PDP11

INTER TEST PROGRAM

MD-11-DZITA-D

EP DZITA D DL B

FICHE 1 OF 1

MAR 1977

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZITA-D-D

PRODUCT NAME: INTERPROCESSOR TEST PROGRAM (ITEP)

PROGRAM DATE: JANUARY 1977

MAINTAINER: DIAGNOSTICS

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR
FIELD SERVICE PERSONEL. IT WILL VERIFY THE PROPER OPERATION
OF A COMPLETE COMMUNICATION LINK FROM ONE PDP-11
SYSTEM TO ANOTHER OR TO A COMMUNICATION TEST CENTER.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

A. PDP-11 SYSTEM WITH AT LEAST 4K OF CORE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE
THIS PROGRAM AND ALL OVERLAYS ARE ASSEMBLED IN ABSOLUTE FORMATS.
THE ABS LOADER IS USED TO LOAD THE PROGRAM AND OVERLAYS.

LOAD THE ITEP PROGRAM AND THE APPROPIATE OVERLAY FOR THE
TYPE OF INTERFACE YOU WISH TO TEST.

4.0 OPERATING PROCEDURES.

IF RUNNING ITEP ON AN LSI-11 ENVIRONMENT :
1. IF THE LINE CLOCK IS TO BE USED IT SHOULD
BE ENABLED PRIOR TO PROGRAM EXECUTION.

A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED
1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE
STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.
*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

B. CONSOLE DIALOGUE PARAMETER INPUT (CURRENT VALUES FOR PARAMETERS ARE FOUND IN OVERLAY)
DN11 AND DM11BB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THIS LISTING.

1. THE PROGRAM WILL TYPEOUT THE NAME OF THE VARIABLE OVERLAY.
A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A CARAGE RETURN
B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
C. IF YOU WISH TO SETUP A DM11BB, TYPE IN DMB.

IF DN OR DMB WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS,
VECTOR, ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DMBB.

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

3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS

B. TYPEIN ACTUAL VECTOR ADDRESS

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

5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE OVERLAY.(SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPEIN ACTUAL VALUE

6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE OVERLAY.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. ENTER ACTUAL VALUE

7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
THE DN-11 WILL USE PARAM #3 AS THE # TO DIAL.
IF USING A MODEM WITHOUT AUTOMATIC HANDSHAKING,
THE NUMBER MUST TERMINATE WITH A
"END-OF-NUMBER" CHARACTER (:).
B. ENTER ACTUAL VALUE.

8. THE PROGRAM WILL RETURN TO STEP B1 IF THIS SETUP
WAS FOR DN11 OR DM11BB.

9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.
NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING
NEW VALUES,THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT
RESTARTS OF THE PROGRAM.

C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
    1. THE PROGRAM HALTS FOR ISR(INTERFACE SERVICE ROUTINE) SPECIFICATION
    SWR14=1 SETUP DM-11BB ISR
    SWR13=1 SETUP DN-11 ISR
    SWR=000000=SETUP VARIABLE ISR (OVERLAY) (NOT DN-11 OR DM11BB)
        SET APPROIATE SWITCHES AND HIT CONTINUE.

    2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
        SETUP SEQUENCE IS: DN11,DM11-BB THEN VARIABLE ISR. (FOR EACH ENTRY SET SWICHES AND THEN HIT CONT.)
        A. HALT FOR BUS ADDRESS OF INTERFACE
        B. HALT FOR VECTOR ADDRESS OF INTERFACE
        C. HALT FOR PRIORITY OF INTERFACE (200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.)
        D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
        E. HALT FOR INTERFACE PARAM #2 (DN11 AND DMBB PARAMETERS ARE DISCUSSED IN SECT. 10.C OF THIS LISTING)
        F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DMB.

    3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
        A. PRESS CONTINUE TO START TESTING

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

(IF) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

D. OPERATIONAL SWITCH SETTINGS.
SW15=1 HALT ON ERROR
SW14=1 SINGLE PASS
    SW14 HAS NO EFFECT IF SW04=0
SW13=1 INHIBIT ERROR TYPEOUTS
SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
    IF SW12=0 AND SW04=1 END PASS IS TYPED
    AND TRANSMITTED/RECEIVED DATA IS TYPED.
SW11=1 USE PREVIOUSLY SPECIFIED DATA
SW10=1 DATA SELECT (WITH SW09)
SW09=1 DATA SELECT (WITH SW10)
    00=1 GET DATA FROM OPERATOR
    01=1 TEST MESSAGE #1 ($A QUICK BROWN FOX)
    10=1 TEST MESSAGE #2 ($B NUMERICS)
    11=1 TEST MESSAGE #3 ($C COMTEST/QUICK BROWN FOX/NUMERICS)
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
SW07=1 DO NOT TEST RECEIVED DATA
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
    * IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
    TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
    RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
    OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.
SW04=1 RETURN TO MONITOR FOR END PASS
    WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.
SW03=1 INTERNAL LOOPBACK MODE
SW02=1 EXTERNAL LOOPBACK MODE
SW01=1 ONE-WAY-IN MODE
SW00=1 ONE-WAY-OUT MODE

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A
REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE.
TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN
↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377)
SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).
    I.E. ABCD↑ 000 123 377↑ EFG (CAR.RETURN)

A TYPICAL SWITCH SETTING FOR HALF-DUPLEX=003150 THIS SETTING USES
INTERNAL LOOPBACK MODE, LOOPS IN OVERLAY, MONITORS TRANSMITTED AND RECEIVED
DATA ON THE CONSOLE TTY, AND TESTS RECEIVED DATA USING TEST MESSAGE #3.

A TYPICAL SWITCH SETTING FOR FULL-DUPLEX=003144 THIS SETTING
IS THE SAME AS ABOVE EXCEPT IT USES THE EXTERNAL LOOPBACK MODE.

ALL STANDARD MESSAGES(TEST MESSAGES 1-3) ARE PRECEDED BY 2 FILL CHARACTERS(177),
AND ARE FOLLOWED BY A CR(015), LF(012),RECEIVE TERMINATING CHARACTER(001),
4 FILLS(177), AND A TRANSMIT TERMINATING CHARACTER(000). DURING TRANSMISSION,
WHEN A 000 CHARACTER IS SEEN THE TRANSMISSION IS STOPPED.DURING RECEPTION,
WHEN A 001 CHARACTER IS RECEIVED, THE RECEIVER IS SHUT OFF.
IF THE MESSAGE WAS INPUTED BY THE OPERATER, THE TERMINATING CHARACTERS ARE ADDED.

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10(SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR
   TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR
   GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
   GO TO STEP 1(SW04=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SW07=0)
3. RETURNS TO MONITOR FOR "END PASS"(SW04=1) OR
   GO TO STEP 1 (SW04=0)

ONE-WAY-OUT MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAITS FOR CLEAR TO SEND
3. TRANSMITS SELECTED DATA
4. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
   GO TO STEP 1 (SW04=0)

E. THE OVERLAY IS THEN ENTERED AND A CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY.

IF ONE-WAY-IN OR INTERNAL LOOPBACK MODES ARE SELECTED.
THE OVERLAY WILL SET DATA TERMINAL READY AND WAIT FOR DATA.

IF ONE-WAY-OUT OR EXTERNAL LOOPBACK MODES WERE SELECTED.
THE OVERLAY WILL SET DATA TERMINAL READY AND REQUEST TO SEND.
THE OVERLAY WILL THEN WAIT FOR CLEAR TO SEND BEFORE ATTEMPTING TO
TRANSMIT DATA.

F. IF SW04=0 THE OVERLAY WILL CONTINUE TO
TRANSMIT/RECEIVE DATA.

IF SW04=1 THE OVERLAY WILL RETURN
TO THE MONITOR AND TYPE "END PASS".

IF BOTH SW04=1 AND SW14=1, THE PROGRAM WILL REQUEST
NEW INTERFACE PARAMS AFTER ONE PASS OF THE SELECTED
TEST MODE.

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING
CHARACTERS ON THE CONSOLE TTY.
LINE FEED = RESTART PROGRAM AT LOCATION 200.
QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)
THEN TYPE EITHER:

*WXXXXXX	TO PRINTOUT THE 8 WORDS AT LOC XXXXXX.
*BXXXXXX	TO PRINTOUT THE 16 BYTES AFTER LOC XXXXXX.
*C	TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.
CARRIAGE RETURN = RESTART AT REQUEST FOR NEW OPERATIONAL SWITCHES.

5.0 PROGRAM AND/OR OPERATOR ACTION

IF THE OPERATOR WISHES TO MANUALY EXAMINE THE TRANSMIT OR RECEIVE
BUFFERS, DO THE FOLLOWING: TO FIND THE STARTING ADDRESS OF THE RECEIVE
BUFFER, LOAD ADDRESS 11020 AND EXAMINE. TO FIND THE STARTING ADDRESS
OF THE TRANSMIT BUFFER, LOAD ADDRESS 11022 AND EXAMINE.

5.1 NORMAL HALTS
SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

THE ONLY ERROR REPORT FROM THE CONTROL PROGRAM OCCURS IF THE
INTERFACE SPECIFIED IS NOT LOADED.

THE ERROR REPORTS FROM THE VARIOUS INTERFACE SERVICE
ROUTINES ARE AS DEFINED IN THEIR DOCUMENTS

7.0 RESTRICTIONS

THE OPERATION OF THIS PROGRAM REQUIRES COORDINATION BETWEEN
THE OPERATOR AND THE OPERATOR OF ANOTHER PDP-11 SYSTEM
UNLESS ONE OF THE SYSTEMS IS ALWAYS OPERATING IN A FIXED
MODE. THE FOLLOWING TABLE LISTS THE VALID COMBINATIONS:

NOTE: ONLY ONE MODE MAY BE SELECTED AT A TIME.

CPU #1	CPU #2
ONE-WAY-OUT	ONE-WAY-IN
ONE-WAY-IN	ONE-WAY-OUT
EXTERNAL-LOOPBACK	INTERNAL-LOOPBACK
INTERNAL-LOOPBACK	EXTERNAL-LOOPBACK
EXTERNAL-LOOPBACK	EXTERNAL-LOOPBACK (FULL-DUPLEX)

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTIONS APPLY:

IF RUNNING IN FULL DUPLEX MODE BOTH SYSTEMS
MUST BE IN EXTERNAL LOOP BACK MODE.

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.
EXAMPLE:
SWITCHES 14,13,7,4 SHOULD BE THE SAME
ON BOTH CPU'S

8.0 MISCELLANEOUS

ITEP WAS CHECKED OUT USING THE FOLLOWING BELL TELEPHONE MODEMS.
201A (HALF-DUPLEX SYNCHRONOUS 2000 BAUD)
202C (HALF-DUPLEX ASYNCHRONOUS 1200 BAUD)
103A (FULL-DUPLEX ASYNCHRONOUS 110 BAUD)

9.0 PROGRAM DESCRIPTION

THE INTERPROCESSOR TEST PROGRAM (ITP) PROVIDES THE
LINKAGE BETWEEN THE OPERATOR AND THE VARIOUS INTERFACE
SERVICE ROUTINES (OVERLAY) WHICH PREFORM THE ACTUAL DATA
MOVEMENT AND VERIFICATION TO AND FROM THE COMMUNICATION
LINK. IN ADDITION, ITP CONTAINS VARIOUS INTERRUPT AND
SUB ROUTINES WHICH ARE USED BY THE OVERLAY'S.

9.1 TRAP CATCHER

THIS IS A SERIES OF JUMP AND HALT INSTRUCTIONS PLACED IN ALL
UNUSED VECTORS TO CATCH UNEXPECTED INTERRUPTS.

9.2 SWITCH REGISTER INPUT ROUTINE (MANIN:)

THIS ROUTINE IS ENTERED ONLY WHEN SWITCH 15 IS SET WHEN
PROGRAM IS STARTED AT LOCATION 200. IT ACCEPTS PARAMETERS
FOR THE ISR'S FROM THE CONSOLE SWITCHES AT A SERIES OF HALTS.
AS SPECIFIED IN OPERATING INSTRUCTIONS.

9.3 PARAMETER INPUT ROUTINE (GETIT:)

THIS ROUTINE SOLICITS PARAMETERS FROM THE OPERATOR ON
THE CONSOLE DEVICE AND PLACES THEM IN THE SPECIFIED ISR'S
PARAMETER TABLE.
NOT USED OPTIONAL PARAMETER WORDS ARE INDICATED BY THE
PRESENSE OF A NEGATIVE VALUE IN THE ISR'S TABLE. THIS SECTION
OF CODE UTILIZES SUB/ROUTINE 'GETANY' WHICH PRINTS OUT THE
WORD POINTED TO BY THE ADDRESS IN REGISTER 0.
IT THEN INPUTS A WORD OR CARRIAGE RETURN FROM THE OPERATOR.
IF ONLY A CARRIAGE RETURN IS TYPED, THE PARAMETER IS
LEFT AS IT IS, OTHERWISE IT IS REPLACED BY THE OPERATORS
TYPE IN AND THE POINTER IN REGISTER 0 IS INCREMENTED TO THE
NEXT WORD.

9.4 TTY INTERRUPT (TTYINT:)

THE TTY INTERRUPT IS USED TO INTERRUPT THE EXECUTION OF
A TEST IN ORDER TO RESTART (TYPE A LINE FEED) OR TO SPECIFY
NEW OPERATIONAL SWITCHES (TYPE A CARRIAGE RETURN)

9.5 SET SWITCH OPTIONS (SWRSET:)

THE PROGRAM WILL HALT (MANUAL PARAMETER ENTRY) OR WAIT FOR
A CARRIAGE RETURN (TTY CONTROL) AT THIS POINT TO PERMIT
THE OPERATOR TO SETUP THE OPERATIONAL SWITCH SETTINGS.
THE TEST MODE(SW00-SW03) AND TEST DATA(SW08-SW11) MAY BE
CHANGED ONLY AT THIS POINT. ALL OTHER SWITCHES MAY BE
CHANGED WHILE A TEST IS RUNNING.
IF NEW VARIABLE DATA IS SPECIFIED, THIS ROUTINE WILL REQUEST
THAT THE DATA BE ENTERED AND UTILIZES THE 'GETSTR' SUB/ROUTINE TO INPUT
THE DATA FROM THE OPERATOR.

9.6 SETUP TIMER (SUTIME:)

THE PROGRAM LOOKS FOR AND UTILIZES EITHER THE LINE CLOCK
OR REAL TIME CLOCK IF EITHER IS PRESENT ON THE SYSTEM. A
BUS ERROR(NO RESPONSE) IS USED TO INDICATE THE ABSENCE OF
A CLOCK. IF NEITHER EXISTS, THE PROGRAM WILL STILL RUN BUT
IS SUBJECT TO WAITING IN UNENDING LOOPS.

9.7 THE INTERFACE SERVICE ROUTINES (ISR'S) ARE ENTERED AT THIS POINT.

9.8 END OF PASS ($EOP:)

THIS SECTION OF CODE WILL PRINT "END OF PASS XXXXXX"
AND THEN SENSE FOR SW14. IF SWITCH 14 IS RESET THE
OVERLAY'S ARE REENTERED.
IF SWITCH 14 IS SET THE PROGRAM CHECKS TO SEE IF IT
WAS LOADED BY A MONITOR (LOCATION 42 NOT EQUAL 0) AND IF
IT WAS, CONTROL IS RETURNED TO THE MONITOR. OTHERWISE
THE PROGRAM REQUESTS NEW PARAMETERS.

9.10 HALT HANDLER ($HLT:)

THIS ROUTINE IS USED TO SENSE THE OPERATIONAL SWITCHES
AND PROVIDE ERROR CONTROL. IT WILL PRINTOUT THE ADDRESS
OF THE ERROR HLT IF SWITCH 13 (DELETE ERROR TYPEOUTS)
IS DOWN (NOT SET)

9.11 READ A CHARACTER ROUTINE ($READC:)
THIS ROUTINE GETS A CHARACTER FROM THE TTY AND PLACES IT
ON THE STACK

9.12 READ A STRING ROUTINE ($READS)

THIS ROUTINE GETS A STRING OF CHARACTERS FROM THE TTY AND PLACES
THEM IN A BUFFER SPECIFIED BY THE ADDRESS FOLLOWING
THE SUB/ROUTINE CALL.
THE ROUTINE WILL ALSO ACCEPT OCTALLY REPRESENTED CHARACTERS
WHEN THEY ARE PRECEEDED AND FOLLOWED BY UP ARROWS,
AND SPERATED BY SPACES OR COMMAS.

9.13 OCTAL INPUT ROUTINE($ACCEPT:)

THIS ROUTINE READS AN OCTALLY REPRESENTED WORD FROM THE TTY
AND PLACES IT IN THE LOCATION INDICATED BY THE ADDRESS
FOLLOWING THE SUB/ROUTINE CALL.

9.14 CLOCK INTERRUPT ROUTINE (TIMER:)

THIS ROUTINE IS ENTERED ON INTERUPTS FROM EITHER THE
LINE CLOCK OR REAL TIME CLOCK EVERY 16 MILLISECONDS IF
EITHER IS PRESENT.
IT WILL INCREMENT LOCATION 'TIME:' IN THE OVERLAY'S PARAMETER
TABLE EVERY SECOND.

9.15 BINARY TO OCTAL ROUTINE ($B2O16)

THIS ROUTINE WILL PRINTOUT THE OCTAL REPRESENTATION OF
A WORD ON THE STACK.

9.16 POWER DOWN ROUTINE '$PWRDN:)

THIS ROUTINE SAVES THE STATUS OF THE MACHINE WHEN
POWER IS LOST.

9.17 POWER UP ROUTINE ($PWRUP:)

THIS ROUTINE RESTORES THE STATE OF THE MACHINE WHEN
POWER IS RESTORED AND RESTARTS AT ADDRESS 200.

9.18 VARIABLE INTERFACE SERVICE ROUTINE (VISR:)

THESE LOCATIONS ARE RESERVED FOR AND WILL BE
OVERLAID BY THE VARIABLE ISR'S.
THE FIRST 2 WORDS CONTAIN A 3 CHARACTER ISR NEMONIC FOLLOWED
BY A ZERO CHARACTER.
THE NEXT 3 WORDS CONTAIN THE BUS ADDRESS, VECTOR ADDRESS AND
PRIORITY.
THE NEXT 2 WORDS MAY CONTAIN OPTIONAL PARAMETERS. THEY WILL
CONTAIN ALL ONES IF THEY ARE NOT REQUIRED
THE NEXT WORD MAY CONTAIN THE ADDRESS OF AN INPUT BUFFER
IF THE ISR REQUIRES AN ASCII PARAMETER. IT WILL CONTAIN

ALL ONES IF THE PARAMETER IS NOT REQUIRED.
LOCATION 'CLOCK:' WILL BE INCREMENTED EVERY SECOND WHILE
THE TEST IS BEING RUN IF THERE IS A LINE CLOCK OR REAL TIME CLOCK
ON THE SYSTEM. IT MAY BE USED AS A ELASPED TIMER BY THE ISR.

10.0 PARAMETERS FOR THE DM11BB AND THE DN11

10.1 DM11BB PARAMETERS

PARAM#1 IS LOADED INTO THE CONTROL AND STATUS REGISTER OF THE DM11BB
TO SELECT THE LINE NUMBER IN OCTAL (BITS 0-3). ALL OTHER BITS MUST BE 0'S.
THIS IS THE ONLY PARARMETER USED BY THE DM11BB.

10.2 DN11 PARAMETERS

ONLY PARAM#3 IS USED BY THE DN11, IT CONTAINS THE NUMBER THE DN WILL DIAL.

N01

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

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


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


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

```
                        000300          PRTY6=  300
                        000340          PRTY7=  340
                                        
                                        ;MISCELLANEOUS BIT ASSIGNMENT
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0
                                        
                                        ;VECTOR ADDRESSES
                        000004          ERRVEC= 4
                        000010          RESVEC= 10
                        000014          TBITVEC=14
                        000014          TRTVEC= 14
                        000014          BPTVEC= 14
                        000020          IOTVEC= 20
                        000024          PWRVEC= 24
                        000030          EMTVEC= 30
                        000034          TRAPVEC=34
                                        .EQUIV  R4,CSR
                                        .EQUIV  R4,RCSR
                                        
                        000000                  .=0
                                        ;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
                                        ;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS
                        000024                  .=24
000024  000024  006672                  $PWRDN
000026  000026  000340                  340
000030  000030  005032                  $HLT
000032  000032  000340                  340
000034  000034  006544                  $TRAP
000036  000026  000340                  340
```

```
                        000100                          .=100
000100  006312                                 TIMER
000102  000340                                 340
                        000174                          .=174
000174  000000                         DISPREG:0
000176  000000                         SWREG:  0
                        000200                          .=200
000200  000137  003254                         JMP     @#BEGIN         ;JUMP TO STARTING ADDRESS OF PROGRAM
000204  000137  004116                         JMP     @#SWPRNT        ;RESTART AT 204. DO THE RESTART.
```

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

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

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

001100  010046                  $TYPE:  MOV     R0,-(SP)        ;SAVE R0
001102  017600  000002                  MOV     @2(SP),R0       ;GET ADDRESS OF ASCIZ STRING
001106  062766  000002  000002          ADD     #2,2(SP)        ;ADJUST RETURN PC
001114  112046                  1$:     MOVB    (R0)+,-(SP)     ;PUSH CHARACTER TO BE TYPED ONTO STACK
001116  001003                          BNE     2$              ;BR IF IT ISN'T THE TERMINATOR
001120  005726                          TST     (SP)+           ;IF TERMINATOR POP IT OFF THE STACK
001122  012600                          MOV     (SP)+,R0        ;RESTORE R0
001124  000002                          RTI                     ;RETURN
001126  004737  001160          2$:     JSR     PC,5$           ;GO TYPE THIS CHARACTER
001132  122726  000012          3$:     CMPB    #12.(SP)+       ;CHECK IF THE CHAR. TYPED WAS A  LINE FEED
001136  001366                          BNE     1$              ;GO GET NEXT CHAR. IF NOT LINE FEED
001140  013746  001524                  MOV     NULL,-(SP)      ;GET # OF FILLER CHARS. NEEDED
                                                                ;AND THE NULL CHAR.
001144  105366  000001          4$:     DECB    1(SP)           ;DOES A NULL NEED TO BE TYPED?
001150  002770                          BLT     3$              ;BR IF NO--GO POP THE NULL OFF OF STACK
001152  004737  001160                  JSR     PC,5$           ;GO TYPE A NULL
001156  000772                          BR      4$              ;LOOP
001160  105777  000334          5$:     TSTB    @TPS            ;WAIT UNTIL PRINTER IS READY
001164  100375                          BPL     5$
001166  116677  000002  000326          MOVB    2(SP),@TPB      ;LOAD CHAR TO BE TYPED INTO DATA REG.
001174  000207                          RTS     PC
```

```
                                ;*****************************************************************
                                ;       DEBUG DUMP ROUTINE
                                ;*****************************************************************
                        001200                  .=1200
001200  013746  011042          DMPHLT: MOV     FLAG,-(SP)
001204  042737  000070  011042          BIC     #70,FLAG        ;INIT FLAGS
001212  104400  001532                  TYPE    ,ASTRK          ;TYPE *
001216  104402                          GETCHR
001220  012637  001314                  MOV     (SP)+,6$
001224  104400  001314                  TYPE    ,6$
001230  122737  000127  001542          CMPB    #127,$CHAR      ;W? FOR WORD
001236  001004                          BNE     1$
001240  052737  000010  011042          BIS     #BIT3,FLAG      ;SET FLAG BIT
001246  000430                          BR      3$
001250  122737  000102  001542  1$:     CMPB    #102,$CHAR      ;B? FOR BYTE
001256  001004                          BNE     2$
001260  052737  000020  011042          BIS     #BIT4,FLAG
001266  000420                          BR      3$
001270  122737  000103  001542  2$:     CMPB    #103,$CHAR      ;C? FOR CONTINUE
001276  001014                          BNE     3$
001300  012637  011042                  MOV     (SP)+,FLAG
001304  052737  000040  011042          BIS     #BIT5,FLAG
001312  000413                          BR      DUMP
001314  000000          6$:     000000
001316  104400  001526          4$:     TYPE    ,QUES
001322  104400  001536                  TYPE    ,CRLF
001326  000724                          BR      DMPHLT
001330  005037  001544          3$:     CLR     WORK
001334  005726                          TST     (SP)+
001336  104406  001544                  ACCEPT  ,WORK
001342  012700  001476          DUMP:   MOV     #DMPLST,R0      ;INIT DUMP LIST
001346  062710  000020                  ADD     #20,    (R0)    ;BUMP ADDRESS
001352  032737  000040  011042          BIT     #BIT5,FLAG
001360  001005                          BNE     L1
001362  013737  001544  001476          MOV     WORK,DMPLST
001370  001001                          BNE     L1
001372  022020                          CMP     (R0)+,  (R0)+   ;SKIP 1ST TWO ENTRIES

001374  012001          L1:     MOV     (R0)+,  R1      ;GET ADDR OF DATA FROM LIST
001376  001700                  BEQ     DMPHLT          ;BR IF END OF LIST
001400  104400  001536                  TYPE    ,CRLF
001404  010146                          MOV     R1,     -(SP)   ;PUSH ADDR ON STACK
001406  004037  006350                  JSR     R0,$B2016       ;PRINT OUT ADDRESS

001412  032737  000010  011042          BIT     #BIT3,FLAG
001420  001014                          BNE     L3
001422  012702  000020                  MOV     #20,    R2      ;SET WORD COUNTER = 8
001426  005046          L2:     CLR     -(SP)
001430  112116                  MOVB    (R1)+,(SP)
001432  104400  007152                  TYPE    ,MSGOO
001436  004037  006336                  JSR     R0,$B20OT
001442     003                          .BYTE   3
001443     001                          .BYTE   1
001444  005302                  DEC     R2              ;DECREMENT WORD COUNTER
001446  001367                  BNE     L2              ;BR IF NOT = 0
001450  000751                  BR      L1              ;GET NEXT ENTRY
```

```
                752
                753  001452  012702  000010          L3:     MOV     #10,R2
                754  001456  012146                  1$:     MOV     (R1)+,-(SP)
                755  001460  104400  007152                  TYPE    ,MSG00
                756  001464  004037  006350                  JSR     R0,$B2016
                757  001470  005302                          DEC     R2              ;DECREMENT THE WORD COUNT
                758  001472  001371                          BNE     1$
                759  001474  000737                          BR      L1              ;GET NEXT ENTRY
                760  001476  000000          DMPLST: 0                               ;RESERVED FOR SW. REG
                761  001500  000000                  0                               ;END OF TABLE FOR SW. REG
                762  001502  000001          .RX:    .BLKW   1
                763  001504  000001          .TX:    .BLKW   1
                764  001506  000000                  0
                765  001510  000000                  0
                766  001512  000000                  0
                767  001514  177560          TKS:    177560                          ;TTY KEYBOARD STATUS REG. ADDRESS
                768  001516  177562          TKB:    177562                          ;TTY KEYBOARD DATA BUFFER REG. ADDRESS
                769  001520  177564          TPS:    177564                          ;TTY PRINTER STATUS REG. ADDRESS
                770  001522  177566          TPB:    177566                          ;TTY PRINTER BUFFER REG. ADDRESS
                771  001524  000000          NULL:   .WORD   0               ;CONTAINS NULL CHARACTER FOR FILLS
```

```
                                        ;*****************************************************************
                                        ;       CONSTANTS AND WORKING STORAGE
                                        ;*****************************************************************


001526  037440  000040          QUES:   .ASCIZ  " ? "
001532  005015  000052          ASTRK:  .ASCIZ  <15><12>"*"
001536     015                  CRLF:   .ASCII  <15>
001537     012     000          LF:     .ASCIZ  <12>
        001542                          .EVEN
        001544                          $WORK=WORK
001542  000000                  $CHAR:  0
001544  000000                  WORK:   0
001546  000000                  WORK1:  0
001550  000000                  WORK2:  0
001552  000000                  WORK3:  0
001554  000000                  WORK4:  0
001556  000000                  WORK5:  0
001560  000000                  WORK6:  0
001562  000000                  FLAGS:  0
                                ;DATA PATTERN ADDRESS TABLE AND PATTERNS
001564  002604                  DAT:    .WORD   VDB             ;ADDRESS OF VARIABLE DATA BUFFER
001566  003004                          .WORD   DP1             ;ADDRESS OF MESSAGE 1
001570  003076                          .WORD   DP2             ;ADDRESS OF MESSAGE 2
001572  003126                          .WORD   DP3             ;ADDRESS OF MESSAGE 3
001574  002604                          .WORD   DP4             ;ADDRESS OF MESSAGE 4
001576  003004                          .WORD   DP5             ;ADDRESS OF MESSAGE 5
001600  003076                          .WORD   DP6             ;ADDRESS OF MESSAGE 6
001602  003126                          .WORD   DP7             ;ADDRESS OF MESSAGE 7

                                ;RECEIVER DATA BUFFER STARTS HERE..
001604  000400                  IBUF:   .BLKW   400
002604                          VDB:
002604  000100                          .BLKW   100             ;VARIABLE DATA BUFFER

003004     177     177          DP1:    .BYTE   177,177
003006  040444  052040  042515 .ASCIZ  '$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.'<15><12><001><177><177><177><1
        003076                          .EVEN
003076     177     177          DP2:    .BYTE   177,177
003100  041044  030040  031061 .ASCIZ  '$B 0123456789'<15><12><001><177><177><177><177>
        003126                          .EVEN
003126     177     177          DP3:    .BYTE   177,177
003130  041444  041440  046517 .ASCII  '$C COM-TEST MAYNARD THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG'
003230  030040  031061  032063 .ASCIZ  ' 0123456789'<15><12,<001><177><177><177><177>
        003254                          .EVEN

        002604                          DP4=VDB                 ;SPARE
        003004                          DP5=DP1                 ;SPARE
        003076                          DP6=DP2                 ;SPARE
        003126                          DP7=DP3                 ;SPARE
```

```
                                ;*****************************************************************
                                ;       START OF PROGRAM
                                ;*****************************************************************
003254                          BEGIN:
003254  012706  001070                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
003260  104414  000340                  STPS,PRTY7
003264  104420                          SUSWR
003266  005037  004774                  CLR     $PASS           ;CLEAR THE PASS COUNT
003272  012737  003254  003462          MOV     #BEGIN,$LPADR   ;INITILIZE THE LOOP ADDRESS FOR SCOPE
003300  005037  001562                  CLR     FLAGS           ;RESET FLAGS
003304  000005                          RESET
                                ;*****************************************************************
                                ;       GET PARAMETERS FROM OPERATOR
                                ;*****************************************************************
003306  022777  100000  005530          CMP     #100000,@SWR    ;MANUAL INPUT??
003314  001063                          BNE     GETIT           ;BR IF NO

                                ;*****************************************************************
                                ;       SWITCH REG INPUT ROUTINE
                                ;*****************************************************************
003316  012701  010164          MANIN:  MOV     #DN+4,R1        ;PRESET POINTER FOR DN-11
003322  005000                          CLR     R0              ;CLEAR DISPLAY
003324  000000                          HALT                    ;HALT FOR ISR REQUEST
003326  017737  005512  001562          MOV     @SWR,   FLAGS   ;SAVE ISR REQUEST INDICATORS
003334  032777  020000  005502          BIT     #20000,@SWR     ;IS DN11 SETUP REQLESTED?
003342  001402                          BEQ     .+6             ;BR IF NO
003344  004737  003404                  JSR     PC,MANBA        ;GO SETUP DN11

003350  012701  010406                  MOV     #DMB+4,R1       ;PRESET DMB ISR ADDRESS
003354  032737  040000  001562          BIT     #40000,FLAGS    ;IS DMB SETUP REQUESTED?
003362  001402                          BEQ     .+6             ;BR IF NO
003364  004737  003404                  JSR     PC,MANBA        ;GO SETUP DM11-B

003370  012701  011004                  MOV     #VISR+4,R1      ;PRESET VARIABLE ISR ADDRESS
003374  004737  003404                  JSR     PC,MANBA        ;GO SETUP VARIABLE ISR

003400  000137  004200                  JMP     SWRSET          ;GO GET OPERATIONAL SWITCHES

003404  011100                  MANBA:  MOV     (R1),   R0      ;DISPLAY BUS ADDR
003406  000000                          HALT                    ;HALT FOR BUS ADDR
003410  017721  005430                  MOV     @SWR,   (R1)+
003414  011100                          MOV     (R1),   R0      ;DISPLAY VECTOR ADDR
003416  000000                          HALT                    ;HALT FOR VECTOR ADDR
003420  017721  005420                  MOV     @SWR,   (R1)+
003424  011100                          MOV     (R1),   R0      ;DISPLAY PRIORITY
003426  000000                          HALT                    ;HALT FOR PRIORITY
003430  017721  005410                  MOV     @SWR,   (R1)+
003434  011100                          MOV     (R1),   R0      ;DISPLAY PARAM #1
003436  100410                          BMI     MANINX          ;BR IF PARAM NOT REQUIRED
003440  000000                          HALT                    ;HALT FOR PARAM #1
003442  017721  005376                  MOV     @SWR,   (R1)+   ;
003446  011100                          MOV     (R1),   R0      ;DISPLAY PARAM #2
003450  100403                          BMI     MANINX          ;BR IF PARAM NOT REQUIRED
003452  000000                          HALT                    ;HALT FOR PARAM #2
003454  017721  005364                  MOV     @SWR,   (R1)-
003460  000207                  MANINX: RTS     PC
```

```
 870  003462  000000                    $LPADR: 0
 871
 872                                    ;*****************************************************************
 873                                    :       ISR PARAM INPUT ROUTINE
 874                                    ;*****************************************************************
 875  003464  012700  011000            GETIT:  MOV     #VISR,  R0      ;PRESET ISR ADDR IN R0
 876  003470  104400  007154                    TYPE    .MSG01          ;<15><12> INTERFACE TYPE
 877  003474  104400  011000                    TYPE    ,VISR           ;PRINTOUT ISR NAME
 878  003500  104400  001526                    TYPE    .QUES
 879
 880  003504  104404  001544                    GETSTR  ,WORK           ;READIN OPERATOR'S RESPONSE
 881  003510  123727  001544  000015            CMPB    WORK,   #15     ;IS IT CAR. RET?
 882  003516  001431                            BEQ     GETBA           ;BR IF YES
 883
 884  003520  012700  010160                    MOV     #DN,    R0      ;PRESET DN ISR ADDR
 885  003524  023737  010160  001544            CMP     DN,     WORK    ;IS IT DN?
 886  003532  001004                            BNE     DMBTST          ;BR IF NO
 887  003534  052737  020000  001562            BIS     #20000,FLAGS    ;SET DN11 FLAG
 888  003542  000417                            BR      GETBA           ;GO GET DN11 PARAMS
 889  003544                            DMBTST:
 890  003544  012700  010402                    MOV     #DMB,   R0      ;PRESET DM11B ISR
 891  003550  023737  010402  001544            CMP     DMB,    WORK    ;IS IT DM?
 892  003556  001004                            BNE     NOISR           ;BR IF NO
 893  003560  052737  040000  001562            BIS     #40000, FLAGS   ;SET DM11-B FLAG
 894  003566  000405                            BR      GETBA           ;GO GET DM11-B PARAMS
 895
 896  003570                            NOISR:
 897  003570  104400  007177                    TYPE    .MSG02          ;<15><12> ISR NOT LOADED!
 898
 899  003574  000000                            HALT                    ;HALT
 900  003576  000137  003254                    JMP     BEGIN           ;TRY AGAIN
 901  003602  010004                    GETBA:  MOV     R0,     R4      ;SAVE POINTER
 902  003604  022020                            CMP     (R0)+,  (R0)+   ;INCREMENT ISR POINTER
 903  003606  104400  007222                    TYPE    ,MSG03          ;<15><12> BUS ADDRESS=
 904  003612  004737  004776                    JSR     PC,GETANY       ;GET THE BUS ADDR
 905
 906  003616  104400  007241            GETVA:  TYPE    ,MSG04          ;<15><12> VECTOR ADDRESS=
 907  003622  004737  004776                    JSR     PC,GETANY       ;GET THE VECTOR ADDR.
 908
 909  003626  104400  007263            GETPRI: TYPE    ,MSG05          ;<15><12> PRIORITY=
 910  003632  004737  004776                    JSR     PC,GETANY       ;GET THE PRIORITY
 911
 912  003636  005710                            TST     (R0)            ;PARAM #1 REQUIRED?
 913  003640  100412                            BMI     GETP3           ;BR IF NO
 914  003642  104400  007277            GETPRM: TYPE    ,MSG06          ;<15><12> PARAMS #1=
 915  003646  004737  004776                    JSR     PC,GETANY       ;GET PARAM
 916
 917  003652  005710                            TST     (R0)            ;PARAM #2 REQUIRED?
 918  003654  100404                            BMI     GETP3           ;BR IF NO
 919  003656  104400  007314                    TYPE    ,MSG07          ;<15><12> PARAMS #2=
 920  003662  004737  004776                    JSR     PC,GETANY       ;GET PARAM
 921
 922  003666  016437  000016  003704    GETP3:  MOV     16(R4), ARIA    ;IS ASCII PARAM REQUIRED
 923  003674  100424                            BMI     GETEX           ;BR IF NO
 924  003676  104400  007331                    TYPE    ,MSG08          ;<15><12> ASCII PARAM=
 925  003702  104400                            TYPE                    ;PRINTOUT ASCII PARAM
```

```
                                ARIA:   0
                                        TYPE    .QUES

                                        GETSTR                  ;GET ASCII INPUT AND
                                        IBUF                    ;PUT IT HERE
                                        MOV     #IBUF,  R2      ;SETUP POINTER
                                        CMPB    #15,    (R2)    ;WAS NEW DATA ENTERED?
                                        BEQ     GETEX           ;BR IF NO

                                        MOV     ARIA,   R3      ;SETUP DEST. POINTER
                                        MOVB    (R2)+,  (R3)+   ;MOV INPUT TO DEST.
                                        CMPB    #15,    (R2)    ;LAST DIGIT?
                                        BNE     .-6             ;LOOP IF NO
                                        CLRB    (R3)            ;INSERT ALL ZERO CHAR


                                GETEX:  CMP     R4,     #VISR   ;WAS THIS THE VARIABLE ISR
                                        BEQ     SWPRNT          ;BR IF YES
                                        JMP     GETIT           ;GET ANOPHER

                                ;TTY INTERUPTS HERE WHEN MODULE IS RUNNING.

                                TTYINT: MOV     @TKB,R1         ;CLEAR TTY BUFFER
                                        BIC     #100,@TKS       ;RESET INT. ENABLE
                                        BIC     #177700,R1      ;STRIP JUNK
                                        CMP     #7,R1
                                        BNE     2$
                                        CMP     #SWREG,SWR
                                        BNE     1$
                                        BIS     #BIT0,FLAG
                                1$:     BIS     #100,@TKS
                                        RTI
                                2$:     CMP     #'?,    R1      ;IS IT ?
                                        BNE     NOQ             ;BR IF NO
                                        MOV     #DMPLST,R0      ;SETUP DUMP LIST
                                        MOV     #IBUF,  (R0)    ;TO PRINTOUT INPUT BUFFER
                                        MOV     @IRDA,.RX       ;IF SWITCH REG. =0 PRINT RX BUFFER.
                                        MOV     @IXDA,.TX       ;" " " "            PRINT TX BUFFER
                                        JMP     LI              ;AND GO PRINT IT

                                NOQ:    CMP     #12,    R1      ;IS IT LINE FEED?
                                        BNE     RSTART          ;BR IF NO
                                        JMP     BEGIN           ;RESTART

                                RSTART: TYPE    ,MFILL
                                        RESET
                                        INCB    #0              ;DELAY HERE FOR AWILE
                                        BNE     .-4
                                SWPRNT: CMP     #SWREG,SWR
                                        BNE     XSWPNT
                                        BIS     #BIT1,FLAG
                                        TYPE    ,MSG09
                                        SETSWI
                                        BR      REST
                                XSWPNT: TYPE    ,MSG09          ;(15) 12) SET SWITCHES
                                        TSTB    @TKS            ;WAIT FOR TTY INPUT
```

```
 004154  100375                        BPL     .-4                 ;LOOP
 004156  017702  175334                MOV     @TKB,R2             ;RESET DONE FLAG
 004162  017746  004656                MOV     @SWR,   -(SP)
 004166  004037  006350                JSR     R0,$B2016           ;PRINTOUT SWITCHES
 004172  104400  001536                TYPE    .CRLF
 004176  000401                        BR      .+4                 ;SKIP OVER HALT
                                ;*****************************************************************
                                ;       SET SWITCH OPTIONS
                                ;*****************************************************************

 004200  000000                SWRSET: HALT                        ;HALT FOR SWITCH SETUP
                                                                    ;SW00-ONE WAY OUT
                                                                    ;SW01=ONE WAY IN
                                                                    ;SW02=EXTERNAL LOOPBACK
                                                                    ;SW03=INTERNAL LOOPBACK
                                                                    ;SW04=LOOP ON DATA
                                                                    ;SW05=MONITOR INPUT
                                                                    ;SW06=MONITOR OUTPUT
                                                                    ;SW07=NO DATA COMPARE
                                                                    ;SW08=EXTERNAL DATA
                                                                    ;SW09=DATA SELECT
                                                                    ;SW10=DATA SELECT
                                                                    ;SW11=DATA SELECT
                                                                    ;SW12=
                                                                    ;SW13=INHIBIT ERROR TYPEOUTS
                                                                    ;SW14=LOOP ON TEST
                                                                    ;SW15=HALT ON ERROR

 004202  012737  004210  003462  REST:   MOV     #RESTRT,$LPADR      ;SETUP LOOP
 004210  017701  004630          RESTRT: MOV     @SWR,R1
 004214  000301                          SWAB    R1
 004216  032777  000017  004620          BIT     #17,@SWR                    ;WAS SOME MODE SELECTED?
 004224  001003                          BNE     .+10                ;BR IF YES
 004226  104400  007637                  TYPE    ,MSG2               ;<15><12>NO MODE SELECTED.
 004232  000723                          BR      RSTART              ;GO GET SWITCH REGISTER.
 004234  042701  177761                  BIC     #177761, R1         ;STRIP JUNK
 004240  016137  001564  011022          MOV     DAT(1), IXDA        ;SETUP INIT DATA ADDR FROM TABLE
 004246  005701                          TST     R1                  ;VARIABLE DATA SPECIFIED?
 004250  001010                          BNE     SUXCC               ;BR IF NO
 004252  032777  000400  004564          BIT     #400,   @SWR        ;USE EXTERNAL DATA?
 004260  001004                          BNE     SUXCC               ;BR IF YES
                                ;*****************************************************************
                                ;       GET VARIABLE DATA
                                ;*****************************************************************
 004262  104400  007372                  TYPE    ,MSG10              ;<15><12> ENTER DATA <15><12>
 004266  104404                          GETSTR                      ;GET DATA
 004270  002604                          VDB                         ;AND PUT IT HERE
 004272  012737  001604  011020  SUXCC:  MOV     #IBUF,  IRDA        ;SETUP READ BUFFER ADDR

 004300  032777  000400  004536          BIT     #400,   @SWR        ;EXTERNAL DATA?
 004306  001403                          BEQ     SWRNXT              ;BR IF NO
 004310  012737  002604  011022          MOV     #VDB,   IXDA        ;SETUP BUFFER ADDRESS
 004316                  SWRNXT:
 004316  012737  003760  000060          MOV     #TTYINT,@#60        ;SETUP TTY VECTOR
 004324  012737  000340  000062          MOV     #340,   @#62        ;
 004332  012777  000100  175154          MOV     #100,   @TKS        ;AND ENABLE INTERRUPTS
```

```
 1038  004340  012702  001604                MOV     #IBUF,  R2
 1039  004344  005022                CLRIB:  CLR     (R2)+           ;CLEAR INPUT BUFFER
 1040  004346  022702  002004                CMP     #IBUF+200,R2
 1041  004352  001374                        BNE     CLRIB
```

```
1042                                    ;*************************
1043                                    ;       SETUP TIMER       *
1044                                    ;*************************
1045
1046  004354  012737  000060  006334  SUTIME: MOV     #60,    MSECS   ;PRESET COUNTER
1047  004362  012737  006312  000100          MOV     #TIMER, 100     ;SETUP LINE CLOCK VECTOR
1048  004370  012737  000340  000102          MOV     #340,   102     ;AND PRIORITY
1049  004376  012737  004422  000004          MOV     #NOLC,  4       ;SETUP BUS ERROR VECTOR
1050  004404  012737  000340  000006          MOV     #340,   6       ;SET UP PRIORITY TO 7
1051  004412  052737  000100  177546          BIS     #100,   177546  ;ENABLE LINE CLOCK
1052  004420  000441                          BR      NORTC
1053
1054                                          ;BUS ERROR RETURNS HERE IF NO LINE CLOCK
1055
1056  004422  012737  006312  000104  NOLC:   MOV     #TIMER, 104     ;SETUP RTC VECTOR
1057  004430  012737  000340  000106          MOV     #340,   106     ;AND PRIORITY
1058  004436  012737  004472  000004          MOV     #1$,    4       ;SETUP BUS ERROR VECTOR
1059  004444  012737  000340  000006          MOV     #340,   6       ;SET PRIORITY
1060  004452  012737  003100  172542          MOV     #1600.. 172542  ;SET COUNTER BUFFER.
1061  004460  012737  000111  172540          MOV     #111,   172540 ;ENABLE REAL TIME TIME CLOCK
1062  004466  000240                          NOP                     ;WAIT
1063  004470  000415                          BR      NORTC           ;CONTINUE.
1064  004472  012737  004516  000100  1$:     MOV     #2$,100         ;TRY LSI CLOCK
1065  004500  104414  000000                  STPS,PRTY0              ;LOWER PSW TO 0
1066  004504  005227  000000          3$:     INC     #0
1067  004510  001375                          BNE     3$
1068  004512  104400  007570                  TYPE,MSG19              ;NO CLOCK AVAILABLE
1069  004516  012737  006312  000100  2$:     MOV     #TIMER,100
1070
1071  004524  000137  004530          NORTC:  JMP     .+4             ;SPARE JUMP
1072  004530  012737  000006  000004          MOV     #6,@#4          ;SET TRAP VECTOR
1073  004536  005037  000006                  CLR     @#6             ;SET BUS ERROR VECTOR
1074  004542  012706  001070                  MOV     #STACK, SP      ;SETUP STACK
1075  004546  104414  000000                  STPS,PRTY0
1076  004552  012737  006350  011030          MOV     #$B2016,B2016   ;SETUP BIN TO OCT ADDR
1077
1078
1079                                    ;*****************************************************************
1080                                    ;       DO TESTING NOW
1081                                    ;*****************************************************************
1082  004560  032737  020000  001562          BIT     #20000, FLAGS   ;WAS A DN11 SETUP
1083  004566  001402                          BEQ     DMCHK           ;BR IF NO
1084  004570  004737  010200                  JSR     PC,DNGO         ;GO TO DN11 ISR
1085
1086  004574  032737  040000  001562  DMCHK:  BIT     #40000, FLAGS   ;WAS A DM11-B SETUP?
1087  004602  001402                          BEQ     VIGO            ;BR IF NO
1088  004604  004737  010422                  JSR     PC,DMBGO        ;GO TO DM11-B ISR
1089
1090                                    ;*********************************
1091                                    ;       GOTO THE MODULE AND RUN   *
1092                                    ;*********************************
1093
1094  004610  004777  004222          VIGO:   JSR     PC,@ISR+36      ;GO TO ISR
```

```
                                        ;*****************************************************************
                                        ;       END OF PASS ROUTINE
                                        ;*****************************************************************
004614  005237  004774          ECP:    INC     $PASS           ;INCREMENT PASS COUNTER
004620  005746                          TST     -(SP)           ;PUSH DOWN AND PROTECT STACK.
004622  104416                          KBDIN
004624  032777  010000  004212          BIT     #SW12,@SWR      ;INHIBIT TYPEOUTS?
004632  001034                          BNE     2$              ;BR IF YES
004634  104400  007411                  TYPE    ,MSG11          ;  5)(12) END OF PASS
004640  013746  004774                  MOV     $PASS, -(SP)
004644  004037  006350                  JSR     R0,$B2016       ;PRINTOUT PASS COUNT
004650  104400  001536                  TYPE    ,CRLF
004654  032700  000002                  BIT     #OWI,MODE       ;SKIP TRANSMIT TYPEOUT IF OWI
004660  001012                          BNE     3$              ;BR IF YES
004662  104400  010060                  TYPE    ,MSG26          ;TRANSMITTED DATA=
004666  013737  011022  004676          MOV     TXDA,4$         ;SET POINTER TO TXBUF
004674  104400                          TYPE                    ;TYPE TXBUFFER
004676  000000                  4$:     0
004700  032700  000001                  BIT     #OWO,MODE       ;SKIP RECEIVE TYPEOUT IF OWO
004704  001007                          BNE     2$              ;BR IF YES
004706  104400  010106          3$:     TYPE    ,MSG27          ;RECEIVED DATA=
004712  013737  011020  004722          MOV     TRDA,5$         ;SET POINTER TO RXBUF
004720  104400                          TYPE                    ;TYPE RXBUFFER
004722  000000                  5$:     0
004724  032777  040000  004112  2$:     BIT     #BIT14,@SWR     ;LOOP ON TEST?
004732  001005                          BNE     1$              ;BR IF NO...
004734  016600  000002                  MOV     2(SP),R0        ;GET RETURN ADDRESS
004740  104414  000000                  STPS,PRTY0
004744  000110                          JMP     (R0)            ;GO BACK TO MODULE.
004746  012706  001070          1$:     MOV     #STACK,SP       ;RESET THE STACK POINTER.
004752  013700  000042                  MOV     @#42,   R0      ;GET MONITOR ADDRESS
004756  001404                          BEQ     $DOAGN          ;BR IF NONE
004760  004710                          JSR     PC,(R0)         ;GO TO MONITOR
004762  000240                          NOP                     ;SAVE ROOM FOR
004764  000240                          NOP                     ;ACT-11
004766  000240                          NOP
004770  000137  000200          $DOAGN: JMP     @#200           ;RESTART TEST
004774  000000          $PASS:  0


                                        ;*****************************************************************
                                        ;       SUBROUTINE TO INPUT OCTAL WORD FROM OPERATOR
                                        ;*****************************************************************
004776  011046          GETANY: MOV     (R0),   -(SP)   ;PUT WORD ON STACK
005000  004037  006350                  JSR     R0,$B2016       ;AND TYPE IT
005004  104400  001526                  TYPE    ,QUES
005010  011037  001544                  MOV     (R0),WORK       ;PRESET FOR DEFAULT (CR)
005014  104406  001544          ANYMOR: ACCEPT  ,WORK           ;OCTAL READIN
005020  013710  001544                  MOV     WORK,   (R0)    ;MOVE IT TO ISR
005024  005720                  ANYEX:  TST     (R0)+           ;BUMP POINTER
005026  000240                          NOP
005030  000207                          RTS     PC              ;SUB/ROUTINE EXIT


                                        ;*****************************************************************
                                        ;       ERROR HLT HANDLER
                                        ;*****************************************************************
```

```
1151  005032                          $HLT:
1152  005032  104414  000140                  STPS,PRTY3              ;LOWER PSW PRIOTITY TO 3
1153  005036  005237  005730          $HLOT:  INC     $ERTTL          ;INCREMENT ERROR COUNTER
1154  005042  001775                  BEQ     $HLOT           ;MAKE SURE ITS NOT ZERO
1155  005044  011637  005726                  MOV     (SP)    ,$HLTAD ;SAVE ADDRESS OF HLT
1156  005050  162737  000002  005726          SUB     #2,     $HLTAD  ;AND BACK IT UP
1157  005056  010146                  MOV     R1,     -(SP)   ;SAVE R1
1158
1159  005060  032777  020000  003756          BIT     #BIT13, @SWR    ;INHIBIT ERR TYPEOUTS?
1160  005066  001070                  BNE     TRX             ;BR IF YES
1161
1162  005070  104400  001536                  TYPE    ,CRLF
1163  005074  117701  000626                  MOVB    @$HLTAD,R1      ;EXTRACT HLT CODE
1164  005100  006301                  ASL     R1              ;AND ALIGN IT
1165  005102  016137  005270  005112          MOV     EMTAB(R1),.+10  ;GET HEADER ADDRESS
1166  005110  104400  005320                  TYPE    ,EMO            ;AND PRINT HEADER
1167  005114  104400  007430                  TYPE    ,MSG12          ; < AT LOC >
1168  005120  013746  005726                  MOV     $HLTAD,-(SP)            ;GET HLT ADDRESS
1169  005124  004037  006350                  JSR     R0,$B2016       ;AND PRINT IT
1170  005130  005701                  TST     R1              ;HLT CODE = 0?
1171  005132  001446                  BEQ     TRX             ;BR IF YES
1172
1173
1174  005134  022701  000016                  CMP     #16.R1          ;IS IT HLT+7?
1175  005140  001023                  BNE     1$              ;BR IF NO
1176  005142  005702                  TST     R2              ;PRINTOUT BAD DATA?
1177  005144  001406                  BEQ     2$              ;BR IF NO
1178  005146  104400  007767                  TYPE    ,MSG23          ;< BAD DATA= ,
1179  005152  110246                  MOVB    R2,-(SP)        ;GET DATA
1180  005154  004037  006336                  JSR     R0,$B20CT       ;AND PRINT IT
1181  005160     003                  .BYTE   3
1182  005161     001                  .BYTE   1
1183  005162  005703          2$:     TST     R3              ;PRINT OUT GOOD DATA?
1184  005164  001410                  BEQ     3$              ;BR IF NO
1185  005166  104400  010003                  TYPE    ,MSG24          ;< GOOD DATA= >
1186  005172  110346                  MOVB    R3,-(SP)        ;GET DATA
1187  005174  004037  006336                  JSR     R0,$B20CT       ;AND PRINT IT
1188  005200     003                  .BYTE   3
1189  005201     001                  .BYTE   1
1190  005202  104400  001536                  TYPE    ,CRLF
1191  005206  000420          3$:     BR      TRX
1192
1193
1194  005210  005702          1$:     TST     R2              ;PRINTOUT RCV CSR?
1195  005212  001405                  BEQ     TR3             ;BR IF NO
1196  005214  104400  007441                  TYPE    ,MSG13          ; < RCV CSR=>
1197  005220  010246                  MOV     R2,     -(SP)   ;GET DATA
1198  005222  004037  006350                  JSR     R0,$B2016       ;AND PRINT IT
1199
1200  005226  005703          TR3:    TST     R3              ;PRINTOUT XMIT CSR?
1201  005230  001407                  BEQ     TRX             ;BR IF NO
1202  005232  104400  007453                  TYPE    ,MSG14          ; < XMIT CSR=>
1203  005236  010346                  MOV     R3,     -(SP)   ;GET DATA
1204  005240  004037  006350                  JSR     R0,$B2016       ;AND PRINT IT
1205  005244  104400  001536                  TYPE    .CRLF
1206
```

```
005250  032777  100000  003566  TRX:    BIT     #BIT15, @SWR    ;HALT ON ERROR?
005256  001401                          BEQ     HLTX            ;BR IF NO
005260  000000                          HALT

005262  104416                  HLTX:   KBDIN
005264  012601                          MOV     (SP)+,  R1      ;RESTORE R1
005266  000002                          RTI                     ;AND RETURN TO PROGRAM

005270  005320                  EMTAB:  EM0
005272  005333                          EM1
005274  005343                          EM2
005276  005363                          EM3
005300  005407                          EM4
005302  005422                          EM5
005304  005446                          EM6
005306  005542                          EM7
005310  005565                          EM10
005312  005642                          EM11
005314  005667                          EM12
005316  005711                          EM13

005320  051105  047522  020122  EM0:    .ASCIZ  "ERROR HALT"
005333     127  044501  044524  EM1:    .ASCIZ  "WAITING"
005343     127  044501  044524  EM2:    .ASCIZ  "WAITING TO XMIT"
005363     104  026516  030461  EM3:    .ASCIZ  "DN-11 NOT AVAILABLE"
005407     104  030516  026461  EM4:    .ASCIZ  "DN11-ERROR"
005422  047104  030461  041440  EM5:    .ASCIZ  "DN11 CALL ABANDONED"
005446  041522  020126  052502  EM6:    .ASCIZ  "RCV BUFFER FULL,END OF MESSAGE CHARACTER(001) WAS NOT FOUND"
005542  040504  040524  041440  EM7:    .ASCIZ  "DATA COMPARE ERROR"
005565     105  051122  051117  EM10:   .ASCIZ  "ERROR  RCV CSR=CONTENTS OF SELECT 0 REGISTER"
005642  047125  054105  042520  EM11:   .ASCIZ  "UNEXPECTED INTERRUPT"
005667     116  046530  050040  EM12:   .ASCIZ  "NXM PRINCIPAL CAR"
005711     116  046530  040440  EM13:   .ASCIZ  "NXM ALT CAR"
        005726                          .EVEN
005726  000000                  $HLTAD: 0
005730  000000                  $ERTTL: 0
```

```
                                ;*****************************************************************
                                ;       READ A CHAR. ROUTINE
                                ;*****************************************************************
                                ;
                                ;       CALL=   GETCHR  ;INPUT A CHAR FROM TTY
                                ;                       ;RETURNS HERE WITH CHAR ON STACK
005732  011646                  $READC: MOV     (SP),-(SP)      ;PUSH THE PC
005734  016666  000004  000002          MOV     4(SP),2(SP)     ;SAVE THE PS
005742  105777  173546                  TSTB    @TKS            ;IS RECEIVE DONE
005746  100375                          BPL     .-4             ;LOOP IF NO
005750  017737  173542  001542          MOV     @TKB    ,$CHAR  ;SAVE THE CHAR.
005756  042737  177600  001542          BIC     #177600,$CHAR   ;$STRIP JUNK
005764  013766  001542  000004          MOV     $CHAR,4(SP)     ;PUT CHAR ON STACK
006772  000002                          RTI                     ;EXIT
                                ;*****************************************************************
                                ;       READ A STRING ROUTINE
                                ;*****************************************************************
                                ;
                                ;       CALL=   GETSTR          ;INPUT A STRING OF CHARS FROM TTY
                                ;               ADDR    ;TO THIS ADDRESS
                                ;                       ;TERMINATE INPUT WITH LINE FEED
006774  011602                  $READS: MOV     (SP),   R2      ;SETUP ADDRESS OF INPUT BUFFER
006776  012201                          MOV     (R2)+,  R1      ;INCREMENT RETURN ADDRESS
007000  010216                          MOV     R2,     (SP)    ;AND PUT BACK ON STACK
                                ;
007002  104402                  GETIC:  GETCHR                  ;GET A CHAR
007004  104400  001542                  TYPE    ,$CHAR
007010  122726  000136                  CMPB    #136,   (SP)+   ;IS IT BINARY DELIMITER
007014  001011                          BNE     GOTIC           ;BR IF NO
                                ;
007016  104406  001544          OCT:    ACCEPT  ,$WORK          ;GET OCT. CHAR
007022  113721  001544                  MOVB    $WORK,(R1)+     ;STORE OCT. CHAR.
007026  122737  000136  001542          CMPB    #136,$CHAR      ;TERMINATOR=BIN. DELIMITER?
007034  001370                          BNE     OCT             ;BR IF NO
007036  000761                          BR      GETIC           ;
007040                          GOTIC:
007040  113721  001542                  MOVB    $CHAR,  (R1)+   ;STORE CHAR. IN BUFFER
007044  022737  000015  001542          CMP     #15,    $CHAR   ;IS IT END OF INPUT (CAR. RETURN)
007052  001353                          BNE     GETIC           ;BR IF NO
007054  112721  000012                  MOVB    #12,(R1)+
007060  104400  001537                  TYPE    ,LF             ;TYPE A LINE FEED
007064  112721  000001                  MOVB    #001,(R1)+      ;INSERT RX TERM.
007070  112721  000177                  MOVB    #177,   (R1)+   ;AND A FILL
007074  112721  000177                  MOVB    #177,   (R1)+   ;INSERT ANOTHER FILL
007100  112721  000177                  MOVB    #177,   (R1)+   ;INSERT 3RD FILL
007104  112721  000177                  MOVB    #177,   (R1)+   ;INSERT 4TH FILL
007110  105011                          CLRB    (R1)            ;PUT ZEROS AT END
007112  000002                          RTI
```

```
1278                                    ;***********************************************
1279                                    ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
1280                                    ;CALL:
1281                                    ;       ACCEPT  .ADDR           ;PUT OCTAL NUMBER IN ADDR
1282                                    ;
1283  006114                            $ACCEPT:
1284  006114  010046                           MOV     R0,-(SP)        ;SAVE R0
1285  006116  010146                           MOV     R1,-(SP)        ;SAVE R1
1286  006120  010246                           MOV     R2,-(SP)        ;SAVE R2
1287  006122  010346                           MOV     R3,-(SP)        ;SAVE R3
1288  006124  016600  000010                   MOV     10(SP),R0               ;GET ADDRESS OF WHERE TO PUT NUMBER
1289  006130  005001                    1$:    CLR     R1              ;CLEAR PARTIAL NUMBER
1290  006132  012702  000006                   MOV     #6,R2           ;MAX. # OF DIGITS ALLOWED
1291  006136  104402                    2$:    GETCHR                  ;GET ONE CHARACTER
1292  006140  112603                           MOVB    (SP)+,R3        ;AND PUT IT IN R3
1293  006142  110337  006310                   MOVB    R3,6$
1294  006146  104400  006310                   TYPE    ,6$             ;ECHO THE CHARACTER
1295  006152  022703  000025                   CMP     #25,R3
1296  006156  001451                           BEQ     8$
1297  006160  022703  000015                   CMP     #15,R3          ;WAS THIS CHARACTER A "CR"?
1298  006164  001427                           BEQ     5$              ;BR IF YES
1299  006166  022703  000040                   CMP     #40,R3          ;WAS "SPACE" HIT?
1300  006172  001433                           BEQ     7$              ;BR IF YES
1301  006174  022703  000136                   CMP     #136,R3         ;WAS " ↑ " HIT?
1302  006200  001430                           BEQ     7$              ;BR IF YES
1303  006202  032703  000110                   BIT     #110,R3         ;INSURE THE CHARACTER IS
1304  006206  001011                           BNE     4$              ;A DIGIT BETWEEN 0 AND 7.
1305                                          ;
1306  006210  005302                           DEC     R2              ;CHECK NUMBER OF CHARACTERS
1307  006212  002407                           BLT     4$              ;BR IF TO MANY
1308  006214  006301                           ASL     R1              ;POSITION PARTIAL NUMBER
1309  006216  006301                           ASL     R1              ;FOR THIS DIGIT
1310  006220  006301                           ASL     R1
1311  006222  042703  177770                   BIC     #^C<7>,R3       ;GET RID OF THE ASCII JUNK
1312  006226  050301                           BIS     R3,R1           ;COMBINE THIS DIGIT WITH PARTIAL
1313  006230  000742                           BR      2$              ;GO GET ANOTHER DIGIT
1314  006232  104400  001526            4$:    TYPE    .QUES           ;TYPE "?"
1315  006236  104400  001536                   TYPE    .CRLF           ;TYPE CARRAGE RETURN AND LINE FEED.
1316  006242  000732                           BR      1$              ;GO START OVER
1317  006244  104400  001537            5$:    TYPE    .LF             ;FOLLOW "CR" WITH A "LF"
1318  006250  022702  000006                   CMP     #6,R2           ;WERE ANY DIGITS INPUT
1319  006254  001002                           BNE     .+6             ;BR IF YES
1320  006256  013001                           MOV     @(R0)+,R1       ;USE OLD DATA
1321  006260  005740                           TST     -(R0)           ;BACKUP R0--
1322  006262  010130                    7$:    MOV     R1,@(R0)+       ;PASS THE NUMBER TO THE USER
1323  006264  010066  000010                   MOV     R0,10(SP)               ;SET FOR RETURN
1324  006270  012603                           MOV     (SP)+,R3        ;RESTORE R3
1325  006272  012602                           MOV     (SP)+,R2        ;RESTORE R2
1326  006274  012601                           MOV     (SP)+,R1        ;RESTORE R1
1327  006276  012600                           MOV     (SP)+,R0        ;RESTORE R0
1328  006300  000002                           RTI
1329  006302  104400  010150            8$:    TYPE,   CTLG
1330  006306  000715                           BR      1$
1331  006310      000      000           6$:    .BYTE   0,0             ;STORAGE FOR ASCII CHAR. AND TERMINATOR
```

```
                                ;*****************************************************************
                                ;       CLOCK INTERRUPT ROUTINE
                                ;*****************************************************************
006312  005337  006334        TIMER:  DEC     MSECS           ;COUNT 60 CYCLES
006316  001005                        BNE     TIMEX           ;BR IF NOT 60
006320  012737  000060  006334        MOV     #60,    MSECS   ;RESTORE COUNT
006326  005237  011032                INC     TIME            ;INCREMENT SECONDS
006332                          TIMEX:
006332  000002                        RTI                     ;RETURN FROM INTERRUPT
006334  000000                MSECS:  0
                                ;*****************************************************
                                ;BINARY TO OCTAL (ASCII) AND TYPE
                                ;$B2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;CALL:
                                ;       MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                                ;       JSR     R0,$B2OCT               ;CALL FOR TYPEOUT
                                ;       .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;       .BYTE   M                       ;M=1 OR 0
                                ;                                               ;1=TYPE LEADING ZEROS
                                ;                                               ;0=SUPPRESS LEADING ZEROS
                                ;
                                ;$B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B5OCT OR $B2O16
                                ;CALL:
                                ;       MOV     NUM,-(SP)
                                ;       JSR     R0,$B201
                                ;
                                ;$B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;CALL:
                                ;       MOV     NUM,-(SP)
                                ;       JSR     R0,$B2016
006336  112037  006543  $B2OCT:  MOVB    (R0)+,$OMODE+1          ;PICKUP THE NUMBER OF DIGITS TO TYPE
006342  112037  006541          MOVB    (R0)+,$OFILL            ;GET THE ZERO FILL SWITCH
006346  000406                  BR      $B201
006350  112737  000001  006541  $B2016:  MOVB    #1,$OFILL       ;SET THE ZERO FILL SWITCH
006356  112737  000006  006543          MOVB    #6,$OMODE+1     ;SET FOR SIX(6) DIGITS
006364  112737  000005  006540  $B201:   MOVB    #5,$OCNT        ;SET THE ITERATION COUNT
006372  010346                  MOV     R3,-(SP)                ;SAVE R3
006374  010446                  MOV     R4,-(SP)                ;SAVE R4
006376  010546                  MOV     R5,-(SP)                ;SAVE R5
006400  113704  006543          MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
006404  005404                  NEG     R4
006406  062704  000006          ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
006412  110437  006542          MOVB    R4,$OMODE               ;SAVE IT FOR LSE
006416  113704  006541          MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
006422  016605  000010          MOV     10(SP),R5               ;PICKUP THE INPUT NUMBER
006426  005003                  CLR     R3                      ;CLEAR THE OUTPUT WORD
006430  006105          1$:     ROL     R5                      ;ROTATE MSB INTO "C"
006432  000404                  BR      3$                      ;GO DO MSB
006434  006105          2$:     ROL     R5                      ;FORM THIS DIGIT
006436  006105                  ROL     R5
006440  006105                  ROL     R5
006442  010503                  MOV     R5,R3
006444  006103          3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
006446  105337  006542          DECB    $OMODE                  ;TYPE THIS DIGIT
006452  100016                  BPL     7$                      ;BR IF NO
```

```
 1388  006454  042703  177770          BIC     #177770,R3          ;GET RID OF JUNK
 1389  006460  001002                  BNE     4$                  ;TEST FOR 0
 1390  006462  005704                  TST     R4                  ;SUPPRESS THIS 0?
 1391  006464  001403                  BEQ     5$                  ;BR IF YES
 1392  006466  005204          4$:     INC     R4                  ;DON'T SUPPRESS ANYMORE 0'S
 1393  006470  052703  000060          BIS     #'0,R3              ;MAKE THIS DIGIT ASCII
 1394  006474  052703  000040  5$:     BIS     #' ,R3              ;MAKE ASCII IF NOT ALREADY
 1395  006500  110337  006536          MOVB    R3,8$               ;SAVE FOR TYPING
 1396  006504  104400  006536          TYPE    ,8$                 ;GO TYPE THIS DIGIT
 1397  006510  105337  006540  7$:     DECB    $OCNT               ;COUNT BY 1
 1398  006514  003347                  BGT     2$                  ;BR IF MORE TO DO
 1399  006516  002402                  BLT     6$                  ;BR IF DONE
 1400  006520  005204                  INC     R4                  ;INSURE LAST DIGIT ISN'T A BLANK
 1401  006522  000744                  BR      2$                  ;GO DO THE LAST DIGIT
 1402  006524  012605          6$:     MOV     (SP)+,R5            ;RESTORE R5
 1403  006526  012604                  MOV     (SP)+,R4            ;RESTORE R4
 1404  006530  012603                  MOV     (SP)+,R3            ;RESTORE R3
 1405  006532  012616                  MOV     (SP)+,(SP)          ;SET THE STACK FOR RETURNING
 1406  006534  000200                  RTS     R0                  ;RETURN
 1407  006536  000             9$:     .BYTE   0                   ;STORAGE FOR ASCII DIGIT
 1408  006537  000                     .BYTE   0                   ;TERMINATOR FOR TYPE ROUTINE
 1409  006540  000             $OCNT:  .BYTE   0                   ;OCTAL DIGIT COUNTER
 1410  006541  000             $OFILL: .BYTE   0                   ;ZERO FILL SWITCH
 1411  006542  000000          $OMODE: 0                           ;NUMBER OF DIGITS TO TYPE
 1412
 1413
                                 ;*****************************************************
                                 ;TRAP HANDLER
 0044  006544  010046  $TRAP:    MOV     R0,-(SP)            ;SAVE R0
 0045  006546  016600  000002    MOV     2(SP),R0            ;GET TRAP ADDRESS
 0046  006552  005740            TST     -(R0)               ;BACKUP BY 2
 0047  006554  111000            MOVB    (R0),R0             ;GET RIGHT BYTE OF TRAP
 0048  006556  016000  006564    MOV     $TRPAD(R0),R0       ;INDEX TO TABLE
 0049  006562  000200            RTS     R0                  ;GO TO ROUTINE

                                 ;TRAP TABLE
                                 ;       ROUTINE
                                 ;       -------

 0053  006564  001100  $TRPAD:  $TYPE
       006566  104400                           TYPE=TRAP+0
 0054  006566  005732           $READC
       006570  104402                           GETCHR=TRAP+2
 0055  006570  005774           $READS
       006572  104404                           GETSTR=TRAP+4
 0056  006572  006114           $ACCEPT
       006574  104406                           ACCEPT=TRAP+6
 0057  006574  006610           $RWAIT
       006576  104410                           RWAIT=TRAP+10
 0058  006576  006620           $XWAIT
       006600  104412                           XWAIT=TRAP+12
 0059  006600  007042           .STPS
       006602  104414                           STPS=TRAP+14
 0060  006602  006732           .KBDIN
       006604  104416                           KBDIN=TRAP+16
 0061  006604  007064           .SUSPR
```

```
1444            104420                          SUSWR=TRAP+20
1445  006606    006750                .SETSWI
1446            104422                          SETSWI=TRAP+22
1447
1448                                  ;*****************************************************************
1449                                  ;       SPECIAL PRINTOUT ROUTINES
1450                                  ;*****************************************************************
1451
1452  006610    104400  007462        $RWAIT: TYPE    ,MSG15          ;<15><12> WAITING AT LOC <SP>
1453  006614    000137  006624                JMP     WAITPO
1454
1455  006620    104400  007505        $XWAIT: TYPE    ,MSG16          ;<15><12> WAITING FOR CLEAR TO SEND AT LOC <SP><SP>
1456
1457  006624    011646                WAITPO: MOV     (SP),   -(SP)   ;SETUP ADDRESS OF CALL FOR PRINTOUT
1458  006626    004037  006350                JSR     R0,$B2016       ;PRINTOUT ADDRESS
1459
1460  006632    104400  010021                TYPE    ,MSG25          ; <DMBB LINE STATUS REG= >
1461
1462  006636    016646  000004                MOV     4(SP)   ,-(SP)  ;MOV CSR TO BOTTOM OF STACK
1463  006642    016666  000004  000006        MOV     4(SP),  6(SP)   ;MOVE PSW UP A WORD
1464  006650    016666  000002  000004        MOV     2(SP),  4(SP)   ;MOVE PC UP A WORD
1465  006656    012616                        MOV     (SP)+,  (SP)    ;MOVE CSR UP A WORD
1466  006660    004037  006350                JSR     R0,$B2016       ;GO PRINT CSR
1467
1468  006664    104400  007564                TYPE    ,MSG18          ;<.><15><12>
1469                                                                    ;GIVE IT LINE FEED
1470
1471  006670    000002                        RTI                     ;AND RETURN TO CALLER
1472
1473                                  ;*****************************************************************
1474                                  ;
1475                                  ;POWER DOWN ROUTINE.
1476                                  ;SINCE INIT IS ISSUED IN A PWR DN/UP SEQUENCE.
1477                                  ;PROGRAM MUST BE RESTARTED AGAIN.
1478                                  ;
1479
1480  006672    012737  006704  000024  $PWRDN: MOV     #$PWRUP,@#24
1481  006700    000000                        HALT
1482  006702    000776                        BR      .-2
1483
1484                                  ;*****************************************************************
1485                                  ;
1486                                  ;POWER UP ROUTINE.
1487                                  ;MESSAGE "POWER HAS FAILED..." WILL BE PRINTED OUT.
1488                                  ;PROGRAM WILL BE RESTARTED.
1489                                  ;
1490
1491  006704    012737  006672  000024  $PWRUP: MOV     #$PWRDN,@#24
1492  006712    012706  001070                MOV     #STACK,SP
1493  006716    104400  007616                TYPE    ,MSG20  ;<15><12> POWER FAILED..
1494  006722    104414  000000                STPS,PRTYO
1495  006726    000137  000200                JMP     @#200
1496
1497  006732    032737  000001  011042  .KBDIN: BIT     #BIT0,FLAG      ;TEST ^G FLAG
1498  006740    001437                        BEQ     OUT             ;NO, EXIT
1499  006742    042777  000100  172544        BIC     #100,@TKS       ;CLEAR TTY IE
```

```
1500  006750  104400  010131          .SETSWI:TYPE    ,SWEQ           ;TYPE SWR=
1501  006754  017746  002064                  MOV     @SWR,-(SP)      ;SET UP OCTAL TYPEOUT
1502  006760  004037  006350                  JSR     R0,$B2016       ;DO IT
1503  006764  104400  010141                  TYPE    ,NEQ            ;TYPE NEW=
1504  006770  017737  002050  001544          MOV     @SWR,WORK       ;SET UP FOR CR DEFAULT
1505  006776  104406  001544                  ACCEPT  ,WORK           ;GET VALUE
1506  007002  013777  001544  002034          MOV     WORK,@SWR       ;REPLACE IT
1507  007010  032737  000002  011042          BIT     #BIT1,FLAG      ;SEE HOW WE GOT HERE
1508  007016  001005                          BNE     1$              ;WRONG WAY?
1509  007020  005777  172472                  TST     @TKB            ;CLEAR BUFFER
1510  007024  052777  000100  172462          BIS     #100,@TKS       ;RESET TTY IE
1511  007032  042737  000003  011042  1$:     BIC     #BIT0+BIT1,FLAG ;CLEAR FLAG BITS
1512  007040  000002                  OUT:    RTI                     ;EXIT
1513
1514  007042  042766  000340  000002  .STPS:  BIC     #PRTY7,2(SP)    ;CLEAR OUT PRIORITY BITS
1515  007050  057666  000000  000002          BIS     @(SP),2(SP)     ;SET NEW PRIORITY
1516  007056  062716  000002                  ADD     #2,(SP)         ;SETUP EXIT
1517  007062  000002                          RTI                     ;EXIT
1518
1519  007064  013746  000006          .SUSWR: MOV     6,-(SP)         ;SAVE 6 ON STACK
1520  007070  013746  000004                  MOV     4,-(SP)         ;SAVE 4 ON STACK
1521  007074  012737  007122  000004          MOV     #1$,4           ;SETUP TIMEOUT
1522  007102  012737  000340  000006          MOV     #340,6          ;SET PRIORITY
1523  007110  022777  177777  001726          CMP     #-1,@SWR        ;TEST FOR 177570
1524  007116  001402                          BEQ     2$              ;NOT ALL 1'S
1525  007120  000407                          BR      3$              ;IT'S THERE - EXIT
1526  007122  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJUST STACK AFTER TRAP
1527  007124  012737  000176  011044  2$:     MOV     #SWREG,SWR      ;REPLACE HARDWARE REGISTERS
1528  007132  012737  000174  011046          MOV     #DISPREG,DISP_AY  ;WITH SOFTWARE REGISTERS
1529  007140  012637  000004          3$:     MOV     (SP)+,4         ;RESTORE 4
1530  007144  012637  000006                  MOV     (SP)+,6         ;RESTORE 6
1531  007150  000002                          RTI                     ;EXIT
```

```
                        ;*****************************************************************
                        ;
                        ;       AREA RESERVED FOR MOST ASCIZ MESSAGES.
                        ;
                        ;*****************************************************************


007152  000040                    MSG00:  .ASCIZ  / /
007154  005015  047111  042524    MSG01:  .ASCIZ  <15><12>/INTERFACE TYPE  /
007177     015  044412  051123    MSG02:  .ASCIZ  <15><12>/ISR NOT LOADED!!/
007222  005015  052502  020123    MSG03:  .ASCIZ  <15><12>/BUS ADDRESS=/
007241     015  053012  041505    MSG04:  .ASCIZ  <15><12>/VECTOR ADDRESS=/
007263     015  050012  044522    MSG05:  .ASCIZ  <15><12>/PRIORITY=/
007277     015  050012  051101    MSG06:  .ASCIZ  <15><12>/PARAMS #1=/
007314  005015  040520  040522    MSG07:  .ASCIZ  <15><12>/PARAMS #2=/
007331     015  040412  041523    MSG08:  .ASCIZ  <15><12>/ASCII PARAM=/
007350  005015  042523  020124    MSG09:  .ASCIZ  <15><12>/SET SWITCHES.../
007372  005015  047105  042524    MSG10:  .ASCIZ  <15><12>/ENTER DATA/<15><12>
007411     015  042412  042116    MSG11:  .ASCIZ  <15><12>/END OF PASS /
007430  040440  020124  047514    MSG12:  .ASCIZ  / AT LOC /
007441     040  041522  020126    MSG13:  .ASCIZ  / RCV CSR=/
007453     040  041530  051123    MSG14:  .ASCIZ  / XCSR=/
007462  005015  053440  044501    MSG15:  .ASCIZ  <15><12>/ WAITING AT LOC /
007505     015  053412  044501    MSG16:  .ASCIZ  <15><12>/WAITING FOR CLEAR TO SEND AT LOC
007551     040  020041  041440    MSG17:  .ASCIZ  / !  CSR=  /
007564  006456  000012            MSG18:  .ASCIZ  /./<15><12>
007570  005015  047516  041440    MSG19:  .ASCIZ  <15><12>/NO CLOCKS AVAILABLE/
007616  005015  047520  042527    MSG20:  .ASCIZ  <15><12>/POWER FAILED../
007637     015  047012  020117    MSG21:  .ASCIZ  <15><12>/NO MODE SELECTED./
007663     015  044412  020106    MSG22:  .ASCII  <15><12>/IF CALLING,DIAL NUMBER/
007713     015  044412  020106            .ASCIZ  <15><12>/IF ANSWERING,PLACE MODEM IN AUTO-ANSWER/<15><12>
007767     073  041040  042101    MSG23:  .ASCIZ  /; BAD DATA=/
010003     040  020040  047507    MSG24:  .ASCIZ  /   GOOD DATA=/
010021     015  020012  046504    MSG25:  .ASCIZ  <15><12>/ DMBB LINE STATUS REGISTER= /
010060  005015  051124  047101    MSG26:  .ASCIZ  <15><12>/TRANSMITTED DATA=/<15><12>
010106  005015  042522  042503    MSG27:  .ASCIZ  <15><12>/RECEIVED DATA=/<15><12>
010131     015  051412  051127    SWEQ:   .ASCIZ  <15><12>/SWR= /
010141     040  042516  036527    NEG:    .ASCIZ  / NEW= /
010150  052536  005015     000    CTLL:   .ASCIZ  /↑U/<15><12>
010155     177  000177            MFILL:  .ASCIZ  <177><177>
                                  .EVEN
```

```
 (1)                                    ;*****************************************************************
 1541                                   ;       DN-11 INTERFACE SERVICE PARAMS
 1542                                   ;*****************************************************************
 1543  010160  047104  000040          DN:     .ASCIZ  "DN "
 1544  010164  175200                  DNBA:   175200                  ;BUS ADDRESS
 1545  010166  000350                  DNIV:   350                     ;INTERRUPT VECTOR
 1546  010170  000200                  DNPRI:  200                     ;PRIORITY
 1547  010172  177777                  DNPAR1: 177777                  ;NOT USED
 1548  010174  177777                  DNPAR2: 177777                  ;NOT USED
 1549  010176  010356                  DNPAR3: DIALNO                  ;ADDRESS OF DIAL #
 1550
 1551
 1552
 1553          100000                          PWI=100000              ;POWER INDICATOR
 1554          040000                          ACR=40000               ;ABANDON CALL AND RETRY
 1555          010000                          DLO=10000               ;DATA LINE OCCUPIED
 1556          000040                          DSS=40                  ;DATA SET STATUS
 1557          000020                          PND=20                  ;PRESENT NEXT DIGIT
 1558          000002                          DP=2                    ;DIGIT PRESENT
 1559          000001                          CRQ=1                   ;CALL REQUEST
 1560
```

```
1561                                    ;*****************************************************************
1562                                    ;       START OF DN-11 CODE
1563                                    ;*****************************************************************
1564  010200  013704  010164   DNGO:    MOV     DNBA,   R4      ;SETUP BUS ADDR
1565  010204  005014                    CLR     (R4)            ;RESET DN-11
1566  010206  005037  011032            CLR     TIME            ;RESET TIMER
1567  010212  032737  000002  011032    BIT     #2,     TIME    ;AND WAIT 2 SECS
1568  010220  001774                    BEQ     .-6
1569  010222  012703  010356            MOV     #DIALNO,R3      ;SETUP DIAL # ADDRESS
1570  010226  012714  000001            MOV     #CRQ,   @CSR    ;SET CALL REQUEST
1571  010232  032714  100000            BIT     #PWI,   @CSR    ;IS DN AVAILABLE
1572  010236  001425                    BEQ     DNL1X           ;BR IF YES
1573
1574  010240  011402                    MOV     @CSR,   R2      ;SETUP CSR FOR PRINTOUT
1575  010242  005003                    CLR     R3
1576  010244  104003                    HLT+3                   ;PRINTOUT "DN NOT AVAILABLE"
1577  010246  000137  010200            JMP     DNGO            ;RESTART
1578
1579  010252  032714  000200   DNL1:    BIT     #DONE,  @CSR    ;IS DONE FLAG SET?
1580  010256  001775                    BEQ     DNL1            ;WAIT IF NO
1581  010260  042714  000200            BIC     #DONE,  @CSR    ;RESET DONE
1582  010264  032714  140000            BIT     #ACR+PWI,@CSR   ;ANY ERRORS?
1583  010270  001003                    BNE     DNL1E           ;BR IF YES
1584
1585
1586  010272  032714  000020            BIT     #PND,   @CSR    ;IS PRESENT NEXT DIGIT SET
1587  010276  001005                    BNE     DNL1X           ;BR IF YES
1588
1589  010300  011402   DNL1E:    MOV     @CSR,   R2      ;SETUP CSR FOR PRINTOUT
1590  010302  005003                    CLR     R3
1591  010304  104004                    HLT+4                   ;PRINTOUT "DN ERROR"
1592  010306  000137  010200            JMP     DNGO            ;RESTART
1593
1594  010312  112364  000001   DNL1X:   MOVB    (R3)+,  1(R4)   ;LOAD NEXT DIGIT
1595  010316  052714  000002            BIS     #DP,    @CSR    ;SET DIGIT PRESENT
1596  010322  105713                    TSTB    (R3)            ;WAS THAT LAST CHAR?
1597  010324  001352                    BNE     DNL1            ;BR IF NO
1598
1599  010326  032714  000200   DNL2:    BIT     #DONE,  @CSR    ;WAIT FOR DONE FLAG
1600  010332  001775                    BEQ     DNL2
1601
1602  010334  032714  040000            BIT     #ACR,   @CSR    ;WAS CALL ABANDONED?
1603  010340  001405                    BEQ     DNEX            ;BR IF NO
1604
1605  010342  011402                    MOV     @CSR,   R2      ;SETUP CSR FOR PRINTOUT
1606  010344  005003                    CLR     R3
1607  010346  104005                    HLT+5                   ;PRINTOUT "CALL ABANDONED"
1608  010350  000137  010200            JMP     DNGO            ;RESTART
1609
1610  010354  000207   DNEX:    RTS     PC              ;RETURN TO ITEP
1611
1612  010356  034470  032467  031460  DIALNO: .ASCIZ  "8975030"              ;NUMBER TO DIAL
1613  010364  000060
1614  010366  000000  000000  000000          0,0,0,0,0,0
1615  010374  000000  000000  000000
```

```
1616                                    ;*****************************************************************
1617                                    ;       DM11-B INTERFACE SERVICE PARAMS
1618                                    ;*****************************************************************
1619
1620
1621  010402  046504  000102           DMB:    .ASCIZ  'DMB'           ;ISR NAME
1622  010406  170500                   MBA:    170500                  ;BUS ADDRESS
1623  010410  000310                   IV:     310                     ;VECTOR ADDRESS
1624  010412  000200                   PRIO:   200                     ;PRIORITY
1625  010414  000000                   PARA1:  0                       ;PARAM #1
1626  010416  177777                   PARA2:  177777                  ;PARAM #2
1627  010420  177777                   PARA3:  177777                  ;PARAM #3
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
```

```
                                ;*****************************************************************
                                ;       DM-11B INTERFACE SERVICE ROUTINE
                                ;*****************************************************************
010422  000240                  DMBGO:  NOP
010424  013704  010406                  MOV     MBA,    R4      ;SETUP BUS ADDR INDEX
010430  005037  011032                  CLR     TIME            ;RESET TIMER

010434  052714  006000                  BIS     #CS+CM, @CSR    ;CLEAR DM-11 B
010440  032714  000020                  BIT     #BUSY,  @CSR    ;WAIT TIL FREE
010444  001375                        BNE     .-4
010446  013714  010414                  MOV     PARA1,  @CSR    ;SELECT LINE #
010452  052764  000003  000002          BIS     #DTR+LE,2(R4)   ;SET DATA TERM RDY & LINE ENABLE
010460  032737  020000  001562          BIT     #20000,FLAGS    ;HAS DM11 MADE CONNECTION YET?
010466  001002                        BNE     1$              ;BR IF YES
010470  104400  007663                  TYPE    .MSG22          ;TYPE "MAKE CONNECTION"

010474  032777  000005  000342  1$:     BIT     #OWO+XLB,@SWR   ;IS MODE = OWO OR XLB
010502  001444                        BEQ     REX1            ;BR IF NO

010504  052764  000004  000002  STARTX: BIS     #RQTS,  2(R4)   ;SET REQUEST TO SEND

010512  032764  000040  000002  CTSW:   BIT     #CTS,   LSTAT(R4)       ;IS CLEAR TO SEND SET?
010520  001016                        BNE     CTSOK           ;BR IF YES
010522  023727  011032  000036          CMP     TIME,   #36     ;30 SECS ELASPED?
010530  103770                        BLO     CTSW            ;BR IF NO
010532  016446  000002                  MOV     LSTAT(R4),-(SP) ;TYPE CONTENTS OF RCSR
010536  032777  010000  000300          BIT     #SW12,@SWR      ;INHIBIT TYPEOUTS?
010544  001001                        BNE     1$              ;BR IF YES
010546  104412                        XWAIT                   ;PRINTOUT 'WAITING FOR CTS'
010550  005037  011032          1$:     CLR     TIME            ;RESET TIMER
010554  000756                        BR      CTSW            ;WAIT SOME MORE

010556                          CTSOK:

010556  012737  177777  011024  REX:    MOV     #-1,SETTLE      ;SET UP DELAY FLAG
010564  005037  010616                  CLR     TEMP1
010570  012737  000014  010620          MOV     #14,TEMP2
010576  062737  000001  010616          ADD     #1,TEMP1
010604  001374                        BNE     .-6
010606  005337  010620                  DEC     TEMP2
010612  001371                        BNE     .-14
010614  000207                  REX1:   RTS     PC              ;RETURN TO CONTROL PROGRAM

010616  000000                  TEMP1:  0
010620  000000                  TEMP2:  0
        000001                          OWO=1
        000002                          OWI=2
        000004                          TLB=4
        000004                          XLB=4
        000010                          ILB=10

                                ;       RCSR EQUATES
                                        .EQUIV  R4,RCSR
                                        .EQUIV  R4,CSR
```

```
1694
1695            100000                  RI=100000               ;RING INDICATOR
1696            000100                  CF=100                  ;CARRIER FLAG
1697            000040                  CTS=40                  ;CLEAR TO SEND
1698            010000                  SRD=10000               ;SEC. RECEIVE DATA
1699            004000                  CS=4000                 ;CLEAR SCAN
1700            002000                  CM=2000                 ;CLEAR MUX
1701            001000                  MM=1000                 ;MAINT MODE
1702            000400                  STEP=400                ;STEP
1703            000200                  DONE=200                ;DONE
1704            000100                  IE=100                  ;INTERRUPT ENABLE
1705            000040                  SE=40                   ;SCAN ENABLE
1706            000020                  BUSY=20         ;BUST
1707            000017                  LINE=17                 ;LINE NUMBER
1708
1709                            ;LINE STATUS REGGRSTER EQUATES
1710            000002                  LSTAT=2
1711
1712            000004                  RQTS=4                  ;REQUEST TO SEND
1713            000002                  DTR=2                   ;DATA TERMINAL READY
1714            000001                  LE=1                    ;LINE ENABLE
1715
```

```
                011000                          .=11000
                                        ;*****************************************************************
                                        ;       THE INTERFACE SERVICE ROUTINE IS LOADED HERE
                                        ;*****************************************************************
                                                        ;THE FOLLOWING 18 WORDS ARE USED AS
                                                        ;THE LINKAGE BETWEEN THE ISR AND THE
                                                        ;CONTROL PROGRAM.
                                                        ;

011000                                  VISR:
011000  000002                         ISR:    .BLKW 2 ;.ASCIZ "DXX"
011004  000001                         BA:     .BLKW 1 ;175610                 ;BUS ADDRESS
011006  000001                         VA:     .BLKW 1 ;300                    ;VECTOR ADDRESS
011010  000001                         PRIOR:  .BLKW 1 ;340                    ;PRIORITY
011012  000001                         PARAM1: .BLKW 1 ;-1                     ;PARAM #1
011014  000001                         PARAM2: .BLKW 1 ;-1                     ;PARAM #2
011016  000001                         PARAM3: .BLKW 1 ;-1                     ;PARAM #3

011020  000001                         IRDA:   .BLKW 1 ;.WORD  0               ;INITIAL READ DATA ADDRESS
011022  000001                         IXDA:   .BLKW 1 ;.WORD  0               ;INITIAL XMIT DATA ADDRESS
011024  000001                         SETTLE: .BLKW 1 ;.WORD  0               ;LINE SETTLE DELAY FLAG
011026  000001                         IRCC:   .BLKW 1 ;.WORD  0               ;INITIAL RCV CHAR COUNT
011030  000001                         B2016:  .BLKW 1 ;.WORD  0               ;ADDR OF BIN TO OCT TYPE ROUTINE
011032  000001                         TIME:   .BLKW 1 ;.WORD  0               ;TIMER
011034  000001                         MODEA:  .BLKW 1 ;.WORD  0               ;ADDR OF ITEP PARAMS
011036  000001                                 .BLKW 1 ;.WORD  START           ;ISR ENTRY ADDRESS
011040                                  TX.TERM:
011040  000001                                 .BLKB 1 ;.BYTE  00C             ;TRANSMITER TERMINATING CHAR.
011041                                  RX.TERM:
011041  000001                                 .BLKB 1 ;.BYTE  012             ;RECEIVER TERMINATING CHAR.

011042  000001                         FLAG:   .BLKW   1
011044  177570                         SWR:    177570
011046  177570                         DISPLAY:177570
011050  000001                         START:  .BLKW 1 ;NOP
011052  000001                                 .BLKW 1 ;NOP
011054  000001                                 .BLKW 1 ;NOP
011056  000001                                 .BLKW 1 ;NOP
011060  000001                         CONT.:  .BLKW 1 ;NOP
011062  000001                                 .BLKW 1 ;NOP
011064  000001                                 .BLKW 1 ;NOP
011066  000001                         FINI:   .BLKW 1 ;MOV    #340,PS         ;LOCK OUT INTERUPTS.
011070  000001                                 .BLKW 1 ;MOV    #ENTER,2(SP)    ;SET FOR RETURN IF SW14=0
011072  000001                                 .BLKW 1 ;JSR    PC,SAVEOS       ;GO SAVE YOUR REGISTERS.
011074  000001                                 .BLKW 1 ;RTS    PC              ;EXIT

011076  000001                         ENTER:  .BLKW 1 ;JSR    PC,RESTOS       ;GO AND RESTORE REGISTERS
011100  000001                                 .BLKW 1 ;MOV    #-1,DELAY       ;INDICATE DELAY FOR TX.
011102  000001                                 .BLKW 1 ;JMP    CONT.   ;CONTINUE IN PROGRAM
000001                                  .END
```

```
ACCEPT= 104406            725    1142    1259    1434*   1505
ACR   = 040000           1554*   1582    1602
ANYEX   005024           1144*
ANYMOR  005014           1142*
ARIA    003704            922*    926*    935
ASTRK   001532            702     778*
BA      011004           1727*
BEGIN   003254            657     817*    822     900     968
BIT0  = 000001            620*    955    1497    1511
BIT00 = 000001            610*    620
BIT01 = 000002            609*    619
BIT02 = 000004            608*    618
BIT03 = 000010            607*    617
BIT04 = 000020            606*    616
BIT05 = 000040            605*    615
BIT06 = 000100            604*    614
BIT07 = 000200            603*    613
BIT08 = 000400            602*    612
BIT09 = 001000            601*    611
BIT1  = 000002            619*    976    1507    1511
BIT10 = 002000            600*
BIT11 = 004000            599*
BIT12 = 010000            598*
BIT13 = 020000            597*   1159
BIT14 = 040000            596*   1179
BIT15 = 100000            595*   1207
BIT2  = 000004            618*
BIT3  = 000010            617*    708     740
BIT4  = 000020            616*    712
BIT5  = 000040            615*    717     728
BIT6  = 000100            614*
BIT7  = 000200            613*
BIT8  = 000400            612*
BIT9  = 001000            611*
BPTVEC= 000014            627*
BUSY  = 000020           1647    1706*
BZ016   011030           1076*   1738*
CF    = 000100           1696*
CLRIB   004344           1039*   1041
COM   = 002000           1646    1700*
CONT.   011060           1754*
CR_F    001536            721     736     779*    986    1106    1162    1190    1205    1315
CR0   = 000001           1559*   1570
CSR   = 004000           1646    1699*
DAT     010150           1329    1540*
DCS   = 000040           1661    1697*
DCSOK   010556           1662    1672*
DCSW    010512           1661*   1664    1670
DAT     001564            793*   1018
DIAGNO  010356           1549    1569    1612*
DISPLA  011046           1528*   1749*
DISPRE  000174            652*   1528
DMA   = 010000           1555*
DMAC    010402            842     890     991    1621*
DMAOUT  010422           1088    1641*
DMAPTR  000544            886     889*
```

```
DMCHK   004574          1083    1086*
DMPHLT  001200           730*    722     735
DMPLST  001476           726     730*    760*    960
DN      010160           834     884     885    1543*
DNBA    010164          1544*   1564
DNEX    010354          1603    1610*
DNGO    010200          1084    1564*   1577    1592    1608
DNIV    010166          1545*
DNL1    010252          1579*   1580    1597
DNL1E   010300          1583    1589*
DNL1X   010312          1572    1527    1594*
DNL2    010326          1599*   1600
DNPAR1  010172          1547*
DNPAR2  010174          1548*
DNPAR3  010176          1549*
DNPRI   010170          1546*
DONE  = 000200          1579    1581    1599    1703*
DP    = 000002          1556*   1595
DP1     003004           794     807*    809
DP2     003076           795     807*    810
DP3     003126           796     807*    811
DP4   = 002604           797     808*
DP5   = 003004           798     809*
DP6   = 003076           799     810*
DP7   = 003126           800     811*
DSS   = 000040          1556*
DTR   = 000002          1650    1713*
EMP     001342           718     726*
MTAB    005270          1165    1215*
MTVEC=  000030           630*
M0      005320          1166    1215    1227*
M1      005330          1216    1227*
M10     005565          1223    1227*
M11     005640          1224    1227*
M12     005667          1225    1227*
M13     005711          1226    1227*
M2      005343          1217    1227*
M3      005363          1218    1227*
M4      005407          1219    1227*
M5      005422          1220    1227*
M6      005446          1221    1227*
M7      005462          1222    1227*
NTER    011076          1762*
OP      004614          1098*
RRVEC=  000004           623*
SNI     011066          1757*
FLAG    011042           700     701*    708*    712*    716*    717*    728     740     955*    976*    1497    1507    1511*
                        1747*
FLAGS   001562           791*    823*    837*    843     887*    893*    1082    1086    1651
GETANY  004776           904     907     910     915     920    1138*
GETBR   003602           882     888     894     901*
GETCHR= 104402           703    1254    1291    1430*
GETEX   003746           923     933     942*
GETIO   006502          1254*   1263    1267
GETWD   003464           829     875*    944
GETWC   003626           906*
```

```
GETPRM  003642            914*
GETP3   003666            913     918     922*
GETSTR= 104404            880     929    1027    1432*
GETVA   003616            906*
GOTIC   006040           1257    1264*
HLTX    005262           1208    1211*
IBUF    001604            803*    930     931     961    1029    1038    1040
IE    = 000100           1704*
IEB   = 000010           1689*
INTVEC= 000020            628*
IRCC    011026           1737*
IRDA    011020            962    1029*   1116    1734*
ISR     011000           1094    1726*
IV      010410           1623*
IXCA    011022            963    1018*   1033*   1110    1735*
KBDIN = 104416           1100    1211    1442*
LF    = 000001           1650    1714*
LP      001537            780*   1269    1317
LINE  = 000017           1707*
LSTAT = 000002           1661    1665    1710*
M..     001374            729     731     734*    751     759     964
M..     001426            743*    750
M..     001452            741     753*
MANBA   003404            840     845     848     852*
MANIN   003316            834*
MANINX  003460            862     866     869*
MBA     010406           1622*   1642
MFILL   010155            970    1540*
MM    = 001000           1701*
MODEA   011034           1740*
MSECS   006334           1046*   1335*   1337*   1341*
MSG00   007152            745     755    1540*
MSG01   007154            876    1540*
MSG02   007177            897    1540*
MSG03   007222            903    1540*
MSG04   007241            906    1540*
MSG05   007263            909    1540*
MSG06   007277            914    1540*
MSG07   007314            919    1540*
MSG08   007331            924    1540*
MSG09   007350            977     980    1540*
MSG10   007372           1026    1540*
MSG11   007411           1103    1540*
MSG12   007430           1167    1540*
MSG13   007441           1196    1540*
MSG14   007453           1202    1540*
MSG15   007462           1455    1540*
MSG16   007505           1455    1540*
MSG17   007551           1640*
MSG18   007564           1468    1540*
MSG19   007570           1068    1540*
MSG20   007616           1493    1540*
MSG21   007637           1015    1540*
MSG22   007663           1540*   1653
MSG23   007767           1178    1540*
MSG24   010003           1185    1540*
```

```
MSG25     010021         1460     1540#
MSG26     010060         1109     1540#
MSG27     010106         1115     1540#
NEG       010141         1503     1540#
NOISR     003570          892      896#
NOLC      004422         1049     1056#
NOG       004070          959      966#
NORTC     004524         1052     1063     1071#
NULL      001524          685      771#
OCT       006016         1259#    1262
OCT4      007040         1498     1512#
OWI     = 000002         1107     1686#
OWO     = 000001         1113     1655     1685#
PARAM1    011012         1730#
PARAM2    011014         1731#
PARAM3    011016         1732#
PARA1     010414         1625#    1649
PARA2     010416         1626#
PARA3     010420         1627#
PND     = 000020         1557#    1586
PRIO      010412         1624#
PRIOR     011010         1729#
PRTY0   = 000000          585#    1065     1075     1122     1494
PRTY1   = 000040          586#
PRTY2   = 000100          587#
PRTY3   = 000140          588#    1152
PRTY4   = 000200          589#
PRTY5   = 000240          590#
PRTY6   = 000300          591#
PRTY7   = 000340          592#     819     1514
PWI     = 100000         1553#    1571     1582
PWRVEC= 000024            629#
QUES      001526          720      777#     878      927     1140     1314
REST      004202          979     1010#
RESTRT    004210         1010     1011#
RESVEC= 000010            624#
REX       010556         1674#
REX1      010614         1656     1691#
RI      = 100000         1695#
RQTS    = 000004         1659     1712#
RSTART    004102          967      970#    1016
RWAIT   = 104410         1436#
RX.TER    011041         1744#
SE      = 000040         1705#
SETSWI= 104422            978     1446#
SETTLE    011024         1674#    1736#
SRC     = 010000         1698#
STACK   = 001070          535#     819     1074     1124     1492
START     011050         1750#
STARTX    010504         1659#
STEP    = 000400         1702#
STPS    = 104414          819     1065     1075     1122     1152     1440#    1494
SUSWR   = 104420          820     1444#
SUTIME    004354         1046#
SUXXX     004272         1020     1022     1029#
SWEG      010131         1500     1540#
```

```
SWPRNT  004116            658     943     974#
SWR     011044            828     837     838     854     857     860     864     868     953     974     984    1011    1013
                         1021    1031    1101    1119    1159    1207    1501    1504    1506#   1523    1527#  1655    1666
                         1748#
SWREG   000176            653#    953     974    1527
SWRNXT  004316           1032    1034#
SWRSET  004200            850     992#
SW0    = 000001            584#
SW00   = 000001            574#    584
SW01   = 000002            573#    583
SW02   = 000004            572#    582
SW03   = 000010            571#    581
SW04   = 000020            570#    580
SW05   = 000040            569#    579
SW06   = 000100            568#    578
SW07   = 000200            567#    577
SW08   = 000400            566#    576
SW09   = 001000            565#    575
SW1    = 000002            583#
SW10   = 002000            564#
SW11   = 004000            563#
SW12   = 010000            562#   1101    1666
SW13   = 020000            561#
SW14   = 040000            560#
SW15   = 100000            559#
SW2    = 000004            582#
SW3    = 000010            581#
SW4    = 000020            580#
SW5    = 000040            579#
SW6    = 000100            578#
SW7    = 000200            577#
SW8    = 000400            576#
SW9    = 001000            575#
TSITVE= 000014             625#
TEMP1   010616            1675#   1677#   1683#
TEMP2   010620            1676#   1679#   1684#
TIME    011032            1338#   1566#   1567    1643#   1663    1669#   1739#
TIMER   006312             648    1047    1056    1069    1335#
TIMEX   006332            1336    1339#
TKB     001516             768#    948     983    1239    1509
TKS     001514             767#    949#    956#    981    1037#   1237    1499#   1510#
TLB    = 000004           1687#
TPB     001522             693#    770#
TPS     001520             691     769#
TRAPVE= 000034             631#
TRTVEC= 000014             626#
TRX     005250            1160    1171    1191    1201    1207#
TR3     005226            1195    1200#
TTYINT  003760             948#   1035
TXTER   011040            1742#
TYPE   = 104400             702     705     720     721     736     745     755     876     877     878     897     903     906
                            909     914     919     924     925     927     970     977     980     986    1015    1020    1068
                           1103    1106    1109    1111    1115    1117    1140    1162    1166    1167    1179    1195    1193
                           1196    1202    1205    1266    1269    1294    1314    1315    1317    1329    1396    1398#   1452
                           1455    1460    1468    1493    1500    1503    1653
.A      011036            1728#
```

```
.VDB     002604            793     804*    808     1028    1033
.VIGC    004610           1087    1094*
.VIER    011000            847     875     877      942    1725*
.WAITPC  006624           1453    1457*
WORK     001544            723*    725     730      782     784*    880     881     885     891    1141*   1142    1143    1504*
                          1505    1506
WORK1    001546            785*
WORK2    001550            786*
WORK3    001552            787*
WORK4    001554            788*
WORK5    001556            789*
WORK6    001560            790*
XLB    = 000004           1655    1688*
XSWPNT   004144            975     980*
XWAIT  = 104412           1438*   1668
$ACCEP   006114           1283*   1433
$B2OCT   006336            746     1180    1187    1363*
$B2O1    006364           1365    1368*
$B2O16   006350            738     756     985     1076    1105    1139    1169    1198    1204    1356*   1458    1456    1502
$CHAR    001542            706     710     714      783*   1239*   1240*   1241    1255    1261    1265    1266
$COPGN   004770           1126    1131*
$ERTTL   005730           1153*   1228*
$HLT     005032            641     1151*
$HLTAD   005726           1155*   1156*   1163     1168    1227*
$HLOT    005036           1153*   1154
$LPADR   003462            822*    870*   1010*
$OCNT    006540           1368*   1397*   1409*
$OMODE   006542           1363*   1367*   1372     1375*   1386*   1411*
$PASS    004774            821*   1098*   1104     1132*
$PWRDN   006672            639    1480*   1491
$PWRUP   006704           1480    1491*
$READC   005732           1235*   1429
$READS   005774           1250*   1431
$RWAIT   006610           1435    1452*
$TRAP    006544            643    1416*
$TRPAD   006564           1420    1427*
$TYPE    001100            674*   1427
$WORK  = 001544            782*   1259    1260
$XWAIT   006620           1437    1455*
$OFILL   006541           1364*   1366*   1376    1410*
.      = 011104            635*    638*    647*    651*    655*    660*    699*    762*    763*    781*    803*    805*    807*
                           839     844     938     973     982     987    1014    1071    1165*   1227*   1238    1319    1482
                          1568    1648    1678    1680    1716*   1726*   1727*   1728*   1729*   1730*   1731*   1732*   1734*
                          1735*   1736*   1737*   1738*   1739*   1740*   1741*   1743*   1745*   1747*   1750*   1751*   1752*
                          1753*   1754*   1755*   1756*   1757*   1758*   1759*   1760*   1762*   1763*   1764*
.VBCIN   006732           1441    1497*
.RX      001532            762*    962*
.SETSW   006750           1445    1500*
.STBS    007042           1439    1514*
.SWBR    007064           1443    1519*
.TY      001504            763*    963*
```

K04


```
BOX       634#    696     772     814     825     831     872     988    1023    1079    1095    1135    1148    1229    1243
         1332    1448    1540    1561    1616    1638    1717
HELLO       1#
HLT       541#    1576    1591    1607

. ABS.  011104     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DSKZ:DZITAD,DZITAD/SOL/CRF/EN:AMA=DZITAD.P11
 RUN-TIME:  9 14 1 SECONDS
 RUN-TIME RATIO: 107/25=4.3
 CORE USED:  11K  (21 PAGES)
```

Spooler runtime 1 Seconds, 16 KCS, 166 disk reads, 4 disk writes, 49 pages