LCP01

LCP01 PRINTER DIAG
CZLCPA0

AH-U023A-MC
1 OF 1 JUL 1985

digital
MADE IN USA

```
\ A  W
A ;;CZLCPA COLOR PRINTER DIAGNOSTIC  MACRO M1113  12-MAR-85 15:38                                SEQ 0001

                                .REM
```

IDENTIFICATION

PRODUCT CODE: AC-U022A-MC

PRODUCT NAME: CZLCPA0 LCP01 PRTR DIAG

DATE : JANUARY 15, 1985

MAINTAINER: CSS PRODUCT GROUP DIAGNOSTIC

CZLCPA COLOR PRINTER DIAGNOSTIC MACRO M1113 12-MAR-85 15:38
TABLE OF CONTENTS

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
TABLE OF CONTENTS

CONTENTS

```
CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
ABSTRACT


  1.0  ABSTRACT


       THIS IS A PDP-11 DIAGNOSTIC WHICH IS DESIGNED TO FUNCTIONALLY EXERCISE
       AN LPC01 PRINTER WHILE ATTACHED TO A PDP-11 PROCESSOR.

       THE DIAGNOSTIC CAUSES THE MICROCODE TO EXECUTE SPECIFIC TESTS.  THE
       DIAGNOSTIC THEN MONITORS THE LCP01 SERIAL LINE OUTPUT, TRACING TEST
       COMPLETION AND ERROR INDICATIONS.

       CZLCP IS AN XXDP+ DIAGNOSTIC.


    1.1  MAINTENANCE HISTORY


      CZLCP IS A NEW PDP-11 DIAGNOSTIC.

       AUTHOR:          DIGITAL EQUIPMENT CORPORATION
                        COMPUTER SPECIAL SYSTEMS
                        HUDSON, NEW HAMPSHIRE

                        PREPARED BY:
                        DICE SYSTEMS, INC.
                        7 1/2 HARRIS ROAD
                        NASHUA, NEW HAMPSHIRE


    EDIT HISTORY:

                NEW VERSION
```

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
REQUIREMENTS

2.0 REQUIREMENTS

2.1 EQUIPMENT

THIS DIAGNOSTIC WILL RUN ON ALL PDP-11 FAMILY COMPUTERS WHICH HAVE EITHER A DL11 OR A DZ11 SERIAL LINE AND AN LCP01 PRINTER.

2.2 STORAGE

THIS PROGRAM REQUIRES A PDP-11 SYSTEM WITH AT LEAST 28K WORDS OF MEMORY.

2.3 SOFTWARE

THIS PROGRAM REQUIRES XXDP+ OPERATING SYSTEM SOFTWARE, VERSION 1.2 OR LATER.

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
TEST OVERVIEW

3.0 TEST OVERVIEW

3.1 SECTION DESCRIPTIONS

THIS DIAGNOSTIC CONSISTS OF ONE SECTION, CONTAINING THREE TESTS, AS FOLLOWS:

1. SERIAL LINE TEST
2. COLOR PRINTER SELF TEST
3. PRINTER DISPLAY TEST

3.2 SERIAL LINE OPERATIONS

THE PROGRAM HAS BEEN DESIGNED TO ALLOW THE DIAGNOSTIC TO TEST THE LCP01 PRINTERS WHILE ATTACHED BY SERIAL LINE INTERFACES.

THE FOLLOWING DIALOGUE ALLOWS FOR SERIAL LINE SELECTION:

```
R CZLCP??
CZLCP.BIN

LCP01 LINE PRINTER DIAGNOSTIC

SERIAL LINE SELECTION MENU
1  DL11 SERIAL LINE
2  DZ11 SERIAL LINE
TYPE MENU SELECTION <1>?
```

THE APPROPRIATE NUMBER (ONE OR TWO) SHOULD BE ENTERED BY THE OPERATOR. TYPING A CARRIAGE RETURN WILL RESULT IN SELECTION OF THE DEFAULT SERIAL LINE, WHICH IS THE DL11.

3.2.1 DL11 SUPPORT -

THE PROGRAM WILL SUPPORT DL11 OF THE FOLLOWING TYPE:

1. ALL UNIBUS DL11S
2. DLV11
3. DLV11-F

THE PROGRAM WILL PROMPT FOR INDIVIDUAL SETUP PARAMETERS AS THEY ARE REQUIRED BY EACH INTERFACE.

3.2.2 DZ11 SUPPORT -

THE PROGRAM WILL SUPPORT BOTH DZ11 AND THE DZV11. THE PROGRAM WILL PROMPT THE OPERATOR FOR THE REQUIRED SETUP PARAMETERS, THESE INCLUDE THE FOLLOWING:

1. CSR ADDRESS
2. THE DZ LINE NUMBER BEING USED (0-7).
3. THE DZ'S BAUD RATE
4. THE NUMBER OF STOP BITS
5. NUMBER OF DATA BITS
6. WHETHER OR NOT PARITY IS BEING USED
7. IF PARITY IS USED, IS IT ODD OR EVEN

4.0 ASSUMPTIONS

THE ONBOARD MICROCODE DIAGNOSTICS HAVE COMPLETE RESPONSIBILITY FOR DEVICE TEST COVERAGE. THE DIAGNOSTIC ONLY VERIFIES THE INTERFACE AND REPORTS ERRORS DETECTED BY THE MICROCODE DIAGNOSTICS.

5.0 OPERATING PROCEDURE

5.1 STARTING ADDRESS OR ADDRESSES

THE INITIAL STARTING ADDRESS TO RUN THE ENTIRE LCP01 DIAGNOSTIC IS LOCATION 200(8). THE RESTART ADDRESS IS 300 (8).

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
OPERATING PROCEDURE

5.2 OPERATIONAL SWITCH SETTINGS - HARDWARE AND SOFTWARE

WHEN THE DIAGNOSTIC IS STARTED AT ADDRESS 200(8),IT WILL DETERMINE WETHER OR NOT THE PROCESSOR HAS A HARDWARE (H/W) SWITCH REGISTER (SWR). IF THERE IS A H/W SWR, THE DIAGNOSTIC WILL USE BOTH THE HARDWARE REGISTER, AT LOCATION 177570, AND THE SOFTWARE (S/W) SWR LOCATED AT ADDRESS 176(8). IF A BIT IS SET IN EITHER REGISTER, EXECUTION WILL BE MODIFIED AS DESCRIBED IN THE 'SWITCH REGISTER BIT DEFINITIONS' SECTION OF THIS DOCUMENT.

5.2.1 CONTROL-G -

THE OPERATOR MAY CHANGE THESE REGISTER VALUES BY ENTERING A CONTROL-G AT THE CONSOLE TERMINAL. THE DIAGNOSTIC WILL PROMPT THE OPERATOR WITH THE MESSAGE:

H/W SWR = XXXXXX SWR = XXXXXX NEW SWR =

AFTER EXECUTION BEGINS, THE OPERATOR MAY CHANGE THE VALUE OF THE SWR, AT ANY TIME, BY ENTERING A CONTROL-G (G) AT THE CONSOLE.

IN RESPONSE TO THE PROMPTS, THE OPERATOR MAY ENTER UP TO SIX (6) OCTAL DIGITS. THE DIGITS MAY BE ANY COMBINATION OF :0,1,2,3,4,5,6,7,OR NO ENTRY AT ALL. ALL SWR VALUES ENTERED WILL BE TRUNCATED TO THE LOWER SIXTEEN (16) BITS. ENTERING MORE THAN SIX DIGITS OR A CHARACTER OTHER THAN A DIGIT RESULTS IN A "?" OUTPUT ON THE CONSOLE AND A REPEAT OF THE PROMPTING MESSAGE.

CARRIAGE RETURN (CR): ENTERS THE NEW SWR VALUE. IF NO DIGITS HAVE BEEN ENTERED, THE SWR VALUE REMAINS UNCHANGED.

5.2.2 CONTROL-U -

ERASES THE SWR VALUE BEING ENTERED. A CARRIAGE RETURN AND LINE FEED WILL BE OUTPUT AT THE CONSOLE. THE CORRECT SWR VALUE MAY THEN BE ENTERED.

5.2.3 CONTROL-H -

PRINTS THE HELP FILE ON THE CONSOLE TERMINAL, THE FOLLOWING INFORMATION IS DISPLAYED:

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
OPERATING PROCEDURE

HELP SWITCH REGISTER BIT DEFINITION

(NOTE: <CTRL>G - ALLOWS CHANGE TO "SOFTWARE" SWITCH REG)

```
15, 14, 13, 12...2, 1, 0
I   I   I   I    I  I  I--LOOP ON SLU TEST 1
I   I   I   I    I  I-----LOOP ON PRINTER "SELF" TEST
I   I   I   I    I--------LOOP ON PRINTER DISPLAY TEST
I   I   I   I-------------PAUSE ON ERROR, PAUSE AT END OF PASS
I   I   I-----------------INHIBIT ERROR REPORTS
I   I---------------------INHIBIT TEST ERROR AND END OF PASS REPORTS
I-------------------------LOOP ON ERROR (OTHERWISE CONTINUE)
```

ENTERING ANY CHARACTER BEFORE A CONTROL-G (G) HAS BEEN ENTERED WILL RESULT IN A "?" OUTPUT AT THE CONSOLE.

NOTE

IT IS POSSIBLE FOR THE DIAGNOSTIC TO OUTPUT MESSAGES AT THE CONSOLE BEFORE THE NEW SWR VALUE HAS BEEN ENTERED. SHOULD THIS HAPPEN, THE OPERATOR SHOULD ENTER A CONTROL-U (U) AND THEN ENTER THE CORRECT SWR VALUE.

5.2.4 CONTROL-C -

ENTRY OF A 'CONTROL-C' COMBINATION ABORTS TESTING AND RESTARTS IT AT LOCATION 200 (OCTAL).

5.2.5 SWITCH REGISTER BIT DEFINITIONS -

BIT0 =1: LOOP ON SLU TEST #1.

BIT1 =1: LOOP ON PRINTER SELF TEST #2

BIT2 =1: LOOP ON THE PRINTER DISPLAY TEST #3

BIT12=1: HALT ON ERROR AND HALT ON END OF PASS. THE OPERATOR MAY CHOOSE TO CONTINUE OR PROCEED BY ENTERINNG THOSE COMMANDS. THE DEFAULT EXECUTION IS TO CONTINUE AFTER AN ERROR OR AN EOP INDICATION IS ENCOUTERED.

BIT13=1: INHIBIT ERROR REPORTING.

BIT14=1: TEST HEADER AND END OF PASS MESSAGES ARE NOT DISPLAYED.

BIT15=1: WHEN AN ERROR IS ENCOUNTERED, LOOP ON ERROR, IF NO ERROR IS ENCOUNTERED EXECUTE TESTS CONTINUOUSLY.

```
345
346                         CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
347                         DEFAULT SECTION TEST DESCRIPTIONS
348
349                           6.0  TEST DESCRIPTIONS
350
351                             6.1  TEST 1: SERIAL LINE TEST
352
353                                  THIS TEST IS DESIGNED TO SHOW THE OPERATOR THAT THE SERIAL LINE SETUP
354                                  HAS BEEN COMPLETED CORRECTLY.  THIS WILL UNCOVER MOST SETUP ERRORS,
355                                  INCLUDING BASIC SERIAL LINE SETUP ERRORS.
356
357                                  FAILURE OF THIS TEST USUALLY SIGNIFIES THAT THE DIAGNOSTIC IS WORKING
358                                  WITH INCORRECT INFORMATION.  FOR INSTANCE, THE CSR ADDRESS SPECIFIED
359                                  MAY BE WRONG.  IF ALL OF THE PROGRAM INFORMATION WAS CORRECT, TEST
360                                  FAILURE INDICATES THAT THE SERIAL LINE DEVICE FAILED.
361
362                                  THE TEST SEQUENCE OPERATES IN MAINTENANCE MODE, WITH THE LOOPBACK
363                                  FEATURE SET.  ERROR NUMBERS 2 AND 3, LISTED BELOW, DEFINE ERRORS
364                                  ENCOUNTERED WHILE ATTEMPTING TO FLOAT A ONE THROUGH A FIELD OF ZEROS.
365                                  ERROR NUMBERS 5 AND 6 WILL BE DISPLAYED IF AN ERROR IS ENCOUNTERED
366                                  WHILE FLOATING A ZERO THROUGH THE ONES FIELD.  SERIAL LINE LOOPBACK
367                                  FAILURE, ERRORS 4 AND 7, WILL BE DISPLAYED IF THE BYTE RETURNED BY
368                                  LOOPBACK DOES NOT COMPARE WITH EXPECTED DATA.
369
370                                   ASSUMPTIONS:
371
372                                   FUNCTIONAL COMMUNICATION INTERFACE
373
374                                   TEST STEPS:
375
376                                         1. CHECK FOR SLU ADDRESS VALIDITY
377                                         2. TEST THE SLU IN 'LOOPBACK' MAINTENANCE MODE
378                                            IF PRESENCE DETETED APPROPRIATELY:
379                                                 A. PERFORM REGISTER TEST - FLOATING ONE BIT
380                                                    IN ZEROS FIELD
381                                                 B. PERFORM REGISTER TEST - FLOATING ZERO BIT
382                                                    IN ONES FIELD
383                                         3. INTERROGATE DEVICE - VERIFY PRESENCE AND DEVICE TYPE
384
385                                  ERRORS:
386                                  1. SERIAL LINE NOT AT THIS ADDRESS
387                                  ERROR NUMBER 0001
388
389                                  2. TIMEOUT WAITING FOR OUTPUT DONE
390                                  ERROR NUMBER 0002
391
392                                  3. TIMEOUT WAITING FOR INPUT RESPONSE
393                                  ERROR NUMBER 0003
394
395                                  4. SERIAL LINE 'LOOPBACK' FAILED
396                                  ERROR NUMBER 0004
397
```

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

5. TIMEOUT WAITING FOR OUTPUT DONE
ERROR NUMBER 0005

6. TIMEOUT WAITING FOR INPUT RESPONSE
ERROR NUMBER 0006

7. SERIAL LINE 'LOOPBACK' FAILED
ERROR NUMBER 0007

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

6.2 TEST 2: COLOR PRINTER SELF TEST

THIS TEST SEQUENCE EXECUTES THE COLOR PRINTER SELF TEST AND MONITORS ITS EXECUTION. IF AN UNEXPECTED STATUS RESPONSE IS ENCOUNTERED, IT IS REPORTED AS AN ERROR. ADDITIONALLY, TIMEOUT ERRORS, ERROR NUMBER 1 AND 3, WILL OCCUR IF THE PRINTER MICROCODE DOES NOT RESPOND WITHIN THE ALLOTTED TIME PERIOD.

ASSUMPTIONS:

FUNCTIONAL COMMUNICATION INTERFACE

TEST STEPS:

1. INITIATE SELF TEST EXECUTION (TRANSMIT ESCAPE SEQUENCE)
2. READ COLOR PRINTER STATUS
3. IF ERROR PRINT MESSAGE ELSE END OF TEST

ERRORS:

1. TIMEOUT WAITING FOR INPUT RESPONSE
   ERROR NUMBER 0010

2. UNEXPECTED RESPONSE TO 'POWER-UP' SELF TEST
   ERROR NUMBER 0011
   VERI
   CPU BAD
   000123 156743 023012 203457 143203 156427 012763 003450

3. TIMEOUT WAITING FOR INPUT RESPONSE
   ERROR NUMBER 0012

NOTE

THE LAST TWO LINES OF THE POWER-UP DIAGNOSTIC FAILURE MESSAGE (ITEM #2, ERROR #0011, ABOVE), WILL VARY ACCORDING TO THE VALUE RETURNED BY THE LCP01. IN ALL INSTANCES A COMPONENT INDICATION ("CPU BAD") AND A PORTION OF THE "VERIFIED" MESSAGE WILL BE DISPLAYED, ALONG WITH A REGISTER DUMP.

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

6.3 TEST 3: PRINTER DISPLAY TEST

THIS TEST IS DESIGNED TO PRINT A "CANNED" DISPLAY FILE. THIS FILE IS PRINTED WHEN THE DIAGNOSTIC SENDS A SPECIAL ESCAPE SEQUENCE TO THE LCP01'S CONTROLLER ("ESC", "[", "6", ";", "2", AND "Y"). THIS DISPLAY HAS BEEN TESTED AND WILL WORK UNLESS THERE IS A SOFTWARE SETUP ERROR OR A HARDWARE FAILURE. THE TEST PRINTS OUT A RADIANT DISPLAY.

TEST THREE IS ONLY EXECUTED ON THE FIRST TEST PASS. IF SUBSEQUENT TEST PASSES ARE SPECIFIED, IT IS NOT EXECUTED. SETTING BIT #2 IN THE SOFTWARE OR HARDWARE SWITCH REGISTER WILL CAUSE EXECUTION TO LOOP CONTINUOUSLY ON THIS TEST.

ASSUMPTIONS:

FUNCTIONAL COMMUNICATION INTERFACE
WORKING LCP01 SOFTWARE DRIVER
WORKING LCP01 PRINTER

TEST STEPS:

1. SEND ESCAPE SEQUENCE TO PRINTER TO REQUEST PRINT

ERRORS:
NO DISTINCT ERROR MESSAGES ARE REQUIRED BY THIS TEST. THE OPERATOR IS ASKED TO VERIFY THE INTEGRITY OF THE PRINTOUT.

NOTE

THE DIAGNOSTIC DOES NOT WAIT FOR PRINT COMPLETION BEFORE CONTINUING. THE PRINT PROCESS TAKES ABOUT TWO MINUTES.

```
505
506                                   CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
507                                   END OF DOCUMENT
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537                                                   END OF DOCUMENT
538
539
540
541
542
543
544
545
546
547
```

```
                        .LIST   SEQ,BIN,LOC
                        ;*****************************************************************;
                        ;                                                                 ;
                        .TITLE CZLCPA COLOR PRINTER DIAGNOSTIC                            ;
                        ;(DECSPEC-11-BDFAD-A-D)                                           ;
                        ;                                                                 ;
                        ; CZLCP-A-0                                                       ;
                        ; CZLCPA COLOR PRINTER DIAGNOSTIC                                 ;
                        ; UNIT IS TEX                                                     ;
                        ;                                                                 ;
                        ;COPYRIGHT (C) 1984 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.      ;
                        ;                                                                 ;
                        ;*****************************************************************;
000000                  R0=%0
000001                  R1=%1
000002                  R2=%2
000003                  R3=%3
000004                  R4=%4
000005                  R5=%5
000006                  R6=%6
000007                  R7=%7
000006                  SP=R6
000007                  PC=R7

100000                  BIT15   =100000
040000                  BIT14   =40000
020000                  BIT13   =20000
010000                  BIT12   =10000
004000                  BIT11   =4000
002000                  BIT10   =2000
001000                  BIT9    =1000
000400                  BIT8    =400
000200                  BIT7    =200
000100                  BIT6    =100
000040                  BIT5    =40
000020                  BIT4    =20
000010                  BIT3    =10
000004                  BIT2    =4
000002                  BIT1    =2
000001                  BIT0    =1

000036                  SL=36           ;START VFU LOAD
000037                  EL=37                   ;END VFU LOAD
```

```
                                ;;;***  .PSECT ABS
000000                                  .ENABLE AMA,ABS
                                        .DSABLE GBL

000000                          BEGIN:
        000000                          .=.+0


        000030                          .=BEGIN+30

000030  004332                         TYP
000032  000340                         340


        000042                          .=BEGIN+42

000042  000000                         0

        000046                          .=BEGIN+46
                                ;       LOGICAL
        000052                          .=BEGIN+52
000052  040000                         BIT14


        000060                          .=BEGIN+60
000060  004754                         TKINT                   ;KEYBOARD INTERRUPT ROUTINE
000062  000300                         300


        000100                          .=BEGIN+100

                                ;       LKSRV                   ;LINE CLOCK SERVICE ROUTINE
000100  000340                         340

                                ;       CONVRT
000102  000340                         340

        000174                          .=BEGIN+174
000174  000000                 DISPREG: 0
000176  000000                 SWREG:   0
```

```
                                .SBTTL PARAMETERS
                                ;*****************************************************************************
                                ;
                                ;  BEGINNING OF PROGRAM-
                                ;
                                        START::
                                        .=BEGIN+200
                                ;
                                ;       THE FOLLOWING INSTRUCTION "MOV #1000,%6" IS OVER LAID BY THE LP11
                                ;       INTERRUPT TESTS. DON'T CHANGE IT WITH OUT BEING AWARE OF THE
                                ;       PROBLEMS THAT MAY OCCUR.
                                ;*****************************************************************************
000200  012706  001000          MOV     #1000,%6
                                ;*****************************************************************************
000204  000137  001306          10$:    JMP     SETUP
                                
                000300          .=BEGIN+300
000300  000137  001464          JMP     TEST1
                                
                001000          .=BEGIN+1000
                                
                                ;LINE PRINTER HARDWARE REGISTERS
                                
001000  177514                  LPS:    177514          ;STATUS REGISTER
                                                        ;BIT 15=ERROR
                                                        ;BIT 7=READY
                                                        ;BIT 6=INTERRUPT ENABLE
                                
001002  177516                  LPB:    177516          ;DATA BUFFER REGISTER
                                                        ;BITS 0-6=7 BIT ASCII CHARACTER BUFFER
                                                        ;BITS 7-15=NOT USED
                                
001004  177570                  SWR:    177570
001006  177570                  DISPLAY:177570
001010  177776                  PSW:    177776
001012  177570                  HWSWR:  177570
001014  177566                  TPB:    177566
001016  177562                  TKB:    177562
001020  177564                  TPS:    177564
001022  177560                  TKS:    177560
001024  172542                  CSBR:   172542
001026  172540                  PLKS:   172540
001030  177546                  LKS:    177546
                                
001032  176500                  DLCSRC: .WORD   176500          ;DL'S DEFAULT CSR
001034  176500                  DLCSR:  .WORD   176500          ;DL'S DEFAULT CSR
001036  160100                  DZCSR:  .WORD   160100          ;DZ'S DEFAULT CSR
001040  160100                  DZCSRC: .WORD   160100          ;DZ'S DEFAULT CSR
001042  000300                  DLVEC:  .WORD   300             ;DL'S DEFAULT VECTOR ADDRESS
001044  000300                  DZVEC:  .WORD   300             ;DZ'S DEFAULT VECTOR ADDRESS
001046  000200                  PTRC:   .WORD   200             ;LP'S DEFAULT VECTOR ADDRESS
001050  000200                  PTRVEC: .WORD   200
001052  000202                  PTRPSW: .WORD   202
                000240          NOP     =240
                000001          N       =1
                000002          M       =2
                000001          W       =1
```

```
                        .SBTTL MACROS
                        ;******************************************************************
                        ;
                        ;     ---MACROS---
                        ;
                            ;MACRO FOR SETTING UP ERROR COUNT
                            ;
                            ; $ERROR = CALL
                            ;       X = ERROR NUMBER
                            ;       Y = LOOP ADDRESS IF SWR BIT SET
                            .LIST ME

                            .MACRO  $ERROR X,Y
                        ERR'X': MOV     #X,     ERCOUNT         ;SET UP ERROR COUNT X
                        .NLIST  ME
                            N=N+1
                        .NLIST
                        .LIST   ME
                        .LIST
                            JSR     %5,STAER                ;REPORT ERROR SET
                        .NLIST  ME
                            .IF     NB Y
                        .NLIST
                        .LIST   ME
                        .LIST

                            $TSWRG  #100000         ;CK SW REG
                            BEQ     CN'X'                   ;CONTINUE IF BIT 15 = 0
                            JMP     Y                       ;OTHERWISE LOOP
                        CN'X':
                        .NLIST  ME
                            .ENDC
                        .NLIST
                        .LIST   ME
                        .LIST
                            .ENDM   $ERROR


                            ;MACRO FOR PRINTING TEST NUMBER AT START OF TEST


                            .MACRO  $PRTSN  Y
                            EMT     +0                      ;PRINT TEST NUMBER
                            TYO'Y'                          ;TEST NUMBER MESSAGE
                            .ENDM   $PRTSN
                        ;
                        ;  WAIT MACROS - -
                        ;       Z = ERROR # (USUALLY "W")
                        ;       X = ERROR JMP ADDRESS, IF NONE - DEFAULT LOOPS BACK TO WAIT'Z'
                        ;       Y = LOOP ON ERROR, IF BIT 15 OF SWR SET
                        ;       #T = 2ND PASS TIME (1 COUNT = ABOUT 1/2 SEC), IF NONE = 100 OCTAL

                            .MACRO  $WAITI  Z,X,Y,T
                                                    ;WAIT FOR INPUT, OR TIMEOUT
                        WAIT'Z':
                        .NLIST  ME
                            .IF     NB T
```

```
                        .NLIST
                        .LIST   ME
                        .LIST
                                MOV     #100,WORK       ;LOAD CNTR
                                CMP     #1,CYCCNT       ;2ND PASS?
                                BNE     WDD'Z'          ;NO, LEAVE COUNT AS IS
                                MOV     T,WORK          ;LOAD CNTR WITH LONG CNT
                        .NLIST  ME
                                .ENDC
                                .IF     B T
                        .NLIST
                        .LIST   ME
                        .LIST
                                MOV     #100,WORK       ;LOAD CNTR
                        .NLIST  ME
                                .ENDC
                        .NLIST
                        .LIST   ME
                        .LIST
                        WDD'Z': MOV     #-1,WORKA               ;LOAD COUNT
                                DEC     WORK                    ;BUMP CNTR
                                BEQ     WER'Z'                  ;ERROR IF TIMEOUT
                        WDE'Z': DEC     WORKA                   ;BUMP COUNT
                                BEQ     WDD'Z'                  ;LOOP IF COUNT 0
                                TSTB    @DLLPS                  ;CK STATUS
                                BPL     WDE'Z'                  ;LOOP TIL DONE
                                BR      WEX'Z'                  ;CONTINUE
                        WER'Z': $TSWRG  #20000          ;CK SW REG
                                BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
                                                        ;BRANCH IF NO MSG WANTED
                                EMT     +0
                                ETIM                            ;TIMED OUT ERROR
                        .NLIST  ME
                                .IF     NB Y
                        .NLIST
                        .LIST   ME
                        .LIST

                                $ERROR  \N,Y
                        .NLIST  ME
                                .ENDC
                                .IF     B Y
                        .NLIST
                        .LIST   ME
                        .LIST

                                $ERROR  \N
                        .NLIST  ME
                                .ENDC
                                .IF     B X
                        .NLIST
                        .LIST   ME
                        .LIST

                                BR      WAIT'Z'                 ;LOOP
                        .NLIST  ME
```

```
                        .ENDC
                        .IF     NB X
                .NLIST
                .LIST   ME
                .LIST

                        JMP     X                       ;EXIT
                .NLIST  ME
                        .ENDC
                .NLIST
                .LIST   ME
                .LIST

                WEX'Z':
                .NLIST  ME
                        W=W+1
                .NLIST
                .LIST   ME
                .LIST
                        .ENDM   $WAITI

                        ;MACRO FOR WAITING FOR OUPUT, DONE

                        .MACRO  $WAITO  Z,X,Y
                WAIT'Z': MOV    #20,TIME                ;WAIT FOR OUPUT, OR TIMEOUT
                WDD'Z': MOV     #-1,TIMER                       ;10 SEC CNTR
                        DEC     TIME                            ;LOAD COUNT
                        BEQ     WER'Z'                          ;BUMP CNTR
                WDE'Z': DEC     TIMER                           ;ERROR IF TIMEOUT
                        BEQ     WDD'Z'                          ;BUMP COUNT
                        TSTB    @LPS                            ;LOOP IF COUNT 0
                        BPL     WDE'Z'                          ;CK STATUS
                        BR      WEX'Z'                          ;LOOP TIL DONE
                WER'Z': $TSWRG  #20000                          ;CONTINUE
                                                        ;CK SW REG
                        BNE     .+6                     ;TEST FOR INHIBIT ERROR MSG
                        EMT     +0                      ;BRANCH IF NO MSG WANTED
                        ETIMO
                .NLIST  ME                                      ;TIMED OUT ERROR
                        .IF     NB Y
                .NLIST
                .LIST   ME
                .LIST

                        $ERROR  \N,Y
                .NLIST  ME
                        .ENDC
                        .IF     B Y
                .NLIST
                .LIST   ME
                .LIST

                        $ERROR  \N
                .NLIST  ME
```

```
                        .ENDC
                        .IF     B X
                .NLIST
                .LIST   ME
                .LIST

                        BR      WAIT'Z'                 ;LOOP
                .NLIST  ME
                        .ENDC
                        .IF     NB X
                .NLIST
                .LIST   ME
                .LIST

                        JMP     X                       ;EXIT
                .NLIST  ME
                        .ENDC
                .NLIST
                .LIST   ME
                .LIST

                WEX'Z':
                .NLIST  ME
                        W=W+1
                .NLIST
                .LIST   ME
                .LIST
                        .ENDM   $WAITO

                        ;
                        ;MACRO FOR ENABLING KEYBOARD INTERUPT
                        ;
                        .MACRO  $ENABLE
        .NLIST  ME
        ;;;@@@   CMP     #176,SWR                ;S/W SWR ?
        ;;;@@@   BNE     .+10                    ;NO- CONTINUE
        .NLIST
        .LIST   ME
        .LIST
                BIS     #100,@TKS               ;ENABLE KEYBOARD INTERRUPT
                MOV     -(SP),-(SP)
                MOV     #0,2(SP)
                MOV     #.+6,(SP)
                RTI
                .ENDM   $ENABLE

                ;MACRO USED TO LOAD THE PSW WITH THE
                ;CORRECT PROCESSOR PRIORITY LEVEL
                ;
                .MACRO  $SETPSW
                MOV     PC,-(SP)        ;MOVE PRESENT LOCATION TO STACK
                ADD     #6,(SP)         ;SET UP FOR NEXT INSTRUCTION
                RTI                     ;LOAD PSW
                .ENDM   $SETPSW

                ;
                ;MACRO USED TO PRINT MESSAGE TO LINE PRINTER
```

```
                        ;
                        .MACRO  $PRINT  V
                        MOV     #'V',PRTMSG     ;LOAD MESSAGE ADDRESS
                        JSR     %4,PRINE        ;PRINT IT
                        .ENDM   $PRINT

                        ;
                        ; MACRO TO TYPE MESSAGE ON THE TERMINAL
                        ;
                        .MACRO  $TYPE   G
                        EMT     +0              ;Call "TYP" interrupt
                        G                       ;address of message
                        .ENDM   $TYPE

                        ;
                        ; MACRO TO TEST HARDWARE AND SOFTWARE SWITCH REGISTERS
                        ;
                        .MACRO  $TSWRG  MSK
                        CMP     #176,SWR        ;HW SWITCH REG THERE?
                        BEQ     .+12            ;NO, SKP HW CHECK
                        BIT     MSK,@HWSWR      ;YES, CK HW REG
                        BNE     .+10            ;IF SET, SKP SW REG CK
                        BIT     MSK,SWREG       ;OTHERWISE, ALSO CK SW REG
                        .ENDM
```

```
                                .SBTTL VARIABLES
                                ;*****************************************************************
                                ;
                                ;MEMORY LOCATIONS USED AS PROGRAM FLAGS AND COUNTERS
                                ;
001054  000000                  SEGCNT: 0
001056  000000                  CHRCNT: 0
001060  000000                  CHRGEN: 0
001062  000000                  LINCNT: 0
001064  000000                  CYCCNT: 0

001066  000000                  DZTCR:  .WORD   0        ;HOLDS DZ'S TCR REGISTER
001070  000000                  DZLPR:  .WORD   0        ;HOLDS DZ'S LPR REGISTER
001072  000000                  DZLNE:  .WORD   0        ;HOLDS DZ'S LINE #

001074  000000                  DLLPR:  .WORD   0        ;HOLDS DL'S BAUD RATE BITS
001076  000000                  DLLPS:  .WORD   0        ;DL'S RECV REG
001100  000000                  DLRBUF: .WORD   0        ;DL'S RECV BUFFER
001102  000000                  BRATE:  .WORD   0        ;HOLD BAUD RATE BITS
001104  000000                  DLRATE: .WORD   0        ; (DL) HOLD BAUD RATE BITS
001106  000000                  DLHERE: .WORD   0        ;SHOWS DL11 PRESENCE

001110  000000                  DZTCRA: .WORD   0        ;ADDRESS OF DZ DTR REGISTER
001112  000000                  EIA:    .WORD   0        ;LINE TYPE 0=20MA  1=EIA

001114  000000                  DLTYPE: .WORD   0        ;DL TYPE 0=ALL OTHERS 1=E/F
001116  000000                  DZRBUF: .WORD   0        ;DZ'S RECEIVER BUFFER ADDRESS
001120  000000                  DZCSRH: .WORD   0        ;HOLDS DZ'S CSR ADDRESS
001122  000000                  MAINTB: .WORD   0        ;MAINTENANCE BIT, IF NOT 0

001124  000000                  WORK:   0
001126  000000                  WORKA:  0
001130  000000                  TIME:   0
001132  000000                  TIMER:  0
001134  000000                  SAVE:   0
001136  000000                  ERCOUNT:0
001140  000000                  STRCHR: 0
001142  000000                  STRCNT: 0
001144  000000                  LEGCHR: 0
001146  000000                  NUMCHR: 0
001150  000000                  OFFSET: 0
001152  000000                  DIGITS: 0
001154  000000                  SIGNAL: 0
001156  000000                  SET:    0
001160  000000                  CHAR:   0
001162  000000                  OCT:    0
001164  000000                  PASSA:  0
001166                          BUFF:   .BLKB   80.
```

```
                                        .SBTTL SETUP SERIAL LINE PARAMETERS
                                        ;*****************************************************************
                                        ;
                                        ;ROUTINE TO TEST THE MECH. OPERATION OF THE LXY11/21-LXV11
                                        ;
001306  004437  004314          SETUP:  JSR     #4,TYPINT       ;PRESET POINTERS
001312  005037  001064                  CLR     CYCCNT          ;CLEAR PASS COUNT
001316  005037  001154                  CLR     SIGNAL          ;CTRL G FLAG
001322  000005                          RESET                   ;CLEAR WORLD
001324  013746  000004                  MOV     4,-(SP)         ;SAVE CURRENT VECTORS
001330  013746  000006                  MOV     6,-(SP)         ;
001334  012737  001350  000004         MOV     #1$,4           ;SET UP TIMEOUT VECTOR
001342  005777  177444                  TST     @HWSWR          ;TRY TO ACCESS HARDWARE SWR
001346  000407                          BR      2$              ;IF THERE, GO TO 2$
001350                          1$:
001350  012737  000176  001004         MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWR
001356  012737  000174  001006         MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY
001364  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
001366  012637  000006          2$:     MOV     (SP)+,6         ;RESTORE TIMEOUT VECTORS
001372  012637  000004                  MOV     (SP)+,4         ;
001376                                  $ENABLE
001376  052777  000100  177416         BIS     #100,@TKS               ;ENABLE KEYBOARD INTERRUPT
001404  014646                          MOV     -(SP),-(SP)
001406  012766  000000  000002         MOV     #0,2(SP)
001414  012716  001422                  MOV     #.+6,(SP)
001420  000002                          RTI
001422  005737  001164                  TST     PASSA
001426  001011                          BNE     3$              ;SKP
001430  112737  000001  001164         MOVB    #1,PASSA
001436  104000                          EMT     +0
001440  006153                          MES1                    ;TYPE DIAGNOSTIC TITLE
001442  104000                          EMT     +0
001444  006167                          MES2                    ;TYPE NAME
001446  104000                          EMT     +0
001450  006230                          MES3                    ;
001452  004737  010234          3$:     JSR     #7,SETSER       ;CHECK FOR SERIAL LINE SELECTION
001456  000005                          RESET                   ;REQUIRED INSURANCE
001460  104000                          EMT     +0
001462  006246                          MES4                    ;TYPE RESTART ADDRESS INFO
```

```
                                        .SBTTL
                                        .SBTTL TEST #   DESCRIPTION
                                        .SBTTL ------   -----------
                                        .SBTTL TEST 1   SERIAL LINE UNIT TEST
                                        ;*******************************************************************************
                                        ;
                                        ;               Test 1  Serial Line Unit test
                                        ;
                                        ;   Test 1-A
                                        ;
                                        ;       - Insure SLU is there, at correct address.
                                        ;
                                        ;   Test 1-B
                                        ;
                                        ;       - If Loopback feature (echo) is present, Test the SLU:
                                        ;
                                        ;               a - with floating one bit field
                                        ;               b - with floating zero bit field
                                        ;
                                        ;
                                        ;
                                        ;   CHECK FOR PRINTER (SLU) ON BUS
                                        ;
                                            .ENABL LSB
001464  052777  000100  177330  TEST1:    BIS     #100,@TKS       ;ENABLE KEYBOARD
001472                                    $TSWRG  #40000          ;CK SW REG
001472  022737  000176  001004            CMP     #176,SWR        ;HW SWITCH REG THERE?
001500  001404                            BEQ     .+12            ;NO, SKP HW CHECK
001502  032777  040000  177302            BIT     #40000,@HWSWR   ;YES, CK HW REG
001510  001003                            BNE     .+10            ;IF SET, SKP SW REG CK
001512  032737  040000  000176            BIT     #40000,SWREG    ;OTHERWISE, ALSO CK SW REG
001520  001002                            BNE     1$              ;IF SET, SKIP TST # HDR MSG
001522                                    $PRTSN  1
001522  104000                            EMT     .0
001524  005571                            TYO1                            ;PRINT TEST NUMBER
                                                                            ;TEST NUMBER MESSAGE
001526  013746  000004            1$:     MOV     @4,-(SP)        ;SAVE VECTORS
001532  013746  000006                    MOV     @6,-(SP)        ;SAVE
001536  012737  001552  000004            MOV     #2$,4           ;RELOAD VECTOR
001544  105777  177230                    TSTB    @LPS            ;IS PRINTER THERE?
001550  000447                            BR      4$              ;YES, SKIP TRAP PROCESSING
001552                            2$:
001552  022626                            CMP     (SP)+,(SP)+     ;IF YES, NEVER GET HERE.
001554  012637  000006                    MOV     (SP)+,6         ;RESTORE STACK
001560  012637  000004                    MOV     (SP)+,4         ;RESTORE VECTORS
001564                                    $TSWRG  #20000          ;RESTORE
001564  022737  000176  001004            CMP     #176,SWR        ;CK SW REG
001572  001404                            BEQ     .+12            ;HW SWITCH REG THERE?
001574  032777  020000  177210            BIT     #20000,@HWSWR   ;NO, SKP HW CHECK
001602  001003                            BNE     .+10            ;YES, CK HW REG
001604  032737  020000  000176            BIT     #20000,SWREG    ;IF SET, SKP SW REG CK
001612  001002                            BNE     .+6             ;OTHERWISE, ALSO CK SW REG
001614  104000                            EMT     .0              ;IF INHIBIT ERR MSG
001616  006704                            ERMS1                   .
001620                                    $ERROR  \N,1$           ;SLU NOT THERE
001620  012737  000001  001136  ERR1:     MOV     #1,     ERCOUNT ;NOTHING THERE
                                                                            ;SET UP ERROR COUNT 1
```

```
        001626  004537  004606                 JSR     @#5,STAER               ;REPORT ERROR SET

        001632
        001632  022737  000176  001004         $TSWRG  #100000         ;CK SW REG
        001640  001404                         CMP     #176,SWR        ;HW SWITCH REG THERE?
        001642  032777  100000  177142         BEQ     .+12            ;NO, SKP HW CHECK
        001650  001003                         BIT     #100000,@HWSWR  ;YES, CK HW REG
        001652  032737  100000  000176         BNE     .+10            ;IF SET, SKP SW REG CK
        001660  001402                         BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
        001662  000137  001526                 BEQ     CN1                     ;CONTINUE IF BIT 15 = 0
        001666                                 JMP     1$                      ;OTHERWISE LOOP
                                       CN1:
  1086
  1087  001666  000607                         BR      SETUP
  1088                                 ;
  1089                                 ;  YES, SLU IS HERE
  1090                                 ;
  1091  001670  012637  000006        4$:      MOV     (SP)+,6         ;RESTORE VECTORS
  1092  001674  012637  000004                 MOV     (SP)+,4         ;RESTORE
```

```
                                        ;*******************************************************************************
                                        ;
                                        ;       Test 1-B
                                        ;
                                        ;       -  If Loopback feature (echo) is present, Test the SLU:
                                        ;
                                        ;               a - with floating one bit field
                                        ;               b - with floating zero bit field
                                        ;
001700  005737  001122                  TST     MAINTB          ;NONZERO IF LOOPBACK FEATURE PRESENT
001704  001002                          BNE     9$              ;HERE, CONTINUE
001706  000137  003016                  JMP     40$             ;NOT HERE, SKIP SUBTEST

                                        ;-------------------------------------------------------------
                                        ;
                                        ;       Test 1-B-a
                                        ;
                                        ;  FLOAT A ONE BIT THRU ALL ZERO BYTE
                                        ;
001712  053777  001122  177060  9$:     BIS     MAINTB,@LPS     ;SET MAINTENANCE (LOOPBACK) BIT
001720  012700  000001                  MOV     #1,R0           ;SET BIT #0
001724  010037  001134          10$:    MOV     R0,SAVE         ;SAVE IT

001730                          12$:    $WAITO  \W,,9$                  ;WAIT FOR OUPUT, OR TIMEOUT
001730  012737  000020  001130  WAIT1: MOV     #20,TIME                ;10 SEC CNTR
001736  012737  177777  001132  WDD1:   MOV     #-1,TIMER               ;LOAD COUNT
001744  005337  001130                  DEC     TIME                    ;BUMP CNTR
001750  001407                          BEQ     WER1                    ;ERROR IF TIMEOUT
001752  005337  001132          WDE1:   DEC     TIMER                   ;BUMP COUNT
001756  001767                          BEQ     WDD1                    ;LOOP IF COUNT 0
001760  105777  177014                  TSTB    @LPS                    ;CK STATUS
001764  100372                          BPL     WDE1                    ;LOOP TIL DONE
001766  000442                          BR      WEX1                    ;CONTINUE
001770                          WER1:   $TSWRG  #20000          ;CK SW REG
001770  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
001776  001404                          BEQ     .+12            ;NO, SKP HW CHECK
002000  032777  020000  177004          BIT     #20000,@HWSWR   ;YES, CK HW REG
002006  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
002010  032737  020000  000176          BIT     #20000,SWREG    ;OTHERWISE, ALSO CK SW REG

002016  001002                          BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
002020  104000                          EMT     +0              ;BRANCH IF NO MSG WANTED
002022  006007                          ETIMO                           ;TIMED OUT ERROR

002024                          $ERROR  \N,9$
002024  012737  000002  001136  ERR2:   MOV     #2,     ERCOUNT         ;SET UP ERROR COUNT 2
002032  004537  004606                  JSR     %5,STAER                ;REPORT ERROR SET

002036                          $TSWRG  #100000         ;CK SW REG
002036  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
002044  001404                          BEQ     .+12            ;NO, SKP HW CHECK
002046  032777  100000  176736          BIT     #100000,@HWSWR  ;YES, CK HW REG
002054  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
002056  032737  100000  000176          BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
002064  001402                          BEQ     CN2             ;CONTINUE IF BIT 15 = 0
002066  000137  001712                  JMP     9$              ;OTHERWISE LOOP
```

```
                002072
                002072  000716                  CN2:
                                                        BR      WAIT1                   ;LOOP
                002074
                002074  110077  176702          WEX1:
                                                        MOVB    R0,@LPB         ;XMIT BYTE
                002100
                002100                          14$:    $WAITI  \W,,9$
                002100                                                  ;WAIT FOR INPUT, OR TIMEOUT
                002100  012737  000100  001124  WAIT2:
                                                        MOV     #100,WORK       ;LOAD CNTR
                002106  012737  177777  001126  WDD2:   MOV     #-1,WORKA               ;LOAD COUNT
                002114  005337  001124                  DEC     WORK                    ;BUMP CNTR
                002120  001407                          BEQ     WER2                    ;ERROR IF TIMEOUT
                002122  005337  001126          WDE2:   DEC     WORKA                   ;BUMP COUNT
                002126  001767                          BEQ     WDD2                    ;LOOP IF COUNT 0
                002130  105777  176742                  TSTB    @DLLPS                  ;CK STATUS
                002134  100372                          BPL     WDE2                    ;LOOP TIL DONE
                002136  000442                          BR      WEX2                    ;CONTINUE
                002140                          WER2:   $TSWRG  #20000          ;CK SW REG
                002140  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
                002146  001404                          BEQ     .+12            ;NO, SKP HW CHECK
                002150  032777  020000  176634          BIT     #20000,@HWSWR   ;YES, CK HW REG
                002156  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
                002160  032737  020000  000176          BIT     #20000,SWREG    ;OTHERWISE, ALSO CK SW REG
                002166  001002                          BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
                                                                                ;BRANCH IF NO MSG WANTED
                002170  104000                          EMT     +0
                002172  005731                          ETIM                            ;TIMED OUT ERROR
                002174
                002174  012737  000003  001136  ERR3:   $ERROR  \N,9$
                002202  004537  004606                  MOV     #3,     ERCOUNT         ;SET UP ERROR COUNT 3
                                                        JSR     %5,STAER                ;REPORT ERROR SET
                002206
                002206  022737  000176  001004          $TSWRG  #100000         ;CK SW REG
                002214  001404                          CMP     #176,SWR        ;HW SWITCH REG THERE?
                002216  032777  100000  176566          BEQ     .+12            ;NO, SKP HW CHECK
                002224  001003                          BIT     #100000,@HWSWR  ;YES, CK HW REG
                002226  032737  100000  000176          BNE     .+10            ;IF SET, SKP SW REG CK
                002234  001402                          BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
                002236  000137  001712                  BEQ     CN3                     ;CONTINUE IF BIT 15 = 0
                002242                                  JMP     9$                      ;OTHERWISE LOOP
                                                CN3:
                002242  000716                          BR      WAIT2                   ;LOOP
                002244                          WEX2:
1121
1122 002244  117701  176630                          MOVB    @DLRBUF,R1      ;GET BYTE IN
1123 002250  120001                                  CMPB    R0,R1           ;SAME?
1124 002252  001431                                  BEQ     16$             ;OK
1125
1126 002254  104000                                  EMT     +0
1127 002256  006756                                  ERMS2                   ;"LOOPBACK FAILED" MSG
1128 002260                                          $ERROR  \N,9$           ;LOOPBACK FAILED
                002260  012737  000004  001136  ERR4:   MOV     #4,     ERCOUNT         ;SET UP ERROR COUNT 4
```

```
002266  004537  004606                          JSR     @#5,STAER               ;REPORT ERROR SET

002272
002272  022737  000176  001004                  $TSWRG  #100000         ;CK SW REG
002300  001404                                  CMP     #176,SWR        ;HW SWITCH REG THERE?
002302  032777  100000  176502                  BEQ     .+12            ;NO, SKP HW CHECK
002310  001003                                  BIT     #100000,@HWSWR  ;YES, CK HW REG
002312  032737  100000  000176                  BNE     .+10            ;IF SET, SKP SW REG CK
002320  001402                                  BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
002322  000137  001712                          BEQ     CN4                     ;CONTINUE IF BIT 15 = 0
002326                                          JMP     9$                      ;OTHERWISE LOOP
1129  002326  013700  001134                CN4:
1130  002332  000137  001730                  MOV     SAVE,R0         ;RESTORE, R0
1131                                          JMP     12$             ;TRY AGAIN
1132  002336  000241                  16$:    CLC                     ;CLEAR CARRY
1133  002340  006100                          ROL     R0              ;SHIFT BIT
1134  002342  105700                          TSTB    R0
1135  002344  100402                          BMI     .+6
1136  002346  000137  001724                  JMP     10$             ;NOT FINISHED
1137
1138                                  ;-------------------------------------------------------
1139                                  ;
1140                                  ;       Test 1-B-b
1141                                  ;
1142                                  ;  FLOAT A ZERO THRU ALL ONES BYTE
1143                                  ;
1144  002352  012700  000176                  MOV     #176,R0         ;SET ALL EXCEPT BIT #0
1145  002356  142700  000200          20$:    BICB    #200,R0         ;CLR BIT #7
1146  002362  010037  001134                  MOV     R0,SAVE         ;SAVE IT
1147
1148  002366                          22$:    $WAITO  \W,,20$

002366  012737  000020  001130  WAIT3:  MOV     #20,TIME                ;WAIT FOR OUPUT, OR TIMEOUT
002374  012737  177777  001132  WDD3:   MOV     #-1,TIMER               ;10 SEC CNTR
002402  005337  001130                  DEC     TIME                    ;LOAD COUNT
002406  001407                          BEQ     WER3                    ;BUMP CNTR
002410  005337  001132          WDE3:   DEC     TIMER                   ;ERROR IF TIMEOUT
002414  001767                          BEQ     WDD3                    ;BUMP COUNT
002416  105777  176356                  TSTB    @LPS                    ;LOOP IF COUNT 0
002422  100372                          BPL     WDE3                    ;CK STATUS
002424  000442                          BR      WEX3                    ;LOOP TIL DONE
002426                          WER3:   $TSWRG  #20000                  ;CONTINUE
002426  022737  000176  001004          CMP     #176,SWR        ;CK SW REG
002434  001404                          BEQ     .+12            ;HW SWITCH REG THERE?
002436  032777  020000  176346          BIT     #20000,@HWSWR   ;NO, SKP HW CHECK
002444  001003                          BNE     .+10            ;YES, CK HW REG
002446  032737  020000  000176          BIT     #20000,SWREG    ;IF SET, SKP SW REG CK
                                                                ;OTHERWISE, ALSO CK SW REG
002454  001002                          BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
002456  104000                          EMT     +0              ;BRANCH IF NO MSG WANTED
002460  006007                          ETIMO                           ;TIMED OUT ERROR

002462
002462  012737  000005  001136  ERR5:   $ERROR  \N,20$
002470  004537  004606                  MOV     #5,     ERCOUNT         ;SET UP ERROR COUNT 5
                                        JSR     @#5,STAER               ;REPORT ERROR SET
002474
                                        $TSWRG  #100000         ;CK SW REG
```

```
                002474  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
                002502  001404                          BEQ     .+12            ;NO, SKP HW CHECK
                002504  032777  100000  176300          BIT     #100000,@HWSWR  ;YES, CK HW REG
                002512  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
                002514  032737  100000  000176          BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
                002522  001402                          BEQ     CN5             ;CONTINUE IF BIT 15 = 0
                002524  000137  002356                  JMP     20$             ;OTHERWISE LOOP
                002530                          CN5:

                002530  000716                          BR      WAIT3                   ;LOOP

                002532                          WEX3:
           1149 002532  110077  176244                  MOVB    R0,@LPB         ;XMIT BYTE
           1150
           1151 002536                          24$:    $WAITI  \W,,20$

                002536                          WAIT4:                          ;WAIT FOR INPUT, OR TIMEOUT
                002536  012737  000100  001124          MOV     #100,WORK       ;LOAD CNTR
                002544  012737  177777  001126  WDD4:   MOV     #-1,WORKA               ;LOAD COUNT
                002552  005337  001124                  DEC     WORK                    ;BUMP CNTR
                002556  001407                          BEQ     WER4                    ;ERROR IF TIMEOUT
                002560  005337  001126          WDE4:   DEC     WORKA                   ;BUMP COUNT
                002564  001767                          BEQ     WDD4                    ;LOOP IF COUNT 0
                002566  105777  176304                  TSTB    @DLLPS                  ;CK STATUS
                002572  100372                          BPL     WDE4                    ;LOOP TIL DONE
                002574  000442                          BR      WEX4                    ;CONTINUE
                002576                          WER4:   $TSWRG  #20000          ;CK SW REG
                002576  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
                002604  001404                          BEQ     .+12            ;NO, SKP HW CHECK
                002606  032777  020000  176176          BIT     #20000,@HWSWR   ;YES, CK HW REG
                002614  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
                002616  032737  020000  000176          BIT     #20000,SWREG    ;OTHERWISE, ALSO CK SW REG
                002624  001002                          BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
                                                                                ;BRANCH IF NO MSG WANTED
                002626  104000                          EMT     +0
                002630  005731                          ETIM                            ;TIMED OUT ERROR

                002632
                002632  012737  000006  001136  ERR6:   $ERROR  \N,20$
                002640  004537  004606                  MOV     #6,     ERCOUNT         ;SET UP ERROR COUNT 6
                                                        JSR     #5,STAER                ;REPORT ERROR SET

                002644                          $TSWRG  #100000         ;CK SW REG
                002644  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
                002652  001404                          BEQ     .+12            ;NO, SKP HW CHECK
                002654  032777  100000  176130          BIT     #100000,@HWSWR  ;YES, CK HW REG
                002662  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
                002664  032737  100000  000176          BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
                002672  001402                          BEQ     CN6             ;CONTINUE IF BIT 15 = 0
                002674  000137  002356                  JMP     20$             ;OTHERWISE LOOP
                002700                          CN6:

                002700  000716                          BR      WAIT4                   ;LOOP

                002702                          WEX4:
           1152
           1153 002702  117701  176172                  MOVB    @DLRBUF,R1      ;GET BYTE IN
```

```
1154 002706  120001                        CMPB    R0,R1           ;SAME?
1155 002710  001431                        BEQ     26$             ;OK
1156
1157 002712  104000                        EMT     +0
1158 002714  006756                        ERMS2                   ;"LOOPBACK FAILED" MSG
1159 002716                                $ERROR  \N,20$          ;LOOPBACK FAILED
     002716  012737  000007  001136  ERR7:  MOV     #7,     ERCOUNT         ;SET UP ERROR COUNT 7
     002724  004537  004606                JSR     %5,STAER                ;REPORT ERROR SET

     002730
     002730  022737  000176  001004        $TSWRG  #100000         ;CK SW REG
     002736  001404                        CMP     #176,SWR        ;HW SWITCH REG THERE?
     002740  032777  100000  176044        BEQ     .+12            ;NO, SKP HW CHECK
     002746  001003                        BIT     #100000,@HWSWR  ;YES, CK HW REG
     002750  032737  100000  000176        BNE     .+10            ;IF SET, SKP SW REG CK
     002756  001402                        BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
     002760  000137  002356                BEQ     CN7             ;CONTINUE IF BIT 15 = 0
     002764                                JMP     20$             ;OTHERWISE LOOP
                                    CN7:
1160 002764  013700  001134                MOV     SAVE,R0         ;RESTORE, R0
1161 002770  000137  002366                JMP     22$             ;TRY AGAIN
1162
1163 002774  000261                26$:    SEC                     ;SET CARRY
1164 002776  006100                        ROL     R0              ;SHIFT BIT
1165 003000  105700                        TSTB    R0
1166 003002  100002                        BPL     .+6
1167 003004  000137  002356                JMP     20$             ;NOT FINISHED
1168 003010  043777  001122  175762        BIC     MAINTB,@LPS     ;CLEAR MAINTENANCE (LOOPBACK) BIT
1169
1170 003016                                ;
1171 003016                         40$:
     003016  022737  000176  001004  59$:    $TSWRG  #1              ;CK SW REG
     003024  001404                        CMP     #176,SWR        ;HW SWITCH REG THERE?
     003026  032777  000001  175756        BEQ     .+12            ;NO, SKP HW CHECK
     003034  001003                        BIT     #1,@HWSWR       ;YES, CK HW REG
     003036  032737  000001  000176        BNE     .+10            ;IF SET, SKP SW REG CK
1172 003044  001402                        BIT     #1,SWREG        ;OTHERWISE, ALSO CK SW REG
1173 003046  000137  001464                BEQ     TEST2           ;CONTINUE IF = 0
1174                                       JMP     TEST1           ;IF SET, LOOP ON TEST
1175
1176
```

```
                                .SBTTL TEST 2   COLOR PRINTER 'SELF' TEST
                                ;******************************************************************************
                                ;
                                ;       Test 2  Initialization Test
                                ;
                                ;       -A  Invoke "Confidence (Self) Test" of Color Printer
                                ;
                                ;       -B  Monitor and report unexpected response:
                                ;
                                ;           Issue request for device "state" or "status" and check for
                                ;           correct response:
                                ;
                                ;               "VERIFIED" message = I'm OK
                                ;               Anything else  = I'm Not OK
                                ;
                                ;
003052                          TEST2:  $TSWRG  #40000          ;CK SW REG
003052  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
003060  001404                          BEQ     .+12            ;NO, SKP HW CHECK
003062  032777  040000  175722          BIT     #40000,@HWSWR   ;YES, CK HW REG
003070  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
003072  032737  040000  000176          BIT     #40000,SWREG    ;OTHERWISE, ALSO CK SW REG
003100  001002                          BNE     61$             ;IF SET, SKIP TST # HDR MSG
003102                                  $PRTSN  2
003102  104000                          EMT     +0                      ;PRINT TEST NUMBER
003104  005630                          TYO2                            ;TEST NUMBER MESSAGE
                                ;
                                ;  SEND "ESCAPE" SEQUENCE TO THE PRINTER
                                ;
003106                          61$:    $PRINT  LCP5            ;INVOKE CONFIDENCE TEST
003106  012737  005542  004312          MOV     #LCP5,PRTMSG    ;LOAD MESSAGE ADDRESS
003114  004437  004250                  JSR     %4,PRINE        ;PRINT IT
                                ;
                                ;  EXPECT "VERIFIED" MESSAGE FROM PRINTER
                                ;
003120  012703  001166          62$:    MOV     #BUFF,R3        ;GET RESPONSE
003124  105013                          CLRB    (R3)            ;INPUT BUFFER ADDRESS
003126  012701  005556                  MOV     #LCP7,R1        ;NULL BYTE
003132  111100                  64$:    MOVB    (R1),R0         ;EXPECTED STRING ADDRESS
003134  001002                          BNE     .+6             ;GET EXPECTED CHAR
003136  000137  003604                  JMP     69$             ;END OF MSG, EXIT
003142                                  $WAITI  \W,,TEST2,#600
                                                                ;WAIT FOR INPUT, OR TIMEOUT
003142                          WAIT5:
003142  012737  000100  001124          MOV     #100,WORK       ;LOAD CNTR
003150  022737  000001  001064          CMP     #1,CYCCNT       ;2ND PASS?
003156  001003                          BNE     WDD5            ;NO, LEAVE COUNT AS IS
003160  012737  000600  001124          MOV     #600,WORK               ;LOAD CNTR WITH LONG CNT
003166  012737  177777  001126  WDD5:   MOV     #-1,WORKA               ;LOAD COUNT
003174  005337  001124                  DEC     WORK                    ;BUMP CNTR
003200  001407                          BEQ     WER5                    ;ERROR IF TIMEOUT
003202  005337  001126          WDE5:   DEC     WORKA                   ;BUMP COUNT
```

```
003206  001767
003210  105777  175662                 BEQ     WDD5                    ;LOOP IF COUNT 0
003214  100372                         TSTB    @DLLPS                  ;CK STATUS
003216  000442                         BPL     WDE5                    ;LOOP TIL DONE
003220                                 BR      WEX5                    ;CONTINUE
003220  022737  000176  001004  WER5:  $TSWRG  #20000          ;CK SW REG
003226  001404                         CMP     #176,SWR        ;HW SWITCH REG THERE?
003230  032777  020000  175554         BEQ     .+12            ;NO, SKP HW CHECK
003236  001003                         BIT     #20000,@HWSWR   ;YES, CK HW REG
003240  032737  020000  000176         BNE     .+10            ;IF SET, SKP SW REG CK
                                       BIT     #20000,SWREG    ;OTHERWISE, ALSO CK SW REG
003246  001002                         BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
                                                               ;BRANCH IF NO MSG WANTED
003250  104000                         EMT     +0
003252  005731                         ETIM                            ;TIMED OUT ERROR

003254
003254  012737  000010  001136  ERR10: $ERROR  \N,TEST2
003262  004537  004606                 MOV     #10,    ERCOUNT         ;SET UP ERROR COUNT 10
                                       JSR     %5,STAER                ;REPORT ERROR SET

003266
003266  022737  000176  001004         $TSWRG  #100000         ;CK SW REG
003274  001404                         CMP     #176,SWR        ;HW SWITCH REG THERE?
003276  032777  100000  175506         BEQ     .+12            ;NO, SKP HW CHECK
003304  001003                         BIT     #100000,@HWSWR  ;YES, CK HW REG
003306  032737  100000  000176         BNE     .+10            ;IF SET, SKP SW REG CK
003314  001402                         BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
003316  000137  003052                 BEQ     CN10                    ;CONTINUE IF BIT 15 = 0
003322                         CN10:   JMP     TEST2                   ;OTHERWISE LOOP

003322  000707                         BR      WAIT5                   ;LOOP

003324                         WEX5:
1216 003324  117702  175550            MOVB    @DLRBUF,R2      ;GET CHAR IN
1217 003330  110223                    MOVB    R2,(R3)+        ;STORE IT
1218 003332  105013                    CLRB    (R3)            ;NULL NEXT BYTE
1219 003334  112100                    MOVB    (R1)+,R0        ;GET EXPECTED CHAR
1220 003336  120002                    CMPB    R0,R2           ;COMPARE
1221 003340  001674                    BEQ     64$             ;OK, WAIT FOR ANOTHER
1222                           ;
1223                           ;  IGNORE UP TO 4 GARBAGE BYTES, LOOKING FOR "V"
1224
1225 003342  020327  001172            CMP     R3,#BUFF+4      ;R3 POINTS TO GARBAGE STORED
1226 003346  003003                    BGT     66$             ;MORE THAN 4 MISCOMPARES
1227 003350  105741                    TSTB    -(R1)           ;RESET TO BEGINING OF MSG
1228 003352  000137  003132            JMP     64$             ;TRY AGAIN
1229
1230 003356  104000            66$:   EMT     +0
1231 003360  007120                    ERMS4                   ;UNEXPECTED RESPONSE
1232 003362                            $ERROR  \N              ;
     003362  012737  000011  001136  ERR11: MOV     #11,    ERCOUNT         ;SET UP ERROR COUNT 11
     003370  004537  004606            JSR     %5,STAER                ;REPORT ERROR SET
1233                           ;
1234                           ;  Print incomplete "VERIFIED" message from "Power-up" diagnostic
1235                           ;
1236 003374  104000                    EMT     +0
1237 003376  001166                    BUFF                    ;Stored message
```

```
1238 003400  012737  000005  001062          MOV     #5,LINCNT       ;Load line count
1239                                 ;
1240                                 ;  Type all subsequent info sent by printer = error info
1241                                 ;
1242 003406                          67$:    $WAITI  \W,79$,TEST2
     003406                          WAIT6:                          ;WAIT FOR INPUT, OR TIMEOUT
     003406  012737  000100  001124          MOV     #100,WORK       ;LOAD CNTR
     003414  012737  177777  001126  WDD6:   MOV     #-1,WORKA               ;LOAD COUNT
     003422  005337  001124          DEC     WORK                    ;BUMP CNTR
     003426  001407                  BEQ     WER6                    ;ERROR IF TIMEOUT
     003430  005337  001126  WDE6:   DEC     WORKA                   ;BUMP COUNT
     003434  001767                  BEQ     WDD6                    ;LOOP IF COUNT 0
     003436  105777  175434          TSTB    @DLLPS                  ;CK STATUS
     003442  100372                  BPL     WDE6                    ;LOOP TIL DONE
     003444  000443                  BR      WEX6                    ;CONTINUE
     003446                  WER6:   $TSWRG  #20000          ;CK SW REG
     003446  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
     003454  001404                  BEQ     .+12            ;NO, SKP HW CHECK
     003456  032777  020000  175326          BIT     #20000,@HWSWR   ;YES, CK HW REG
     003464  001003                  BNE     .+10            ;IF SET, SKP SW REG CK
     003466  032737  020000  000176          BIT     #20000,SWREG    ;OTHERWISE, ALSO CK SW REG
     003474  001002                          BNE     .+6             ;TEST FOR INHIBIT ERROR MSG
                                                                     ;BRANCH IF NO MSG WANTED
     003476  104000                          EMT     +0
     003500  005731                          ETIM                            ;TIMED OUT ERROR
     003502                          $ERROR  \N,TEST2
     003502  012737  000012  001136  ERR12:  MOV     #12,    ERCOUNT         ;SET UP ERROR COUNT 12
     003510  004537  004606                  JSR     %5,STAER                ;REPORT ERROR SET
     003514
     003514  022737  000176  001004          $TSWRG  #100000         ;CK SW REG
     003522  001404                          CMP     #176,SWR        ;HW SWITCH REG THERE?
     003524  032777  100000  175260          BEQ     .+12            ;NO, SKP HW CHECK
     003532  001003                          BIT     #100000,@HWSWR  ;YES, CK HW REG
     003534  032737  100000  000176          BNE     .+10            ;IF SET, SKP SW REG CK
     003542  001402                          BIT     #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
     003544  000137  003052                  BEQ     CN12
     003550                                  JMP     TEST2                   ;CONTINUE IF BIT 15 = 0
                                                                             ;OTHERWISE LOOP
                                     CN12:
     003550  000137  003604          WEX6:   JMP     79$                     ;EXIT
     003554
1243
1244 003554  117737  175320  004464          MOVB    @DLRBUF,TYPDAT  ;Character from printer
1245 003562  004737  004400                  JSR     %7,TYPD         ;Type it
1246 003566  022737  000015  004464          CMP     #15,TYPDAT      ; CR?
1247 003574  001304                          BNE     67$             ;No, loop
1248
1249 003576  005337  001062                  DEC     LINCNT          ;Bump line count
1250 003602  001301                          BNE     67$             ;Loop if not done
1251
1252
1253 003604                           69$:
1254 003604                          79$:    $TSWRG  #2              ;CK SW REG
```

```
        003604  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
        003612  001404                          BEQ     .+12            ;NO, SKP HW CHECK
        003614  032777  000002  175170          BIT     #2,@HWSWR       ;YES, CK HW REG
        003622  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
        003624  032737  000002  000176          BIT     #2,SWREG        ;OTHERWISE, ALSO CK SW REG
  1255  003632  001402                          BEQ     TEST3           ;CONTINUE IF = 0
  1256  003634  000137  003052                  JMP     TEST2           ;IF SET, LOOP ON TEST
  1257
  1258
```

```
 1260                                   .SBTTL TEST 3   PRINTER DISPLAY TEST
 1261                                   ;******************************************************************************
 1262                                   ;
 1263                                   ;       Test 3  Display Test
 1264                                   ;
 1265                                   ;       -A  Send "Escape Sequence" to color printer
 1266                                   ;           to initiate Pattern display
 1267                                   ;
 1268                                   ;
 1269 003640  005737  001064            TEST3:  TST     CYCCNT          ;CK PASS COUNT
 1270 003644  001416                            BEQ     82$             ;IF 1ST PASS, EXECUTE TEST ALWAYS
 1271
 1272 003646                                    $TSWRG  #4              ;ONLY IF LOOP ON TST BIT, CONTINUE
      003646  022737  000176  001004            CMP     #176,SWR        ;HW SWITCH REG THERE?
      003654  001404                            BEQ     .+12            ;NO, SKP HW CHECK
      003656  032777  000004  175126            BIT     #4,@HWSWR       ;YES, CK HW REG
      003664  001003                            BNE     .+10            ;IF SET, SKP SW REG CK
      003666  032737  000004  000176            BIT     #4,SWREG        ;OTHERWISE, ALSO CK SW REG
 1273 003674  001002                            BNE     82$             ;SET, CONTINUE TEST
 1274 003676  000137  004070                    JMP     TSEND           ;OTHERWISE, EXIT>>>>>>>>>>
 1275
 1276 003702                            82$:    $TSWRG  #40000          ;CK SW REG
      003702  022737  000176  001004            CMP     #176,SWR        ;HW SWITCH REG THERE?
      003710  001404                            BEQ     .+12            ;NO, SKP HW CHECK
      003712  032777  040000  175072            BIT     #40000,@HWSWR   ;YES, CK HW REG
      003720  001003                            BNE     .+10            ;IF SET, SKP SW REG CK
      003722  032737  040000  000176            BIT     #40000,SWREG    ;OTHERWISE, ALSO CK SW REG
 1277 003730  001002                            BNE     83$             ;IF SET, SKIP TST # HDR MSG
 1278 003732                                    $PRTSN  3
      003732  104000                            EMT     +0                      ;PRINT TEST NUMBER
      003734  005673                            TY03                            ;TEST NUMBER MESSAGE
 1279
 1280                                   ;
 1281                                   ;  DELAY FOR A SEC
 1282                                   ;
 1283 003736  012737  000004  001124    83$:    MOV     #4,WORK                 ;2 SEC CNTR
 1284 003744  012737  177777  001126    84$:    MOV     #-1,WORKA               ;LOAD COUNT
 1285 003752  005337  001124                    DEC     WORK                    ;BUMP CNTR
 1286 003756  001404                            BEQ     87$                     ;TIMEOUT
 1287 003760  005337  001126            86$:    DEC     WORKA                   ;BUMP COUNT A
 1288 003764  001767                            BEQ     84$                     ;LOOP IF COUNT 0
 1289 003766  000774                            BR      86$                     ;LOOP TIL DONE
 1290
 1291                                   ;
 1292                                   ;  SEND "ESCAPE" SEQUENCE TO THE PRINTER
 1293                                   ;
 1294 003770                            87$:
 1295
 1296 003770                                    $PRINT  LCP4            ;INVOKE DISPLAY TEST
      003770  012737  005533  004312            MOV     #LCP4,PRTMSG    ;LOAD MESSAGE ADDRESS
      003776  004437  004250                    JSR     %4,PRINE        ;PRINT IT
 1297
 1298                                   ;
 1299                                   ;  DELAY FOR A COUPLE OF MINUTES
 1300                                   ;
 1301 004002  012737  000400  001124    93$:    MOV     #400,WORK               ;TWO MIN CNTR
 1302 004010  012737  177777  001126    94$:    MOV     #-1,WORKA               ;LOAD COUNT
```

```
 1303 004016  005337  001124                    DEC     WORK            ;BUMP CNTR
 1304 004022  001404                            BEQ     99$             ;TIMEOUT
 1305 004024  005337  001126          96$:      DEC     WORKA           ;BUMP COUNT A
 1306 004030  001767                            BEQ     94$             ;LOOP IF COUNT 0
 1307 004032  000774                            BR      96$             ;LOOP TIL DONE
 1308
 1309                                           ;
 1310 004034                           99$:     $TSWRG  #4              ;CK SW REG
      004034  022737  000176  001004            CMP     #176,SWR        ;HW SWITCH REG THERE?
      004042  001404                            BEQ     .+12            ;NO, SKP HW CHECK
      004044  032777  000004  174740            BIT     #4,@HWSWR       ;YES, CK HW REG
      004052  001003                            BNE     .+10            ;IF SET, SKP SW REG CK
      004054  032737  000004  000176            BIT     #4,SWREG        ;OTHERWISE, ALSO CK SW REG
 1311 004062  001402                            BEQ     TSEND           ;CONTINUE IF = 0
 1312 004064  000137  003640                    JMP     TEST3           ;IF SET, LOOP ON TEST
 1313
 1314                                           .DSABL LSB
 1315
```

```
1317                                            .SBTTL
1318                                            .SBTTL END OF TEST SEQUENCE
1319                                            ;*****************************************************************************
1320                                            ;
1321                                            ;  END OF TEST SEQUENCE, WAIT FOR KEY INPUT.
1322                                            ;
1323 004070  005237  001064            TSEND:  INC     CYCCNT          ;BUMP PASS COUNT
1324 004074                                    $TSWRG  #40000          ;CK SW REG
     004074  022737  000176  001004            CMP     #176,SWR        ;HW SWITCH REG THERE?
     004102  001404                            BEQ     .+12            ;NO, SKP HW CHECK
     004104  032777  040000  174700            BIT     #40000,@HWSWR   ;YES, CK HW REG
     004112  001003                            BNE     .+10            ;IF SET, SKP SW REG CK
     004114  032737  040000  000176            BIT     #40000,SWREG    ;OTHERWISE, ALSO CK SW REG
1325 004122  001045                            BNE     TSRST           ;DON'T PRINT IF = SET
1326 004124  004537  004466                    JSR     %5,CONV         ;CONVERT PASS #
1327 004130  001064                            CYCCNT
1328 004132  006475                            MES9
1329 004134  000004                            4
1330 004136  104000                            EMT     +0              ;PRINT IT
1331 004140  006457                            MES8
1332 004142                                    $TSWRG  #10000          ;CK SW REG
     004142  022737  000176  001004            CMP     #176,SWR        ;HW SWITCH REG THERE?
     004150  001404                            BEQ     .+12            ;NO, SKP HW CHECK
     004152  032777  010000  174632            BIT     #10000,@HWSWR   ;YES, CK HW REG
     004160  001003                            BNE     .+10            ;IF SET, SKP SW REG CK
     004162  032737  010000  000176            BIT     #10000,SWREG    ;OTHERWISE, ALSO CK SW REG
1333 004170  001002                            BNE     TSEAB           ;PAUSE IF = SET
1334 004172  000137  001464                    JMP     TEST1           ;LOOP TO BEGINING OF TEST
1335
1336 004176  104000            TSEAB:  EMT     +0
1337 004200  006416                    MES7
1338 004202  105777  174614    TSEDA:  TSTB    @TKS            ;ANY KEY TO RESTART
1339 004206  100375                    BPL     TSEDA           ;KEY INPUT?
1340 004210  117700  174602            MOVB    @TKB,R0         ;IF NOT LOOP
1341 004214  142700  000140            BICB    #140,R0         ;GET INPUT CHAR
1342 004220  122700  000004            CMPB    #4,R0           ;CLR UPPER 2 BITS
1343 004224  001004                    BNE     TSRST           ;CTRL D, $, D, OR d
1344 004226  004737  014066            JSR     %7,CLRTTY       ;IF NOT
1345                                            ;CLR PENDING CHARACTERS
1346 004232  000137  001306    TSRSS:  JMP     SETUP
1347
1348 004236  000137  001464    TSRST:  JMP     TEST1
1349
```

```
                                        .SBTTL MISC. SUBROUTINES
                                        ;*****************************************************************************
                                        ;
                                        ;   MISC. ROUTINES
                                        ;
                                        ;*****************************************************************************
                                        ;
                                        ;   ROUTINE TO OUTPUT ASCII MESSAGES ON THE LINE PRINTER
                                        ;
004242  012737  006570  004312  PRNNT:  MOV     #MES14,PRTMSG   ;PRINT TEST NUMBER
004250  004737  014122          PRINE:  JSR     %7,ERCHK        ;TEST FOR ERROR
004254  100007                          BPL     RINT            ;BRANCH IF OK
004256  104000                          EMT     +0
004260  007305                          ERMS6                   ;STATUS ERROR
004262                                  $ERROR  \N
004262  012737  000013  001136  ERR13:  MOV     #13,    ERCOUNT         ;SET UP ERROR COUNT 13
004270  004537  004606                  JSR     %5,STAER                ;REPORT ERROR SET
                                        ;
004274  013737  001000  001020  RINT:   MOV     LPS,TPS ;SET VECTORS -
004302  013737  001002  001014          MOV     LPB,TPB ;TO PRINT ON LINE PRINTER
004310  104000                          EMT     +0              ;PRINT
004312  006570                  PRTMSG: MES14                   ;MESSAGE
004314  012737  177564  001020  TYPINT: MOV     #177564,TPS     ;RESET VECTORS
004322  012737  177566  001014          MOV     #177566,TPB     ;FOR TTY
004330  000204                          RTS     %4              ;RETURN
                                        ;*****************************************************************************
                                        ;
                                        ;INTERRUPT CALLED ROUTINE TO OUTPUT ASCII MESSAGES ON TELETYPE PRINTER
                                        ;
                                        ;       EMT     +0
                                        ;       POINTER TO MESSAGE
                                        ;
004332  011600                  TYP:    MOV     @%6,%0          ;GET ADDR. THAT CONTAINS MESS.
004334  062716  000002                  ADD     #2,@%6          ;SET UP EXIT
004340  011000                          MOV     @%0,%0          ;ADDRESS OF MESSAGE IN R0
004342  112037  004464          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
004346  001001                          BNE     TYPC            ;BRANCH IF NOT DONE
004350  000002                          RTI                     ;EXIT
004352  122737  000045  004464  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR "%"
004360  001426                          BEQ     TYPF            ;BRANCH IF "%"
004362  122737  000043  004464          CMPB    #43,TYPDAT      ;CHECK FOR "#"
004370  001427                          BEQ     TYPG            ;BRANCH IF "#"
004372  004737  004400                  JSR     %7,TYPD         ;TYPE CHARACTER IN TYPDAT
004376  000761                          BR      TYPA            ;NEXT CHAR IN MESSAGE
004400  113777  004464  174406  TYPD:   MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
004406  023737  001020  001120          CMP     TPS,DZCSRH      ;ARE WE REALLY TALKING TO A DZ
004414  001004                          BNE     TYPDO           ;BR. IF NOT DZ
004416  005777  174376                  TST     @TPS            ;IF DZ BIT 15 IS READY BIT NOT BIT 7
004422  100375                          BPL     .-4             ;WAIT UNTIL DONE IS SET
004424  000403                          BR      TYPD01          ;SKIP OTHER "TSTB"
004426  105777  174366          TYPDO:  TSTB    @TPS
004432  100375                          BPL     .-4
004434  000207                  TYPD01: RTS     %7
004436  112737  000012  004464  TYPF:   MOVB    #12,TYPDAT      ;CHAR. TYPED EXIT
004444  004737  004400                  JSR     %7,TYPD         ;OUTPUT LF
004450  112737  000015  004464  TYPG:   MOVB    #15,TYPDAT      ;GO TYPE CHAR.
004456  004737  004400                  JSR     %7,TYPD         ;OUTPUT CR
                                                                ;GO TYPE CHAR.
```

```
1406 004462  000727                    BR      TYPA
1407 004464  000000            TYPDAT: 0
1408
1409                           ;*****************************************************************************
1410                           ;
1411                           ;ROUTINE TO CONVERT OCTAL TO ASCII
1412                           ;
1413                           ;ENTER ROUTINE AS FOLLOWS
1414                           ;       JSR     %5,CONV
1415                           ;XXXXXX=ADDRESS OF NUMBER TO BE CONVERTED
1416                           ;XXXXXX=ADDRESS OF ASCII MESSAGE
1417                           ;XXXXXX=NUMBER OF OCTAL NO.'S TO BE CONVERTED
1418                           ;
1419 004466  010137  004600   CONV:   MOV     R1,CONR1        ;SAVE REG
1420 004472  010237  004602           MOV     R2,CONR2        ;SAVE REG
1421 004476  010337  004604           MOV     R3,CONR3        ;SAVE REG
1422
1423 004502  013537  004576           MOV     @(5)+,ACNVX     ;ADDRSS OF NO. TO BE CONVERTED
1424 004506  012501                   MOV     (5)+,%1         ;ADDRESS OF MESSAGE
1425 004510  012502                   MOV     (5)+,%2         ;NUMBER OF ASCII CHARACTERS
1426 004512  060201                   ADD     %2,%1           ;FIRST CHAR ADDRESS
1427 004514  013703  004576   ACVN:   MOV     ACNVX,%3        ;STORE NUMBER
1428 004520  042703  177770           BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT BIT
1429 004524  062703  000060           ADD     #60,%3          ;SET UP ASCII CHARACTER
1430 004530  110341                   MOVB    %3,-(1)         ;STORE CHARACTER
1431 004532  000241                   CLC                     ;GET NEXT SIGNIFICANT BIT ...
1432 004534  006037  004576           ROR     ACNVX
1433 004540  000241                   CLC
1434 004542  006037  004576           ROR     ACNVX
1435 004546  000241                   CLC
1436 004550  006037  004576           ROR     ACNVX
1437 004554  005302                   DEC     %2              ;-1 FROM ASCII CHAR. CNT
1438 004556  001356                   BNE     ACVN            ;CONVERT NEXT CHARACTER
1439 004560  013701  004600           MOV     CONR1,R1        ;RESTORE REG
1440 004564  013702  004602           MOV     CONR2,R2        ;RESTORE REG
1441 004570  013703  004604           MOV     CONR3,R3        ;RESTORE REG
1442
1443 004574  000205                   RTS     %5              ;EXIT! CONVERSION DONE
1444
1445 004576  000000            ACNVX:  0                       ;WORK REGISTER
1446 004600  000000            CONR1:  0
1447 004602  000000            CONR2:  0
1448 004604  000000            CONR3:  0
1449
1450                           ;*****************************************************************************
1451                           ;
1452                           ;ROUTINE TO REPORT ERROR COUNT
1453                           ;
1454 004606  010537  004752   STAER:  MOV     %5,STARN        ;SAVE R5
1455 004612  004537  004466           JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
1456 004616  001136                   ERCOUNT
1457 004620  006104                   HED2
1458 004622  000003                   3
1459 004624                           $TSWRG  #20000          ;CK SW REG
     004624  022737  000176  001004   CMP     #176,SWR        ;HW SWITCH REG THERE?
     004632  001404                   BEQ     .+12            ;NO, SKP HW CHECK
     004634  032777  020000  174150   BIT     #20000,@HWSWR   ;YES, CK HW REG
```

```
                004642  001003
                004644  032737  020000  000176          BNE     .+10            ;IF SET, SKP SW REG CK
                                                        BIT     #20000,SWREG    ;OTHERWISE, ALSO CK SW REG
1460                                                                            ;TEST FOR INHIBIT ERROR MSG
1461            004652  001002                          BNE     .+6             ;BRANCH IF SET
1462            004654  104000                          EMT     +0              ;TYPE ERROR MESSAGE
1463            004656  006063                          HED1
1464
1465            004660                                  $TSWRG  #10000          ;CK SW REG
                004660  022737  000176  001004          CMP     #176,SWR        ;HW SWITCH REG THERE?
                004666  001404                          BEQ     .+12            ;NO, SKP HW CHECK
                004670  032777  010000  174114          BIT     #10000,@HWSWR   ;YES, CK HW REG
                004676  001003                          BNE     .+10            ;IF SET, SKP SW REG CK
                004700  032737  010000  000176          BIT     #10000,SWREG    ;OTHERWISE, ALSO CK SW REG
1466                                                                            ;TEST FOR HALT ON ERROR
1467            004706  001416                          BEQ     STEXT           ;BRANCH IF NO HALT WANTED
1468                                            ;       HALT                    ;HALT ON ERROR
1469            004710  104000                          EMT     +0              ;PRINT IT
1470            004712  006504                          MES10
1471
1472            004714  105777  174102          STEDA:  TSTB    @TKS            ;KEY INPUT?
1473            004720  100375                          BPL     STEDA           ;IF NOT LOOP
1474            004722  117705  174070                  MOVB    @TKB,R5         ;GET INPUT CHAR
1475            004726  142705  000140                  BICB    #140,R5         ;CLR UPPER 2 BITS
1476            004732  122705  000004                  CMPB    #4,R5           ;CTRL D, $, D, OR d
1477            004736  001002                          BNE     STEXT           ;IF NOT
1478            004740  004737  014066                  JSR     %7,CLRTTY       ;CLR PENDING CHARACTERS
1479
1480            004744  013705  004752  STEXT:  MOV     STARN,%5        ;RESTORE R5
1481            004750  000205                  RTS     %5              ;RETURN
1482                                            .EVEN
1483            004752  000000          STARN:  .WORD   0               ;RETURN STORAGE
```

```
                                        .SBTTL KEYBOARD INTERRUPT ROUTINE
                                        ;*************************************************************************************
                                        ;
                                        ;KEYBOARD INTERRUPT ROUTINE
                                        ;FOR ACCESS TO THE S/W SWITCH REGISTER
                                        ;
004754  010046                  TKINT:  MOV     %0,-(SP)        ;SAVE REGISTERS
004756  010146                          MOV     %1,-(SP)        ;
004760  010246                          MOV     %2,-(SP)        ;
004762  010346                          MOV     %3,-(SP)        ;
004764  010446                          MOV     %4,-(SP)        ;
004766  010546                          MOV     %5,-(SP)        ;
                                        ;
004770  017737  174022  001160  ;;;@@@   TST     SIGNAL          ;PREVIOUS CONTROL-G INPUT ?
004776  042737  177600  001160  ;;;@@@   BEQ     CNTRLG          ;NO-
005004  022737  000007  001160          MOV     @TKB,CHAR       ;GET INPUT CHARACTER
005012  001527                          BIC     #177600,CHAR    ;STRIP OFF PARITY BIT
005014  022737  000015  001160          CMP     #7,CHAR         ;CONTROL-G INPUT?
005022  001466                          BEQ     TYPSWR          ;YES-PRINT HEADER
005024  022737  000025  001160          CMP     #15,CHAR        ;CARRIAGE RETURN ?
005032  001557                          BEQ     DGTS            ;YES-TERMINATE SWR CHANGE
005034  022737  000010  001160          CMP     #25,CHAR        ;CONTROL-U INPUT ?
005042  001576                          BEQ     TK4             ;YES-CLEAR PREVIOUS ENTRY
                                        CMP     #10,CHAR        ;CONTROL-H INPUT ?
                                        BEQ     TKHLP           ;YES-PRINT HELP
005044  022737  000003  001160          CMP     #3,CHAR         ;CONTROL-C INPUT ?
005052  001564                          BEQ     TK9             ;YES-ABORT
                                        ;
005054  023727  001160  000060          CMP     CHAR,#60        ;ILLEGAL # CHECK: LESS THAN 60 ?
005062  100001                          BPL     TK1             ;NO-CONTINUE
005064  000466                          BR      WT3             ;YES-PRINT "?"
005066  022737  000067  001160  TK1:    CMP     #67,CHAR        ;ILLEGAL # CHECK: GREATER THAN 67 ?
005074  100001                          BPL     TK2             ;NO-CONTINUE
005076  000461                          BR      WT3             ;YES-PRINT "?"
005100  005237  001152          TK2:    INC     DIGITS          ;NEXT DIGIT OF SWR INPUT
005104  022737  000006  001152          CMP     #6,DIGITS       ;MORE THAN SIX DIGITS ?
005112  100453                          BMI     WT3             ;YES-PRINT "?"
005114  105777  173700          WT2:    TSTB    @TPS            ;TTY PRINTER READY ?
005120  100375                          BPL     WT2             ; NO-WAIT
005122  013777  001160  173664          MOV     CHAR,@TPB       ;PRINT CHARACTER
005130  162737  000060  001160          SUB     #60,CHAR        ;CONVERT TO OCTAL
005136  022737  000001  001152          CMP     #1,DIGITS       ;FIRST DIGIT ?
005144  001411                          BEQ     TK5             ;YES-CONTINUE
005146  000241                          CLC                     ;ROTATE LEFT THREE
005150  006137  001162                  ROL     OCT             ;TIMES
005154  000241                          CLC                     ;THIS WILL SHIFT
005156  006137  001162                  ROL     OCT             ;SWR VALUE ONE
005162  000241                          CLC                     ;PLACE LEFT
005164  006137  001162                  ROL     OCT             ;OCTAL.
005170  063737  001160  001162  TK5:    ADD     CHAR,OCT        ;NEW VALUE OF SWR
005176  000503                          BR      TK6             ;RETURN FROM INTERRUPT
005200  005737  001152          DGTS:   TST     DIGITS          ;SWR VALUE CHANGED ?
005204  001470                          BEQ     TK3             ;NO-RETURN ,NO CHANGE TO SWR
005206  013737  001162  000176          MOV     OCT,SWREG       ;YES-ENTER NEW SWR VALUE
005214  000464                          BR      TK3             ;RETURN FROM INTERRUPT
```

```
1542 005216  017737  173574  001160  CNTRLG: MOV     @TKB,CHAR       ;GET CHARACTER
1543 005224  042737  177600  001160          BIC     #177600,CHAR    ;STRIP OFF PARITY BIT
1544 005232  022737  000007  001160          CMP     #7,CHAR         ;CONTROL-G INPUT ?
1545 005240  001414                          BEQ     TYPSWR          ;YES-PRINT HEADER
1546 005242  105777  173552          WT3:    TSTB    @TPS            ;TTY PRINTER READY ?
1547 005246  100375                          BPL     WT3             ;NO-WAIT
1548 005250  013777  001160  173536          MOV     CHAR,@TPB       ;PRINT CHARACTER
1549 005256  104000                          EMT     +0              ;PRINT "?"
1550 005260  006614                          MES22
1551 005262  005737  001154                  TST     SIGNAL          ;BAD VALUE ?
1552 005266  001001                          BNE     TYPSWR          ;YES-PRINT HEADER
1553 005270  000442                          BR      TK7             ;RETURN FROM INTERRUPT
1554 005272  012737  000001  001154  TYPSWR: MOV     #1,SIGNAL       ;SET FLAG: CONTROL-G ENTERED
1555 005300  104000                          EMT     +0
1556 005302  006612                          MES21                   ;CR
1557 005304  022737  000176  001004          CMP     #176,SWR        ;H/W SW REG THERE?
1558 005312  001411                          BEQ     TYPSWX          ;NO, SKIP H/W REG DUMP
1559 005314  104000                          EMT     +0
1560 005316  006656                          MES26                   ;H/W SW REG HDR
1561 005320  004537  004466                  JSR     %5,CONV
1562 005324  177570                          177570
1563 005326  006672                          MES27
1564 005330  000006                          6
1565 005332  104000                          EMT     +0
1566 005334  006672                          MES27
1567 005336  104000          TYPSWX: EMT     +0              ;PRINT HEADER
1568 005340  006620                          MES23
1569 005342  004537  004466                  JSR     %5,CONV         ;CONVERT SWR VALUE TO ASCII
1570 005346  000176                          176
1571 005350  006647                          MES25
1572 005352  000006                          6
1573 005354  104000                          EMT     +0              ;PRINT SWR VALUE
1574 005356  006647                          MES25
1575 005360  104000                          EMT     +0              ;PRINT HEADER
1576 005362  006630                          MES24
1577 005364  000404                          BR      TK7             ;RETURN FROM INTERRUPT
1578 005366  005037  001154  TK3:    CLR     SIGNAL          ;CLEAR CONTROL-G FLAG
1579 005372  104000  TK4:    EMT     +0              ; PRINT LINE FEED AND CARRIAGE RETURN
1580 005374  006612                          MES21
1581 005376  005037  001152  TK7:    CLR     DIGITS          ;CLEAR DIGIT COUNT
1582 005402  005037  001162          CLR     OCT             ;CLEAR SWR INPUT
1583 005406  012605  TK6:    MOV     (SP)+,%5        ;RESTORE REGISTERS
1584 005410  012604          MOV     (SP)+,%4        ;
1585 005412  012603          MOV     (SP)+,%3        ;
1586 005414  012602          MOV     (SP)+,%2        ;
1587 005416  012601          MOV     (SP)+,%1        ;
1588 005420  012600          MOV     (SP)+,%0        ;
1589 005422  000002          RTI                     ; RETURN FROM INTERRUPT
1590
1591 005424  012746  000340  TK9:    MOV     #340,-(SP)      ; ABORT AND GO TO 200, RESTART
1592 005430  012746  005440          MOV     #.+10,-(SP)
1593 005434  000137  001306          JMP     SETUP
1594
1595                          ;
1596                          ;  Help -  reached by <ctrl>H
1597 005440  104000          ;
1598 005442  007350  TKHLP:  EMT     +0
                             HLP0
```

```
1599 005444  104000                EMT     +0
1600 005446  007420                HLP1
1601 005450  104000                EMT     +0
1602 005452  007514                HLP2
1603 005454  104000                EMT     +0
1604 005456  007541                HLP3
1605 005460  104000                EMT     +0
1606 005462  007611                HLP4
1607 005464  104000                EMT     +0
1608 005466  007672                HLP4A
1609 005470  104000                EMT     +0
1610 005472  007752                HLP5
1611 005474  104000                EMT     +0
1612 005476  010040                HLP6
1613 005500  104000                EMT     +0
1614 005502  010100                HLP7
1615 005504  104000                EMT     +0
1616 005506  010163                HLP8
1617 005510  000726                BR      TK3
1618
```

```
                                .SBTTL LOCAL MESSAGES
                                .NLIST  BEX
                                .EVEN
                                ;*****************************************************************************
                                ;
                                ;   DEVICE "ESCAPE SEQUENCES"
                                ;
                                ;       AND OPERATOR MESSAGES.
                                ;
005512  033     133     143     DAR1:   .ASCIZ  <33>\[c\        ;What are you?
005516  033     133     060     DAR2:   .ASCIZ  <33>\[0c\       ;What are you?
005523  033     133     077     LCP3:   .ASCIZ  <33>\[??;0c\    ;<answer> I am a LCP0 (Color printer).
005533  033     133     066     LCP4:   .ASCIZ  <33>\[6;2y\     ;Display pattern
005542  033     133     066     LCP5:   .ASCIZ  <33>\[6;1y\     ;Run Confidence test on clr ptr.
005551  033     133     065     DAR6:   .ASCIZ  <33>\[5n\       ;Report your status.
005556  126     105     122     LCP7:   .ASCIZ  \VERIFIED\      ;<answer> I'm OK (Color Printer).
005567  021     000             DAR9:   .ASCIZ  <21>            ;XON
005571  045     124     105     TY01:   .ASCIZ  \%TEST 1, SERIAL LINE UNIT TEST\
005630  045     124     105     TY02:   .ASCIZ  \%TEST 2, COLOR PRINTER 'SELF' TEST\
005673  045     124     105     TY03:   .ASCIZ  \%TEST 3, PRINTER DISPLAY TEST\

005731  045     105     122     ETIM:   .ASCIZ  \%ERROR - TIMEOUT, WAITING FOR INPUT, RESPONSE\
006007  045     105     122     ETIMO:  .ASCIZ  \%ERROR - TIMEOUT, WAITING FOR OUTPUT, DONE$\
006063  045     040     055     HED1:   .ASCII  /% - ERROR NUMBER /
006104  040     040     040     HED2:   .ASCIZ  /       %/
006114  045     123     124     MESDD:  .ASCIZ  /%STARTING EVFU PRINTING TESTS%/
006153  045     045     103     MES1:   .ASCIZ  \%%CZLCP-A-0\
006167  045     103     132     MES2:   .ASCIZ  \%CZLCP0 COLOR PRINTER DIAGNOSTIC\
006230  045     125     116     MES3:   .ASCIZ  \%UNIT IS TEX%\
006246  045     122     105     MES4:   .ASCIZ  \%RESTART ADDRESS 200%\
006274  045     116     157     MES5:   .ASCIZ  \%Now begins the Color Printer Diagnostic%%\
006347  045     103     157     MES6:   .ASCIZ  \%Color Printer Diagnostic Completed%%\
                                .EVEN
006416  045     120     162     MES7:   .ASCIZ  \%Press Any Key to Restart test..\
006457  045     105     116     MES8:   .ASCII  \%END OF PASS: \
006475  040     040     040     MES9:   .ASCIZ  \       %\
006504  045     120     101     MES10:  .ASCIZ  \%PAUSE (HALT) ON ERROR, Press Any Key to Continue..\

                                .EVEN
006570  012     012     124     MES14:  .ASCIZ  <12><12>\TEST NUMBER \
006607  040     040     000     MES20A: .ASCIZ  /  /
                                .EVEN
006612  045     000             MES21:  .ASCIZ  /%/
006614  040     077     045     MES22:  .ASCIZ  / ?%/
006620  123     127     122     MES23:  .ASCIZ  /SWR =  /
006630  040     040     040     MES24:  .ASCIZ  /   NEW SWR =  /
006647  040     040     040     MES25:  .ASCIZ  /       /
006656  050     110     057     MES26:  .ASCIZ  \(H/W SWR = \
006672  040     040     040     MES27:  .ASCIZ  \       ), \
                                .EVEN
006704  045     105     122     ERMS1:  .ASCIZ  \%ERROR - SERIAL LINE NOT AT THIS ADDRESS%\
006756  045     105     122     ERMS2:  .ASCIZ  \%ERROR - SERIAL LINE "LOOPBACK" FAILED%\
007026  045     105     122     ERMS3:  .ASCIZ  \%ERROR - UNEXPECTED RESPONSE, TO "WHAT ARE YOU?" REQUEST%\
007120  045     105     122     ERMS4:  .ASCIZ  \%ERROR - UNEXPECTED RESPONSE, TO "POWER-UP" (SELF) TEST%\
007211  045     105     122     ERMS5:  .ASCIZ  \%ERROR - UNEXPECTED RESPONSE, TO "POWER-UP" STATUS REQUEST%\
007305  045     117     125     ERMS6:  .ASCIZ  \%OUTPUT, SERIAL LINE STATUS ERROR%\
```

```
1677
1678 007350     045     045     040  hlp0:   .asciz  \%% HELP, Switch Register Bit definition\
1679 007420     045     040     040  hlp1:   .asciz  \%  (Note: <CTRL>G - Allows change to "Software" Switch Reg)\
1680 007514     045     061     065  hlp2:   .asciz  \%15)14,13,12...2,1,0\
1681 007541     045     040     174  hlp3:   .asciz  \%  }  }  }  }   } } }_Loop on SLU Test 1\
1682 007611     045     040     174  hlp4:   .asciz  \%  }  }  }  }   } }_Loop on Printer "Self" Test 2\
1683 007672     045     040     174  hlp4a:  .asciz  \%  }  }  }  }   }_Loop on Printer Display Test 3\
1684 007752     045     040     174  hlp5:   .asciz  \%  }  }  }  }_Pause on Error, and Pause at End of Pass\
1685 010040     045     040     174  hlp6:   .asciz  \%  }  }  }_Inhibit Error Reports\
1686 010100     045     040     174  hlp7:   .asciz  \%  }  }_Inhibit Test Number and End of Pass Reports\
1687 010163     045     040     174  hlp8:   .asciz  \%  }_Loop On Error (otherwise continue)%\
```

```
                                .SBTTL SERIAL LINE SETUP ROUTINES
                                .EVEN
                                ;*****************************************************************************
                                ;
                                ;       THIS SUBROUTINE SETS UP THE SERIAL LINE INTERFACE IF THERE IS ONE
                                ;
                                ;*****************************************************************************
010234                          SETSER:
010234  005037  001120                  CLR     DZCSRH                  ;CLEAR DZ'S CSR HOLDER
010240  005037  001066                  CLR     DZTCR                   ;CLEAR PSEUDO DZ TCR REG
010244  005037  001070                  CLR     DZLPR                   ;CLEAR PSEUDO DZ LPR REG
010250  005037  001106                  CLR     DLHERE                  ;CLEAR DL'S PRESENCE AREA
010254  005037  001102                  CLR     BRATE                   ;CLEAR DZ'S BAUD RATE HOLDER
010260                          20$:    ; CMP   SERSW,#1                ;HAVE WE ALREADY SET UP SLU
                                ;       BNE     30$                     ;BR, NOT SET UP
                                ;       JMP     200$                    ;LEAVE ROUTINE ALREADY SET UP
010260  042777  000100  170534  30$:    BIC     #100,@TKS               ;SHUT OFF IE IN TTY
                                ;       MOV     #1,SERSW                ;SET SW TO BEEN SET UP
010266  012737  000010  001122          MOV     #10,MAINTB              ;PRELOAD BIT 3 SET
010274  104000                          EMT     +0                      ;PRINT MESSAGE CALL
010276  015723                          MENU1
010300  104000                          EMT     +0                      ;PRINT MESSAGE CALL
010302  015760                          MENU2
010304  104000                          EMT     +0                      ;PRINT MESSAGE CALL
010306  016006                          MENU3
010310  104000                          EMT     +0                      ;PRINT MESSAGE CALL
010312  016034                          MENU4
010314  105777  170502          40$:    TSTB    @TKS                    ;READ TTY'S STATUS
010320  100375                          BPL     40$                     ;BR, IF NOT DONE
010322  117700  170470                  MOVB    @TKB,%0                 ;PICK UP CHAR. TYPED
010326  105777  170466          42$:    TSTB    @TPS                    ;CHECK FOR BUSY
010332  100375                          BPL     42$                     ;LOOP IF TTY IS BUSY (BR)
010334  110077  170454                  MOVB    %0,@TPB                 ;ECHO CHARACTER TO TTY
010340  042700  177700                  BIC     #177700,%0              ;ONLY 6 BITS ALLOWED
010344  120027  000061                  CMPB    %0,#61                  ;WAS IT AN ASCII 1
010350  001422                          BEQ     47$                     ;BR, IF IT WAS (DL11 SEL.)
010352  120027  000062                  CMPB    %0,#62                  ;WAS IT AN ASCII 2
010356  001007                          BNE     45$
010360  004737  013642                  JSR     %7,GETCR                ;GO GET LF OR CR
010364  020027  123456                  CMP     %0,#123456              ;CHECK FOR BAD INPUT
010370  001733                          BEQ     30$                     ;TRY AGAIN BAD INPUT (NOT CR)
010372  000137  011340                  JMP     300$
010376  122700  000012          45$:    CMPB    #12,%0                  ;BR, IF IT WAS (DZ11 SEL.)
010402  001412                          BEQ     49$                     ;CHK FOR <LF> DEFAULT
010404  122700  000015                  CMPB    #15,%0                  ;BR, IF DEFAULT (DL11 SEL.)
010410  001407                          BEQ     49$                     ;CHK FOR <CR> DEFAULT
010412  000137  010260                  JMP     30$                     ;BR, IF DEFAULT (DL11 SEL.)
                                                                        ;WASN'T CORRECT TYPE-IN
                                ;
                                ;       IT'S A DL11
                                ;
010416  004737  013642          47$:    JSR     %7,GETCR                ;GO GET LF OR CR
010422  020027  123456                  CMP     %0,#123456              ;CHECK FOR BAD INPUT
010426  001714                          BEQ     30$                     ;TRY AGAIN BAD INPUT (NOT CR)
010430  012737  000001  001106  49$:    MOV     #1,DLHERE               ;SHOW DEC PRESENCE
010436  005037  001102                  CLR     BRATE                   ;JUST TO BE SURE
010442  004737  014066                  JSR     %7,CLRTTY               ;PICK UP PENDING CHARACTERS
010446  012737  000004  001122          MOV     #4,MAINTB               ;PRELOAD BIT # 2
```

```
1746 010454  104000                        EMT     +0
1747 010456  016070                        MENUD1                  ;PRINT MESSAGE CALL
1748 010460  104000                        EMT     +0
1749 010462  016156                        MENUD2                  ;PRINT MESSAGE CALL
1750 010464  104000                        EMT     +0
1751 010466  016221                        MENUD3                  ;PRINT MESSAGE CALL
1752 010470  104000                        EMT     +0
1753 010472  016263                        MENUD4                  ;PRINT MESSAGE CALL
1754 010474  105777  170322          51$:  TSTB    @TKS              ;READ TTY'S STATUS
1755 010500  100375                        BPL     51$               ;BR, IF NOT DONE
1756 010502  117700  170310                MOVB    @TKB,%0           ;PICK UP CHAR. TYPED
1757 010506  105777  170306          52$:  TSTB    @TPS              ;CHECK FOR BUSY
1758 010512  100375                        BPL     52$               ;LOOP IF TTY IS BUSY (BR)
1759 010514  110077  170274                MOVB    %0,@TPB           ;ECHO CHARACTER TO TTY
1760 010520  042700  177700                BIC     #177700,%0        ;ONLY 6 BITS ALLOWED
1761 010524  120027  000061                CMPB    %0,#61            ;WAS IT AN ASCII 1
1762 010530  001424                        BEQ     54$               ;BR, IF IT WAS (DL11 SEL.)
1763 010532  120027  000062                CMPB    %0,#62            ;WAS IT AN ASCII 2
1764 010536  001011                        BNE     53$
1765 010540  004737  013642                JSR     %7,GETCR          ;GO GET LF OR CR
1766 010544  020027  123456                CMP     %0,#123456        ;CHECK FOR BAD INPUT
1767 010550  001727                        BEQ     49$               ;TRY AGAIN BAD INPUT (NOT CR)
1768 010552  005037  001122                CLR     MAINTB            ;NO MAINT FEATURE
1769 010556  000137  010614                JMP     55$               ;BR
1770 010562  122700  000012          53$:  CMPB    #12,%0            ;CHK FOR <LF> DEFAULT
1771 010566  001412                        BEQ     55$               ;BR, IF DEFAULT (DL11 SEL.)
1772 010570  122700  000015                CMPB    #15,%0            ;CHK FOR <CR> DEFAULT
1773 010574  001407                        BEQ     55$               ;BR, IF DEFAULT (DL11 SEL.)
1774 010576  000137  010430                JMP     49$               ;WASN'T CORRECT TYPE-IN
1775                                ;
1776 010602  004737  013642          54$:  JSR     %7,GETCR          ;GO GET LF OR CR
1777 010606  020027  123456                CMP     %0,#123456        ;CHECK FOR BAD INPUT
1778 010612  001706                        BEQ     49$               ;TRY AGAIN BAD INPUT (NOT CR)
1779 010614  012737  000001  001106  55$:  MOV     #1,DLHERE         ;SHOW DEC PRESENCE
1780 010622  005037  001102                CLR     BRATE             ;JUST TO BE SURE
1781 010626  004737  014066                JSR     %7,CLRTTY         ;PICK UP PENDING CHARACTERS
1782
1783 010632  104000                        EMT     +0
1784 010634  016317                        DLCSRM                  ;PRINT MESSAGE TO TTY
1785 010636  105777  170160          56$:  TSTB    @TKS              ;CHK TTY IN STATUS
1786 010642  100375                        BPL     56$               ;WAIT FOR DONE
1787 010644  117700  170146                MOVB    @TKB,%0           ;PICK UP CHARACTER TYPED IN
1788 010650  105777  170144          57$:  TSTB    @TPS              ;CHECK FOR BUSY
1789 010654  100375                        BPL     57$               ;LOOP IF TTY IS BUSY (BR)
1790 010656  110077  170132                MOVB    %0,@TPB           ;ECHO CHARACTER
1791 010662  042700  177700                BIC     #177700,%0        ;ONLY 6 BIT PASS
1792 010666  122700  000012                CMPB    #12,%0            ;WAS DEFAULT SEL. <LF>
1793 010672  001551                        BEQ     60$               ;BR, IF DEFAULT SELECTED
1794 010674  122700  000015                CMPB    #15,%0            ;WAS DEFAULT SEL. <CR>
1795 010700  001546                        BEQ     60$               ;BR, IF DEFAULT SELECTED
1796 010702  122700  000061                CMPB    #61,%0            ;WAS IT AN ASCII 1
1797 010706  001403                        BEQ     59$               ;BR, IT WAS A 1
1798 010710  122700  000067                CMPB    #67,%0            ;WAS IT AN ASCII 7
1799 010714  001332                        BNE     54$               ;BR, IF IT WASN'T A 1 OR 7
1800 010716  005037  001124          59$:  CLR     WORK              ;CLEAR WORK AREA
1801 010722  052737  100000  001124        BIS     #100000,WORK      ;ALL THAT FOR BIT 15
1802 010730  004737  013560                JSR     %7,GETOCT         ;GO GET AN OCTAL NUMBER
```

```
1803 010734  020027  123456          CMP     %0,#123456      ;CHECK FOR NON-OCTAL #
1804 010740  001720                  BEQ     54$             ;BR, IN NO GOOD #
1805 010742  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1806         000014                  .REPT   12.
1807                                 ROL     %0              ;ROTATE RO
1808                                 .ENDR
     010744  006100                  ROL     %0              ;ROTATE RO
     010746  006100                  ROL     %0              ;ROTATE RO
     010750  006100                  ROL     %0              ;ROTATE RO
     010752  006100                  ROL     %0              ;ROTATE RO
     010754  006100                  ROL     %0              ;ROTATE RO
     010756  006100                  ROL     %0              ;ROTATE RO
     010760  006100                  ROL     %0              ;ROTATE RO
     010762  006100                  ROL     %0              ;ROTATE RO
     010764  006100                  ROL     %0              ;ROTATE RO
     010766  006100                  ROL     %0              ;ROTATE RO
     010770  006100                  ROL     %0              ;ROTATE RO
     010772  006100                  ROL     %0              ;ROTATE RO
1809 010774  060037  001124          ADD     %0,WORK         ;"OR" THE BITS IN
1810 011000  004737  013560          JSR     %7,GETOCT       ;GO GET AN OCTAL NUMBER
1811 011004  020027  123456          CMP     %0,#123456      ;CHECK FOR NON-OCTAL #
1812 011010  001674                  BEQ     54$             ;BR, IN NO GOOD #
1813 011012  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1814         000011                  .REPT   9.
1815                                 ROL     %0              ;ROTATE RO
1816                                 .ENDR
     011014  006100                  ROL     %0              ;ROTATE RO
     011016  006100                  ROL     %0              ;ROTATE RO
     011020  006100                  ROL     %0              ;ROTATE RO
     011022  006100                  ROL     %0              ;ROTATE RO
     011024  006100                  ROL     %0              ;ROTATE RO
     011026  006100                  ROL     %0              ;ROTATE RO
     011030  006100                  ROL     %0              ;ROTATE RO
     011032  006100                  ROL     %0              ;ROTATE RO
     011034  006100                  ROL     %0              ;ROTATE RO
1817 011036  060037  001124          ADD     %0,WORK         ;"OR" THE BITS IN
1818 011042  004737  013560          JSR     %7,GETOCT       ;GO GET AN OCTAL NUMBER
1819 011046  020027  123456          CMP     %0,#123456      ;CHECK FOR NON-OCTAL #
1820 011052  001653                  BEQ     54$             ;BR, IN NO GOOD #
1821 011054  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1822         000006                  .REPT   6
1823                                 ROL     %0              ;ROTATE RO
1824                                 .ENDR
     011056  006100                  ROL     %0              ;ROTATE RO
     011060  006100                  ROL     %0              ;ROTATE RO
     011062  006100                  ROL     %0              ;ROTATE RO
     011064  006100                  ROL     %0              ;ROTATE RO
     011066  006100                  ROL     %0              ;ROTATE RO
     011070  006100                  ROL     %0              ;ROTATE RO
1825 011072  060037  001124          ADD     %0,WORK         ;"OR" THE BITS IN
1826 011076  004737  013560          JSR     %7,GETOCT       ;GO GET AN OCTAL NUMBER
1827 011102  020027  123456          CMP     %0,#123456      ;CHECK FOR NON-OCTAL #
1828 011106  001635                  BEQ     54$             ;BR, IN NO GOOD #
1829 011110  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1830         000003                  .REPT   3
1831                                 ROL     %0              ;ROTATE RO
1832                                 .ENDR
```

```
                011112  006100
                011114  006100
                011116  006100
     1833 011120  060037  001124
     1834 011124  004737  013560
     1835 011130  020027  123456
     1836 011134  001622
     1837 011136  060037  001124
     1838 011142  013737  001124  001076
     1839 011150  013737  001124  001100
     1840 011156  062737  000002  001100
     1841 011164  013737  001124  001000
     1842 011172  062737  000004  001000
     1843 011200  013737  001000  001002
     1844 011206  062737  000002  001002
     1845 011214  000426
     1846 011216  013737  001032  001000  60$:   MOV     DLCSRC,LPS
     1847 011224  062737  000004  001000        ADD     #4,LPS
     1848 011232  013737  001032  001076        MOV     DLCSRC,DLLPS
     1849 011240  013737  001032  001100        MOV     DLCSRC,DLRBUF
     1850 011246  062737  000002  001100        ADD     #2,DLRBUF
     1851 011254  013737  001000  001002        MOV     LPS,LPB
     1852 011262  062737  000002  001002        ADD     #2,LPB
     1853 011270  000407                        BR      67$
     1854 011272  004737  013642          65$:   JSR     %7,GETCR
     1855 011276  020027  123456                CMP     %0,#123456
     1856 011302  001004                        BNE     66$
     1857 011304  000137  010602                JMP     54$
     1858 011310  004737  014066          67$:   JSR     %7,CLRTTY
     1859 011314                          66$:
     1860 011314  004737  014364                JSR     %7,DLSET
     1861 011320  004737  014066          96$:   JSR     %7,CLRTTY
     1862 011324  052777  000100  167470  99$:   BIS     #100,@TKS
     1863 011332  104000                        EMT     .0
     1864 011334  016543                        DLCRLF
     1865 011336  000207                200$:   RTS     %7
     1866                                ;
     1867                                ;       IT'S A DZ11
     1868                                ;
     1869 011340                          300$:
     1870 011340  004737  014066          350$:  JSR     %7,CLRTTY
     1871 011344  104000                        EMT     .0
     1872 011346  016431                        DZCSRM
     1873 011350  005037  001106                CLR     DLHERE
     1874 011354  105777  167442          355$:  TSTB    @TKS
     1875 011360  100375                        BPL     355$
     1876 011362  117700  167430                MOVB    @TKB,%0
     1877 011366  105777  167426          356$:  TSTB    @TPS
     1878 011372  100375                        BPL     356$
     1879 011374  110077  167414                MOVB    %0,@TPB
     1880 011400  042700  177700                BIC     #177700,%0
     1881 011404  122700  000012                CMPB    #12,%0
     1882 011410  001545                        BEQ     360$
     1883 011412  122700  000015                CMPB    #15,%0
     1884 011416  001542                        BEQ     360$
     1885 011420  122700  000061                CMPB    #61,%0
     1886 011424  001403                        BEQ     357$

                                     ROL     %0          ;ROTATE RO
                                     ROL     %0          ;ROTATE RO
                                     ROL     %0          ;ROTATE RO
                                     ADD     %0,WORK     ;"OR" THE BITS IN
                                     JSR     %7,GETOCT   ;GO GET AN OCTAL NUMBER
                                     CMP     %0,#123456  ;CHECK FOR NON-OCTAL #
                                     BEQ     54$         ;BR, IN NO GOOD #
                                     ADD     %0,WORK     ;"OR" THE BITS IN
                                     MOV     WORK,DLLPS  ;ADDRESS OF DL'S RECV STATUS
                                     MOV     WORK,DLRBUF ;ADDRESS OF DL'S RECV BUF
                                     ADD     #2,DLRBUF   ;ADDRESS OF DL'S RECV BUFFER
                                     MOV     WORK,LPS    ;NEW ADDRESS FOR CSR
                                     ADD     #4,LPS      ;MUST POINT TO TRANSMITTER BUF
                                     MOV     LPS,LPB     ;GET STATUS ADDRESS
                                     ADD     #2,LPB      ;POINT TO DATA BUFFER ADDRESS
                                     BR      65$         ;SKIP OVER DEFAULT
                                                         ;MOVE DEFAULT CSR IN
                                                         ;TRANSMITTER STATUS
                                                         ;ADDRESS OF DL'S RECV STATUS
                                                         ;ADDRESS OF DL'S RECV BUF
                                                         ;ADDRESS OF DL'S RECV BUFFER
                                     MOV     LPS,LPB     ;SET UP THE DATA BUFFER
                                     ADD     #2,LPB      ;SET TO CORRECT ADDRESS
                                     BR      67$         ;SKIP OVER CR GET
                                     JSR     %7,GETCR    ;GO GET LF OR CR
                                     CMP     %0,#123456  ;CHECK FOR BAD INPUT
                                     BNE     66$         ;BR, IF CR RECEIVED (GOOD)
                                     JMP     54$         ;JMP, IF BAD CHARACTER RECD.
                                     JSR     %7,CLRTTY   ;PICK UP PENDING CHARACTERS
                                     JSR     %7,DLSET    ;FIND OUT WHAT TYPE DL11
                                     JSR     %7,CLRTTY   ;PICK UP PENDING CHARACTERS
                                     BIS     #100,@TKS   ;TURN BACK ON
                                     EMT     .0          ;MESSAGE ADDRESS
                                     DLCRLF              ;MESSAGE ADDRESS
                                     RTS     %7          ;RETURN TO CALLEE
                                     ;
                                     ;       IT'S A DZ11
                                     ;
                                     JSR     %7,CLRTTY   ;PICK UP PENDING CHARACTERS
                                     EMT     .0          ;PRINT MESSAGE TO TTY
                                     DZCSRM
                                     CLR     DLHERE      ;JUST TO BE SURE
                                     TSTB    @TKS        ;CHK TTY IN STATUS
                                     BPL     355$        ;WAIT FOR DONE
                                     MOVB    @TKB,%0     ;PICK UP CHARACTER TYPED IN
                                     TSTB    @TPS        ;CHECK FOR BUSY
                                     BPL     356$        ;LOOP IF TTY IS BUSY (BR)
                                     MOVB    %0,@TPB     ;ECHO CHARACTER
                                     BIC     #177700,%0  ;ONLY 6 BIT PASS
                                     CMPB    #12,%0      ;WAS DEFAULT SEL. <LF>
                                     BEQ     360$        ;BR, IF DEFAULT SELECTED
                                     CMPB    #15,%0      ;WAS DEFAULT SEL. <CR>
                                     BEQ     360$        ;BR, IF DEFAULT SELECTED
                                     CMPB    #61,%0      ;WAS IT AN ASCII 1
                                     BEQ     357$        ;BR, IT WAS A 1
```

```
1887 011426  122700  000067          CMPB    #67,R0          ;WAS IT AN ASCII 7
1888 011432  001342                  BNE     350$            ;BR, IF IT WASN'T A 1 OR 7
1889 011434  042700  177770  357$:   BIC     #177770,R0      ;ONLY 3 BITS NEEDED
1890 011440  005037  001124          CLR     WORK            ;CLEAR WORK AREA
1891 011444  052737  100000  001124  BIS     #100000,WORK    ;ALL THAT FOR BIT 15
1892 011452  004737  013560          JSR     R7,GETOCT       ;PICK UP AN OCTAL NUMBER
1893 011456  020027  123456          CMP     R0,#123456      ;CHECK FOR BAD DATA
1894 011462  001726                  BEQ     350$            ;BR, IF DATA WAS BAD
1895 011464  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1896         000014                  .REPT   12.
1897                                 ROL     R0              ;ROTATE R0
1898                                 .ENDR
     011466  006100                  ROL     R0              ;ROTATE R0
     011470  006100                  ROL     R0              ;ROTATE R0
     011472  006100                  ROL     R0              ;ROTATE R0
     011474  006100                  ROL     R0              ;ROTATE R0
     011476  006100                  ROL     R0              ;ROTATE R0
     011500  006100                  ROL     R0              ;ROTATE R0
     011502  006100                  ROL     R0              ;ROTATE R0
     011504  006100                  ROL     R0              ;ROTATE R0
     011506  006100                  ROL     R0              ;ROTATE R0
     011510  006100                  ROL     R0              ;ROTATE R0
     011512  006100                  ROL     R0              ;ROTATE R0
     011514  006100                  ROL     R0              ;ROTATE R0
1899 011516  060037  001124          ADD     R0,WORK         ;"OR" THE BITS IN
1900 011522  004737  013560          JSR     R7,GETOCT       ;GO GET AN OCTAL NUMBER
1901 011526  020027  123456          CMP     R0,#123456      ;CHECK FOR NON-OCTAL #
1902 011532  001702                  BEQ     350$            ;BR, IN NO GOOD #
1903 011534  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1904         000011                  .REPT   9.
1905                                 ROL     R0              ;ROTATE R0
1906                                 .ENDR
     011536  006100                  ROL     R0              ;ROTATE R0
     011540  006100                  ROL     R0              ;ROTATE R0
     011542  006100                  ROL     R0              ;ROTATE R0
     011544  006100                  ROL     R0              ;ROTATE R0
     011546  006100                  ROL     R0              ;ROTATE R0
     011550  006100                  ROL     R0              ;ROTATE R0
     011552  006100                  ROL     R0              ;ROTATE R0
     011554  006100                  ROL     R0              ;ROTATE R0
     011556  006100                  ROL     R0              ;ROTATE R0
1907 011560  060037  001124          ADD     R0,WORK         ;"OR" THE BITS IN
1908 011564  004737  013560          JSR     R7,GETOCT       ;GO GET AN OCTAL NUMBER
1909 011570  020027  123456          CMP     R0,#123456      ;CHECK FOR NON-OCTAL #
1910 011574  001661                  BEQ     350$            ;BR, IN NO GOOD #
1911 011576  000241                  CLC                     ;CLEAR CARRY FOR ROTATE
1912         000006                  .REPT   6
1913                                 ROL     R0              ;ROTATE R0
1914                                 .ENDR
     011600  006100                  ROL     R0              ;ROTATE R0
     011602  006100                  ROL     R0              ;ROTATE R0
     011604  006100                  ROL     R0              ;ROTATE R0
     011606  006100                  ROL     R0              ;ROTATE R0
     011610  006100                  ROL     R0              ;ROTATE R0
     011612  006100                  ROL     R0              ;ROTATE R0
1915 011614  060037  001124          ADD     R0,WORK         ;"OR" THE BITS IN
1916 011620  004737  013560          JSR     R7,GETOCT       ;GO GET AN OCTAL NUMBER
```

```
1917 011624  020027  123456                CMP     #0,#123456          ;CHECK FOR NON-OCTAL #
1918 011630  001643                        BEQ     350$                ;BR, IN NO GOOD #
1919 011632  000241                        CLC                         ;CLEAR CARRY FOR ROTATE
1920         000003                        .REPT   3
1921                                       ROL     #0                  ;ROTATE R0
1922                                       .ENDR
     011634  006100                        ROL     #0                  ;ROTATE R0
     011636  006100                        ROL     #0                  ;ROTATE R0
     011640  006100                        ROL     #0                  ;ROTATE R0
1923 011642  060037  001124                ADD     #0,WORK             ;"OR" THE BITS IN
1924 011646  004737  013560                JSR     #7,GETOCT           ;GO GET AN OCTAL NUMBER
1925 011652  020027  123456                CMP     #0,#123456          ;CHECK FOR NON-OCTAL #
1926 011656  001630                        BEQ     350$                ;BR, IN NO GOOD #
1927 011660  060037  001124                ADD     #0,WORK             ;"OR" THE BITS IN
1928 011664  013737  001124  001000        MOV     WORK,LPS            ;NEW ADDRESS FOR CSR
1929 011672  013737  001000  001002        MOV     LPS,LPB             ;GET STATUS ADDRESS
1930 011700  062737  000006  001002        ADD     #6,LPB              ;POINT TO DATA BUFFER ADDRESS
1931 011706  004737  013642                JSR     #7,GETCR            ;GO GET LF OR CR
1932 011712  020027  123456                CMP     #0,#123456          ;CHECK FOR BAD INPUT
1933 011716  001013                        BNE     365$                ;BR, IF CR RECEIVED (GOOD)
1934 011720  000137  011340                JMP     350$                ;JMP, IF NO CR (BAD)
1935 011724  013737  001040  001000  360$: MOV     DZCSRC,LPS          ;MOVE DEFAULT CSR IN
1936 011732  013737  001000  001002        MOV     LPS,LPB             ;SET UP THE DATA BUFFER
1937 011740  062737  000006  001002        ADD     #6,LPB              ;SET TO CORRECT BUFFER
1938 011746                          365$:
1939 011746  013746  000004                MOV     @#4,-(SP)       ;IS DZ LINE VALID???
1940 011752  013746  000006                MOV     @#6,-(SP)       ;SAVE VECTORS
1941 011756  012737  011772  000004        MOV     #372$,4         ;SAVE
1942 011764  105777  167010                TSTB    @LPS            ;RELOAD VECTOR
1943 011770  000411                        BR      374$            ;IS PRINTER THERE?
1944                                                               ;YES, SKIP TRAP PROCESSING
1945 011772                          372$:
1946 011772  022626                        CMP     (SP)+,(SP)+     ;IF YES, NEVER GET HERE.
1947 011774  012637  000006                MOV     (SP)+,6         ;RESTORE STACK
1948 012000  012637  000004                MOV     (SP)+,4         ;RESTORE VECTORS
1949 012004  104000                        EMT     +0              ;RESTORE
1950 012006  006704                        ERMS1                   ;SLU NOT THERE
1951 012010  000137  010260                JMP     30$             ;TRY AGAIN
1952                                  ;
1953                                  ;  YES, SLU IS HERE
1954                                  ;
1955 012014  012637  000006          374$: MOV     (SP)+,6         ;RESTORE VECTORS
1956 012020  012637  000004                MOV     (SP)+,4         ;RESTORE
1957
1958 012024  012777  000020  166746        MOV     #20,@LPS                ;SET MASTER CLEAR IN DZ11
1959 012032  013737  001000  001120  367$: MOV     LPS,DZCSRH              ;HOLD CSR ADDRESS FOR LATER
1960 012040  032777  000020  166732  366$: BIT     #20,@LPS                ;WAIT FOR MC TO DROP
1961 012046  001374                        BNE     366$                    ;BR, IF MASTER CLEAR IS SET
1962 012050  004737  014066                JSR     #7,CLRTTY               ;PICK UP PENDING CHARACTERS
1963 012054                          369$:
1964                                  ;
1965                                  ;       GET THE DZ'S LINE NUMBER
1966                                  ;
1967 012054  005037  001124          395$: CLR     WORK                    ;CLEAR THE WORK AREA
1968 012060  004737  014066                JSR     #7,CLRTTY               ;PICK UP PENDING CHARACTERS
1969 012064  104000                        EMT     +0                      ;CALL PRINTER
1970 012066  016547                        DZLINE                          ;MESSAGE ADDRESS
```

```
 1971 012070  105777  166726          470$:  TSTB    @TKS            ;CHK TTY IN STATUS
 1972 012074  100375                         BPL     470$            ;WAIT FOR DONE
 1973 012076  117700  166714                 MOVB    @TKB,%0         ;PICK UP CHARACTER TYPED IN
 1974 012102  105777  166712          480$:  TSTB    @TPS            ;CHECK FOR BUSY
 1975 012106  100375                         BPL     480$            ;LOOP UNTIL NOT BUSY
 1976 012110  110077  166700                 MOVB    %0,@TPB         ;ECHO CHARACTER
 1977 012114  042700  177700                 BIC     #177700,%0      ;ONLY 6 BIT PASS
 1978 012120  122700  000012                 CMPB    #12,%0          ;WAS DEFAULT SEL. <LF>
 1979 012124  001425                         BEQ     490$            ;BR, IF DEFAULT SELECTED
 1980 012126  122700  000015                 CMPB    #15,%0          ;WAS DEFAULT SEL. <CR>
 1981 012132  001422                         BEQ     490$            ;BR, IF DEFAULT SELECTED
 1982 012134  005037  001124                 CLR     WORK            ;CLEAR WORK AREA
 1983 012140  042700  177700                 BIC     #177700,%0      ;ONLY 6 BIT PASS
 1984 012144  120027  000070                 CMPB    %0,#70          ;ERROR IF 8 OR MORE
 1985 012150  002341                         BGE     395$
 1986 012152  120027  000060                 CMPB    %0,#60          ;ERROR IF LESS THAN 0
 1987 012156  002736                         BLT     395$
 1988 012160  042700  177770                 BIC     #177770,%0      ;ONLY THREE BITS PASS
 1989 012164  060037  001124                 ADD     %0,WORK         ;"OR" THE BITS IN
 1990 012170  013737  001124  001072         MOV     WORK,DZLNE      ;SET UP NEW LINE NUMBER
 1991 012176  000406                         BR      495$            ;SKIP OVER DEFAULT
 1992 012200  005037  001072          490$:  CLR     DZLNE           ;SET WITH DEFAULT VECTOR
 1993 012204  052737  000001  001066         BIS     #1,DZTCR        ;SET LINE 0 POSITION
 1994 012212  000423                         BR      498$            ;SKIP OVER CR GET STUFF
 1995 012214                          495$:
 1996 012214  053737  001072  001070         BIS     DZLNE,DZLPR     ;SET LINE NUMBER UP
 1997 012222  013703  001072                 MOV     DZLNE,%3        ;NUMBER OF LOOPS
 1998 012226  062703  000001                 ADD     #1,%3           ;SET UP COUNTER FOR SOB
 1999 012232  005001                         CLR     %1              ;CLEAR WORK AREA
 2000 012234  000261                         SEC                     ;SET CARRY BIT
 2001 012236  006101                  496$:  ROL     %1              ;MOV BIT INTO POSITION
 2002 012240  000241                         CLC                     ;DON'T SHIFT IN CARRY
 2003 012242  077303                         SOB     %3,496$         ;KEEP GOING IF NOT ZERO
 2004 012244  010137  001066                 MOV     %1,DZTCR        ;BIT FOR LINE NUMBER
 2005 012250  004737  013642                 JSR     %7,GETCR        ;GO GET LF OR CR
 2006 012254  020027  123456                 CMP     %0,#123456      ;CHECK FOR BAD INPUT
 2007 012260  001675                         BEQ     395$            ;TRY AGAIN BAD INPUT (NOT CR)
 2008                                 ;
 2009                                 ;      PICK UP DZ'S BAUD RATE
 2010                                 ;
 2011 012262  004737  014066          498$:  JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
 2012 012266  104000                         EMT     +0              ;CALL TO THE TTY PRINTER
 2013 012270  016615                         LN1                     ;BAUD RATE MENU PRINTOUTS
 2014 012272  004737  014066                 JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
 2015 012276  104000                         EMT     +0              ;CALL TO THE TTY PRINTER
 2016 012300  016701                         LN2                     ;BAUD RATE MENU PRINTOUTS
 2017 012302  004737  014066                 JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
 2018 012306  104000                         EMT     +0              ;CALL TO THE TTY PRINTER
 2019 012310  016722                         LN3                     ;BAUD RATE MENU PRINTOUTS
 2020 012312  004737  014066                 JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
 2021 012316  104000                         EMT     +0              ;CALL TO THE TTY PRINTER
 2022 012320  016743                         LN4                     ;BAUD RATE MENU PRINTOUTS
 2023 012322  004737  014066                 JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
 2024 012326  104000                         EMT     +0              ;CALL TO THE TTY PRINTER
 2025 012330  016764                         LN5                     ;BAUD RATE MENU PRINTOUTS
 2026 012332  004737  014066                 JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
 2027 012336  104000                         EMT     +0              ;CALL TO THE TTY PRINTER
```

```
2028 012340  017005
2029 012342  004737  014066          LN6
2030 012346  104000                  JSR     %7,CLRTTY       ;BAUD RATE MENU PRINTOUTS
2031 012350  017026                  EMT     +0              ;PICK UP PENDING CHARACTERS
2032 012352  004737  014066          LN7                     ;CALL TO THE TTY PRINTER
2033 012356  104000                  JSR     %7,CLRTTY       ;BAUD RATE MENU PRINTOUTS
2034 012360  017047                  EMT     +0              ;PICK UP PENDING CHARACTERS
2035 012362  004737  014066          LN8                     ;CALL TO THE TTY PRINTER
2036 012366  104000                  JSR     %7,CLRTTY       ;BAUD RATE MENU PRINTOUTS
2037 012370  017111                  EMT     +0              ;PICK UP PENDING CHARACTERS
2038 012372  105777  166424          LN10                    ;CALL TO THE TTY PRINTER
2039 012376  100375          500$:   TSTB    @TKS            ;BAUD RATE MENU PRINTOUTS
2040 012400  117700  166412          BPL     500$            ;CHK TTY IN STATUS
2041 012404  105777  166410          MOVB    @TKB,%0         ;WAIT FOR DONE
2042 012410  100375          507$:   TSTB    @TPS            ;PICK UP CHARACTER TYPED IN
2043 012412  110077  166376          BPL     507$            ;CHECK FOR BUSY
2044 012416  042700  177700          MOVB    %0,@TPB         ;LOOP UNTIL NOT BUSY
2045 012422  122700  000012          BIC     #177700,%0      ;ECHO CHARACTER
2046 012426  001467                  CMPB    #12,%0          ;ONLY 6 BIT PASS
2047 012430  122700  000015          BEQ     544$            ;WAS DEFAULT SEL. <LF>
2048 012434  001464                  CMPB    #15,%0          ;BR, IF DEFAULT SELECTED
2049 012436  005037  001124          BEQ     544$            ;WAS DEFAULT SEL. <CR>
2050 012442  042700  177770          CLR     WORK            ;BR, IF DEFAULT SELECTED
2051 012446  122700  000001          BIC     #177770,%0      ;CLEAR WORK AREA
2052 012452  001003                  CMPB    #1,%0           ;ONLY THREE BITS PASS
2053 012454  012737  002000  001124  BNE     510$            ;CHECK FOR A "1" TYPED
2054 012462                          MOV     #2000,WORK      ;BR, IF IT WASN'T A ONE
2055 012462  122700  000002  510$:                           ;SET FOR 150 BAUD
2056 012466  001003                  CMPB    #2,%0
2057 012470  012737  002400  001124  BNE     520$            ;CHECK FOR A TWO TYPED
2058 012476                          MOV     #2400,WORK      ;BR, IF IT WASN'T A "2"
2059 012476  122700  000003  520$:                           ;SET FOR 300 BAUD
2060 012502  001003                  CMPB    #3,%0
2061 012504  012737  003000  001124  BNE     521$            ;CHECK FOR A THREE TYPED
2062 012512                          MOV     #3000,WORK      ;BR, IF IT WASN'T A "3"
2063 012512  122700  000004  521$:                           ;SET FOR 600 BAUD
2064 012516  001003                  CMPB    #4,%0
2065 012520  012737  003400  001124  BNE     522$            ;CHECK FOR A FOUR TYPED
2066 012526                          MOV     #3400,WORK      ;BR, IF IT WASN'T A "4"
2067 012526  122700  000005  522$:                           ;SET FOR 1200 BAUD
2068 012532  001003                  CMPB    #5,%0
2069 012534  012737  005000  001124  BNE     523$            ;CHECK FOR A FIVE TYPED
2070 012542                          MOV     #5000,WORK      ;BR, IF IT WASN'T A "5"
2071 012542  122700  000006  523$:                           ;SET FOR 2400 BAUD
2072 012546  001003                  CMPB    #6,%0
2073 012550  012737  006000  001124  BNE     524$            ;CHECK FOR A SIX TYPED
2074 012556                          MOV     #6000,WORK      ;BR, IF IT WASN'T A "6"
2075 012556  122700  000007  524$:                           ;SET FOR 4800 BAUD
2076 012562  001015                  CMPB    #7,%0
2077 012564  012737  007000  001124  BNE     525$            ;CHECK FOR A SEVEN TYPED
2078 012572  004737  013642  543$:   MOV     #7000,WORK      ;BR, IF IT WASN'T A "7"
2079 012576  020027  123456          JSR     %7,GETCR        ;SET FOR 9600 BAUD
2080 012602  001627                  CMP     %0,#123456      ;GO GET LF OR CR
2081 012604  000415                  BEQ     498$            ;CHECK FOR BAD INPUT
2082 012606  012737  007000  001124  BR      530$            ;BR, IF CR RECEIVED (GOOD)
2083 012614  000411          544$:   MOV     #7000,WORK      ;LEAVE
2084 012616                  525$:   BR      530$            ;DEFAULT IS 9600
                                                             ;LEAVE
```

```
2085 012616  023727  001124  000000  527$:   CMP     WORK,#0                 ;CHECK FOR NOT SET
2086 012624  001616                          BEQ     498$                    ;BR, IF NUMBERS NOT VALID
2087 012626  004737  013642                  JSR     %7,GETCR                ;GO GET LF OR CR
2088 012632  020027  123456                  CMP     %0,#123456              ;CHECK FOR BAD INPUT
2089 012636  001611                          BEQ     498$                    ;TRY AGAIN BAD INPUT (NOT CR)
2090 012640                          530$:
2091 012640  013737  001124  001102  540$:   MOV     WORK,BRATE              ;STORE BAUD RATE FOR LATER
2092 012646  053737  001124  001070          BIS     WORK,DZLPR              ;PUT BAUD RATE IN PLACE
2093                                 ;
2094                                 ;       THIS CODE SETS UP THE NUMBER OF STOP BITS
2095                                 ;
2096 012654                          600$:
2097 012654  004737  014066                  JSR     %7,CLRTTY               ;PICK UP PENDING CHARACTERS
2098 012660  104000                          EMT     +0                      ;CALL PRINT ROUTINE
2099 012662  017153                          STOPM                           ;"TYPE NUMBER OF STOP BITS ETC."
2100 012664  105777  166132          604$:   TSTB    @TKS                    ;CK TTY IN STATUS
2101 012670  100375                          BPL     604$                    ;WAIT FOR DONE
2102 012672  117700  166120                  MOVB    @TKB,%0                 ;PICK UP CHARACTER
2103 012676  105777  166116          607$:   TSTB    @TPS                    ;CHECK FOR BUSY
2104 012702  100375                          BPL     607$                    ;LOOP IF TTY IS BUSY (BR)
2105 012704  110077  166104                  MOVB    %0,@TPB                 ;ECHO CHARACTER
2106 012710  042700  177700                  BIC     #177700,%0              ;ONLY 6 BIT PASS
2107 012714  122700  000012                  CMPB    #12,%0                  ;CHK FOR <LF> DEFAULT
2108 012720  001413                          BEQ     610$                    ;BR, IF DEFAULT (ONE STOP BIT)
2109 012722  122700  000015                  CMPB    #15,%0                  ;CHK FOR <CR> DEFAULT
2110 012726  001410                          BEQ     610$                    ;BR, IF DEFAULT (ONE STOP BIT)
2111 012730  122700  000061                  CMPB    #61,%0                  ;WAS AN ASCII 1 TYPED
2112 012734  001011                          BNE     620$                    ;BR, IF IT WASN'T A ONE
2113 012736  004737  013642                  JSR     %7,GETCR                ;GO GET LF OR CR
2114 012742  020027  123456                  CMP     %0,#123456              ;CHECK FOR BAD INPUT
2115 012746  001742                          BEQ     600$                    ;TRY AGAIN BAD INPUT (NOT CR)
2116 012750  042737  000040  001070  610$:   BIC     #40,DZLPR               ;1 STOP BIT = 0
2117 012756  000413                          BR      630$                    ;SKIP OVER
2118 012760  122700  000062          620$:   CMPB    #62,%0                  ;CHECK FOR A TWO
2119 012764  001333                          BNE     600$                    ;BR, IF NOT A TWO (ERROR)
2120 012766  004737  013642                  JSR     %7,GETCR                ;GO GET LF OR CR
2121 012772  020027  123456                  CMP     %0,#123456              ;CHECK FOR BAD INPUT
2122 012776  001726                          BEQ     600$                    ;TRY AGAIN BAD INPUT (NOT CR)
2123 013000  052737  000040  001070          BIS     #40,DZLPR               ;2 STOP BIT = 1
2124 013006                          630$:
2125 013006  004737  014066                  JSR     %7,CLRTTY               ;PICK UP PENDING CHARACTERS
2126                                 ;
2127                                 ;       THIS CODE SETS UP THE NUMBER OF DATA BITS
2128                                 ;
2129 013012                          700$:
2130 013012  004737  014066                  JSR     %7,CLRTTY               ;PICK UP PENDING CHARACTERS
2131 013016  104000                          EMT     +0                      ;CALL PRINT ROUTINE
2132 013020  017224                          DATAM                           ;"TYPE MUNBER OF DATA BITS ETC."
2133 013022  105777  165774          704$:   TSTB    @TKS                    ;CK TTY IN STATUS
2134 013026  100375                          BPL     704$                    ;WAIT FOR DONE
2135 013030  117700  165762                  MOVB    @TKB,%0                 ;PICK UP CHARACTER
2136 013034  105777  165760          707$:   TSTB    @TPS                    ;CHECK FOR BUSY
2137 013040  100375                          BPL     707$                    ;LOOP IF TTY IS BUSY (BR)
2138 013042  110077  165746                  MOVB    %0,@TPB                 ;ECHO CHARACTER
2139 013046  042700  177700                  BIC     #177700,%0              ;ONLY 6 BIT PASS
2140 013052  122700  000012                  CMPB    #12,%0                  ;CHK FOR <LF> DEFAULT
2141 013056  001407                          BEQ     710$                    ;BR, IF DEFAULT (7 DATA BITS)
```

```
2142 013060  122700  000015                    CMPB    #15,%0          ;CHK FOR <CR> DEFAULT
2143 013064  001404                            BEQ     710$            ;BR, IF DEFAULT (7 DATA BITS)
2144 013066  122700  000067                    CMPB    #67,%0          ;WAS AN ASCII 7 TYPED
2145 013072  001016                            BNE     720$            ;BR, IF IT WASN'T A SEVEN
2146 013074  000404                            BR      715$            ;SKIP OVER
2147 013076  052737  000020  001070  710$:     BIS     #20,DZLPR       ;7 DATA BITS = 10
2148 013104  000424                            BR      730$            ;SKIP OVER CR GET
2149 013106  052737  000020  001070  715$:     BIS     #20,DZLPR       ;7 DATA BITS = 10
2150 013114  004737  013642                    JSR     %7,GETCR        ;GO GET LF OR CR
2151 013120  020027  123456                    CMP     %0,#123456      ;CHECK FOR BAD INPUT
2152 013124  001732                            BEQ     700$            ;TRY AGAIN BAD INPUT (NOT CR)
2153 013126  000413                            BR      730$            ;SKIP OVER
2154 013130  122700  000070            720$:   CMPB    #70,%0          ;CHECK FOR A EIGHT
2155 013134  001326                            BNE     700$            ;BR, IF NOT A EIGHT (ERROR)
2156 013136  004737  013642                    JSR     %7,GETCR        ;GO GET LF OR CR
2157 013142  020027  123456                    CMP     %0,#123456      ;CHECK FOR BAD INPUT
2158 013146  001721                            BEQ     700$            ;TRY AGAIN BAD INPUT (NOT CR)
2159 013150  052737  000030  001070            BIS     #30,DZLPR       ;8 DATA BITS = 30
2160 013156                            730$:
2161 013156  004737  014066                    JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
2162                                    ;
2163                                    ;       THIS CODE SETS UP PARITY BITS
2164                                    ;
2165 013162                            800$:
2166 013162  004737  014066                    JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
2167 013166  104000                            EMT     +0              ;CALL PRINT ROUTINE
2168 013170  017275                            PARITY                  ;"IS PARITY SELECTED "
2169 013172  105777  165624            804$:   TSTB    @TKS            ;CK TTY IN STATUS
2170 013176  100375                            BPL     804$            ;WAIT FOR DONE
2171 013200  117700  165612                    MOVB    @TKB,%0         ;PICK UP CHARACTER
2172 013204  105777  165610            807$:   TSTB    @TPS            ;CHECK FOR BUSY
2173 013210  100375                            BPL     807$            ;LOOP IF TTY IS BUSY (BR)
2174 013212  110077  165576                    MOVB    %0,@TPB         ;ECHO CHARACTER
2175 013216  042700  177600                    BIC     #177600,%0      ;ONLY 7 BIT PASS
2176 013222  122700  000012                    CMPB    #12,%0          ;CHK FOR <LF> DEFAULT
2177 013226  001427                            BEQ     825$            ;BR, IF DEFAULT (NO)
2178 013230  122700  000015                    CMPB    #15,%0          ;CHK FOR <CR> DEFAULT
2179 013234  001424                            BEQ     825$            ;BR, IF DEFAULT (NO)
2180 013236  122700  000131                    CMPB    #131,%0         ;WAS AN ASCII Y TYPED
2181 013242  001011                            BNE     820$            ;BR, IF IT WASN'T A Y
2182 013244  004737  013642            810$:   JSR     %7,GETCR        ;GO GET LF OR CR
2183 013250  020027  123456                    CMP     %0,#123456      ;CHECK FOR BAD INPUT
2184 013254  001742                            BEQ     800$            ;TRY AGAIN BAD INPUT (NOT CR)
2185 013256  052737  000100  001070            BIS     #100,DZLPR      ;YES PARITY IS USED
2186 013264  000415                            BR      830$            ;SKIP OVER
2187 013266  122700  000116            820$:   CMPB    #116,%0         ;CHECK FOR A N
2188 013272  001333                            BNE     800$            ;BR, IF NOT A N (ERROR)
2189 013274  004737  013642                    JSR     %7,GETCR        ;GO GET LF OR CR
2190 013300  020027  123456                    CMP     %0,#123456      ;CHECK FOR BAD INPUT
2191 013304  001726                            BEQ     800$            ;TRY AGAIN BAD INPUT (NOT CR)
2192 013306  042737  000100  001070  825$:     BIC     #100,DZLPR      ;NO PARITY
2193 013314  000137  013462                    JMP     950$            ;NO MORE PARITY ?
2194 013320                            830$:
2195 013320  004737  014066                    JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
2196                                    ;
2197                                    ;       THIS CODE SETS UP THE ODD EVEN PARITY BIT
2198                                    ;
```

```
2199  013324
2200  013324  004737  014066                  900$:
2201  013330  104000                                  JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
2202  013332  017340                                  EMT     +0              ;CALL PRINT ROUTINE
                                                      PARITZ                  ;"IS PARITY ODD OR EVEN"
2203  013334  105777  165462                  904$:   TSTB    @TKS            ;CK TTY IN STATUS
2204  013340  100375                                  BPL     904$            ;WAIT FOR DONE
2205  013342  117700  165450                          MOVB    @TKB,%0         ;PICK UP CHARACTER
2206  013346  105777  165446                  907$:   TSTB    @TPS            ;CHECK FOR BUSY
2207  013352  100375                                  BPL     907$            ;LOOP IF TTY IS BUSY (BR)
2208  013354  110077  165434                          MOVB    %0,@TPB         ;ECHO CHARACTER
2209  013360  042700  177600                          BIC     #177600,%0      ;ONLY 7 BIT PASS
2210  013364  122700  000012                          CMPB    #12,%0          ;CHK FOR <LF> DEFAULT
2211  013370  001413                                  BEQ     910$            ;BR, IF DEFAULT (ODD)
2212  013372  122700  000015                          CMPB    #15,%0          ;CHK FOR <CR> DEFAULT
2213  013376  001410                                  BEQ     910$            ;BR, IF DEFAULT (ODD)
2214  013400  122700  000117                          CMPB    #117,%0         ;WAS AN ASCII O TYPED
2215  013404  001011                                  BNE     920$            ;BR, IF IT WASN'T A ONE
2216  013406  004737  013642                          JSR     %7,GETCR        ;GO GET LF OR CR
2217  013412  020027  123456                          CMP     %0,#123456      ;CHECK FOR BAD INPUT
2218  013416  001742                                  BEQ     900$            ;TRY AGAIN BAD INPUT (NOT CR)
2219  013420  042737  000200  001070  910$:   BIC     #200,DZLPR      ;ODD PARITY = 0
2220  013426  000413                                  BR      930$            ;SKIP OVER
2221  013430  122700  000105                  920$:   CMPB    #105,%0         ;CHECK FOR A E
2222  013434  001333                                  BNE     900$            ;BR, IF NOT A TWO (ERROR)
2223  013436  004737  013642                          JSR     %7,GETCR        ;GO GET LF OR CR
2224  013442  020027  123456                          CMP     %0,#123456      ;CHECK FOR BAD INPUT
2225  013446  001726                                  BEQ     900$            ;TRY AGAIN BAD INPUT (NOT CR)
2226  013450  052737  000200  001070                  BIS     #200,DZLPR      ;EVEN PARITY = 1
2227  013456                                  930$:
2228  013456  004737  014066                          JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
2229                                          ;
2230  013462                                  950$:
2231                                          ;       SET UP ACTUAL DZ REGISTERS
2232                                          ;
2233  013462  013703  001000                          MOV     LPS,%3          ;GET STATUS REGISTER ADDRESS
2234  013466  062703  000006                          ADD     #6,%3           ;POINT TO TRANS BUFFER ADDR
2235  013472  010337  001002                          MOV     %3,LPB          ;PRINTER DATA BUFFER ADDR
2236  013476  013703  001000                          MOV     LPS,%3          ;GET STATUS REG ADDRESS
2237  013502  062703  000002                          ADD     #2,%3           ;POINT TO LPR REGISTER
2238  013506  010337  001116                          MOV     %3,DZRBUF       ;ADDRESS OF RECEIVER BUFFER
2239  013512  052737  010000  001070                  BIS     #10000,DZLPR    ;SET RECEIVER ON BIT
2240  013520  013713  001070                          MOV     DZLPR,(%3)      ;SET SPEED, LINE, PARITY ETC
2241  013524  052777  000040  165246                  BIS     #40,@LPS        ;SET MASTER SCAN ENABLE
2242  013532  062703  000002                          ADD     #2,%3           ;POINT TO TCR REGISTER
2243  013536  113713  001066                          MOVB    DZTCR,(%3)      ;SET TRANS LINE NUMBER
2244  013542  010337  001110                          MOV     %3,DZTCRA       ;ADDRESS FOR DTR CHECK
2245  013546  062737  000001  001110                  ADD     #1,DZTCRA       ;POINT TO UPPER HALF
2246                                          ;       SHOULD BE ALL SET
2247                                          ;
2248  013554  000137  011324                          JMP     99$             ;RETURN TO CALLEE
```

```
2250
2251
2252
2253
2254
2255
2256 013560
2257 013560  105777  165236  ;*****************************************************************************
2258 013564  100375          ;
2259 013566  117700  165224  ;  OTHER SUBROUTINES
2260 013572  105777  165222  ;
2261 013576  100375
2262 013600  110077  165210  GETOCT:
2263 013604  042700  177700  1$:     TSTB    @TKS            ;CK TTY IN STATUS
2264 013610  120027  000070          BPL     1$              ;WAIT FOR DONE
2265 013614  002007          MOVB    @TKB,%0         ;PICK UP CHARACTER
2266 013616  120027  000060  5$:     TSTB    @TPS            ;CHECK FOR BUSY
2267 013622  002404          BPL     5$              ;LOOP IF TTY IS BUSY (BR)
2268 013624  042700  177770          MOVB    %0,@TPB         ;ECHO CHARACTER
2269 013630  000137  013640          BIC     #177700,%0      ;ONLY 6 BIT PASS
2270 013634  012700  123456          CMPB    %0,#70          ;ERROR IF 8 OR MORE
2271 013640  000207                  BGE     10$
2272                                 CMPB    %0,#60
2273                                 BLT     10$             ;ERROR IF LESS THAN 0
2274                                 BIC     #177770,%0      ;ONLY THREE BITS PASS
2275                                 JMP     20$             ;OCTAL # OK
2276 013642  10$:    MOV     #123456,%0      ;WAS NOT OCTAL #
2277 013642  105777  165154  20$:    RTS     %7
2278 013646  100375          ;*****************************************************************************
2279 013650  117700  165142  ;
2280 013654  105777  165140  ;       THIS ROUTINE WAITS FOR A CR OR LF
2281 013660  100375          ;
2282 013662  110077  165126  GETCR:
2283 013666  042700  177700  1$:     TSTB    @TKS            ;CK TTY IN STATUS
2284 013672  120027  000015          BPL     1$              ;WAIT FOR DONE
2285 013676  001405          MOVB    @TKB,%0         ;PICK UP CHARACTER
2286 013700  120027  000012  5$:     TSTB    @TPS            ;CHECK FOR BUSY
2287 013704  001402          BPL     5$              ;LOOP IF TTY IS BUSY (BR)
2288 013706  012700  123456          MOVB    %0,@TPB         ;ECHO CHARACTER
2289 013712  004737  014066          BIC     #177700,%0      ;ONLY 6 BIT PASS
2290 013716  000207                  CMPB    %0,#15          ;WAS IT A CR
2291 013720  004737  014066          BEQ     20$             ;BR, IF IT WAS
2292 013724  104000                  CMPB    %0,#12          ;WAS IT A LF
2293 013726  015613                  BEQ     20$             ;BR, IF IT WAS
2294 013730  104000          10$:    MOV     #123456,%0      ;WAS NOT OCTAL #
2295 013732  015643          20$:    JSR     %7,CLRTTY       ;CLEAR OUT ANY WAITING CHARA
2296 013734  104000                  RTS     %7
2297 013736  015654
2298 013740  104000          EIACHK: JSR     %7,CLRTTY       ;PICK UP PENDING CHARACTERS
2299 013742  015667                  EMT     +0              ;CALL TO THE TTY PRINTER
2300 013744  105777  165052          MENU10                  ;"LINE TYPE"
2301 013750  100375                  EMT     +0              ;CALL TO THE TTY PRINTER
2302 013752  117700  165040          MENU20                  ;" 1 EIA"
2303 013756  105777  165036          EMT     +0              ;CALL TO THE TTY PRINTER
2304 013762  100375                  MENU30                  ;" 20MA"
2305 013764  110077  165024          EMT     +0              ;CALL TO THE TTY PRINTER
2306 013770  042700  177700          MENU40                  ;"SERIAL LINE TYPE <1>?"
                             1$:     TSTB    @TKS            ;CHK TTY IN STATUS
                                     BPL     1$              ;WAIT FOR DONE
                                     MOVB    @TKB,%0         ;PICK UP CHARACTER TYPED IN
                             5$:     TSTB    @TPS            ;CHECK FOR BUSY
                                     BPL     5$              ;LOOP UNTIL NOT BUSY
                                     MOVB    %0,@TPB         ;ECHO CHARACTER
                                     BIC     #177700,%0      ;ONLY 6 BIT PASS
```

```
2307 013774  122700  000012            CMPB    #12,%0          ;WAS DEFAULT SEL. <LF>
2308 014000  001426                    BEQ     20$             ;BR, IF DEFAULT SELECTED
2309 014002  122700  000015            CMPB    #15,%0          ;WAS DEFAULT SEL. <CR>
2310 014006  001423                    BEQ     20$             ;BR, IF DEFAULT SELECTED
2311 014010  042700  177770            BIC     #177770,%0      ;ONLY THREE BITS PASS
2312 014014  122700  000001            CMPB    #1,%0           ;CHECK FOR A "1" TYPED
2313 014020  001006                    BNE     10$             ;BR, IF IT WASN'T A ONE
2314 014022  012737  000001  001112    MOV     #1,EIA          ;SET FOR EIA
2315 014030  004737  013642            JSR     %7,GETCR        ;WAIT FOR CR
2316 014034  000413                    BR      100$            ;EXIT
2317 014036  122700  000002    10$:    CMPB    #2,%0           ;CHECK FOR 20MA
2318 014042  001340                    BNE     1$              ;BR, NEITHER
2319 014044  005037  001112            CLR     EIA             ;SET TO 20 MA
2320 014050  004737  013642            JSR     %7,GETCR        ;WAIT FOR CR
2321 014054  000403                    BR      100$            ;EXIT
2322 014056  012737  000001  001112  20$:    MOV     #1,EIA          ;CR = DEFAULT = EIA
2323 014064  000207            100$:   RTS     %7              ;EXIT SUBROUTINE
2324
2325
2326                            ;*****************************************************************************
2327                            ;
2328                            ;       THIS SUBROUTINE PICKS CHARACTERS OUT OF THE TTY RECEIVE BUFFER
2329                            ;       I.E. GETS RID OF THE "EXTRA" ONES
2330                            ;
2331 014066                    CLRTTY:
2332 014066  012703  040000            MOV     #40000,%3       ;BIG COUNTER
2333 014072  105777  164724    1$:     TSTB    @TKS            ;CHECK FOR RECEIVER DONE SET
2334 014076  100007                    BPL     5$              ;BR, IF DONE IS NOT SET
2335 014100  117700  164712            MOVB    @TKB,%0         ;PICK UP THE CHARACTER
2336 014104  105777  164710    3$:     TSTB    @TPS            ;CHECK FOR TRANSMITTER READY
2337 014110  100375                    BPL     3$              ;WAIT UNTIL READY
2338 014112  110077  164676            MOVB    %0,@TPB         ;ECHO THE "EXTRA" CHARACTER
2339 014116  077313    5$:     SOB     %3,1$           ;LOOP COUNTER
2340 014120  000207            RTS     %7              ;RETURN TO CALLEE
```

```
                                ;*****************************************************************************
                                ;
                                ;   TEST FOR ERROR,
                                ;       THIS ROUTINE TAKES THE PLACE OF ALL THE "TST @LPS" CODES.
                                ;       THIS WAS REQUIRED FOR THE DZ11. IF THE DZ IS NOT SELECTED
                                ;       THE ROUTINE SIMPLY DOES THE "TST @LPS" AND RETURNS. IF THERE
                                ;       IS A DZ11 IT ONLY CHECKS A DUMMY WORD TO SHOW NO ERROR.
                                ;       THERE ARE NO TRANSMIT ERRORS ON THE DZ11.
014122                          ERCHK:
014122  005737  001102                  TST     BRATE                   ;CHECK FIRST FOR DZ
014126  001403                          BEQ     50$                     ;BR, IF NO DZ SELECTED
014130  005737  014144                  TST     TSTWRD                  ;DUMMY CONDITION CODE SETUP
014134  000207                  20$:    RTS     %7                      ;RETURN WITH NO MINUS SET
014136  005777  164636          50$:    TST     @LPS                    ;NOT DZ SO DO REGULAR CHECK
014142  000774                          BR      20$                     ;RETURN WITH CODES ETC
014144  000200                  TSTWRD: .WORD   000200                  ;NO MINUS BIT SET
                                ;*****************************************************************************
                                ;
                                ;   TEST FOR READY,
                                ;       THIS ROUTINE TAKES THE PLACE OF ALL THE "TSTB @LPS" CODES.
                                ;       THIS WAS REQUIRED FOR THE DZ11. IF THE DZ IS NOT SELECTED
                                ;       THE ROUTINE SIMPLY DOES THE "TSTB @LPS" AND RETURNS. IF THERE
                                ;       IS A DZ11 IT CHECKS THE REAL READY BIT (BIT15).
                                ;       THE ROUTINE DOES A "TST @LPS" WHICH SETS THE CORRECT CONDITION
                                ;       CODES
                                ;
014146                          ERCHKB:
014146  005737  001106                  TST     DLHERE                  ;ARE WE TALKING TO A DL
014152  001047                          BNE     80$                     ;BR, IF DL IS BEING USED
014154  005737  001102                  TST     BRATE                   ;CHECK FIRST FOR DZ
014160  001416                          BEQ     50$                     ;BR, IF NO DZ SELECTED
014162  005737  001112                  TST     EI4                     ;WHAT MODE ARE WE IN
014166  001404                          BEQ     5$                      ;BR, IF IN 20 MA MODE
014170  137737  164714  001066          BITB    @DZTCRA,DZTCR           ;IS DATA TERMINAL READY SET
014176  001465                          BEQ     100$                    ;BR IF IT IS NOT SET (ERROR)
014200  037727  164574  000200  5$:     BIT     @LPS,#200               ;RECEIVER DONE SET
014206  001006                          BNE     60$                     ;BR, IF RECVR HAS CHARACTER
014210  005777  164564          10$:    TST     @LPS                    ;LOOKS AT REAL READY BIT
014214  000207                  20$:    RTS     %7                      ;RETURN WITH NO MINUS SET
014216  105777  164556          50$:    TSTB    @LPS                    ;NOT DZ SO DO REGULAR CHECK
014222  000774                          BR      20$                     ;RETURN WITH CODES ETC
014224  017705  164666          60$:    MOV     @DZRBUF,%5              ;PICK UP CHARACTER
014230  042705  177700                  BIC     #177700,%5              ;ONLY 6 BITS PASS
014234  120527  000023                  CMPB    %5,#23                  ;CHK FOR XOFF (CNTL S)
014240  001363                          BNE     10$                     ;BR, IF NOT XOFF (RETURN)
014242  037727  164532  000200  70$:    BIT     @LPS,#200               ;LOOK FOR ANOTHER CHARACTER
014250  001774                          BEQ     70$                     ;WAIT HERE FOR NEXT CHARACTER
014252  017705  164640                  MOV     @DZRBUF,%5              ;IT ARRIVED
014256  042705  177700                  BIC     #177700,%5              ;ONLY SIX BITS PASS
014262  120527  000021                  CMPB    %5,#21                  ;WAS IT XON (CNTL Q)
014266  001365                          BNE     70$                     ;BR, IF IT WASN'T XON (LOOP)
014270  000747                          BR      10$                     ;IT WAS XON RETURN
014272  032777  000200  164576  80$:    BIT     #200,@DLLPS             ;HAVE WE RECEIVED A RECV CHARA
014300  001746                          BEQ     50$                     ;BR, IF WE HAVE NOT
014302  117705  164572                  MOVB    @DLRBUF,%5              ;PICK UP THE CHARACTER
014306  042705  177700                  BIC     #177700,%5              ;ONLY SIX BITS
```

```
 2399 014312  120527  000023                    CMPB    %5,#23          ;WAS IT AN XOFF
 2400 014316  001337                            BNE     50$             ;BR, IF NOT (FORGET IT)
 2401 014320  032777  000200  164550  90$:      BIT     #200,@DLLPS     ;DID ANOTHER ARIVE
 2402 014326  001774                            BEQ     90$             ;BR, IF NO OTHER CHARACTER
 2403 014330  117705  164544                    MOVB    @DLRBUF,%5      ;WELL PICK IT UP
 2404 014334  042705  177700                    BIC     #177700,%5      ;ONLY SIX BITS
 2405 014340  120527  000021                    CMPB    %5,#21          ;WAS IT AN XON
 2406 014344  001365                            BNE     90$             ;LOOP IF NOT XON
 2407 014346  000137  014216                    JMP     50$             ;XON = DO SOME MORE
 2408 014352  104000                  100$:     EMT     +0              ;CALL TYP ROUTINE
 2409 014354  015536                            PTRDTR                  ;"PRINTER IS NOT SENDING DTR"
 2410 014356  000000                            HALT                    ;STOP
 2411 014360  000137  014146                    JMP     ERCHKB          ;LOOP AROUND
```

```
                        ;*****************************************************************************
                        ;
                        ;
                        ;       THIS ROUTINE CHECKS FOR DL11S THAT NEED ADDITIONAL PROGRAMMING
                        ;       FEATURES I.E. PROGRAMABLE BAUD RATES ON THE DLV11-E/F MODULES.
                        ;
014364                  DLSET:
014364  104000                  EMT     +0                              ;PRINT CALL
014366  015302                  DLASK1                                  ;"DL11 TYPE MENU"
014370  104000                  EMT     +0                              ;PRINT MESSAGE CALL
014372  015335                  DLASK2                                  ;"1  DLV11-F OR DLV11-E"
014374  104000                  EMT     +0                              ;PRINT MESSAGE CALL
014376  015365                  DLASK3                                  ;"2  DLV11, DL11 OR DLV11-J"
014400  104000                  EMT     +0                              ;PRINT MESSAGE CALL
014402  015421                  DLASK4                                  ;"ENTER MENU SELECTION"
014404  105777  164412    40$:  TSTB    @TKS                            ;READ TTY'S STATUS
014410  100375                  BPL     40$                             ;BR, IF NOT DONE
014412  117700  164400          MOVB    @TKB,%0                         ;PICK UP CHAR. TYPED
014416  105777  164376    45$:  TSTB    @TPS                            ;CHECK FOR BUSY
014422  100375                  BPL     45$                             ;LOOP IF TTY IS BUSY (BR)
014424  110077  164364          MOVB    %0,@TPB                         ;ECHO CHARACTER TO TTY
014430  042700  177700          BIC     #177700,%0                      ;ONLY 6 BITS ALLOWED
014434  122700  000012          CMPB    #12,%0                          ;CHK FOR <LF> DEFAULT
014440  001416                  BEQ     75$                             ;BR, IF DEFAULT (NO)
014442  122700  000015          CMPB    #15,%0                          ;CHK FOR <CR> DEFAULT
014446  001413                  BEQ     75$                             ;BR, IF DEFAULT (NO)
014450  120027  000061          CMPB    %0,#61                          ;WAS IT AN ASCII 1
014454  001421                  BEQ     100$                            ;BR, IF IT WAS (DL11 SEL.)
014456  120027  000062          CMPB    %0,#62                          ;WAS IT AN ASCII 2
014462  001350                  BNE     40$
014464  004737  013642          JSR     %7,GETCR                        ;GO GET LF OR CR
014470  020027  123456          CMP     %0,#123456                      ;CHECK FOR BAD INPUT
014474  001743                  BEQ     40$                             ;TRY AGAIN BAD INPUT (NOT CR)
014476  004737  014066    75$:  JSR     %7,CLRTTY                       ;CLEAR OUT THE TTY
014502  000207                  RTS     %7                              ;RETURN TO CALLEE
014504  004737  013642    76$:  JSR     %7,GETCR                        ;LOOK FOR CR
014510  020027  123456          CMP     %0,#123456                      ;WAS THERE AN ERROR
014514  001403                  BEQ     105$                            ;BR, IF AN ERROR RESTART
014516  000767                  BR      75$                             ;EXIT NO ERROR
014520                  100$:
014520  004737  013642          JSR     %7,GETCR                        ;GET CR
014524  104000    105$: EMT     +0                                      ;CALL PRINT ROUTINE
014526  015455                  DLASK5                                  ;"DOES DL HAVE PROG BAUD RATE"
014530  012737  000001  001102  MOV     #1,BRATE                        ;SWITCH FOR BAUD RATE NEEDED
014536  105777  164260    110$: TSTB    @TKS                            ;CK TTY IN STATUS
014542  100375                  BPL     110$                            ;WAIT FOR DONE
014544  117700  164246          MOVB    @TKB,%0                         ;PICK UP CHARACTER
014550  105777  164244    115$: TSTB    @TPS                            ;CHECK FOR BUSY
014554  100375                  BPL     115$                            ;LOOP IF TTY IS BUSY (BR)
014556  110077  164232          MOVB    %0,@TPB                         ;ECHO CHARACTER
014562  042700  177600          BIC     #177600,%0                      ;ONLY 7 BIT PASS
014566  122700  000012          CMPB    #12,%0                          ;CHK FOR <LF> DEFAULT
014572  001741                  BEQ     75$                             ;BR, IF DEFAULT (NO)
014574  122700  000015          CMPB    #15,%0                          ;CHK FOR <CR> DEFAULT
014600  001736                  BEQ     75$                             ;BR, IF DEFAULT (NO)
014602  122700  000131          CMPB    #131,%0                         ;WAS AN ASCII Y TYPED
```

```
2470 014606  001412
2471 014610  122700  000116          120$:  CMPB  #116,%0      ;BR, IF IT WASN'T A Y
2472 014614  001343                         BNE   105$         ;CHECK FOR A N
2473 014616  004737  013642                 JSR   %7,GETCR     ;BR, IF NOT A N (ERROR)
2474 014622  020027  123456                 CMP   %0,#123456   ;GO GET LF OR CR
2475 014626  001736                         BEQ   105$         ;CHECK FOR BAD INPUT
2476 014630  000137  014476                 JMP   75$          ;TRY AGAIN BAD INPUT (NOT CR)
2477 014634  004737  013642          497$:  JSR   %7,GETCR     ;EXIT ROUTINE
2478 014640  020027  123456                 CMP   %0,#123456   ;GO GET LF OR CR
2479 014644  001727                         BEQ   105$         ;CHECK FOR BAD INPUT
2480 014646  004737  014066          498$:  JSR   %7,CLRTTY     ;TRY AGAIN BAD INPUT (NOT CR)
2481 014652  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2482 014654  016647                         LN1A               ;CALL TO THE TTY PRINTER
2483 014656  004737  014066                 JSR   %7,CLRTTY    ;BAUD RATE MENU PRINTOUTS
2484 014662  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2485 014664  016701                         LN2                ;CALL TO THE TTY PRINTER
2486 014666  004737  014066                 JSR   %7,CLRTTY    ;BAUD RATE MENU PRINTOUTS
2487 014672  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2488 014674  016722                         LN3                ;CALL TO THE TTY PRINTER
2489 014676  004737  014066                 JSR   %7,CLRTTY    ;BAUD RATE MENU PRINTOUTS
2490 014702  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2491 014704  016743                         LN4                ;CALL TO THE TTY PRINTER
2492 014706  004737  014066                 JSR   %7,CLRTTY    ;BAUD RATE MENU PRINTOUTS
2493 014712  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2494 014714  016764                         LN5                ;CALL TO THE TTY PRINTER
2495 014716  004737  014066                 JSR   %7,CLRTTY    ;BAUD RATE MENU PRINTOUTS
2496 014722  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2497 014724  017005                         LN6                ;CALL TO THE TTY PRINTER
2498 014726  104000                         EMT   +0           ;BAUD RATE MENU PRINTOUTS
2499 014730  017026                         LN7                ;CALL TO THE TTY PRINTER
2500 014732  004737  014066                 JSR   %7,CLRTTY    ;BAUD RATE MENU PRINTOUTS
2501 014736  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2502 014740  017047                         LN8                ;CALL TO THE TTY PRINTER
2503 014742  104000                         EMT   +0           ;BAUD RATE MENU PRINTOUTS
2504 014744  017070                         LN8A               ;CALL TO SUBROUTINE
2505 014746  004737  014066                 JSR   %7,CLRTTY    ;"8  =19200 BAUD"
2506 014752  104000                         EMT   +0           ;PICK UP PENDING CHARACTERS
2507 014754  017111                         LN10               ;CALL TO THE TTY PRINTER
2508 014756  105777  164040          500$:  TSTB  @TKS         ;BAUD RATE MENU PRINTOUTS
2509 014762  100375                         BPL   500$         ;CHK TTY IN STATUS
2510 014764  117700  164026                 MOVB  @TKB,%0      ;WAIT FOR DONE
2511 014770  105777  164024          505$:  TSTB  @TPS         ;PICK UP CHARACTER TYPED IN
2512 014774  100375                         BPL   505$         ;CHECK FOR BUSY
2513 014776  110077  164012                 MOVB  %0,@TPB      ;LOOP UNTIL NOT BUSY
2514 015002  042700  177700                 BIC   #177700,%0   ;ECHO CHARACTER
2515 015006  122700  000012                 CMPB  #12,%0       ;ONLY 6 BIT PASS
2516 015012  001475                         BEQ   544$         ;WAS DEFAULT SEL. <LF>
2517 015014  122700  000015                 CMPB  #15,%0       ;BR, IF DEFAULT SELECTED
2518 015020  001472                         BEQ   544$         ;WAS DEFAULT SEL. <CR>
2519 015022  005037  001124                 CLR   WORK         ;BR, IF DEFAULT SELECTED
2520 015026  042700  177760                 BIC   #177760,%0   ;CLEAR WORK AREA
2521 015032  122700  000001                 CMPB  #1,%0        ;ONLY FOUR BITS PASS
2522 015036  001003                         BNE   510$         ;CHECK FOR A "1" TYPED
2523 015040  012737  040000  001124        MOV   #40000,WORK  ;BR, IF IT WASN'T A ONE
2524 015046                          510$:                     ;SET FOR 150 BAUD
2525 015046  122700  000002                 CMPB  #2,%0        ;CHECK FOR A TWO TYPED
2526 015052  001003                         BNE   520$         ;BR, IF IT WASN'T A "2"
```

```
2527 015054 012737 050000 001124          520$:  MOV     #50000,WORK     ;SET FOR 300 BAUD
2528 015062
2529 015062 122700 000003                        CMPB    #3,%0           ;CHECK FOR A THREE TYPED
2530 015066 001003                               BNE     521$            ;BR, IF IT WASN'T A "3"
2531 015070 012737 060000 001124                 MOV     #60000,WORK     ;SET FOR 600 BAUD
2532 015076                               521$:
2533 015076 122700 000004                        CMPB    #4,%0           ;CHECK FOR A FOUR TYPED
2534 015102 001003                               BNE     522$            ;BR, IF IT WASN'T A "4"
2535 015104 012737 070000 001124                 MOV     #70000,WORK     ;SET FOR 1200 BAUD
2536 015112                               522$:
2537 015112 122700 000005                        CMPB    #5,%0           ;CHECK FOR A FIVE TYPED
2538 015116 001003                               BNE     523$            ;BR, IF IT WASN'T A "5"
2539 015120 012737 120000 001124                 MOV     #120000,WORK    ;SET FOR 2400 BAUD
2540 015126                               523$:
2541 015126 122700 000006                        CMPB    #6,%0           ;CHECK FOR A SIX TYPED
2542 015132 001003                               BNE     524$            ;BR, IF IT WASN'T A "6"
2543 015134 012737 140000 001124                 MOV     #140000,WORK    ;SET FOR 4800 BAUD
2544 015142                               524$:
2545 015142 122700 000007                        CMPB    #7,%0           ;CHECK FOR A SEVEN TYPED
2546 015146 001003                               BNE     525$            ;BR, IF IT WASN'T A "7"
2547 015150 012737 160000 001124          543$:  MOV     #160000,WORK    ;SET FOR 9600 BAUD
2548 015156                               525$:
2549 015156 122700 000010                        CMPB    #8.,%0          ;WAS AN ASCII EIGHT TYPED
2550 015162 001015                               BNE     526$            ;IT WASN'T AN EIGHT
2551 015164 012737 170000 001124                 MOV     #170000,WORK    ;SET FOR 19200KC
2552 015172 004737 013642                        JSR     %7,GETCR        ;GO GET LF OR CR
2553 015176 020027 123456                        CMP     %0,#123456      ;CHECK FOR BAD INPUT
2554 015202 001621                               BEQ     498$            ;BR, IF CR RECEIVED (GOOD)
2555 015204 000415                               BR      530$            ;LEAVE
2556 015206 012737 160000 001124          544$:  MOV     #160000,WORK    ;DEFAULT IS 9600
2557 015214 000411                               BR      530$            ;LEAVE
2558 015216                               526$:
2559 015216 023727 001124 000000          527$:  CMP     WORK,#0         ;CHECK FOR NOT SET
2560 015224 001610                               BEQ     498$            ;BR, IF NUMBERS NOT VALID
2561 015226 004737 013642                        JSR     %7,GETCR        ;GO GET LF OR CR
2562 015232 020027 123456                        CMP     %0,#123456      ;CHECK FOR BAD INPUT
2563 015236 001603                               BEQ     498$            ;TRY AGAIN BAD INPUT (NOT CR)
2564 015240                               530$:
2565 015240 013737 001124 001102          540$:  MOV     WORK,BRATE      ;STORE BAUD RATE FOR LATER
2566 015246 053737 001124 001074                 BIS     WORK,DLLPR      ;PUT BAUD RATE IN PLACE
2567                                      ;      SET UP ACTUAL DL REGISTERS
2568                                      ;
2569 015254 013703 001000                        MOV     LPS,%3          ;GET STATUS REG ADDRESS
2570 015260 062703 000002                        ADD     #2,%3           ;POINT TO LPR REGISTER
2571 015264 052737 004000 001074                 BIS     #4000,DLLPR     ;SET PBR ENB BIT
2572 015272 013713 001074                        MOV     DLLPR,(%3)      ;SET SPEED, LINE, PARITY ETC
2573                                      ;      SHOULD BE ALL SET
2574                                      ;
2575 015276 000137 014476                        JMP     75$             ;NO MORE DL QUESTIONS
```

```
                                        .SBTTL SERIAL LINE SETUP MESSAGES

                                        ;******************************************************************************
                                        ;
                                        ;       OPERATOR MESSAGES
                                        ;
                                        ;
015302      012     015     124 DLASK1: .ASCIZ  <12><15>/TYPE OF DL11 MODULE MENU/
015335      012     015     061 DLASK2: .ASCIZ  <12><15>/1  DLV11-E OR DLV11-F/
015365      012     015     062 DLASK3: .ASCIZ  <12><15>/2  DLV11, DL11 OR DLV11-J/
015421      012     015     124 DLASK4: .ASCIZ  <12><15>/TYPE MENU SELECTION <2>? /
015455      012     015     104 DLASK5: .ASCIZ  <12><15>/DOES DL HAVE PROGRAM SELECTABLE BAUD RATE <N>?/
015536      012     015     120 PTRDTR: .ASCIZ  <12><15>/PRINTER IS NOT SENDING DATA TERMINAL READY/
015613      012     015     123 MENU10: .ASCIZ  <12><15>/SERIAL LINE TYPE MENU/
015643      012     015     061 MENU20: .ASCIZ  <12><15>/1  EIA/
015654      012     015     062 MENU30: .ASCIZ  <12><15>/2  20 MA/
015667      012     015     124 MENU40: .ASCIZ  <12><15>/TYPE MENU SELECTION <1>? /
015723      012     015     123 MENU1:  .ASCIZ  <12><15>/SERIAL LINE SELECTION MENU/
01576C      012     015     061 MENU2:  .ASCIZ  <12><15>/1  DL11 SERIAL LINE/
016006      012     015     062 MENU3:  .ASCIZ  <12><15>/2  DZ11 SERIAL LINE/
016034      012     015     124 MENU4:  .ASCIZ  <12><15>/TYPE MENU SELECTION <1>? /
016070      012     015     042 MENUD1: .ASCIZ  <12><15>/"DL" PROGRAMMABLE MAINTENANCE LOOPBACK FEATURE MENU/
016156      012     015     061 MENUD2: .ASCIZ  <12><15>)1  YES (SUCH AS DL11, DLV11-E/F))
016221      012     015     062 MENUD3: .ASCIZ  <12><15>)2  NO  (SUCH AS DLV11, DLV11-J))
016263      012     015     124 MENUD4: .ASCIZ  <12><15>/TYPE MENU SELECTION <1>? /
016317      012     015     124 DLCSRM: .ASCIZ  <12><15>/TYPE DL11'S CSR ADDRESS <776500>? /
016364      012     015     124 DLVECM: .ASCIZ  <12><15>/TYPE DL11'S VECTOR ADDRESS <300>? /
016431      012     015     124 DZCSRM: .ASCIZ  <12><15>/TYPE DZ11'S CSR ADDRESS <760100>? /
016476      012     015     124 DZVECM: .ASCIZ  <12><15>/TYPE DZ11'S VECTOR ADDRESS <300>? /
016543      012     015     000 DLCRLF: .ASCIZ  <12><15><00>
016547      012     015     124 DZLINE: .ASCIZ  <12><15>/TYPE DZ11'S LINE NUMBER 0-7 < 0 >? /
016615      012     015     040 LN1:    .ASCIZ  <12><15>/ ENTER DZ11'S BAUD RATE/
016647      012     015     040 LN1A:   .ASCIZ  <12><15>/ ENTER DL11'S BAUD RATE/
016701      012     015     040 LN2:    .ASCIZ  <12><15>/ 1 =  150 BAUD/
016722      012     015     040 LN3:    .ASCIZ  <12><15>/ 2 =  300 BAUD/
016743      012     015     040 LN4:    .ASCIZ  <12><15>/ 3 =  600 BAUD/
016764      012     015     040 LN5:    .ASCIZ  <12><15>/ 4 = 1200 BAUD/
017005      012     015     040 LN6:    .ASCIZ  <12><15>/ 5 = 2400 BAUD/
017026      012     015     040 LN7:    .ASCIZ  <12><15>/ 6 = 4800 BAUD/
017047      012     015     040 LN8:    .ASCIZ  <12><15>/ 7 = 9600 BAUD/
017070      012     015     040 LN8A:   .ASCIZ  <12><15>/ 8 =19200 BAUD/
017111      012     015     124 LN10:   .ASCIZ  <12><15>/TYPE BAUD RATE SELECTION < 7 >?/
017153      012     015     124 STOPM:  .ASCIZ  <12><15>/TYPE NUMBER OF STOP BITS 1 OR 2 < 1 >?/
017224      012     015     124 DATAM:  .ASCIZ  <12><15>/TYPE NUMBER OF DATA BITS 7 OR 8 < 7 >?/
017275      012     015     111 PARITY: .ASCIZ  <12><15>/IS PARITY SELECTED Y OR N < N >?/
017340      012     015     111 PARITZ: .ASCIZ  <12><15>/IS PARITY ODD (O) OR EVEN (E) < O >?/
                                        .EVEN
                                        .NLIST ME
                                        .LIST BEX
017410  000000                  LSTADR: .WORD   0                       ;LAST LOCATION IN PROGRAM
        000200                          .END    START
```

CZLCPA COLOR PRINTER DIAGNOSTIC MACRO M1113 12-MAR-85 15:38 SEQ 0066
SYMBOL TABLE

```
ACNVX   004576
ACVN    004514
BEGIN   000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BRATE   001102
BUFF    001166
CHAR    001160
CHRCNT  001056
CHRGEN  001060
CLRTTY  014066
CNTRLG  005216
CN1     001666
CN10    003322
CN12    003550
CN2     002072
CN3     002242
CN4     002326
CN5     002530
CN6     002700
CN7     002764
CONR1   004600
CONR2   004602
CONR3   004604
CONV    004466
CSBR    001024
CYCCNT  001064
DAR1    005512
DAR2    005516
DAR6    005551
DAR9    005567
DATAM   017224
DGTS    005200
DIGITS  001152
DISPLA  001006
DISPRE  000174
DLASK1  015302
DLASK2  015335
DLASK3  015365
DLASK4  015421
DLASK5  015455
DLCRLF  016543
DLCSR   001034
DLCSRC  001032
DLCSRM  016317
DLHERE  001106
DLLPR   001074
DLLPS   001076
DLRATE  001104
DLRBUF  001100
DLSET   014364
DLTYPE  001114
DLVEC   001042
DLVECM  016364
DZCSR   001036
DZCSRC  001040
DZCSRH  001120
DZCSRM  016431
DZLINE  016547
DZLNE   001072
DZLPR   001070
DZRBUF  001116
DZTCR   001066
DZTCRA  001110
DZVEC   001044
DZVECM  016476
EIA     001112
EIACHK  013720
EL    = 000037
ERCHK   014122
ERCHKB  014146
ERCOUN  001136
ERMS1   006704
ERMS2   006756
ERMS3   007026
ERMS4   007120
ERMS5   007211
ERMS6   007305
ERR1    001620
ERR10   003254
ERR11   003362
ERR12   003502
ERR13   004262
ERR2    002024
ERR3    002174
ERR4    002260
ERR5    002462
ERR6    002632
ERR7    002716
ETIM    005731
ETIMO   006007
GETCR   013642
GETOCT  013560
HED1    006063
HED2    006104
HLP0    007350
HLP1    007420
HLP2    007514
HLP3    007541
HLP4    007611
HLP4A   007672
HLP5    007752
HLP6    010040
HLP7    010100
HLP8    010163
HWSWR   001012
LCP3    005523
LCP4    005533
LCP5    005542
LCP7    005556
LEGCHR  001144
LINCNT  001062
LKS     001030
LN1     016615
LN1A    016647
LN10    017111
LN2     016701
LN3     016722
LN4     016743
LN5     016764
LN6     017005
LN7     017026
LN8     017047
LN8A    017070
LPB     001002
LPS     001000
LSTADR  017410
M     = 000002
MAINTB  001122
MENUD1  016070
MENUD2  016156
MENUD3  016221
MENUD4  016263
MENU1   015723
MENU10  015613
MENU2   015760
MENU20  015643
MENU3   016006
MENU30  015654
MENU4   016034
MENU40  015667
MESDD   006114
MES1    006153
MES10   006504
MES14   006570
MES2    006167
MES20A  006607
MES21   006612
MES22   006614
MES23   006620
MES24   006630
MES25   006647
MES26   006656
MES27   006672
MES3    006230
MES4    006246
MES5    006274
MES6    006347
MES7    006416
MES8    006457
MES9    006475
N     = 000014
NOP   = 000240
NUMCHR  001146
OCT     001162
OFFSET  001150
PARITY  017275
PARITZ  017340
PASSA   001164
PLKS    001026
PRINE   004250
PRNNT   004242
PRTMSG  004312
PSW     001010
PTRC    001046
PTRDTR  015536
PTRPSW  001052
PTRVEC  001050
RINT    004274
R6    =%000006
R7    =%000007
SAVE    001134
SEGCNT  001054
SET     001156
SETSER  010234
SETUP   001306
SIGNAL  001154
SL    = 000036
STAER   004606
STARN   004752
START   000200 G
STEDA   004714
STEXT   004744
STOPM   017153
STRCHR  001140
STRCNT  001142
SWR     001004
SWREG   000176
TEST1   001464
TEST2   003052
TEST3   003640
TIME    001130
TIMER   001132
TKB     001016
TKHLP   005440
TKINT   004754
TKS     001022
TK1     005066
TK2     005100
TK3     005366
TK4     005372
TK5     005170
TK6     005406
TK7     005376
TK9     005424
TPB     001014
TPS     001020
TSEAB   004176
TSEDA   004202
TSEND   004070
TSRSS   004232
TSRST   004236
TSTWRD  014144
TYO1    005571
TYO2    005630
TYO3    005673
TYP     004332
TYPA    004342
TYPC    004352
TYPD    004400
TYPDAT  004464
TYPDO   004426
TYPDO1  004434
TYPF    004436
TYPG    004450
TYPINT  004314
TYPSWR  005272
TYPSWX  005336
W     = 000007
WAIT1   001730
WAIT2   002100
WAIT3   002366
WAIT4   002536
WAIT5   003142
WAIT6   003406
WDD1    001736
WDD2    002106
WDD3    002374
WDD4    002544
WDD5    003166
WDD6    003414
WDE1    001752
WDE2    002122
WDE3    002410
WDE4    002560
WDE5    003202
WDE6    003430
WER1    001770
WER2    002140
WER3    002426
WER4    002576
WER5    003220
WER6    003446
WEX1    002074
WEX2    002244
WEX3    002532
WEX4    002702
WEX5    003324
WEX6    003554
WORK    001124
WORKA   001126
```

```
CZLCPA COLOR PRINTER DIAGNOSTIC  MACRO M1113  12-MAR-85 15:38
SYMBOL TABLE                                                    SEQ 0067

WT2     005114          WT3     005242

. ABS.  017412      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  4362 WORDS  ( 18 PAGES)
DYNAMIC MEMORY:  7630 WORDS  ( 29 PAGES)
ELAPSED TIME:  00:01:37
CZLCPA,CZLCPA/-SP/CR=CZLCPA
```

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0068

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES
ACNVX	004576	*25-1423 25-1427 *25-1432 *25-1434 *25-1436 #25-1445
ACVN	004514	#25-1427 25-1438
BEGIN	000000	#15-608 15-613 15-619 15-623 15-625 15-629 15-634 15-642 16-652 16-662 16-665
BIT0	= 000001	#14-599
BIT1	= 000002	#14-598
BIT10	= 002000	#14-589
BIT11	= 004000	#14-588
BIT12	= 010000	#14-587
BIT13	= 020000	#14-586
BIT14	= 040000	#14-585 15-626
BIT15	= 100000	#14-584
BIT2	= 000004	#14-597
BIT3	= 000010	#14-596
BIT4	= 000020	#14-595
BIT5	= 000040	#14-594
BIT6	= 000100	#14-593
BIT7	= 000200	#14-592
BIT8	= 000400	#14-591
BIT9	= 001000	#14-590
BRATE	001102	#18-975 *28-1701 *28-1743 *28-1780 *28-2091 30-2352 30-2372 *31-2457 *31-2565
BUFF	001166	#18-1004 22-1209 22-1225 22-1237
CHAR	001160	#18-1001 *26-1501 *26-1502 26-1503 26-1505 26-1507 26-1509 26-1512 26-1515 26-1518 26-1526 *26-1527 26-1536 *26-1542 *26-1543 26-1544 26-1548
CHRCNT	001056	#18-963
CHRGEN	001060	#18-964
CLRTTY	014066	24-1344 25-1478 28-1744 28-1781 28-1858 28-1861 28-1870 28-1962 28-1968 28-2011 28-2014 28-2017 28-2020 28-2023 28-2026 28-2029 28-2032 28-2035 28-2097 28-2125 28-2130 28-2161 28-2166 28-2195 28-2200 28-2228 29-2289 29-2291 #29-2331 31-2447 31-2480 31-2483 31-2486 31-2489 31-2492 31-2495 31-2500 31-2505
CNTRLG	005216	#26-1542
CN1	001666	20-1085 #20-1085
CN10	003322	22-1215 #22-1215
CN12	003550	22-1242 #22-1242
CN2	002072	21-1117 #21-1117
CN3	002242	21-1120 #21-1120
CN4	002326	21-1128 #21-1128
CN5	002530	21-1148 #21-1148
CN6	002700	21-1151 #21-1151
CN7	002764	21-1159 #21-1159
CONR1	004600	*25-1419 25-1439 #25-1446
CONR2	004602	*25-1420 25-1440 #25-1447
CONR3	004604	*25-1421 25-1441 #25-1448
CONV	004466	24-1326 #25-1419 25-1455 26-1561 26-1569
CSBR	001024	#16-686
CYCCNT	001064	#18-966 *19-1013 22-1215 23-1269 *24-1323 24-1327
DAR1	005512	#27-1629
DAR2	005516	#27-1630
DAR6	005551	#27-1634
DAR9	005567	#27-1636
DATAM	017224	28-2132 #32-2620

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0069

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES
DGTS	005200	26-1506 #26-1538
DIGITS	001152	#18-998 *26-1521 26-1522 26-1528 26-1538 *26-1581
DISPLA	001006	#16-679 *19-1023
DISPRE	000174	#15-643 19-1023
DLASK1	015302	31-2422 #32-2584
DLASK2	015335	31-2424 #32-2585
DLASK3	015365	31-2426 #32-2586
DLASK4	015421	31-2428 #32-2587
DLASK5	015455	31-2456 #32-2588
DLCRLF	016543	28-1864 #32-2606
DLCSR	001034	#16-691
DLCSRC	001032	#16-690 28-1846 28-1848 28-1849
DLCSRM	016317	28-1784 #32-2602
DLHERE	001106	#18-977 *28-1700 *28-1742 *28-1779 *28-1873 30-2370
DLLPR	001074	#18-972 *31-2566 *31-2571 31-2572
DLLPS	001076	#18-973 21-1120 21-1151 22-1215 22-1242 *28-1838 *28-1848 30-2395 30-2401
DLRATE	001104	#18-976
DLRBUF	001100	#18-974 21-1122 21-1153 22-1216 22-1244 *28-1839 *28-1840 *28-1849 *28-1850 30-2397 30-2403
DLSET	014364	28-1860 #31-2420
DLTYPE	001114	#18-982
DLVEC	001042	#16-694
DLVECM	016364	#32-2603
DZCSR	001036	#16-692
DZCSRC	001040	#16-693 28-1935
DZCSRH	001120	#18-984 25-1394 *28-1697 *28-1959
DZCSRM	016431	28-1872 #32-2604
DZLINE	016547	28-1970 #32-2607
DZLNE	001072	#18-970 *28-1990 *28-1992 28-1996 28-1997
DZLPR	001070	#18-969 *28-1699 *28-1996 *28-2092 *28-2116 *28-2123 *28-2147 *28-2149 *28-2159 *28-2185 *28-2192 *28-2219 *28-2226 *28-2239 28-2240
DZRBUF	001116	#18-983 *28-2238 30-2384 30-2390
DZTCR	001066	#18-968 *28-1698 *28-1993 *28-2004 28-2243 30-2376
DZTCRA	001110	#18-979 *28-2244 *28-2245 30-2376
DZVEC	001044	#16-695
DZVECM	016476	#32-2605
EIA	001112	#18-980 *29-2314 *29-2319 *29-2322 30-2374
EIACHK	013720	#29-2291
EL	= 000037	#14-602
ERCHK	014122	25-1361 #30-2351
ERCHKB	014146	#30-2369 30-2411
ERCOUN	001136	#18-992 *20-1085 *21-1117 *21-1120 *21-1128 *21-1148 *21-1151 *21-1159 *22-1215 *22-1232 *22-1242 *25-1365 25-1456
ERMS1	006704	20-1084 #27-1671 28-1950
ERMS2	006756	21-1127 21-1158 #27-1672
ERMS3	007026	#27-1673
ERMS4	007120	22-1231 #27-1674
ERMS5	007211	#27-1675
ERMS6	007305	25-1364 #27-1676
ERR1	001620	#20-1085
ERR10	003254	#22-1215
ERR11	003362	#22-1232

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0070

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES								
ERR12	003502	#22-1242								
ERR13	004262	#25-1365								
ERR2	002024	#21-1117								
ERR3	002174	#21-1120								
ERR4	002260	#21-1128								
ERR5	002462	#21-1148								
ERR6	002632	#21-1151								
ERR7	002716	#21-1159								
ETIM	005731	21-1120	21-1151	22-1215	22-1242	#27-1642				
ETIMO	006007	21-1117	21-1148	#27-1643						
GETCR	013642	28-1727	28-1739	28-1765	28-1776	28-1854	28-1931	28-2005	28-2078	28-2087
		28-2113	28-2120	28-2150	28-2156	28-2182	28-2189	28-2216	28-2223	#29-2276
		29-2315	29-2320	31-2444	31-2449	31-2454	31-2473	31-2477	31-2552	31-2561
GETOCT	013560	28-1802	28-1810	28-1818	28-1826	28-1834	28-1892	28-1900	28-1908	28-1916
		28-1924	#29-2256							
HED1	006063	25-1463	#27-1644							
HED2	006104	25-1457	#27-1645							
HLP0	007350	26-1598	#27-1678							
HLP1	007420	26-1600	#27-1679							
HLP2	007514	26-1602	#27-1680							
HLP3	007541	26-1604	#27-1681							
HLP4	007611	26-1606	#27-1682							
HLP4A	007672	26-1608	#27-1683							
HLP5	007752	26-1610	#27-1684							
HLP6	010040	26-1612	#27-1685							
HLP7	010100	26-1614	#27-1686							
HLP8	010163	26-1616	#27-1687							
HWSWR	001012	#16-681	19-1019	20-1067	20-1081	20-1085	21-1117	21-1117	21-1120	21-1120
		21-1128	21-1148	21-1148	21-1151	21-1151	21-1159	21-1171	22-1194	22-1215
		22-1215	22-1242	22-1242	22-1254	23-1272	23-1276	23-1310	24-1324	24-1332
		25-1459	25-1465							
LCP3	005523	#27-1631								
LCP4	005533	23-1296	#27-1632							
LCP5	005542	22-1202	#27-1633							
LCP7	005556	22-1211	#27-1635							
LEGCHR	001144	#18-995								
LINCNT	001062	#18-965	*22-1238	*22-1249						
LKS	001030	#16-688								
LN1	016615	28-2013	#32-2608							
LN1A	016647	31-2482	#32-2609							
LN10	017111	28-2037	31-2507	#32-2618						
LN2	016701	28-2016	31-2485	#32-2610						
LN3	016722	28-2019	31-2488	#32-2611						
LN4	016743	28-2022	31-2491	#32-2612						
LN5	016764	28-2025	31-2494	#32-2613						
LN6	017005	28-2028	31-2497	#32-2614						
LN7	017026	28-2031	31-2499	#32-2615						
LN8	017047	28-2034	31-2502	#32-2616						
LN8A	017070	31-2504	#32-2617							
LPB	001002	#16-674	21-1118	21-1149	25-1368	*28-1843	*28-1844	*28-1851	*28-1852	*28-1929
		*28-1930	*28-1936	*28-1937	*28-2235					
LPS	001000	#16-669	20-1074	21-1113	21-1117	21-1148	21-1168	25-1367	*28-1841	*28-1842

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0071

SYMBOL CROSS REFERENCE CREF V01

SYMBOL VALUE REFERENCES

```
                 28-1843   *28-1846   *28-1847   28-1851   *28-1928   28-1929   *28-1935   28-1936   28-1942
                 28-1958    28-1959    28-1960   28-2233    28-2236   28-2241    30-2356   30-2378   30-2380
                 30-2382    30-2388    31-2569
LSTADR   017410 #32-2626
M      = 000002 #16-701
MAINTB   001122 #18-985     21-1103    21-1113   21-1168   *28-1707   *28-1745   *28-1768
MENUD1   016070  28-1747   #32-2598
MENUD2   016156  28-1749   #32-2599
MENUD3   016221  28-1751   #32-2600
MENUD4   016263  28-1753   #32-2601
MENU1    015723  28-1709   #32-2594
MENU10   015613  29-2293   #32-2590
MENU2    015760  28-1711   #32-2595
MENU20   015643  29-2295   #32-2591
MENU3    016006  28-1713   #32-2596
MENU30   015654  29-2297   #32-2592
MENU4    016034  28-1715   #32-2597
MENU40   015667  29-2299   #32-2593
MESDD    006114 #27-1646
MES1     006153  19-1032   #27-1647
MES10    006504  25-1470   #27-1657
MES14    006570  25-1360    25-1370  #27-1660
MES2     006167  19-1034   #27-1648
MES20A   006607 #27-1661
MES21    006612  26-1556    26-1580  #27-1663
MES22    006614  26-1550   #27-1664
MES23    006620  26-1568   #27-1665
MES24    006630  26-1576   #27-1666
MES25    006647  26-1571    26-1574  #27-1667
MES26    006656  26-1560   #27-1668
MES27    006672  26-1563    26-1566  #27-1669
MES3     006230  19-1036   #27-1649
MES4     006246  19-1040   #27-1650
MES5     006274 #27-1651
MES6     006347 #27-1652
MES7     006416  24-1337   #27-1654
MES8     006457  24-1331   #27-1655
MES9     006475  24-1328   #27-1656
N      = 000014 #16-700     20-1085    20-1085  #20-1085   21-1117   21-1117  #21-1117   21-1120   21-1120
                 #21-1120    21-1128    21-1128  #21-1128   21-1148   21-1148  #21-1148   21-1151   21-1151
                 #21-1151    21-1159    21-1159  #21-1159   22-1215   22-1215  #22-1215   22-1232   22-1232
                 #22-1232    22-1242    22-1242  #22-1242   25-1365   25-1365  #25-1365
NOP    = 000240 #16-699
NUMCHR   001146 #18-996
OCT      001162 #18-1002   *26-1531   *26-1533  *26-1535   *26-1536   26-1540  *26-1582
OFFSET   001150 #18-997
PARITY   017275  28-2168   #32-2621
PARITZ   017340  28-2202   #32-2622
PASSA    001164 #18-1003    19-1028  *19-1030
PLKS     001026 #16-687
PRINE    004250  22-1202    23-1296  #25-1361
PRNNT    004242 #25-1360
```

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0072

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE		REFERENCES								
PRTMSG	004312		*22-1202	*23-1296	*25-1360	#25-1370					
PSW	001010		#16-680								
PTRC	001046		#16-696								
PTRDTR	015536		30-2409	#32-2589							
PTRPSW	001052		#16-698								
PTRVEC	001050		#16-697								
RINT	004274		25-1362	#25-1367							
R6	=%000006		#14-579	14-581							
R7	=%000007		#14-580	14-582							
SAVE	001134		#18-991	*21-1115	21-1129	*21-1146	21-1160				
SEGCNT	001054		#18-962								
SET	001156		#18-1000								
SETSER	010234		19-1037	#28-1696							
SETUP	001306		16-660	#19-1012	20-1087	24-1346	26-1593				
SIGNAL	001154		#18-999	*19-1014	26-1551	*26-1554	*26-1578				
SL	= 000036		#14-601								
STAER	004606		20-1085	21-1117	21-1120	21-1128	21-1148	21-1151	21-1159	22-1215	22-1232
			22-1242	25-1365	#25-1454						
STARN	004752		*25-1454	25-1480	#25-1483						
START	000200	G	#16-651	32-2627							
STEDA	004714		#25-1472	25-1473							
STEXT	004744		25-1467	25-1477	#25-1480						
STOPM	017153		28-2099	#32-2619							
STRCHR	001140		#18-993								
STRCNT	001142		#18-994								
SWR	001004		#16-678	*19-1022	20-1067	20-1081	20-1085	21-1117	21-1117	21-1120	21-1120
			21-1128	21-1148	21-1148	21-1151	21-1151	21-1159	21-1171	22-1194	22-1215
			22-1215	22-1242	22-1242	22-1254	23-1272	23-1276	23-1310	24-1324	24-1332
			25-1459	25-1465	26-1557						
SWREG	000176		#15-644	19-1022	20-1067	20-1081	20-1085	21-1117	21-1117	21-1120	21-1120
			21-1128	21-1148	21-1148	21-1151	21-1151	21-1159	21-1171	22-1194	22-1215
			22-1215	22-1242	22-1242	22-1254	23-1272	23-1276	23-1310	24-1324	24-1332
			25-1459	25-1465	*26-1540						
TEST1	001464		16-663	#20-1066	21-1173	24-1334	24-1348				
TEST2	003052		21-1172	#22-1194	22-1215	22-1242	22-1256				
TEST3	003640		22-1255	#23-1269	23-1312						
TIME	001130		#18-989	*21-1117	*21-1117	*21-1148	*21-1148				
TIMER	001132		#18-990	*21-1117	*21-1117	*21-1148	*21-1148				
TKB	001016		#16-683	24-1340	25-1474	26-1501	26-1542	28-1718	28-1756	28-1787	28-1876
			28-1973	28-2040	28-2102	28-2135	28-2171	28-2205	29-2259	29-2279	29-2302
			29-2335	31-2431	31-2460	31-2510					
TKHLP	005440		26-1510	#26-1597							
TKINT	004754		15-630	#26-1492							
TKS	001022		#16-685	19-1027	20-1066	24-1338	25-1472	28-1705	28-1716	28-1754	28-1785
			28-1862	28-1874	28-1971	28-2038	28-2100	28-2133	28-2169	28-2203	29-2257
			29-2277	29-2300	29-2333	31-2429	31-2458	31-2508			
TK1	005066		26-1516	#26-1518							
TK2	005100		26-1519	#26-1521							
TK3	005366		26-1539	26-1541	#26-1578	26-1617					
TK4	005372		26-1508	#26-1579							
TK5	005170		26-1529	#26-1536							
TK6	005406		26-1537	#26-1583							

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0073

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES
TK7	005376	26-1553 26-1577 #26-1581
TK9	005424	26-1513 #26-1591
TPB	001014	#16-682 *25-1368 *25-1372 25-1393 26-1526 26-1548 28-1721 28-1759 28-1790 28-1879 28-1976 28-2043 28-2105 28-2138 28-2174 28-2208 29-2262 29-2282 29-2305 29-2338 31-2434 31-2463 31-2513
TPS	001020	#16-684 *25-1367 *25-1371 25-1394 25-1396 25-1399 26-1524 26-1546 28-1719 28-1757 28-1788 28-1877 28-1974 28-2041 28-2103 28-2136 28-2172 28-2206 29-2260 29-2280 29-2303 29-2336 31-2432 31-2461 31-2511
TSEAB	004176	24-1333 #24-1336
TSEDA	004202	#24-1338 24-1339
TSEND	004070	23-1274 23-1311 #24-1323
TSRSS	004232	#24-1346
TSRST	004236	24-1325 24-1343 #24-1348
TSTWRD	014144	30-2354 #30-2358
TYO1	005571	20-1069 #27-1637
TYO2	005630	22-1196 #27-1638
TYO3	005673	23-1278 #27-1639
TYP	004332	15-615 #25-1381
TYPA	004342	#25-1384 25-1392 25-1406
TYPC	004352	25-1385 #25-1387
TYPD	004400	22-1245 25-1391 #25-1393 25-1403 25-1405
TYPDAT	004464	*22-1244 22-1246 *25-1384 25-1387 25-1389 25-1393 *25-1402 *25-1404 #25-1407
TYPDO	004426	25-1395 #25-1399
TYPDO1	004434	25-1398 #25-1401
TYPF	004436	25-1388 #25-1402
TYPG	004450	25-1390 #25-1404
TYPINT	004314	19-1012 #25-1371
TYPSWR	005272	26-1504 26-1545 26-1552 #26-1554
TYPSWX	005336	26-1558 #26-1567
W	= 000007	#16-702 *21-1117 21-1117 #21-1117 *21-1120 21-1120 #21-1120 *21-1148 21-1148 #21-1148 *21-1151 21-1151 #21-1151 22-1215 22-1215 #22-1215 *22-1242 22-1242 #22-1242
WAIT1	001730	#21-1117 21-1117
WAIT2	002100	#21-1120 21-1120
WAIT3	002366	#21-1148 21-1148
WAIT4	002536	#21-1151 21-1151
WAIT5	003142	#22-1215 22-1215
WAIT6	003406	#22-1242
WDD1	001736	#21-1117 21-1117
WDD2	002106	#21-1120 21-1120
WDD3	002374	#21-1148 21-1148
WDD4	002544	#21-1151 21-1151
WDD5	003166	22-1215 #22-1215 22-1215
WDD6	003414	#22-1242 22-1242
WDE1	001752	#21-1117 21-1117
WDE2	002122	#21-1120 21-1120
WDE3	002410	#21-1148 21-1148
WDE4	002560	#21-1151 21-1151
WDE5	003202	#22-1215 22-1215
WDE6	003430	#22-1242 22-1242
WER1	001770	21-1117 #21-1117
WER2	002140	21-1120 #21-1120

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES								
WER3	002426	21-1148	#21-1148							
WER4	002576	21-1151	#21-1151							
WER5	003220	22-1215	#22-1215							
WER6	003446	22-1242	#22-1242							
WEX1	002074	21-1117	#21-1117							
WEX2	002244	21-1120	#21-1120							
WEX3	002532	21-1148	#21-1148							
WEX4	002702	21-1151	#21-1151							
WEX5	003324	22-1215	#22-1215							
WEX6	003554	22-1242	#22-1242							
WORK	001124	#18-987	*21-1120	*21-1120	*21-1151	*21-1151	*22-1215	*22-1215	*22-1215	*22-1242
		*22-1242	*23-1283	*23-1285	*23-1301	*23-1303	*28-1800	*28-1801	*28-1809	*28-1817
		*28-1825	*28-1833	*28-1837	28-1838	28-1839	28-1841	*28-1890	*28-1891	*28-1899
		*28-1907	*28-1915	*28-1923	*28-1927	28-1928	*28-1967	*28-1982	*28-1989	28-1990
		*28-2049	*28-2053	*28-2057	*28-2061	*28-2065	*28-2069	*28-2073	*28-2077	*28-2082
		28-2085	28-2091	28-2092	*31-2519	*31-2523	*31-2527	*31-2531	*31-2535	*31-2539
		*31-2543	*31-2547	*31-2551	*31-2556	31-2559	31-2565	31-2566		
WORKA	001126	#18-988	*21-1120	*21-1120	*21-1151	*21-1151	*22-1215	*22-1215	*22-1242	*22-1242
		*23-1284	*23-1287	*23-1302	*23-1305					
WT2	005114	#26-1524	26-1525							
WT3	005242	26-1517	26-1520	26-1523	#26-1546	26-1547				

CZLCPA CREATED BY MACRO ON 12-MAR-85 AT 15:39 SEQ 0075

MACRO CROSS REFERENCE CREF V01

MACRO NAME REFERENCES

$ENABL #17-907 19-1027
$ERROR #17-716 20-1085 21-1117 21-1120 21-1128 21-1148 21-1151 21-1159 22-1215 22-1232
  22-1242 25-1365
$PRINT #17-933 22-1202 23-1296
$PRTSN #17-745 #20-1069 #22-1196 #23-1278
$SETPS #17-924
$TSWRG #17-949 20-1067 20-1081 20-1085 21-1117 21-1117 21-1120 21-1120 21-1128 21-1148
  21-1148 21-1151 21-1151 21-1159 21-1171 22-1194 22-1215 22-1215 22-1242 22-1242
  22-1254 23-1272 23-1276 23-1310 24-1324 24-1332 25-1459 25-1465
$TYPE #17-941
$WAITI #17-756 21-1120 21-1151 22-1215 22-1242
$WAITO #17-843 #21-1117 #21-1148