007464  001402                        BEQ     12$
1211  007466  005205                        INC     R5              ;INC BYTE COUNT
1212  007470  000774                        BR      11$             ;START OVER
1213  007472  010537  001650        12$:    MOV     R5,COUNT+2
1214  007476  000704                        BR      2$
1215
1216
```

```
1217                                    ;*****************************************************************
1218                                    ;
1219                                    ;       AREA RESERVED FOR MOST ASCIZ MESSAGES.
1220                                    ;
1221                                    ;*****************************************************************
1222
1223
1224
1225  007500  000040                    MSG00:  .ASCIZ  / /
      007502  005015  052126  030466    MSG01:  .ASCIZ  <15><12>/VT61-T AND VT71-T LOOPBACK TEST/<15><12>
      007546  005015  051511  020122    MSG02:  .ASCIZ  <15><12>/ISR NOT LOADED!!/
      007571     015  041012  051525    MSG03:  .ASCIZ  <15><12>/BUS ADDRESS=/
      007610  005015  042526  052103    MSG04:  .ASCIZ  <15><12>/VECTOR ADDRESS=/
      007632  005015  051120  047511    MSG05:  .ASCIZ  <15><12>/PRIORITY=/
      007646  005015  044514  042516    MSG06:  .ASCIZ  <15><12>/LINE # (OCTAL)=/
      007670  005015  040520  040522    MSG07:  .ASCIZ  <15><12>/PARAMS #1=/
      007705     015  040412  041523    MSG08:  .ASCIZ  <15><12>/ASCII PARAM=/
      007724  005015  042523  020124    MSG09:  .ASCIZ  <15><12>/SET SWITCHES.../
      007746  005015  047105  042524    MSG10:  .ASCIZ  <15><12>/ENTER DATA/<15><12>
      007765     015  042412  042116    MSG11:  .ASCIZ  <15><12>/END OF PASS /
      010004  040440  020124  047514    MSG12:  .ASCIZ  / AT LOC /
      010015     040  041522  020126    MSG13:  .ASCIZ  / RCV CSR=/
      010027     040  041530  051123    MSG14:  .ASCIZ  / XCSR=/
      010036  020440  020040  051503    MSG17:  .ASCIZ  / !  CSR=  /
      010051     056  005015     000    MSG18:  .ASCIZ  /./<15><12>
      010055     015  047012  020117    MSG19:  .ASCIZ  <15><12>/NO CLOCKS AVAILABLE/
      010103     015  050012  053517    MSG20:  .ASCIZ  <15><12>/POWER FAILED../
      010124  020073  040502  020104    MSG23:  .ASCIZ  /; BAD DATA=/
      010140  020040  043440  047517    MSG24:  .ASCIZ  /   GOOD DATA=/
      010156  005015  051124  047101    MSG26:  .ASCIZ  <15><12>/TRANSMITTED DATA=/<15><12>
      010204  005015  042522  042503    MSG27:  .ASCIZ  <15><12>/RECEIVED DATA=/<15><12>
      010227     015  051412  051127    SWEQ:   .ASCIZ  <15><12>/SWR= /
      010237     040  042516  036527    NEQ:    .ASCIZ  / NEW= /
      010246  052536  005015     000    CTLU:   .ASCIZ  /↑U/<15><12>
      010253     177  000177            MFILL:  .ASCIZ  <177><177>
      010256  037440  000040            QUES:   .ASCIZ  " ? "
      010262  005015  000052            ASTRK:  .ASCIZ  <15><12>"*"
      010266     015                    CRLF:   .ASCII  <15>
      010267     012     000            LF:     .ASCIZ  <12>
      010271     015  005012     000    TUCRLF: .ASCIZ  <15><12><12>
      010275     105  051122  051117    EM0:    .ASCIZ  "ERROR HALT ON LINE #"
      010322  040527  052111  047111    EM1:    .ASCIZ  "WAITING TO RCV ON LINE #"
      010353     127  044501  044524    EM2:    .ASCIZ  "WAITING TO XMIT ON LINE #"
      010405     122  053103  041040    EM6:    .ASCIZ  "RCV BUFFER FULL"
      010425     104  052101  020101    EM7:    .ASCIZ  "DATA COMPARE ERROR LINE(8) #"
      010462  051105  047522  020122    EM10:   .ASCIZ  "ERROR  RCV CSR=CONTENTS OF SELECT 0 REGISTER"
      010537     015  052012  042510    MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
      010620  005015     000            MSG1:   .ASCIZ  <15><12>
      010623     015  051012  041505    MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
      010650  005015  040504  040524    MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
      010673     015  042412  051122    MFULL:  .ASCIZ<15><12>/ERROR! RECEIVER BUFFER FULL/
      010731     015  042412  051122    SILO:   .ASCIZ<15><12>/ERROR! SILO OVERFLOW/
      010760  005015  047516  020116    NONEX:  .ASCIZ<15><12>/NON EXISTENT MEMORY ERROR/
      011014  005015  042524  052123    TESTXX: .ASCIZ<15><12>/TESTING LINE (8)#/
      011040  000003                    CTLC:   .ASCIZ  <003>
                                        .EVEN
```

```
1226
1227
1228                                    ;*****************************************************************
1229                                    ;       DH11-X INTERFACE SERVICE ROUTINE
1230                                    ;*****************************************************************
1231  011042  000240                   START:  NOP
1232  011044  017700  170712                   MOV     @SWR,   R0      ;SETUP MODE IN R0
1233  011050  042700  177400                   BIC     #177400,        R0      ;STRIP JUNK
1234  011054  013702  001722                   MOV     VA,     R2      ;SETUP
1235  011060  012722  012412                   MOV     #RISR,  (R2)+   ;INTERRUPT
1236  011064  013722  001724                   MOV     PRIOR,  (R2)+   ;VECTORS
1237  011070  012722  012200                   MOV     #XISR,  (R2)+   ;
1238  011074  013722  001724                   MOV     PRIOR,  (R2)+   ;
1239  011100  013704  001720                   MOV     BA,     R4      ;SETUP BUS ADDR INDEX
1240  011104  005737  001726                   TST     PARAM1
1241  011110  001001                           BNE     4$
1242  011112  000000                           HALT                    ;NO LINES SELECTED
1243  011114  012714  004000           4$:     MOV     #MC,@RCSR
1244  011120  004737  012560                   JSR     PC,NEEXT
1245
1246
1247  011124  005037  001746           GO:     CLR     TIME
1248  011130  005037  001670                   CLR     DELAY
1249  011134  005037  001674                   CLR     STOP
1250
1251  011140  104412                   $XLB:   KBDIN
1252  011142  013737  001650  001646           MOV     COUNT+2,COUNT
1253  011150  013705  001650                   MOV     COUNT+2,R5
1254  011154  032737  040000  001760           BIT     #ESC,FLAG
1255  011162  001403                           BEQ     1$
1256  011164  012737  000004  001646           MOV     #4,COUNT
1257  011172  004737  012324           1$:     JSR     PC,STARTR
1258  011176  004737  012110                   JSR     PC,STARTX
1259  011202  005037  001746                   CLR     TIME
1260  011206  032700  100000           2$:     BIT     #XFLG,STAT
1261  011212  001016                           BNE     3$
1262  011214  032700  040000           7$:     BIT     #RFLG,STAT
1263  011220  001024                           BNE     4$
1264  011222  023737  001746  001572           CMP     TIME,$TYME
1265  011230  103766                           BLO     2$
1266  011232  011402                           MOV     @RCSR,R2
1267  011234  016403  000000                   MOV     XCSR(R4),R3
1268  011240  104001                           HLT     1               ;ERROR MESSAGE HERE COULD IMPLY THAT
1269  011242  005037  001746                   CLR     TIME            ;TERMINAL IS OFFLINE-MESSAGE NOT
1270  011246  000757                           BR      2$              ;LOOPBACKED BEFORE TIMER EXPIRED
1271  011250  032737  040000  001730   3$:     BIT     #HALF.DUPLEX,PARAM2
1272  011256  001756                           BEQ     7$
1273  011260  042700  100000                   BIC     #XFLG,STAT
1274  011264  004737  012324                   JSR     PC,STARTR
1275  011270  000746                           BR      2$
1276  011272  032700  100000           4$:     BIT     #XFLG,STAT
1277  011276  001013                           BNE     6$
1278  011300  023737  001746  001572           CMP     TIME,$TYME
1279  011306  103771                           BLO     4$
1280  011310  011402                           MOV     @RCSR,R2
```

```
1281  011312  016403  000000                        MOV     XCSR(R4),R3
1282  011316  104002                               HLT     2               ;COULD BE PROBLEM WITH DH-MESSAGE
1283  011320  005037  001746                       CLR     TIME            ;NOT SENT BEFORE TIMER EXPIRED
1284  011324  000762                               BR      4$
1285  011326  042700  100000              6$:      BIC     #XFLG STAT
1286  011332  042700  040000              8$:      BIC     #RFLG,STAT
1287  011336  005037  001746                       CLR     TIME
1288  011342  032777  000200  170412               BIT     #NODAT,@SWR
1289  011350  001002                               BNE     5$
1290  011352  004737  011522                       JSR     PC,TESTD
1291  011356  012737  011140  001672      5$:      MOV     #$XLB,BACK
1292
1293  011364                              EOP:
1294  011364  104410  000340                       STPS,PRTY7              ;SET PS PRIORITY TO 7
1295  011370  016437  000000  001652               MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
1296  011376  042737  157777  001652               BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
1297  011404  042764  020000  000000               BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
1298  011412  012766  011452  000002               MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
1299  011420  010037  001654                       MOV     R0,SAVR0        ;SAVE REGISTER 0
1300  011424  010137  001656                       MOV     R1,SAVR1        ;SAVE REGISTER 1
1301  011430  010237  001660                       MOV     R2,SAVR2        ;SAVE REGISTER 2
1302  011434  010337  001662                       MOV     R3,SAVR3        ;SAVE REGISTER 3
1303  011440  010437  001664                       MOV     R4,SAVR4        ;SAVE REGISTER 4
1304  011444  010537  001666                       MOV     R5,SAVR5        ;SAVE REGISTER 5
1305  011450  000207                               RTS     PC              ;RETURN TO CONTROL PROGRAM
1306
1307  011452                              ENTER:
1308  011452  013700  001654                       MOV     SAVR0,R0        ;RESTORE R0
1309  011456  013701  001656                       MOV     SAVR1,R1        ;RESTORE R1
1310  011462  013702  001660                       MOV     SAVR2,R2        ;RESTORE R2
1311  011466  013703  001662                       MOV     SAVR3,R3        ;RESTORE R3
1312  011472  013704  001664                       MOV     SAVR4,R4        ;RESTORE R4
1313  011476  013705  001666                       MOV     SAVR5,R5        ;RESTORE R5
1314  011502  012737  177777  001670               MOV     #-1,DELAY
1315  011510  053764  001652  000000               BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
1316  011516  000177  170150                       JMP     @BACK
1317
1318
1319
1320                                      ;***********************************
1321                                      ;       SUBROUTINE TO CHECK
1322                                      ;       RECEIVER DATA.
1323                                      ;***********************************
1324  011522  013746  001774              TESTD:  MOV     ERDBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
1325  011526  001413                               BEQ     TSTDAT          ;BR IF NO
1326  011530  032777  020000  170224               BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
1327  011536  001007                               BNE     TSTDAT          ;BR IF YES
1328  011540  104400  010537                       TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
1329  011544  004077  170174                       JSR     R0,@82016       ;PRINT CONTENTS OF RBUF
1330  011550  005746                               TST     -(SP)
1331  011552  104400  010620                       TYPE    ,MSG1           ;<15><12>
1332  011556  032737  040000  001760  TSTDAT:     BIT     #ESC,FLAG       ;ESC MODE?
1333  011564  001405                               BEQ     1$              ;NO
1334  011566  012701  004224                       MOV     #LPB,R1
1335  011572  012705  000004                       MOV     #4,R5           ;SET COUNT
1336  011576  000404                               BR      2$
```

```
1337  011600  013701  001736          1$:     MOV     IXDA,   R1      ;SETUP XMIT DATA ADDR
1338  011604  013705  001650                  MOV     COUNT+2,R5      ;SETUP COUNT
1339  011610  005205                  2$:     INC     R5              ;ADJUST COUNT
1340  011612  005037  001676                  CLR     $ERROR          ;CLEAR ERROR CNT
1341  011616  013702  001734                  MOV     IRDA,   R2      ;SETUP RCV DATA ADDR
1342  011622  005305                  SCAN4:  DEC     R5              ;DEC COUNT
1343  011624  001422                          BEQ     TESTDX          ;ALL DONE
1344  011626  122122                          CMPB    (R1)+,(R2)+     ;DATA OK
1345  011630  001774                          BEQ     SCAN4           ;YES NEXT CHAR PLEASE
1346  011632  032777  000010  170122          BIT     #FSTERR,@SWR    ;PRINT JUST FIRST ERROR
1347  011640  001403                          BEQ     1$              ;NO
1348  011642  005737  001676                  TST     $ERROR          ;IS THIS THE FIRST
1349  011646  001365                          BNE     SCAN4           ;NO
1350  011650  005237  001676          1$:     INC     $ERROR          ;INC ERROR CNT
1351  011654  010246                          MOV     R2,-(SP)        ;SAVE R2
1352  011656  114103                          MOVB    -(R1),R3
1353  011660  114202                          MOVB    -(R2),R2
1354  011662  104003                          HLT     3
1355  011664  012602                          MOV     (SP)+,R2        ;RESTORE R2
1356  011666  105721                          TSTB    (R1)+           ;ADJUST POINTER
1357  011670  000754                          BR      SCAN4
1358  011672  005726                  TESTDX: TST     (SP)+           ;POP STACK
1359  011674  042737  040000  001760          BIC     #ESC,FLAG
1360  011702  000207                          RTS     PC              ;RETURN FROM SUB/ROUT
1361
1362
1363                                  ;ROUTINE TO INPUT LINE NUMBERS TO TEST
1364  011704  104400  007646          GETPRM: TYPE    ,MSG06          ;ASK FOR LINE NUMBERS
1365  011710  005010                          CLR     (R0)            ;CLEAR LINE #
1366  011712  052737  100000  001570          BIS     #BIT15,FLAGS
1367  011720  104404                          GETSTR
1368  011722  002024                          IBUF                    ;STORE THEN AT IBUF
1369  011724  010146                          MOV     R1,-(SP)
1370  011726  010246                          MOV     R2,-(SP)
1371  011730  010346                          MOV     R3,-(SP)
1372  011732  010446                          MOV     R4,-(SP)
1373  011734  012701  002024                  MOV     #IBUF,R1        ;SET UP POINTERS
1374  011740  012702  001546          9$:     MOV     #WORK,R2
1375  011744  020227  001550          1$:     CMP     R2,#WORK+2      ;HAVE TWO CHARS YET
1376  011750  001002                          BNE     4$
1377  011752  000342                          SWAB    -(R2)
1378  011754  000414                          BR      5$
1379  011756  112122                  4$:     MOVB    (R1)+,(R2)+     ;GET A CHARACTER
1380  011760  124227  000015                  CMPB    -(R2),#15       ;<CR>?
1381  011764  001443                          BEQ     8$
1382  011766  123727  001546  000054          CMPB    WORK,#54        ;IS 1ST BYTE A SPACE
1383  011774  001761                          BEQ     9$
1384  011776  122227  000054          7$:     CMPB    (R2)+,#54       ;SPACE DELIMITER?
1385  012002  001360                          BNE     1$              ;RECYCLE
1386  012004  105042                          CLRB    -(R2)
1387  012006  012702  001546          5$:     MOV     #WORK,R2        ;PROCESS LINE NUMBER
1388  012012  042712  177370                  BIC     #177370,(R2)    ;CLEAR JUNK
1389  012016  032712  000400                  BIT     #400,(R2)       ;LINE BETWEEN 10-17?
1390  012022  001402                          BEQ     2$              ;NO
1391  012024  052712  000010                  BIS     #10,(R2)        ;ELSE ADD 10 TO LOW BYTE
1392  012030  005003                  2$:     CLR     R3
```

```
1393  012032  105212                          INCB    (R2)
1394  012034  000261                          SEC
1395  012036  006103                  3$:     ROL     R3              ;SET BIT IN PARAM1 FOR LINE #
1396  012040  000241                          CLC
1397  012042  105312                          DECB    (R2)
1398  012044  001374                          BNE     3$
1399  012046  050310                          BIS     R3,(R0)         ;DO IT HERE
1400  012050  000735                          BR      1$              ;RECYCLE
1401  012052  005720                  11$:    TST     (R0)+           ;BUMP THE POINTER BY 2
1402  012054  012604                          MOV     (SP)+,R4
1403  012056  012603                          MOV     (SP)+,R3
1404  012060  012602                          MOV     (SP)+,R2
1405  012062  012601                          MOV     (SP)+,R1
1406  012064  042737  100000  001570          BIC     #BIT15,FLAGS
1407  012072  000207                          RETURN                  ;EXIT
1408  012074  123727  001546  000015  8$:     CMPB    WORK,#15
1409  012102  001763                          BEQ     11$
1410  012104  105012                          CLRB    (R2)
1411  012106  000737                          BR      5$
1412
```

```
1413
1414                                    ;*****************************************************************
1415                                    ;       TRANSMITTER INITIALIZATION SUBROUTINE
1416                                    ;*****************************************************************
1417
1418  012110  032737  040000  001760  STARTX: BIT     #ESC,FLAG       ;ESCAPE MODE YET
1419  012116  001407                          BEQ     7$
1420  012120  012764  004224  000006          MOV     #LPB,CAR(R4)
1421  012126  012764  177774  000010          MOV     #-4,BCR(R4)
1422  012134  000411                          BR      8$
1423  012136  013737  001736  002006  7$:     MOV     IXDA,XDA
1424  012144  013764  002006  000006          MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
1425  012152  012764  177777  000010          MOV     #-1,BCR(R4)     ;LOAD BYTE COUNT REG
1426  012160  042700  100000          8$:     BIC     #XFLG,STAT      ;CLEAR XMIT DONE FLAG
1427  012164  052714  020000                  BIS     #TIE,@RCSR      ;SET INTERUPT ENABLE
1428  012170  013764  001704  000012          MOV     BARTMP,BAR(R4)  ;LOAD BAR REG
1429  012176  000207                          RTS     PC
1430
1431  012200  042714  100000          XISR:   BIC     #TI,@RCSR       ;CLEAR XMIT DONE
1432  012204  032714  002000                  BIT     #NEM,@RCSR      ;NON-EXISTENT MEM ERROR?
1433  012210  001407                          BEQ     1$              ;BR IF NO
1434  012212  011402                          MOV     @RCSR,R2        ;SAVE CSR FOR TYPE OUT
1435  012214  005003                          CLR     R3
1436  012216  104005                          HLT     5
1437  012220  104400  010760                  TYPE    ,NONEX          ;TYPE ERROR MESS
1438  012224  000000                          HALT
1439  012226  000776                          BR      .-2             ;BR HALT
1440  012230  032737  040000  001760  1$:     BIT     #ESC,FLAG
1441  012236  001002                          BNE     4$
1442  012240  005305                          DEC     R5
1443  012242  001005                          BNE     NOXMON
1444  012244  052700  100000          4$:     BIS     #XFLG,STAT      ;SET XMIT DONE FLAG
1445  012250  042714  020000                  BIC     #TIE,@RCSR      ;CLEAR INTERUPT ENABLE
1446  012254  000416                  3$:     BR      XISR2
1447  012256  005237  002006          NOXMON: INC     XDA             ;INC TXBUF POINTER
1448  012262  013764  002006  000006          MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
1449  012270  005737  001710                  TST     TRNFLG          ;IS THIS FIRST TIME?
1450  012274  001006                          BNE     XISR2           ;BR IF YES
1451  012276  012764  177777  000010          MOV     #-1,BCR(R4)     ;LOAD BYTE COUNT REG
1452  012304  013764  001704  000012          MOV     BARTMP,BAR(R4)  ;SET BAR BIT
1453  012312  005037  001746          XISR2:  CLR     TIME
1454  012316  005037  001710                  CLR     TRNFLG
1455  012322  000002                          RTI
1456                                    ;*****************************************************************
1457                                    ;       RECEIVER INITIALIZATION SUBROUTINE
1458                                    ;*****************************************************************
1459
1460  012324  032737  040000  001730  STARTR: BIT     #HALF.DUPLEX,PARAM2     ;HALF DUPLEX?
1461  012332  001005                          BNE     2$              ;BR IF YES
1462  012334  005037  001700                  CLR     TEMP1           ;START DELAY
1463  012340  005237  001700          1$:     INC     TEMP1
1464  012344  001375                          BNE     1$
1465  012346  042700  040000          2$:     BIC     #RFLG,STAT      ;CLEAR RFLG
1466  012352  013737  001734  002004          MOV     IRDA,RDA        ;SET UP RECEIVER DATA ADD
1467  012360  012737  001000  002002          MOV     #1000,RCC       ;SET UP BUFFER LIMIT
1468  012366  012737  177777  001706          MOV     #-1,SNCFLG      ;SET SOFTWARE FLAG
```

```
1469  012374  005037  001772                  CLR     ERCSR           ;CLEAR ERROR RECORDS
1470  012400  005037  001774                  CLR     ERDBR
1471  012404  052714  010100                  BIS     #RIE+SIE,@RCSR  ;SET INTERUPT ENABLES
1472  012410  000207                          RTS     PC
1473
1474  012412  032714  040000          RISR:   BIT     #SI,@RCSR       ;SILO OVERFLOW?
1475  012416  001407                          BEQ     1$              ;BR IF NO
1476  012420  011402                          MOV     @RCSR,R2        ;SAVE CSR FOR TYPEOUT
1477  012422  005003                          CLR     R3
1478  012424  104005                          HLT     5               ;ERROR HLT
1479  012426  104400  010731                  TYPE    ,SILO           ;TYPE ERROR MESS
1480  012432  000000                          HALT
1481  012434  000776                          BR      .-2             ;BR HALT
1482  012436  016401  000002          1$:     MOV     RRCR(R4),R1     ;PUT CHAR IN R1
1483  012442  042701  000200                  BIC     #200,R1         ;STRIP A BIT
1484  012446  005701                          TST     R1              ;VALID DATA?
1485  012450  100403                          BMI     4$              ;BR IF YES
1486  012452  011402                          MOV     @RCSR,R2        ;SAVE CSR FOR TYPEOUT
1487  012454  005003                          CLR     R3
1488  012456  104005                          HLT     5               ;ERROR HLT
1489  012460  032701  070000          4$:     BIT     #DO+FE+PE,R1    ;OVERRUN,FRAMING OR PARITY ERROR?
1490  012464  001404                          BEQ     3$              ;BR IF NO
1491  012466  011437  001772                  MOV     @RCSR,ERCSR     ;SAVE CSR
1492  012472  010137  001774                  MOV     R1,ERDBR        ;SAVE CHAR
1493  012476  110177  167302          3$:     MOVB    R1,@RDA         ;STORE CHAR IN BUFFER
1494  012502  005237  002004          2$:     INC     RDA             ;INC RECEIVER BUFFER POINTER
1495  012506  105077  167272                  CLRB    @RDA            ;CLEAR NEXT LOCATION
1496  012512  005337  002002                  DEC     RCC             ;DEC CHAR COUNT
1497  012516  001004                          BNE     5$              ;BR IF BUFFER NOT FULL
1498  012520  000005                          RESET
1499  012522  104004                          HLT     4
1500  012524  000000                          HALT
1501  012526  000776                          BR      .-2             ;BR HALT
1502  012530  005337  001646          5$:     DEC     COUNT
1503  012534  001004                          BNE     6$
1504  012536  042714  010100          8$:     BIC     #RIE+SIE,@RCSR  ;CLEAR INTERUPT ENABLES
1505  012542  052700  040000                  BIS     #RFLG,STAT      ;SET RCV DONE FLAG
1506  012546  005037  001746          6$:     CLR     TIME
1507  012552  005037  001706                  CLR     SNCFLG          ;CLEAR FLAG
1508  012556  000002                          RTI
1509
1510
1511  012560  105737  001565          NEEXT:  TSTB    LINE+1          ;FIRST TIME THRU
1512  012564  001010                          BNE     1$              ;NO BR
1513  012566  105137  001565                  COMB    LINE+1          ;YES SET TO NONZERO
1514  012572  012737  000001  001566          MOV     #1,LINMSK       ;SET FOR LINE 0
1515  012600  105037  001564                  CLRB    LINE            ;CLEAR OUT LINE #
1516  012604  000405                          BR      2$
1517  012606  105237  001564          1$:     INCB    LINE            ;NEXT?
1518  012612  006337  001566                  ASL     LINMSK          ;SHIFT FOR NEXT LINE
1519  012616  001405                          BEQ     3$              ;ALL DONE EXIT
1520  012620  033737  001566  001726  2$:     BIT     LINMSK,PARAM1   ;THIS LINE?
1521  012626  001767                          BEQ     1$              ;NO
1522  012630  000410                          BR      4$              ;PROCEED
1523  012632  005037  001564          3$:     CLR     LINE            ;INIT TAGS
1524  012636  052737  000100  001760          BIS     #BIT6,FLAG      ;SET ONCE THRU FLAG
```

```
 1525  012644  162716  000004                    SUB     #4,(SP)          ;ADJUST STACK TO CALL SELINE
 1526  012650  000420                            BR      5$               ;DO IT
 1527  012652  052714  004000            4$:     BIS     #MC,@RCSR        ;CLEAR OUT LINE #
 1528  012656  153714  001564                    BISB    LINE,@RCSR       ;SET NEW LINE #
 1529  012662  053764  001730  000004            BIS     PARAM2,LPR(R4)   ;SET PARAMS
 1530  012670  013737  001566  001704            MOV     LINMSK,BARTMP    ;SET BAR BITS
 1531  012676  005737  001576                    TST     $PASS            ;FIRST PASS
 1532  012702  001003                            BNE     5$               ;IF NOT
 1533  012704  052737  040000  001760            BIS     #ESC,FLAG        ;SET ESCAPE FLAG
 1534  012712  000207                    5$:     RTS     PC               ;EXIT
 1535          000001                    .END
```

```
ACCEPT= 104406  424   773   901   1073#  1116
ANYMOR  005566  773#
ASTRK   010262  400   1225#
BA      001720  526#  1239
BACK    001672  514#  1291*  1316
BAR   = 000012  294#  571*  1428*  1452*
BARTMP  001704  519#  571   1428   1452   1530*
BCR   = 000010  293#  571*  1421*  1425*  1451*
BEGIN   004466  349   575#  581
BIT0  = 000001  284#  571   1108   1122
BIT00 = 000001  274#  284
BIT01 = 000002  273#  283
BIT02 = 000004  272#  282
BIT03 = 000010  271#  281
BIT04 = 000020  270#  280
BIT05 = 000040  269#  279
BIT06 = 000100  268#  278
BIT07 = 000200  267#  277
BIT08 = 000400  266#  276
BIT09 = 001000  265#  275
BIT1  = 000002  283#  622   1118   1122
BIT10 = 002000  264#  304
BIT11 = 004000  263#  299
BIT12 = 010000  262#  300   307
BIT13 = 020000  261#  302   306   313#  789   1326
BIT14 = 040000  260#  297   298   301   305   741
BIT15 = 100000  259#  303   841   1366  1406
BIT2  = 000004  282#  583   590   593
BIT3  = 000010  281#  309   407   439
BIT4  = 000020  280#  411
BIT5  = 000040  279#  404   416   427
BIT6  = 000100  278#  296   571   719   721   1524
BIT7  = 000200  277#
BIT8  = 000400  276#
BIT9  = 001000  275#
BPTVEC= 000014  319#
B2016   001744  536#  699*  1329
CAR   = 000006  292#  571*  1420*  1424*  1448*
CLRIB   005044  662#  664
COUNT   001646  505#  1157*  1213*  1252*  1253  1256*  1338  1502*
CRLF    010266  392   420   435   632   792   824   839   914   954   1225#
CTLC    011040  571   1225#
CTLU    010246  968   1225#
DAT     002020  563#  644
DELAY   001670  513#  1248*  1314*
DISPLA  001764  548#  1139*
DISPRE  000174  344#  1139
DMPHLT  001220  398#  421   434
DMPLST  001524  425   429*  459#  571
DO    = 040000  305#  1489
DP1     003224  563   571#  1155
DSFLG = 020000  312#
DSSTAT  001776  553#
DUMP    001370  417   425#
EMTAB   006050  795   849#
EMTVEC= 000030  322#
```

```
EM0     010275            796     849    1225#
EM1     010322            850    1225#
EM10    010462            854    1225#
EM2     010353            851    1225#
EM6     010405            853    1225#
EM7     010425            852    1225#
ENTER   011452           1298    1307#
EOP     011364           1293#
EOPPY   005260            711#
ERCSR   001772            551#   1469*  1491*
ERDBR   001774            552#   1324   1470*  1492*
ERRVEC= 000004            315#
ESC   = 040000            298#    721   1254   1332   1359   1418   1440   1533
ESC.TE  001756            544#
FE    = 020000            306#   1489
FILL    001712            522#
FLAG    001760            398     399*   407*   411*   415*   416*   427    439    546#   571*   583*   590    593*
                          622*    719    721*  1108   1118   1122*  1254   1332   1359*  1418   1440   1524*  1533*
FLAGS   001570            481#    582*   910   1366*  1406*
FSTERR= 000010            309#   1346
GETAMY  005550            597     600    603    608    769#
GETBA   004560            591     594#
GETCHR= 104402            401     896    930   1069#
GETIC   006164            896#    905    909
GETIT   004532            589#
GETPRI  004604            602#
GETPRM  011704            605    1364#
GETSTR= 104404            651    1071#  1367
GETVA   004574            599#
GO      011124           1247#
GOTIC   006222            899     906#
H       001644            504#   1178   1200*
HALF.D= 040000            297#   1271   1460
HLTX    006042            842     845#
IBUF    002024            567#    571    653    661   1368   1373
IOTVEC= 000020            320#
IRCC    001742            535#
IRDA    001734            532#    571    653*   738   1341   1466
ISR     001714            525#
IXDA    001736            533#    571    644*   734   1337   1423
KBDIN = 104412            722     845   1077#  1251
L       001642            503#   1177   1199*
LF      010267            956    1225#
LINE    001564            479#    571*   579*   619*   749    797   1511   1513*  1515*  1517*  1523*  1528
LINMSK  001566            480#   1514*  1518*  1520   1530
LOOP  = 000020            287#    746
LPB     004224            571#   1334   1420
LPR   = 000004            291#   1529*
L1      001422            428     430    433#   450    458    571
L2      001454            442#    449
L3      001500            440     452#
MC    = 004000            299#   1243   1527
MFILL   010253            615    1225#
MFULL   010673           1225#
MODEA   001750            538#
MSECS   006500            669*    974*   976*   980#
```

```
MSG0     010537            1225#   1328
MSG00    007500             444     454    1225#
MSG01    007502             592    1225#
MSG02    007546            1225#
MSG03    007571             596    1225#
MSG04    007610             599    1225#
MSG05    007632             602    1225#
MSG06    007646            1225#   1364
MSG07    007670             607    1225#
MSG08    007705            1225#
MSG09    007724             623     626    1225#
MSG1     010620            1225#   1331
MSG10    007746             650    1225#
MSG11    007765             727    1225#
MSG12    010004             801    1225#
MSG13    010015             830    1225#
MSG14    010027             836    1225#
MSG17    010036            1225#
MSG18    010051            1225#
MSG19    010055             691    1225#
MSG2     010623            1225#
MSG20    010103            1104    1225#
MSG23    010124             812    1225#
MSG24    010140             819    1225#
MSG26    010156             733    1225#
MSG27    010204             737    1225#
MSG3     010650            1225#
NEEXT    012560             571     718    1244    1511#
NEM   =  002000             304#   1432
NEQ      010237            1114    1225#
NODAT =  000200             288#   1288
NOLC     005122             672     679#
NONEX    010760            1225#   1437
NORTC    005224             675     686     694#
NOXMON   012256            1443    1447#
NRCR  =  000002             290#   1482
NULL     001374             382     483#
OCT      006200             901#    904
PARAM1   001726             529#   1240    1520
PARAM2   001730             530#    861    1271    1460    1529
PARAM3   001732             531#
PE    =  010000             307#   1489
PRIOR    001724             528#   1236    1238
PRTY0 =  000000             249#    688     698     754    1105
PRTY1 =  000040             250#
PRTY2 =  000100             251#
PRTY3 =  000140             252#    571
PRTY4 =  000200             253#
PRTY5 =  000240             254#
PRTY6 =  000300             255#
PRTY7 =  000340             256#    577    1125    1294
PWRVEC=  000024             321#
QTPIE    001652             506#   1295#   1296#   1315
QUES     010256             419     771     953    1225#
RCC      003002             555#   1467#   1496#
RDA      002004             556#   1466#   1493#   1494#   1495#
```

```
REST    004736              625     640#
RESTRT  004744              640     641#
RESVEC= 000010              316#
RFLG  = 040000              311#    1262    1286    1465    1505
RIE   = 000100              296#    1471    1504
RISR    012412             1235     1474#
RSTART  004632              350      615#
RX.TER  001755              542#
SAVR0   001654              507#    1299*   1308
SAVR1   001656              508#    1300*   1309
SAVR2   001660              509#    1301*   1310
SAVR3   001662              510#    1302*   1311
SAVR4   001664              511#    1303*   1312
SAVR5   001666              512#    1304*   1313
SCAN4   011622             1342#    1345    1349    1357
SETBUF  007230              654     1150#
SETSWI= 104416              624     1081#
SETTLE  001740              534#
SI    = 040000              301#    1474
SIE   = 010000              300#    1471    1504
SILO    010731             1225#    1479
SNCFLG  001706              520#    1468*   1507*
SRCSR   001770              550#
SSR   = 000016              295#
STACK = 001070              202#     576     697     756    1103
START   011042              539      707    1231#
STARTR  012324             1257     1274    1460#
STARTX  012110             1258     1418#
STOP    001674              515#    1249*
STPS  = 104410              571      577     688     698     754    1075#   1105    1294
SUSWR = 104414              578     1079#
SUTIME  005054              669#
SUXCC   005000              646      653#
SWEQ    010227             1111     1225#
SWPRNT  004652              609      620#
SWR     001762              547#     571     620     630     641     725     731     741     746     789     841    1112    1115
                           1117#    1134    1138*   1232    1288    1326    1346
SWREG   000176              345#     571     620    1138
SWRNXT  005016              657#
SWRSET  004734              638#
SW0   = 000001              248#
SW00  = 000001              238#     248
SW01  = 000002              237#     247
SW02  = 000004              236#     246     731
SW03  = 000010              235#     245
SW04  = 000020              234#     244
SW05  = 000040              233#     243
SW06  = 000100              232#     242
SW07  = 000200              231#     241
SW08  = 000400              230#     240
SW09  = 001000              229#     239
SW1   = 000002              247#
SW10  = 002000              228#
SW11  = 004000              227#
SW12  = 010000              226#     725
SW13  = 020000              225#
```

```
SW14  = 040000   224#
SW15  = 100000   223#
SW2   = 000004   246#
SW3   = 000010   245#
SW4   = 000020   244#
SW5   = 000040   243#
SW6   = 000100   242#
SW7   = 000200   241#
SW8   = 000400   240#
SW9   = 001000   239#
SXCSR   001766   549#
TBITVE= 000014   317#
TEMP1   001700   517#  1462*  1463*
TEMP2   001702   518#
TESTD   011522  1290  1324#
TESTDX  011672  1343  1358#
TESTXX  011014   748  1225#
TI    = 100000   303#  1431
TIE   = 020000   302#   571  1296  1297  1427  1445
TIME    001746   537#   977* 1247* 1259* 1264  1269* 1278  1283* 1287* 1453* 1506*
TIMER   006456   340    670   679   692   974#
TIMEX   006476   975    978#
TKB     002012   559#   571   629   881  1120
TKS     002010   558#   571*  627   660*  879  1110* 1121*
TOOUT   007116  1109  1123#
TPB     002016   390*   561#
TPS     002014   388    560#
TRAPVE= 000034   323#
TRNFLG  001710   521#  1449  1454*
TRTVEC= 000014   318#
TRX     006030   790    805   825   835   841#
TR3     006006   829    834#
TSTDAT  011556  1325  1327  1332#
TTYINT  004232   571#   658
TUCRLF  010271   730  1225#
TX.TER  001754   540#
TYPE  = 104400   392    400   403   419   420   435   444   454   592   596   599   602   607
                 615    623   626   632   650   691   727   730   733   735   737   739   748
                 771    792   796   801   812   819   824   830   836   839   897   914   933
                 953    954   956   968  1035  1067#  1104  1111  1114  1328  1331  1364  1437
                1479
VA      001722   527#  1234
VDB     003024   564    568#   652   663  1208
VIGO    005254   707#
VISR    001714   524#   589
WORK    001546   422*   424   429   469   472#   772*  773   774  1115* 1116  1117  1374  1375
                1382  1387  1408
WORK1   001550   473#
WORK2   001552   474#
WORK3   001554   475#
WORK4   001556   476#
WORK5   001560   477#
WORK6   001562   478#
XCC     000000   554#
XCSR  = 000000   289#  1267  1281  1295  1297*  1315#
XDA     002006   557#  1423* 1424  1447* 1448
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
XFLG  = 100000            310#   571   1260  1273  1276  1285  1426  1444
XISR    012200           1237  1431#
XISR2   012312           1446  1450   1453#
XSWPNT  004700            621   626#
$ACCEP  006260            922#  1072
$BAUD   006064            655   861#
$B20CT  006502            445   750    798   814   821  1002#
$B201   006530           1004  1007#
$B2016  006514            437   455    631   699   729   770   803   832   838  1005#  1113
$CHAR   001544            404*  405    409   413   471#  881*  882*  883   897   903   907   908   913
$DOAGN  005544            758   763#
$ERROR  001676            516#  1340*  1348  1350*
$ERTTL  001604            488#  783*
$HLT    005600            333   780#
$HLTAD  001602            487#  785*   786*  793   802
$HLOT   005604            783#  784
$ICNT   001600            485#  714*   715   717*
$LPADR  001542            470#  581*   640*
$OCNT   006704           1007*  1036*  1048#
$OMODE  006706           1002*  1006*  1011  1014*  1025*  1050#
$PASS   001576            484#  580*   712   723   728   743*  744   1531
$PWRDN  006750            331   1091#  1102
$PWRUP  006762           1091   1102#
$READC  006114            877#  1068
$READS  006156            892#  1070
$TIMES  001601            486#   715
$TRAP   006710            335   1055#
$TRPAD  006730           1059   1066#
$TYME   001572            482#   867*  1264  1278
$TYPE   001100            366#  1066
$WORK  = 001546            469#   901   902
$XLB    011140           1251#  1291
$OFILL  006705           1003*  1005*  1015  1049#
.      = 012714            327#  330#   339#  343#  347#  352#  461#  462#  523#  546#  567#  569#  571#
                           618   628    633   795*  880   958   1093  1439  1481  1501
.BAUD   001606            489#   867
.CTLC   004356            571#   763
.KBDIN  007010           1076   1108#
.RX     001530            461#   571*
.SETSW  007026           1080   1111#
.STPS   007120           1074   1125#
.SUSWR  007142           1078   1130#
.TX     001532            462#   571*
```

```
BOX       326#   395    466    571    572    586    634    647    708    766    777    857    871    885    971
         1146
HLT       205#  1268   1282   1354   1436   1478   1488   1499

. ABS.  012714     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZVTLA,DZVTLA=DZVTLA
 RUN-TIME:  1  3  .5 SECONDS
 RUN-TIME RATIO: 29/5=5.4
 CORE USED:  7K  (14 PAGES)
```