DMC11

ITEP OVERLAY

MD-11-DZDMO-A

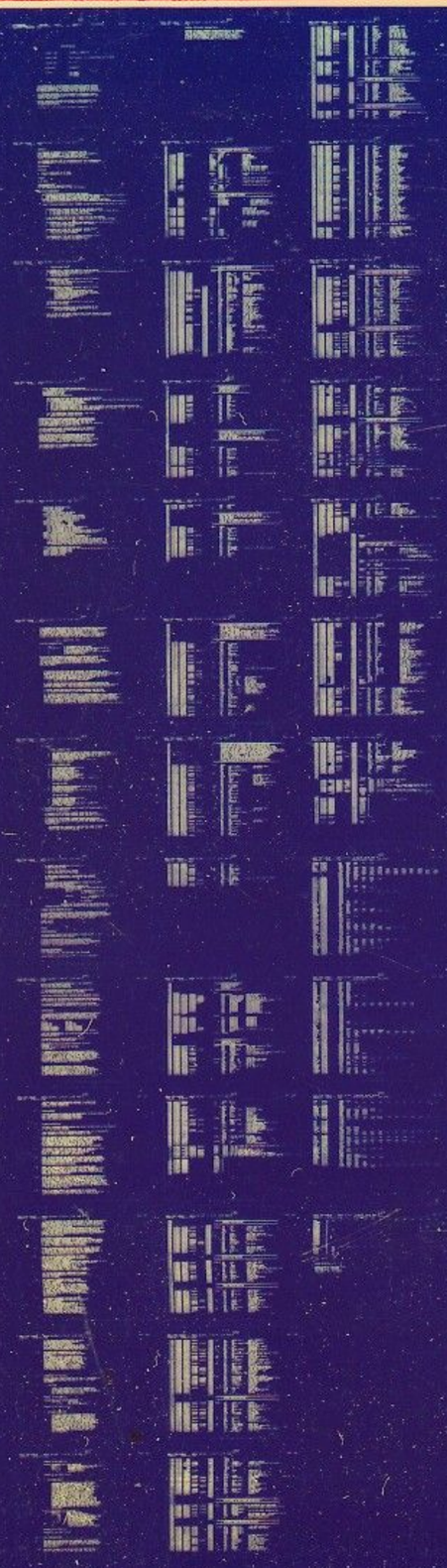
EP-DZDMO-A-DL-A

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA



IDENTIFICATION

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

PRODUCT NAME: DMC11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: MAY 1977

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
JOHN EGOLF
FAY BASHAW

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.

THIS PROGRAM MUST BE USED IN CONJUNCTION WITH THE INTERPROCESSOR
TEST PROGRAM(DZITP) ON A PDP-11 SYSTEM WITH A DL-11 INTERFACE.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

A. PDP-11 SYSTEM WITH 4K OF CORE.
B. A DMC11 COMMUNICATION INTERFACE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE
THIS PROGRAM IS IN ABSOLUTE FORMAT.
THE ABS LOADER MUST BE USED TO LOAD THE PROGRAM.

4.0 OPERATING PROCEDURES.

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)

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. TYPE IN ACTUAL VALUE

5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE ISR.(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 ISR.
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=SETUP DM-11B ISR
SWR13=SETUP DN-11 ISR
SWR=000000=SETUP VARIABLE ISR

2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11,DM11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWICHES THEN HIT CONTINUE
A. HALT FOR BUS ADDRESS OF INTERFACE
B. HALT FOR VECTOR ADDRESS OF INTERFACE
C. HALT FOR PRIORITY OF INTERFACE
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. 10.0 OF THE MON
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

BEFORE ATTEMPTING TO RUN THIS PROGRAM, THE OPERATOR MUST
ACCERTAIN THE COMPLETE COMMUNICATION LOOP AND PROCEEDURES
TO BE USED, INCLUDING THE TYPE OF MODEMS, THE TYPE OF
INTERFACE BEING USED AT THE OTHER CPU AND THE MODES OF OPERATION,
DATA AND PARAMETERS TO BE USED AT EACH CPU.

THIS WILL REQUIRED VOCAL COMMUNICATION WITH THE OPERATOR
AT THE OTHER CPU UNLESS ITS CONFIGURATION AND OPERATION
ARE FIXED AS A TEST CENTER.

AFTER DETERMINING THAT THE EQUIPMENTS ARE COMPATIBLE AND
AGREEING ON THE MODE AND VARIABLE PARAMETERS TO BE USED,
THE SYSTEM WHICH IS TO RECEIVE DATA FIRST SHOULD BE
LOADED AND STARTED. IF THE MODEM BEING USED ON THIS SYSTEM
HAS AN AUTOMATIC ANSWER FEATURE, IT SHOULD BE ENABLED.

THE SYSTEM WHICH IS TO TRANSMIT FIRST SHOULD THEN BE LOADED
AND STARTED AND THE CONNECTION ESTABLISHED EITHER MANUALLY
OR AUTOMATICALLY (VIA DN-11).

```
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
```

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

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.

THE PROGRAM WILL PRINTOUT A "WAITING FOR CLEAR TO SEND" MESSAGE AND THE CONTENTS OF THE XMIT CSR EVERY 60 SECS. UNTIL CLEAR TO SEND IS ASSERTED.

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.

IF DATA IS RECEIVED AND SWITCH 7 (NO DATA COMPARE)
IS RESET, THE DATA WILL BE COMPARED AGAINST THE PRESELECTED
DATA AFTER A LINE FEED CHARACTER IS RECEIVED. IF THERE IS A
MISMATCH, THE FOLLOWING ERROR REPORT IS PRINTED:

```
RECEIVED DATA=RRRRRR
DATA SHOULD BE TTTTTT
DATA COMPARE ERROR;  BAD DATA=BBB  GOOD DATA=GGG
```

WHERE RRRRRR IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)

TTTTTT IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
BBB IS THE BAD DATA CHARACTER
GGG IS THE GOOD DATA CHARACTER

IF THE INTERFACE DETECTS A DATA ERROR, THE FOLLOWING
WILL BE PRINTED BEFORE THE DATA IS COMPARED:

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

WHERE XXXXXX IS THE CONTENTS OF THE RECEIVER DATA REGISTER
THE LOW BYTE IS THE DATA, AND THE HIGH BYTE IS THE ERROR BITS.

IF A RECEIVE TERMINATING CHARACTER<001> IS NOT DETECTED
WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

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:

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
RESTRICTION 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

IF PROGRAM IS WAITING IN A SCAN ROUTINE AND TYPES OUT
A "WAITING MESSAGE", IF AN INCOMING MESSAGE STARTS DURING
THE TYPE OUT, IT WILL BE LOST BECAUSE THE TYPEOUT PRIORITY
IS AT LEVEL 7. THIS WILL RESULT IN OVERRUN OR SILO OVER-
RUN ERRORS, DEPENDING ON THE DEVICE.TO AVOID THIS SITUATION
RUN WITH SWITCH 13 UP. IF OVERRUN DOES OCCURE DURING A
TYPEOUT THE PROGRAM SHOULD BE RESTARTED.

IF USING AN ASYNCRONOUS DEVICE, MODEMS AND THE
MAYNARD TEST STATION AND INITALIZE DOES NOT CLEAR THE
CONNECTION (EXAMPLE THE DJ11) IF THE PROGRAM IS RESTARTED
IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING CR
AN IMMEDEATE ERROR MESSAGE FROM MAYNARD WILL BE RE-
CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING

FOR THE REST OF THE INTERUPTED MESSAGE. TO AVOID THIS
ERROR , RESTART PROGRAM ONLY AT THE END OF THE MESSAGE
CURRENTLY BEING TRANSMITTED.

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

9.1 THE DMC11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR,
BY THE ITEP CONTROL PROGRAM.

TIME: PROVIDES A MEANS OF MEASURING ELASPED TIME. IT IS INCREMENTED
EVERY SECOND BY A CLOCK INTERRUPT ROUTINE IN ITEP.

9.2 WHEN THE OVERLAY IS FIRST ENTERED BY ITEP AT LOCATION START:,
THE CONTENTS OF THE SWITCH REGISTER ARE STORED IN REGISTER 0.
THE MODE AND DATA SELECTIONS ARE FIXED AT THIS TIME AND CANNOT
BE ALTERED WITHOUT RETURNING TO THE CONTROL PROGRAM.
THE INTERRUPT VECTORS AND VARIABLES ARE THEN SETUP.
THE SELECTED ROUTINE DETERMINED BY THE MODE IS THEN ENTERED

9.3 THE OVERLAY THEN LOOPS IN ROUTINES: $OWI ,IF "ONE WAY IN" MODE
WAS SELECTED. $OWO,IF "ONE WAY OUT" MODE WAS SELECTED.
$ILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED.
$XLB,IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 $OWI: IN THIS ROUTINE THE RECEIVER IS INITIALIZED AND PROGRAM
LOOPS WAITING FOR THE RECEIVER TO FINISH.IF NOTHING IS RECEIVED
FOR 60 SECS A "WAITING" MESSAGE IS TYPED. WHEN THE RECEIVER IS
DONE,THE PROGRAM CHECKS DATA IF SWITCHES PERMIT,AND TYPES END
PASS DEPENDING ON SWITCH SETTINGS.

9.32 $OWO: THE TRANSMITTER IS INITIALIZED AND PROGRAM LOOPS
WAITING FOR TRANSMITTER TO FINISH,A "WAITING" MESSAGE IS TYPED
EVERY 60 SECS IF THERE IS NO ACTION. WHEN THE TRANSMITTER IS
DONE, THE PROGRAM EITHER LOOPS BACK TO $OWO OR TYPES END PASS
DEPENDING ON SWITCH SETTINGS.

9.33 $ILB: THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR
RECEIVER TO FINISH, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC
IF NO ACTION. WHEN RECEIVER IS DONE PROGRAM CHECKS DATA IF SWITCH
SETTINGS PERMIT, AND END PASS IS TYPED IF SWITCH SETTINGS PERMIT.
THEN THE TRANSMITTER IS INITIALIZED, A "WAITING" MESSAGE IS TYPED
EVERY 60 SEC IF NO ACTION. WHEN TRANSMITTER IS DONE PROGRAM RETURNS
TO START OF ROUTINE. ($ILB)

9.34 $XLB: IF IN HALF DUPLEX THE TRANSMITTER IS INITIALIZED,
A "WAITING MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION
WHEN THE TRANSMITTER IS DONE THE RECEIVER IS INITIALIZED

A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION.
WHEN THE RECEIVER IS DONE,DATA IS CHECKED IF SWITCH SETTINGS
PERMIT AND END PASS IS TYPED IF SWITCHES ALLOW.THE PROGRAM NOW
REPEATS CYCLE STARTING AT $XLB.
IF IN FULL DUPLEX THE RECEIVER AND TRANSMITTER ARE INITIALIZED
A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO
ACTION. WHEN BOTH THE RECEIVER AND TRANSMITTER ARE DONE,DATA IS
CHECKED, END PASS IS TYPED AND PROGRAM LOOPS TO $XLB DEPENDING
ON THE SWITCH SETTINGS.

9.4 THE RETURN TO MONITOR ROUTINE FOR END PASS AT EOP:
LOCKS OUT INTERUPTS AND SAVES THE TRANSMITTER INTERUPT ENABLE
BIT AND P/L GENERAL REGISTERS. IT THEN RETURNS TO THE MONITOR
TO TYPE "END PASS". THE MONITOR CHECKS SW14 IF UP IT RETURNS
TO ENTER:, OTHERWISE IT RESTARTS THE PROGRAM.

9.5 ENTER: IS ENTERED FROM THE MONITOR AFTER TYPEING "END PASS",
IT RESTORES THE GENERAL REGISTERS AND THE TRANSMITTER CSR
AS SAVED IN EOP. THE DELAY FLAG IS SET AND PROGRAM RETURNS TO
THE SCAN ROUTINE(OWO,OWI,ILB,XLB) WHERE IT CAME FROM.

9.6 THE INITIALIZE TRANSMIT SUBROUTINE AT STARTX:
SETS UP THE INTERFACE AND POINTERS NECESSARY TO
INITIATE A TRANSMIT OPERATION.
AFTER SETTING "DATA TERMINAL READY" AND "REQUEST TO SEND" A CHECK
IS MADE ON PARAM2 TO DETERMINE IF HALF DUPLEX OPERATION
WAS SELECTED BY THE OPERATOR. IF IT WAS, THE
SUBROUTINE WAITS FOR CLEAR TO SEND.
A 'WAITING FOR CLEAR TO SEND' PRINTOUT OCCURS
EVERY 30 SECONDS UNTIL CLEAR TO SEND IS ASSERTED.

9.7 THE INITIALIZE RECEIVED SUBROUTINE AT STARTR:
SETS UP THE INTERFACE AND POINTERS NECESSARY TO
RECEIVE A MESSAGE.

9.8 THE TRANSMIT INTERRUPT SERVICE ROUTINE,
AT XISR:, IS ENTERED VIA TRANSMIT INTERRUPTS
FROM THE INTERFACE.
A TEST IS MADE TO SEE IF THE LAST CHARACTER
TRANSMITTED WAS A NULL (ALL ZEROS) CHARACTER.
IF IT WAS; THE TRANSMIT LOGIC IN THE INTERFACE
IS RESET AND THE TRANSMIT COMPLETE FLAG IS SET.
AT XISR1: THE NEXT CHARACTER IS TRANSMITTED
AND PRINTED ON THE TTY IF THE MONITOR TRANSMIT
SWITCH IS SET.

9.9 THE RECEIVE INTERRUPT SERVICE ROUTINE
AT RISR:, IS ENTERED VIA RECEIVER INTERRUPTS
FROM THE INTERFACE.
THE RECEIVED CHARACTER IS STORED IN
THE INPUT BUFFER AND PRINTED ON THE TTY IF
THE MONITOR RECEIVER SWITCH IS SET.
IF THE INPUT BUFFER IS FULL, A 'BUFFER FULL'
PRINTOUT WILL OCCUR. THIS INDICATES THAT A
LINE FEED CHARACTER WAS NOT RECOGNIZED
IN THE RECEIVED DATA (WITHIN 1000 CHARACTERS).

IF THE RECEIVED CHARACTER IS A LINE FEED,
THE RECEIVED LOGIC IS RESET AND THE
RECEIVE COMPLETE FLAG IS SET.
IF A 'RECEIVE ERROR' IS DETECTED AT RISR:, THE
CSR AND DBR WILL BE SAVED AND PRINTED OUT
AFTER THE COMPLETE MESSAGE HAS BEEN RECEIVED.

9.10 THE DATA TEST SUBROUTINE AT TESTD: IS
ENTERED AFTER A COMPLETE MESSAGE HAS BEEN
RECEIVED.
IF A 'RECEIVE ERROR' HAD BEEN DETECTED,
THE CONTENTS OF THE 'RECEIVE BUFFER' AT THE
TIME THE ERROR OCCURRED WILL BE PRINTED.
THE DATA IS COMPARED UNTIL A 'ALL ZEROS'
CHARACTER IS RECOGNIZED. 'FILL' (ALL ONES)
CHARACTERS ARE IGNORED. IF A MISMATCH
IS DETECTED, THE COMPLETE CONTENTS OF THE
INPUT BUFFER AND GOOD DATA IS PRINTED.

10.0 PARAMETERS FOR THE DMC11

DZDMOA PROVIDES THREE TESTS FOR THE DMC-11, SELECTABLE BY
THE PARAMETER LOCATIONS PROVIDED IN ITEP. THE THREE
TESTS ARE:

10.1 1.) LINK TEST

NORMAL ITEP OPERATION, THE ONLY RESTRICTION
IS IT MUST BE DMC TO DMC. IT IS NORMAL TO GET
SOFT ERRORS DURING THE LINK TEST. THE PARAMETERS
FOR THE LINK TEST ARE AS FOLLOWS:

PARAM#1 IS NOT USED (0)

PARAM#2 FULL/HALF DUPLEX SELECTION
BIT0 = 0 HALF DUPLEX
= 1 FULL DUPLEX (DEFAULT)

PARAM#3 IS NOT USED (177777)

10.2 2.) SECONDARY MODE TEST

THIS TEST CHECKS THE SECONDARY STATION DELAY.
IF RUNNING THIS TEST, EXTERNAL LOOP BACK IS
THE ONLY LEGAL MODE OF OPERATION. ALSO BOTH
DMC-11'S MUST HAVE HALF-DUPLEX SELECTED AND THE
SECONDARY MODE BIT SET IN THE PARAMETER WORD.
ADDITIONALLY ONE DMC IS SET TO BE THE SECONDARY
STATION AND THE OTHER THE PRIMARY STATION.
AGAIN IT IS NORMAL TO GET SOFT ERRORS DURING
THE SECONDARY MODE TEST AS IN THE LINK TEST.
THE PARAMETERS FOR THE SECONDARY MODE TEST
ARE AS FOLLOWS:

PARAM#1 IS NOT USED (0)

PARAM#2 SECONDARY MODE TEST SELECTION
BIT0 = 0 HALF DUPLEX(MUST BE 0 FOR THIS TEST
BIT1 = 1 SECONDARY MODE TEST(MUST BE 1)
BIT2 = 0 PRIMARY STATION
= 1 SECONDARY STATION

PARAM#3 IS NOT USED (177777)

10.3 3.) BOOTSTRAP TEST

THIS WILL TEST THE ABILITY OF A DMC TO BOOT ANOTHER DMC USING MOP MESSAGES. THIS TEST REQUIRES A M9301-YJ AT ONE STATION. THE STATION WITH THE M9301-YJ IS THE BOOT STATION, THE OTHER IS THE ORIGINATING STATION. THERE ARE TWO VARIATIONS OF THIS TEST: 1) AUTOMATIC MODE IN WHICH THE ORIGINATING STATION SENDS THE BOOT MESSAGE TO THE BOOT STATION DMC WHICH CAUSES THE DMC TO BOOT THE M9301-YJ. 2) MANUAL MODE, IN THIS MODE THE BOOT STATION M9301 IS MANUALLY BOOTED. IN ADDITTION TO THE PARAMETERS THE SWITCH REGISTER MUST BE SET AS FOLLOWS: SW09=1 SW02=1. MAKE ALL NECESSARY MODEM CONNECTIONS AND CALLS BEFORE STARTING THE BOOTSTRAP TEST. THE PARAMETERS FOR THE BOOTSTRAP TEST ARE AS FOLLOWS:

PARAM#1 BOOTSTRAP TEST SELECTION
BIT8 = 1 BOOT STRAP TEST
BITS 10&9 MODE & STATION SELECT
= 00 ORIGINATING STATION-AUTOMATIC MODE
= 01 BOOT STATION-AUTOMACTIC MODE
= 10 ORIGINATING STATION-MANUAL MODE
= 11 BOOT STATION-MANUAL MODE
BITS 0-7 SWITCH PAC SETTING OF BOOT STATION DMC-11 DDCMP LINE#
(USED AS PASSWORD IN MOP MESSAGE)

PARAM#2 IS NOT USED (0)

PARAM#3 IS NOT USED (177777)

11.0 DMC11 RESTRICTIONS

11.1 THE DMC11 IS A DMA DEVICE AND THEREFORE THE TRANSMITTED AND RECEIVED DATA CAN NOT BE MONITORED ON A PER CHARACTER BASIS BY THE CONSOLE TTY. BECAUSE OF THIS, SW05 AND SW06 HAVE NO EFFECT.

11.2 DMC ITEP IS MEANT TO BE AN ON LINE LINK TEST FOR TWO DMC11S. DMC ITEP WILL NOT WORK WITH ANY OTHER DEVICE

RUNNING ITEP EXCEPT ANOTHER DMC11.

11.3 BECAUSE THE DMC11 SUPPORTS DDCMP OPERATION IN THE FIRMWARE, THE PDP-11 PROGRAM (ITEP) IS UNABLE TO CONTROL OR KNOW EXACTLY WHAT IS BEING TRANSMITTED AT ANY GIVEN TIME. ALL DATA MESSAGES ARE ENCLOSED IN A DDCMP ENVELOPE AND THERE MAY ALSO BE CONTROL MESSAGES (AKS NAKS ETC) BEING TRANSMITTED. BECAUSE OF THIS PLEASE BEWARE IF YOU ARE SCOPING DATA.

```
                                ;*****************************************************************
                                ;       DMC11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                                        .=11000
011000  046504  000103  DMC11:  .ASCIZ  /DMC/           ;ISR NAME
011004  160010          BA:     160010                  ;BUS ADDRESS
011006  000300          RIV:    300                     ;VECTOR ADDRESS
011010  000240          PRIOR:  240                     ;PRIORITY
011012  000000          PARAM1: 0                       ;PARAM #1
011014  000001          PARAM2: 1                       ;PARAM #2
011016  177777          PARAM3: 177777                  ;PARAM #3
011020  000000          IRDA:   .WORD   0               ;INITIAL READ DATA ADDRESS
011022  000000          IXDA:   .WORD   0               ;INITIAL XMIT DATA ADDRESS
011024  000000          SETTLE: .WORD   0               ;LINE SETTLE DELAY FLAG
                                .WORD   0               ;
011026  000000          B2016:  .WORD   0               ;ADDR OF BIN TO OCT TYPE ROUTINE
011030  000000          TIME:   .WORD   0               ;TIMER
011032  000000                  .WORD   0               ;
011034  000000                  .WORD   START           ;ADDR OF START OF PROGRAM
011036  011106  TX.TERM:
                                .BYTE   000             ;TRANSMITTER TERMINATING CHAR.
011040      000 RX.TERM:
                                .BYTE   001             ;RECEIVER TERMINATING CHAR.
011041      001 FLAG:   .WORD   0
011042  000000          SWR:    177570
011044  177570          DISPLAY:177570
011046  177570
                                ;*****************************************************************
                                ;       CONSTANTS + WORKING STORAGE
                                ;*****************************************************************
                                        STAT=R0
000000                          XFLG=100000             ;XMIT COMPLETE FLAG
100000                          RFLG=40000              ;RCV COMPLETE FLAG
040000                          DSFLG=20000             ;DATA SET STATUS CHANGE FLAG
020000                          BIT13=20000             ;INHIBIT PRINTOUTS
020000
011050  000000          SXCSR:  0                       ;SAVED XMIT CSR
011052  000000          SRCSR:  0                       ;SAVED RCV CSR
011054  000000          ERCSR:  0                       ;RCV CSR SAVED ON ERROR
011056  000000          ERDBR:  0                       ;RCV DATA REG SAVED ON ERROR
011060  000000          DSSTAT: 0                       ;RCV CSR SAVED ON DS CHANGE

011062  000000  01400    TXWC:  0
011064  000000  01500    RXWC:  0
011066  000000          XCC:    0                       ;XMIT CHAR COUNT
011070  000000          RCC:    0                       ;RCV CHAR COUNT
011072  000000          RDA:    0                       ;RCV DATA ADDR.
011074  000000          XDA:    0                       ;XMIT DATA ADDR.

011076  177560          TKS:    177560
011100  177562          TKB:    177562
011102  177564          TPS:    177564
011104  177566          TPB:    177566

000001                  FULL.DUPLEX=000001
```

```
                                                ;*****************************************************************
                                                ;       DMC11-X INTERFACE SERVICE ROUTINE
                                                ;*****************************************************************
011106  000240                          START:  NOP
011110  017700  177730                          MOV     @SWR,   R0      ;SETUP MODE IN R0
011114  042700  177400                          BIC     #177400,R0      ;STRIP JUNK
011120  013702  011006                          MOV     RIV,    R2      ;SETUP
011124  012722  014032                          MOV     #RISR,  (R2)+   ;INTERRUPT
011130  013722  011010                          MOV     PRIOR,  (R2)+   ;VECTORS
011134  012722  013544                          MOV     #XISR,  (R2)+   ;
011140  013722  011010                          MOV     PRIOR,  (R2)+   ;
011144  013704  011004                          MOV     BA,     R4      ;SETUP BUS ADDR INDEX
011150  013702  011006          01700           MOV     RIV,R2          ;ADJUST VECTORS FOR
011154  012712  013544          01800           MOV     #XISR,(R2)      ;INPUT SERVICE ROUTINE
011160  012762  014032  000004  01900           MOV     #RISR,4(R2)     ;OUTPUT SERVICE ROUTINE
011166  032737  000400  011012  02000           BIT     #BIT8,PARAM1    ;BOOT MODE?
011174  001405                  02100           BEQ     3$              ;BR IF NO
011176  012712  015126          02200           MOV     #IISR,(R2)      ;LOAD BOOT VECTORS
011202  012762  015312  000004  02300           MOV     #OISR,4(R2)     ;
011210  012714  040000          02400   3$:     MOV     #BIT14,(R4)     ;MASTER CLEAR DMC11
011214  013702  011022          02500           MOV     IXDA,R2         ;CALCULATE WORD COUNTS
011220  005003                  02600           CLR     R3              ;CLEAR COUNT
011222  005203                  02700   1$:     INC     R3              ;INC COUNT
011224  123722  011040          02800           CMPB    TX.TERM,(R2)+   ;LAST CHARACTER?
011230  001374                  02900           BNE     1$              ;BR IF NO
011232  010337  011062          03000           MOV     R3,TXWC         ;STORE XMIT COUNT
011236  062703  000010          03100           ADD     #10,R3          ;ADD 10 TO IT
011242  010337  011064          03200           MOV     R3,RXWC         ;STORE REC COUNT
011246  005714                  03300   2$:     TST     (R4)            ;WAIT FOR RUN
011250  100376                  03400           BPL     2$              ;BR IF RUN NOT SET
011252  005037  016744          03500           CLR     BASEFG          ;CLEAR BASE LOAD FLAG
011256  005037  016754          03600           CLR     RFLAG           ;CLEAR RECEIVE FLAG
011262  005037  016752          03700           CLR     XFLAG           ;CLEAR XMIT FLAG
011266  005714                  03800   4$:     TST     (R4)            ;RUN SET?
011270  100376                  03900           BPL     4$              ;BR IF NO
011272  052764  000100  000002  04000           BIS     #100,2(R4)      ;SET OIE
011300  052714  000143          04100           BIS     #143,(R4)       ;ASK FOR BASE TRANSFER
011304  005037  011054          04200           CLR     ERCSR           ;CLEAR ERROR LOCATIONS
011310  005037  011056          04300           CLR     ERDBR           ;
011314  005037  016766          04400           CLR     ERRCNT          ;CLEAR SOFT ERROR STORAGE
011320  005037  016770          04500           CLR     ERRCNT+2        ;LOCATIONS
011324  005037  016772          04600           CLR     ERRCNT+4        ;
011330  005037  016774          04700           CLR     ERRCNT+6        ;
011334  105037  017001          04800           CLRB    BASE+3          ;CLEAR BASE ERROR COUNT LOCATIONS
011340  005037  017002          04900           CLR     BASE+4          ;
011344  005037  017004          05000           CLR     BASE+6          ;
011350  005037  017006          05100           CLR     BASE+10         ;
011354  105037  017010          05200           CLRB    BASE+12         ;
011360  005037  016750          05300           CLR     TEMP2           ;
011364  005037  016742          05400           CLR     RESUME          ;CLEAR RESUME FLAG
011370  032737  000400  011012  05500           BIT     #BIT8,PARAM1    ;BOOT MODE?
011376  001402                  05600           BEQ     5$              ;BR IF NO
011400  000137  014254          05700           JMP     BOOT            ;GO TO START OF BOOT CODE
011404                          05800   5$:     ;
                                05900           ;
```

```
                                        ;****************************************
                                        ;       ROUTINE USED TO GOTO
                                        ;       SUBROUTINE DEPENDENT
                                        ;       ON MODE SELECTED.
                                        ;****************************************

011404  005037  011032          GO:     CLR     TIME
011410  005037  013376                  CLR     DELAY
011414  005037  013402                  CLR     STOP
011420  032700  000001                  BIT     #OWO,MODE
011424  001402                          BEQ     1$
011426  000137  011606                  JMP     $OWO
011432  032700  000002          1$:     BIT     #OWI,MODE
011436  001402                          BEQ     2$
011440  000137  011474                  JMP     $OWI
011444  032700  000010          2$:     BIT     #ILB,MODE
011450  001402                          BEQ     3$
011452  000137  011710                  JMP     $ILB
011456  032700  000004          3$:     BIT     #XLB,MODE
011462  001402                          BEQ     4$
011464  000137  012144                  JMP     $XLB
011470  000000                  4$:     HALT
011472  000776                          BR      .-2


                                        ;****************************************************************************
                                        ;       ROUTINE USED IF "ONE WAY IN" MODE WAS SELECTED.
                                        ;       NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE
                                        ;       ONLY MODE AVAILABLE.
                                        ;       "ONE WAY IN" MEANS THAT ONLY THE RECEIVER IS
                                        ;       ENABLED. THE TRANSMITTER IS NEVER "TURNED ON".
                                        ;****************************************************************************


011474  104416          $OWI:   KBDIN
011476  004737  013474          JSR     PC,STARTR
011502  032700  040000          1$:     BIT     #RFLG,STAT
011506  001013                  BNE     2$
011510  023727  011032  000100  CMP     TIME,#100
011516  103771                  BLO     1$
011520  011402                  MOV     @RCSR,R2
011522  016403  000002          MOV     XCSR(R4),R3
011526  104001                  HLT     1
011530  005037  011032          CLR     TIME
011534  000762                  BR      1$

011536  032777  000200  177300  2$:     BIT     #NODAT,@SWR
011544  001002                  BNE     3$
011546  004737  012622          JSR     PC,TESTD
011552  042700  040000          3$:     BIC     #RFLG,STAT
011556  004737  016176          JSR     PC,CKBASE       ;CKECK DMC SOFT ERROR COUNTERS
011562  032777  000020  177254  BIT     #LOOP,@SWR
011570  001405                  BEQ     4$
```

```
                                                MOV     #4$,BACK
                                                JMP     EOP
                                        4$:     BR      $OWI

                                        ;*****************************************************************
                                        ;       ROUTINE USED IF "ONE WAY OUT" WAS SELECTED.
                                        ;       NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE ONLY
                                        ;       MODE AVAILABLE.
                                        ;       "ONE WAY OUT" MEANS THAT ONLY THE TRANSMITTER IS
                                        ;       ENABLED. THE RECEIVER IS NEVER "TURNED ON.
                                        ;*****************************************************************

011606  104416                          $OWO:   KBDIN
011610  004737  013404                          JSR     PC,STARTX
011614  005037  011032                          CLR     TIME
011620  032700  100000                  1$:     BIT     #XFLG,STAT
011624  001013                                  BNE     2$
011626  023727  011032  000100                  CMP     TIME,#100
011634  103771                                  BLO     1$
011636  011402                                  MOV     @RCSR,R2
011640  016403  000002                          MOV     XCSR(R4),R3
011644  104001                                  HLT     1
011646  005037  011032                          CLR     TIME
011652  000762                                  BR      1$
011654  042700  100000                  2$:     BIC     #XFLG,STAT
011660  004737  016176                          JSR     PC,CKBASE       ;CKECK DMC SOFT ERROR COUNTERS
011664  032777  000020  177152                  BIT     #LOOP,@SWR
011672  001405                                  BEQ     3$
011674  012737  011706  013400                  MOV     #3$,BACK
011702  000137  012462                          JMP     EOP
011706  000737                          3$:     BR      $OWO
```

```
                                        ;*************************************************************************
                                        ;       ROUTINE USED IF INTERNAL LOOP BACK" WAS SELECTED.
                                        ;       NOTE THAT WHEN IN THIS MODE; HALF DUPLEX IS THE
                                        ;       ONLY MODE AVAILABLE.
                                        ;       "INTERNAL LOOP BACK" MEANS THAT THE RECEIVER IS "TURNED ON"
                                        ;       AND A COMLPETE MESSAGE IS RECEIVED. IF DATA IS TO BE CHECKED
                                        ;       IT IS; IF "END PASS" IS DESIRED; IT IS GIVEN.
                                        ;       THEN THE TRANSMITTER IS ENABLED. AFTER THE WHOLE MESSAGE
                                        ;       IS TRANSMITTED; THE CYCLE IS REPETED AS ABOVE.
                                        ;*************************************************************************

011710  104416                  $ILB:   KBDIN
011712  004737  013474                  JSR     PC,STARTR
011716  005037  011032                  CLR     TIME
011722  032700  040000          1$:     BIT     #RFLG,STAT
0'1726  001013                          BNE     2$
011730  023727  011032  000100          CMP     TIME,#100
011736  103771                          BLO     1$
011740  011402                          MOV     @RCSR,R2
011742  016403  000002                  MOV     XCSR(R4),R3
011746  104001                          HLT     1
011750  005037  011032                  CLR     TIME
011754  000762                          BR      1$
011756  032777  000200  177060  2$:     BIT     #NODAT,@SWR
011764  001002                          BNE     3$
011766  004737  012622                  JSR     PC,TESTD
011772  042700  040000          3$:     BIC     #RFLG,STAT
011776  004737  016176                  JSR     PC,CKBASE       ;CKECK DMC SOFT ERROR COUNTERS
012002  032777  000020  177034          BIT     #LOOP,@SWR
012010  001405                          BEQ     4$
012012  012737  012024  013400          MOV     #4$,BACK
012020  000137  012462                  JMP     EOP
012024  032777  000400  177012  4$:     BIT     #400,   @SWR    ;USE EXTERNAL DATA?
012032  001416                          BEQ     7$              ;BR IF NO
012034  013702  011020                  MOV     IRDA,   R2      ;SET POINTER
012040  013703  011022                  MOV     IXDA,   R3      ;SET POINTER
012044  010337  011074                  MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
012050  112223                          MOVB    (R2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
012052  001376                          BNE     .-2             ;LOOP IF NOT ZERO CHAR
012054  112743  000177                  MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
012060  005203                          INC     R3              ;BUMP ADDRESS
012062  112723  000177                  MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
012066  105023                          CLRB    (R3)+           ;INSERT ZERO CHAR
012070  005037  011032          7$:     CLR     TIME
012074  004737  013404                  JSR     PC,STARTX
012100  032700  100000          5$:     BIT     #XFLG,STAT
012104  001013                          BNE     6$
012106  023727  011032  000100          CMP     TIME,#100
012114  103771                          BLO     5$
012116  011402                          MOV     @RCSR,R2
012120  016403  000002                  MOV     XCSR(R4),R3
012124  104001                          HLT     1
012126  005037  011032                  CLR     TIME
012132  000762                          BR      5$
012134  042700  100000          6$:     BIC     #XFLG,STAT
012140  000137  011710                  JMP     $ILB
```

```
                        ;*****************************************************************
                        ;       ROUTINE USED IF "EXTERNAL LOOP BACK" WAS SELECTED.
                        ;       EITHER HALF OR FULL DUPLEX MAY BE SELECTED IN THIS MODE.
                        ;       "EXTERNAL LOOP BACK" MEANS THAT THE TRANSMITTER IS FIRST
                        ;       TURNED ON (IF HALF DUPLEX) AND THE WHOLE MESSAGE IS TRANSMITTED;
                        ;       THEN THE RECEIVER IS ENABLED. AFTER THE WHOLE MESSAGE IS RECEIVED
                        ;       DATA WILL THEN BE CHECKED IF DESIRED AND END PASS WILL
                        ;       BE GIVEN IF DESIRED.    THEN THE CYCLE IS REPEATED
                        ;       AS ABOVE.  IF RUNNING IN FULL DUPLEX THE PROGRAM
                        ;       WAITS FOR BOTH THE RECEIVER AND TRANSMITTER TO
                        ;       FINISH THEN RESTARTS THE RECEIVER AND TRANSMITTER.
                        ;*****************************************************************

012144  104416          $XLB:   KBDIN
012146  032737  000002  011014          BIT     #BIT1,PARAM2            ;SECONDARY MODE?
012154  001004                        BNE     9$                      ;BR IF YES
012156  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
012164  001417                        BEQ     1$
012166  004737  013474          9$:     JSR     PC,STARTR
012172  032737  000004  011014          BIT     #BIT2,PARAM2            ;SECONDARY STATION?
012200  001411                        BEQ     1$                      ;BR IF NO
012202  005737  016750                TST     TEMP2                   ;FIRST TIME HERE?
012206  001403                        BEQ     11$                     ;BR IF YES
012210  032700  040000                BIT     #RFLG,STAT              ;WAIT FOR RECEIVE BEFORE
012214  001774                        BEQ     .-6                     ;TRANSMITTING
012216  012737  177777  016750  11$:   MOV     #-1,TEMP2               ;SET FIRST TIME FLAG
012224  004737  013404          1$:     JSR     PC,STARTX
012230  005037  011032                CLR     TIME
012234  032700  100000          2$:     BIT     #XFLG,STAT
012240  001016                        BNE     3$
012242  032700  040000          7$:     BIT     #RFLG,STAT
012246  001030                        BNE     4$
012250  023727  011032  000100        CMP     TIME,#100
012256  103766                        BLO     2$
012260  011402                        MOV     @RCSR,R2
012262  016403  000002                MOV     XCSR(R4),R3
012266  104001                        HLT     1
012270  005037  011032                CLR     TIME
012274  000757                        BR      2$
012276  032737  000002  011014  3$:     BIT     #BIT1,PARAM2            ;SECONDARY MODE?
012304  001356                        BNE     7$                      ;BR IF YES
012306  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
012314  001352                        BNE     7$
012316  042700  100000                BIC     #XFLG,STAT
012322  004737  013474                JSR     PC,STARTR
012326  000742                        BR      2$
012330  032737  000002  011014  4$:     BIT     #BIT1,PARAM2            ;SECONDARY MODE?
012336  001004                        BNE     10$                     ;BR IF YES
012340  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
012346  001420                        BEQ     8$
012350  032700  100000          10$:    BIT     #XFLG,STAT
012354  001013                        BNE     6$
012356  023727  011032  000100        CMP     TIME,#100
012364  103761                        BLO     4$
012366  011402                        MOV     @RCSR,R2
012370  016403  000002                MOV     XCSR(R4),R3
```

```
  974  012374  104001                          HLT     1
  975  012376  005037  011032                  CLR     TIME
  976  012402  000752                          BR      4$
  977  012404  042700  100000          6$:     BIC     #XFLG,STAT
  978  012410  042700  040000          8$:     BIC     #RFLG,STAT
  979  012414  005037  011032                  CLR     TIME
  980  012420  032777  000200  176416          BIT     #NODAT,@SWR
  981  012426  001002                          BNE     5$
  982  012430  004737  012622                  JSR     PC,TESTD
  983  012434  004737  016176          5$:     JSR     PC,CKBASE       ;CHECK DMC SOFT ERROR COUNTERS
  984  012440  032777  000020  176376          BIT     #LOOP,@SWR
  985  012446  001636                          BEQ     $XLB
  986  012450  012737  012144  013400          MOV     #$XLB,BACK
  987  012456  000137  012462                  JMP     EOP
```

```
                                        ;*****************************
                                        ;       ROUTINE TO RETURN
                                        ;       TO MONITOR FOR
                                        ;       END PASS.
                                        ;*****************************

012462                                  EOP:
                                                STPS,PRTY7              ;SET PS PRIORITY TO 7
012462  104414  000340                        MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012466  016437  000002  012620                BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012474  042737  177677  012620                BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012502  042764  000100  000002                MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012510  012766  012550  000002                MOV     R0,SAVR0        ;SAVE REGISTER 0
012516  010037  013362                        MOV     R1,SAVR1        ;SAVE REGISTER 1
012522  010137  013364                        MOV     R2,SAVR2        ;SAVE REGISTER 2
012526  010237  013366                        MOV     R3,SAVR3        ;SAVE REGISTER 3
012532  010337  013370                        MOV     R4,SAVR4        ;SAVE REGISTER 4
012536  010437  013372                        MOV     R5,SAVR5        ;SAVE REGISTER 5
012542  010537  013374                        RTS     PC              ;RETURN TO CONTROL PROGRAM
012546  000207
012550                                  ENTER:
012550  013700  013362                        MOV     SAVR0,R0        ;RESTORE R0
012554  013701  013364                        MOV     SAVR1,R1        ;RESTORE R1
012560  013702  013366                        MOV     SAVR2,R2        ;RESTORE R2
012564  013703  013370                        MOV     SAVR3,R3        ;RESTORE R3
012570  013704  013372                        MOV     SAVR4,R4        ;RESTORE R4
012574  013705  013374                        MOV     SAVR5,R5        ;RESTORE R5
012600  012737  177777  013376                MOV     #-1,DELAY
012606  053764  012620  000002                BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012614  000177  000560                        JMP     @BACK
012620  000000                          QTPIE:  000000



                                        ;*********************************
                                        ;       SUBROUTINE TO CHECK
                                        ;       RECEIVER DATA.
                                        ;*********************************
012622  013746  011056                  TESTD:  MOV     ERDBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012626  001413                                BEQ     TSTDAT          ;BR IF NO
012630  032777  020000  176206                BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012636  001007                                BNE     TSTDAT          ;BR IF YES
012640  104400  013022                        TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012644  004077  176160                        JSR     R0,@#2016       ;PRINT CONTENTS OF RBUF
012650  005746                                TST     -(SP)
012652  104400  013103                        TYPE    ,MSG1           ;<15><12>
012656  013701  011022                  TSTDAT: MOV     IXDA,   R1      ;SETUP XMIT DATA ADDR
012662  013702  011020                        MOV     IRDA,   R2      ;SETUP RCV DATA ADDR
012666  122122                          SCAN4:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012670  001776                                BEQ     SCAN4           ;BR IF OK
012672  123741  011040                        CMPB    TX.TERM,-(R1)   ;IS IT END OF DATA
012676  001447                                BEQ     TESTDX          ;BR IF YES
012700  122742  000002                        CMPB    #002,-(R2)
012704  001005                                BNE     2$
012706  010237  012714                        MOV     R2,1$
012712  104400                                TYPE
```

```
1044  012714  000000                  1$:     .WORD   0
1045  012716  000437                          BR      TESTDX
1046  012720                          2$:
1047  012720  105712                          TSTB    (R2)            ;
1048  012722  001435                          BEQ     TESTDX          ;BR IF YES
1049  012724  122721  000177                  CMPB    #177    (R1)+   ;IS IT FILL CHAR?
1050  012730  001756                          BEQ     SCAN4           ;BR IF YES
1051  012732  005301                          DEC     R1              ;BACKUP
1052  012734  122722  000177                  CMPB    #177,   (R2)+   ;IS IT FILL?
1053  012740  001752                          BEQ     SCAN4           ;BR IF YES
1054  012742  000240                  SCAN5:  NOP                     ;DATA ERROR
1055  012744  032777  020000  176072          BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS
1056  012752  001016                          BNE     DERR            ;BR IF YES
1057  012754  104400  013106                  TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
1058  012760  013737  011020  012770          MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
1059  012766  104400                          TYPE                    ;PRINT RECEIVED DATA
1060  012770  000000                  RDAX:   0                       ;RECEIVED DATA ADDR.
1061  012772  104400  013133                  TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
1062  012776  013737  011022  013006          MOV     IXDA,   .+10    ;SETUP ADDR.
1063  013004  104400                          TYPE                    ;PRINT GOOD DATA
1064  013006  011022                          IXDA
1065  013010  111103                  DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
1066  013012  114202                          MOVB    -(R2),R2        ;SETUP RCV DATA
1067  013014  104007                          HLT+7                   ;DATA ERROR HALT
1068  013016  005726                  TESTDX: TST     (SP)+           ;POP STACK
1069  013020  000207                          RTS     PC              ;RETURN FROM SUB/ROUT
1070
1071  013022  005015  044124  051105  MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
 (1)  013103     015  000012          MSG1:   .ASCIZ  <15><12>
 (1)  013106  005015  042522  042503  MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
 (1)  013133     015  042012  052101  MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
 (1)  013156  005015  046120  040505  MSG4:   .ASCII  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
 (1)  013225     015  053412  042510          .ASCIZ  <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
 (1)  013310  005015  046120  040505  MSG5:   .ASCIZ  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
 (1)                                  .EVEN
 (1)  013362  000000                  SAVR0:  0
1072  013364  000000                  SAVR1:  0
1073  013366  000000                  SAVR2:  0
1074  013370  000000                  SAVR3:  0
1075  013372  000000                  SAVR4:  0
1076  013374  000000                  SAVR5:  0
1077  013376  000000                  DELAY:  0
1078  013400  000000                  BACK:   0
1079  013402  000000                  STOP:   0
1080
```

```
1081                                                    ;*****************************************************************
1082                                                    ;       TRANSMIT INIT ROUTINE
1083                                                    ;*****************************************************************
1084                                            06300
1085  013404  005737  013402                   06400   STARTX: TST     STOP            ;FIRST TIME HERE?
1086  013410  001005                           06500           BNE     1$              ;BR IF NOT
1087  013412  104400  013156                   06600           TYPE    ,MSG4           ;TYPE CONNECT MESS
1088  013416  000000                           06700           HALT
1089  013420  005137  013402                   06800           COM     STOP            ;SET FIRST TIME FLAG
1090  013424  005737  016744                   06900   1$:     TST     BASEFG          ;BASE AND CNTL IN ALL DONE?
1091  013430  001775                           07000           BEQ     1$              ;BR IF NO
1092  013432  005037  016744                   07100           CLR     BASEFG          ;CLEAR FLAG
1093  013436  005037  016746                   07200           CLR     TEMP1           ;GET SET TO DELAY
1094  013442  012737  000025  016750           07300           MOV     #25,TEMP2       ;
1095  013450  062737  000001  016746           07400           ADD     #1,TEMP1        ;INC DELAY
1096  013456  001374                           07500           BNE     .-6             ;
1097  013460  005337  016750                   07600           DEC     TEMP2           ;DEC DELAY COUNTER
1098  013464  001371                           07700           BNE     .-14            ;BR IF NOT DONE
1099  013466  152714  000140                   07800           BISB    #140,(R4)       ;ASK FOR XMIT BUFFER
1100  013472  000207                           07900           RTS     PC              ;RETURN
1101                                            08000
1102                                                    ;*****************************************************************
1103                                                    ;       RECEIVE INIT ROUTINE
1104                                                    ;*****************************************************************
1105                                            08200
1106  013474  005737  013402                   08300   STARTR: TST     STOP            ;FIRST TIME HERE?
1107  013500  001004                           08400           BNE     1$              ;BR IF NOT
1108  013502  104400  013310                   08500           TYPE    ,MSG5           ;TYPE CONNECT MESS
1109  013506  005137  013402                   08600           COM     STOP            ;SET FIRST TIME FLAG
1110  013512  005737  016754                   08700   1$:     TST     RFLAG           ;HAS AN REC BUFFER ALREADY BEEN GIVEN
1111  013516  001007                           08800           BNE     2$              ;BR IF YES
1112  013520  005737  016744                   08900           TST     BASEFG          ;BASE AND CNTL IN ALL DONE?
1113  013524  001772                           09000           BEQ     1$              ;BR IF NO
1114  013526  005037  016744                   09100           CLR     BASEFG          ;CLEAR FLAG
1115  013532  152714  000144                   09200           BISB    #144,(R4)       ;ASK FOR REC BUFFER
1116  013536  005037  016754                   09300   2$:     CLR     RFLAG           ;CLEAR FLAG
1117  013542  000207                           09400           RTS     PC              ;RETURN
1118                                            09500
1119                                                    ;*****************************************************************
1120                                                    ;       INPUT SERVICE ROUTINE
1121                                                    ;*****************************************************************
1122                                            09700
1123  013544  032714  000002                   09800   XISR:   BIT     #BIT1,(R4)      ;BASE REQUEST?
1124  013550  001426                           09900           BEQ     1$              ;BR IF NO
1125  013552  032714  000004                   10000           BIT     #BIT2,(R4)      ;IS IT REALLY A SHUT DOWN?
1126  013556  001403                           10100           BEQ     9$              ;BR IF NO
1127  013560  004737  014020                   10200           JSR     PC,4$           ;YES, CLEAR RQI
1128  013564  000002                           10300           RTI                     ;AND RETURN
1129  013566  012764  016776  000004           10400   9$:     MOV     #BASE,4(R4)     ;LOAD BASE ADDRESS
1130  013574  005064  000006                   10500           CLR     6(R4)           ;CLEAR SEL 6
1131  013600  005737  016742                   10600           TST     RESUME          ;RESUME FLAG SET?
1132  013604  001403                           10700           BEQ     8$              ;BR IF NOT
1133  013606  012764  010000  000006           10800           MOV     #BIT12,6(R4)    ;OTHERWISE SET RESUME BIT
1134  013614  004737  014020                   10900   8$:     JSR     PC,4$           ;CLEAR RQI
1135  013620  152714  000141                   11000           BISB    #141,(R4)       ;ASK FOR CNTL I
1136  013624  000002                           11100           RTI                     ;RETURN
```

```
 1137  013626  032714  000001          11200  1$:     BIT     #BIT0,(R4)      ;CNTL I REQUEST?
 1138  013632  001430                  11300          BEQ     2$              ;BR IF NO
 1139  013634  032737  000001  011014  11400          BIT     #FULL.DUPLEX,PARAM2;FULL OR HALF?
 1140  013642  001403                  11500          BEQ     5$              ;BR IF HALF
 1141  013644  005064  000006          11600          CLR     6(R4)           ;SELECT FULL DUPLEX
 1142  013650  000413                  11700          BR      6$              ;CONTINUE
 1143  013652  032737  000004  011014  11800  5$:     BIT     #BIT2,PARAM2    ;SECONDARY STATION?
 1144  013660  001404                  11900          BEQ     7$              ;BR IF NO
 1145  013662  012764  006000  000006  12000          MOV     #BIT10!BIT11,6(R4);SET SEC MODE AND HALF DUPLEX
 1146  013670  000403                  12100          BR      6$              ;CONTINUE
 1147  013672  012764  002000  000006  12200  7$:     MOV     #BIT10,6(R4)    ;SELECT HALF DUPLEX
 1148  013700  004737  014020          12300  6$:     JSR     PC,4$           ;CLEAR RQI
 1149  013704  012737  177777  016744  12400          MOV     #-1,BASEFG      ;SET BASE LOADED FLAG
 1150  013712  000002                  12500          RTI                     ;RETURN
 1151  013714  032714  000004          12600  2$:     BIT     #BIT2,(R4)      ;RECEIVE BUFFER REQUEST?
 1152  013720  001416                  12700          BEQ     3$              ;BR IF NO
 1153  013722  042700  040000          12800          BIC     #RFLG,STAT      ;CLEAR RECEIVE FLAG BIT
 1154  013726  013764  011020  000004  12900          MOV     IRDA,4(R4)      ;LOAD REC BUFFER ADDRESS
 1155  013734  013764  011064  000006  13000          MOV     RXWC,6(R4)      ;LOAD REC BYTE COUNT
 1156  013742  004737  014020          13100          JSR     PC,4$           ;CLEAR RQI
 1157  013746  012737  177777  016744  13200          MOV     #-1,BASEFG      ;SET FLAG
 1158  013754  000002                  13300          RTI                     ;RETURN
 1159  013756  042700  100000          13400  3$:     BIC     #XFLG,STAT      ;CLEAR XMIT FLAG BIT
 1160  013762  012737  177777  016752  13500          MOV     #-1,XFLAG       ;SET XMIT FLAG(FOR TIMEOUT ERROR)
 1161  013770  013764  011022  000004  13600          MOV     IXDA,4(R4)      ;LOAD XMIT BUFFER ADDRESS
 1162  013776  013764  011062  000006  13700          MOV     TXWC,6(R4)      ;LOAD XMIT BYTE COUNT
 1163  014004  004737  014020          13800          JSR     PC,4$           ;CLEAR RQI
 1164  014010  012737  177777  016744  13900          MOV     #-1,BASEFG      ;SET FLAG
 1165  014016  000002                  14000          RTI                     ;RETURN
 1166  014020  142714  000040          14100  4$:     BICB    #40,(R4)        ;CLEAR RQI
 1167  014024  105714                  14200          TSTB    (R4)            ;WAIT FOR RQI TO DROP
 1168  014026  100776                  14300          BMI     .-2             ;BR IF STILL SET
 1169  014030  000207                  14400          RTS     PC              ;RETURN
 1170                                  14500
 1171                                          ;*****************************************************************
 1172                                          ;       OUTPUT SERVICE ROUTINE
 1173                                          ;*****************************************************************
 1174                                  14700
 1175  014032  032764  000001  000002  14800  RISR:  BIT     #BIT0,2(R4)     ;ERROR?
 1176  014040  001463                  14900          BEQ     1$              ;BR IF NO
 1177  014042  005737  016742          15000          TST     RESUME          ;RESUME FLAG SET?
 1178  014046  001413                  15100          BEQ     5$              ;BR IF NOT
 1179  014050  022764  001000  000006  15200          CMP     #BIT9,6(R4)     ;IS PROC. ERROR BIT SET?
 1180  014056  001007                  15300          BNE     5$              ;BR IF NOT
 1181  014060  142764  000207  000002  15400          BICB    #207,2(R4)      ;CLEAR DONE
 1182  014066  012737  177777  016744  15500          MOV     #-1,BASEFG      ;SET BASEFG
 1183  014074  000002                  15600          RTI                     ;RETURN
 1184  014076  022764  000004  000006  15700  5$:     CMP     #4,6(R4)        ;OVERUN ERROR?
 1185  014104  001003                  15800          BNE     3$              ;BR IF NO
 1186  014106  152714  000144          15900          BISB    #144,(R4)       ;REQUEUE XMIT BUFFER
 1187  014112  000432                  16000          BR      4$
 1188  014114  016437  000004  011054  16100  3$:     MOV     4(R4),ERCSR     ;SAVE SEL4
 1189  014122  016437  000006  011056  16200          MOV     6(R4),ERDBR     ;SAVE SEL6 (ERROR BITS,
 1190  014130  104400  016366          16300          TYPE    ,DMCER          ;ERROR MESSAGE
 1191  014134  013746  011054          16400          MOV     ERCSR,-(SP)     ;PUSH SEL4 ON STACK FOR TYPOUT
 1192  014140  004037  015770          16500          JSR     R0,$B2OCT       ;TYPE IT OUT
```

```
                                                16600           .BYTE   6
                                                16700           .BYTE   1
                                                16800           TYPE    SPACE3          ;INSERT 3 SPACES
                                                16900           MOV     ERRBR,-(SP)     ;SPUSH SELB ON STACK FOR "YPEOU"
                                                17000           JSR     R0,$B2OCT       ;TYPE IT OUT
                                                17100           .BYTE   6
                                                17200           .BYTE   1
                                                17300           TST     @SWR            ;CHECK BIT .5
                                                17400           BPL     .+4             ;SKIP HALT IF = 0
                                                17500           HALT                    ;HALT IF  SWR15 = 1
                                                17600           CLR     ERRBR           ;CLR  ERRBR LOCATION
                                                17700   4$:     BICB    @207,2(R4)      ;CLEAR DONE
                                                17800           RTI                     ;RETURN
                                                17900   1$:     BIT     @BIT2,2(R4)     ;REC DONE?
                                                18000           BEQ     2$              ;BR IF NO
                                                18100           BIS     @RFLG,STAT      ;SET REC DONE FLAG
                                                18200           BICB    @207,2(R4)      ;CLEAR DONE
                                                18300           RTI                     ;RETURN
                                                18400   2$:     BIS     @XFLG,STAT      ;SET XMIT DONE FLAG
                                                18500           CLR     XFLAG           ;CLEAR XFLAG
                                                18600           BICB    @207,2(R4)      ;CLEAR DONE
                                                18700           RTI                     ;RETURN
                                                18800
                                                18900
                                                19100
                                                19200   BOOT:   BIT     @BIT10!BIT9,PARAM1;DETERMINE WHICH BOOT MODE
                                                19300           BEQ     AUTORG          ;BR IF AUTO MODE, ORIGINATING STATION
                                                19400           BIT     @BIT10,PARAM1
                                                19500           BEQ     AUTBOO          ;BR IF AUTO MODE, BOOT STATION
                                                19600           BIT     @BIT9,PARAM1
                                                19700           BEQ     MANORG          ;BR IF MANUAL MODE, ORIGINATING STATION
                                                19800           TYPE    ,BOOMSG         ;MANUAL MODE, BOOT STATION
                                                19900           BR      .               ;WAIT FOR MANUAL BOOT
                                                20000
                                                20200
                                                20300   AUTORG: MOV     @MOP1+1,R1      ;LOAD MOP1 MESSAGE WITH
                                                20400           MOVB    PARAM1,(R1)+    ;PASSWORD FOUR TIMES
                                                20500           MOVB    PARAM1,(R1)+
                                                20600           MOVB    PARAM1,(R1)+
                                                20700           MOVB    PARAM1,(R1)+    ;
                                                20800           TST     BASEFG          ;IS BASE REQUEST DONE?
                                                20900           BEQ     AUTORG          ;BR IF NO
                                                21000           CLR     BASEFG          ;CLEAR LOCK FLAG
                                                21100           MOV     @2400,SEL6      ;MAINT. MODE BIT(MOP)
                                                21200           BIS     @141,(R4)       ;ASK FOR CNTL IN
                                                21300   1$:     TST     BASEFG          ;IS CNTL IN DONE?
                                                21400           BEQ     1$              ;BR IF NO
                                                21500           CLR     BASEFG          ;CLEAR LOCK FLAG
                                                21600           CLR     RFLAG           ;CLEAR RECEIVE FLAG
                                                21700           BIS     @144,(R4)       ;ASK FOR REC BA/CC
```

;***
; ENTER HERE IF BOOT MODE WAS SELECTED
;***

;***
; AUTOMATIC MODE ORIGINATING STATION
;***

```
1249  014404  005737  016744          21800  2$:   TST     BASEFG          ;IS REQUEST DONE?
1250  014410  001775                  21900         BEQ     2$              ;BR IF NO
1251  014412  005037  016744          22000         CLR     BASEFG          ;CLEAR LOCK FLAG
1252  014416  005037  016752          22100         CLR     XFLAG           ;CLEAR XMIT FLAG
1253  014422  012737  015602  016762  22200         MOV     #MOP1,SEL4      ;SET FOR DMC LOAD
1254  014430  012737  000005  016764  22300         MOV     #5,SEL6         ;SET FOR DMC LOAD
1255  014436  052714  000140          22400         BIS     #140,(R4)       ;ASK FOR XMIT BA/CC
1256  014442  012737  000005  016746  22500         MOV     #5,TEMP1        ;SET UP DELAY COUNT
1257  014450  005037  013376          22600         CLR     DELAY
1258  014454  005737  016754          22700  3$:   TST     RFLAG           ;RECEIVED ANYTHING?
1259  014460  001012                  22800         BNE     4$              ;BR IF YES
1260  014462  005337  013376          22900         DEC     DELAY           ;DEC DELAY COUNTER
1261  014466  001372                  23000         BNE     3$              ;WAIT TO RECEIVE
1262  014470  005337  016746          23100         DEC     TEMP1           ;DEC SECOND COUNT
1263  014474  001367                  23200         BNE     3$              ;BR IF NOT DONE DELAY
1264  014476  005737  016752          23300         TST     XFLAG           ;WAS XMIT COMPLETED ?
1265  014502  001340                  23400         BNE     2$              ;BR IF YES, SEND MOP1 AGAIN
1266  014504  000000                  23500         HALT                    ;ERROR, MOP1 WAS NEVER SENT OUT
1267                                  23600
1268                                  23700         ;CHECK TO SEE IF RECEIVED MESSAGE IS MOP2
1269                                  23800
1270  014506  012701  015610          23900  4$:   MOV     #MOP2,R1        ;STARTING ADDRESS OF MOP2
1271  014512  013702  011020          24000         MOV     IRDA,R2         ;RECEIVE BUFFER ADDRESS
1272  014516  005003                  24100         CLR     R3              ;CLEAR COUNT
1273  014520  122122                  24200  5$:   CMPB    (R1)+,(R2)+     ;COMPARE DATA
1274  014522  001317                  24300         BNE     1$              ;IF NOT MOP2 TRY AGAIN
1275  014524  005203                  24400         INC     R3              ;DATA OK, BUMP COUNTER
1276  014526  022703  000004          24500         CMP     #4,R3           ;DONE YET?
1277  014532  001372                  24600         BNE     5$              ;BR IF NO
1278  014534  004737  015506          24700         JSR     PC,MP3          ;IT WAS MOP2, SO SEND MOP3
1279                                  24800
1280                                  ;*****************************************************************
1281                                  ;       AUTOMATIC MODE BOOT STATION
1282                                  ;*****************************************************************
1283                                  25000
1284  014540  005737  016744          25100  AUTBOO: TST     BASEFG          ;BASE COMPLETED?
1285  014544  001775                  25200         BEQ     AUTBOO          ;BR IF NO
1286  014546  005037  016744          25300         CLR     BASEFG          ;CLEAR LOCK FLAG
1287  014552  005037  016760          25400         CLR     MFLAG           ;CLEAR MAINT FLAG
1288  014556  005037  016754          25500         CLR     RFLAG           ;CLEAR REC FLAG
1289  014562  052714  000144          25600         BIS     #144,(R4)       ;ASK FOR REC BA/CC
1290  014566  005737  016760          25700  1$:   TST     MFLAG           ;ARE WE IN MAINT MODE?
1291  014572  001362                  25800         BNE     AUTBOO          ;BR IF YES
1292  014574  005737  016754          25900  2$:   TST     RFLAG           ;DID WE RECEIVE ANYTHING?
1293  014600  001357                  26000         BNE     AUTBOO          ;YES REQUEUE RECEIVE BUFFER
1294  014602  000771                  26100         BR      1$              ;KEEP ON TRUCKIN'
1295                                  26200
1296                                  ;*****************************************************************
1297                                  ;       MANUAL MODE ORGINATING STATION
1298                                  ;*****************************************************************
1299                                  26400
1300  014604  005037  016760          26500  MANORG: CLR     MFLAG           ;CLEAR MAINT FLAG
1301  014610  005037  016756          26600         CLR     SFLAG           ;CLEAR DDCMP START RECEIVED ERROR FLAG
1302  014614  005737  016744          26700  7$:   TST     BASEFG          ;BASE LOADED?
1303  014620  001775                  26800         BEQ     7$              ;BR IF NO
1304  014622  005037  016744          26900         CLR     BASEFG          ;RESET FLAG
```

```
1305  014626  005037  016764          27000          CLR     SEL6            ;LOAD SEL6 FOR FULL-DUPLEX
1306  014632  052714  000141          27100          BIS     #141,(R4)       ;ASK FOR CNTLI
1307  014636  005737  016744          27200  8$:     TST     BASEFG          ;CNTLI DONE?
1308  014642  001775                  27300          BEQ     8$              ;BR IF NOT
1309  014644  005037  016744          27400          CLR     BASEFG          ;RESET FLAG
1310  014650  005037  016754          27500          CLR     RFLAG           ;CLEAR RECEIVE FLAG
1311  014654  052714  000144          27600          BIS     #144,(R4)       ;ASK FOR REC BA/CC
1312  014660  005737  016756          27700  1$:     TST     SFLAG           ;DOCMP START ERROR?
1313  014664  001437                  27800          BEQ     4$              ;BR IF NO
1314  014666  012714  040000          27900          MOV     #BIT14,(R4)     ;INITIALIZE DMC
1315  014672  005714                  28000  2$:     TST     (R4)            ;RUN SET?
1316  014674  100376                  28100          BPL     2$              ;BR IF NO
1317  014676  005037  016756          28200          CLR     SFLAG           ;CLEAR FLAG
1318  014702  005037  016744          28300          CLR     BASEFG          ;CLEAR LOCK FLAG
1319  014706  052764  000100  000002  28400          BIS     #100,2(R4)      ;SET INT ENABLE
1320  014714  052714  000143          28500          BIS     #143,(R4)       ;ASK FOR BASE
1321  014720  005737  016744          28600  3$:     TST     BASEFG          ;BASE DONE?
1322  014724  001775                  28700          BEQ     3$              ;BR IF NO
1323  014726  005037  016744          28800          CLR     BASEFG          ;CLEAR FLAG
1324  014732  005037  016764          28900          CLR     SEL6            ;SET UP FOR FULL-DUPLEX
1325  014736  052714  000141          29000          BIS     #141,(R4)       ;ASK FOR CNTLI
1326  014742  005737  016744          29100  9$:     TST     BASEFG          ;CNTLI FINISHED?
1327  014746  001775                  29200          BEQ     9$              ;BR IF NOT
1328  014750  005037  016744          29300          CLR     BASEFG          ;CLEAR FLAG
1329  014754  005037  016754          29400          CLR     RFLAG           ;CLEAR RECEIVER FLAG
1330  014760  052714  000144          29500          BIS     #144,(R4)       ;ASK FOR REC BA/CC
1331  014764  005737  016760          29600  4$:     TST     MFLAG           ;ARE WE IN MAINT MODE?
1332  014770  001733                  29700          BEQ     1$              ;BR IF NO
1333  014772  012714  040000          29800          MOV     #BIT14,(R4)     ;INITIALIZE DMC
1334  014776  005714                  29900  10$:    TST     (R4)            ;RUN SET?
1335  015000  100376                  30000          BPL     10$             ;BR IF NO
1336  015002  005037  016744          30100          CLR     BASEFG          ;CLEAR LOCK FLAG
1337  015006  052764  000100  000002  30200          BIS     #100,2(R4)      ;SET INT ENABLE
1338  015014  052714  000143          30300          BIS     #143,(R4)       ;ASK FOR BASE
1339  015020  005737  016744          30400  11$:    TST     BASEFG          ;BASE DONE?
1340  015024  001775                  30500          BEQ     11$             ;BR IF NO
1341  015026  005037  016744          30600          CLR     BASEFG          ;CLEAR FLAG
1342  015032  012737  002400  016764  30700          MOV     #2400,SEL6      ;MAINT. MODE (MOP)
1343  015040  052714  000141          30800          BIS     #141,(R4)       ;ASK FOR CNTLI
1344  015044  005737  016744          30900  12$:    TST     BASEFG          ;CNTLI FINISHED?
1345  015050  001775                  31000          BEQ     12$             ;BR IF NOT
1346  015052  005037  016744          31100          CLR     BASEFG          ;CLEAR FLAG
1347  015056  005037  016754          31200          CLR     RFLAG           ;CLEAR RECEIVER FLAG
1348  015062  052714  000144          31300          BIS     #144,(R4)       ;ASK FOR REC BA/CC
1349  015066  005737  016754          31400  5$:     TST     RFLAG           ;HAVE WE RECEIVED ANYTHING?
1350  015072  001775                  31500          BEQ     5$              ;BR IF NO
1351                                  31600
1352                                  31700          ;CHECK TO SEE IF RECEIVED MESSAGE IS MOP2
1353                                  31800
1354  015074  012701  015610          31900          MOV     #MOP2,R1        ;MOP2 STARTING ADDRESS
1355  015100  013702  011020          32000          MOV     IRDA,R2         ;RECEVE BUFFER ADDRESS
1356  015104  005003                  32100          CLR     R3              ;CLEAR COUNT
1357  015106  122122                  32200  6$:     CMPB    (R1)+,(R2)+     ;COMPARE DATA
1358  015110  001343                  32300          BNE     11$             ;IT ISN'T MOP2, TRY AGAIN
1359  015112  005203                  32400          INC     R3              ;DATA OK, BUMP COUNT
1360  015114  022703  000004          32500          CMP     #4,R3           ;DONE YET?
```

```
1361  015120  001372                  32600          BNE     6$              ;BR IF NO
1362  015122  004737  015506          32700          JSR     PC,MP3          ;IT WAS MOP2, SO SEND OUT MOP3
1363                                  32800
1364                                          ;*****************************************************************
1365                                          ;       INPUT INTERRUPT SERVICE ROUTINE (BOOT MODE)
1366                                          ;*****************************************************************
1367                                  33000
1368  015126  032714  000002          33100  IISR:   BIT     #BIT1,(R4)      ;IS IT A BASE REQUEST?
1369  015132  001413                  33200          BEQ     1$              ;NO
1370  015134  012764  016776  000004  33300          MOV     #BASE,4(R4)     ;YES, LOAD BASE ADDRESS
1371  015142  005064  000006          33400          CLR     6(R4)           ;CLEAR SEL6
1372  015146  004737  015300          33500          JSR     PC,4$           ;CLEAR RQI
1373  015152  012737  177777  016744  33600          MOV     #-1,BASEFG      ;SET FLAG
1374  015160  000002                  33700          RTI                     ;RETURN
1375  015162  032714  000001          33800  1$:     BIT     #BIT0,(R4)      ;IS IT CNTL IN?
1376  015166  001411                  33900          BEQ     2$              ;BR IF NO
1377  015170  013764  016764  000006  34000          MOV     SEL6,6(R4)      ;LOAD SEL6 FOR CNTLI
1378  015176  004737  015300          34100          JSR     PC,4$           ;CLEAR RQI
1379  015202  012737  177777  016744  34200          MOV     #-1,BASEFG      ;SET FLAG
1380  015210  000002                  34300          RTI                     ;RETURN
1381  015212  032714  000004          34400  2$:     BIT     #BIT2,(R4)      ;IS IT RECEIVE REQUEST?
1382  015216  001414                  34500          BEQ     3$              ;NO
1383  015220  013764  011020  000004  34600          MOV     IRDA,4(R4)      ;YES, LOAD REC BA
1384  015226  012764  000010  000006  34700          MOV     #10,6(R4)       ;CC
1385  015234  004737  015300          34800          JSR     PC,4$           ;CLEAR RQI
1386  015240  012737  177777  016744  34900          MOV     #-1,BASEFG      ;SET FLAG
1387  015246  000002                  35000          RTI                     ;RETURN
1388  015250  013764  016762  000004  35100  3$:     MOV     SEL4,4(R4)      ;XMIT REQUEST, LOAD XMIT BA
1389  015256  013764  016764  000006  35200          MOV     SEL6,6(R4)      ;XMIT CC
1390  015264  004737  015300          35300          JSR     PC,4$           ;CLEAR RQI
1391  015270  012737  177777  016744  35400          MOV     #-1,BASEFG      ;SET FLAG
1392  015276  000002                  35500          RTI                     ;RETURN
1393  015300  142714  000040          35600  4$:     BICB    #40,(R4)        ;CLEAR RQI
1394  015304  105714                  35700          TSTB    (R4)            ;RQI CLEAR?
1395  015306  100776                  35800          BMI     .-2             ;NO
1396  015310  000207                  35900          RTS     PC              ;RETURN
1397                                  36000
1398                                          ;*****************************************************************
1399                                          ;       OUTPUT INTERRUPT SERVICE ROUTINE (BOOT MODE)
1400                                          ;*****************************************************************
1401                                  36200
1402  015312  032764  000001  000002  36300  OISR:   BIT     #BIT0,2(R4)     ;ERROR?
1403  015320  001453                  36400          BEQ     3$              ;NO
1404  015322  022764  000010  000006  36500          CMP     #10,6(R4)       ;YES, MAINT MODE ENTERED?
1405  015330  001004                  36600          BNE     1$              ;NO
1406  015332  012737  177777  016760  36700          MOV     #-1,MFLAG       ;YES, SET MFLAG
1407  015340  000456                  36800          BR      5$              ;RETURN
1408  015342  022764  000200  000006  36900  1$:     CMP     #200,6(R4)      ;DOCMP START RECEIVED ERROR?
1409  015350  001004                  37000          BNE     2$              ;NO
1410  015352  012737  177777  016756  37100          MOV     #-1,SFLAG       ;YES,SET SFLAG
1411  015360  000446                  37200          BR      5$              ;RETURN
1412  015362  016437  000004  011054  37300  2$:     MOV     4(R4),ERCSR     ;SAVE SEL4
1413  015370  016437  000006  011056  37400          MOV     6(R4),ERDBR     ;SAVE SEL6 (ERROR BITS)
1414  015376  104400  016366          37500          TYPE    ,DMCER          ;ERROR MESSAGE
1415  015402  013746  011054          37600          MOV     ERCSR,-(SP)     ;PUSH SEL4 ON STACK FOR TYPEOUT
1416  015406  004037  015770          37700          JSR     R0,$B2OCT       ;TYPE IT OUT
```

```
1417  015412  000006                  37800            .BYTE   6
1418  015413  000001                  37900            .BYTE   1
1419  015414  104400  016562          38000            TYPE    ,SPACE3         ;INSERT 3 SPACES
1420  015420  013746  011056          38100            MOV     ERDBR,-(SP)     ;SPUSH SEL6 ON STACK FOR TYPEOUT
1421  015424  004037  015770          38200            JSR     R0,$$20CT       ;TYPE IT OUT
1422  015430  000006                  38300            .BYTE   6
1423  015431  000001                  38400            .BYTE   1
1424  015432  005777  173406          38500            TST     @SWR            ;CHECK BIT 15
1425  015436  100001                  38600            BPL     .+4             ;SKIP HALT IF = 0
1426  015440  000000                  38700            HALT                    ;HALT IF  SWR15 = 1
1427  015442  005037  011056          38800            CLR     ERDBR           ;CLR  ERDBR LOCATION
1428  015446  000413                  38900            BR      5$              ;RETURN
1429  015450  032764  000004  000002  39000  3$:       BIT     #BIT2,2(R4)     ;RECEIVE DONE?
1430  015456  001404                  39100            BEQ     4$              ;BR IF NO
1431  015460  012737  177777  016754  39200            MOV     #-1,RFLAG       ;SET RECEIVE FLAG
1432  015466  000403                  39300            BR      5$              ;RETURN
1433  015470  012737  177777  016752  39400  4$:       MOV     #-1,XFLAG       ;XMIT DONE, SET XMIT FLAG
1434  015476  142764  000207  000002  39500  5$:       BICB    #207,2(R4)      ;CLEAR DONE
1435  015504  000002                  39600            RTI                     ;RETURN
1436                                    39700
1437                                    ;*****************************************************************
1438                                    ;       SUBROUTINE TO SEND MOP3 MESSAGE
1439                                    ;*****************************************************************
1440                                    39900
1441  015506  005737  016744          40000  MP3:      TST     BASEFG          ;IS IT OK TO REQUEST
1442  015512  001775                  40100            BEQ     MP3             ;BR IF NO
1443  015514  005037  016744          40200            CLR     BASEFG          ;CLEAR LOCK FLAG
1444  015520  005037  016752          40300            CLR     XFLAG           ;CLEAR XMIT FLAG
1445  015524  012737  015614  016762  40400            MOV     #MOP3,SEL4      ;MOP3 ADDRESS
1446  015532  012737  000154  016764  40500            MOV     #MOP3ED-MOP3,SEL6 ;MOP3 COUNT
1447  015540  052714  000140          40600            BIS     #140,(R4)       ;ASK FOR XMIT BA/CC
1448  015544  005037  013376          40700            CLR     DELAY           ;START DELAY COUNT
1449  015550  005737  016752          40800  1$:       TST     XFLAG           ;XMIT DONE?
1450  015554  001004                  40900            BNE     2$              ;BR IF YES
1451  015556  005337  013376          41000            DEC     DELAY           ;DEC DELAY COUNT
1452  015562  001372                  41100            BNE     1$              ;BR IF NO DONE
1453  015564  000000                  41200            HALT                    ;ERROR, MOP3 SEND NOT DONE
1454  015566  104400  016635          41300  2$:       TYPE    ,ORGOK          ;OK MOP3 SEND DONE
1455  015572  005726                  41400            TST     (SP)+           ;POP STACK (ENTERED BY JSR)
1456  015574  000000                  41500            HALT                    ;ALL DONE HIT CONT TO DO IT AGAIN
1457  015576  000137  011106          41600            JMP     START
1458                                    41700
1459  015602  000006  000     000     41800  MOP1:     .BYTE   6,0,0,0,0
1460  015605  000000  000
1461          015610                  41900  .EVEN
1462  015610  000010  014     001     42000  MOP2:     .BYTE   10,12.,1,0
1463  015613  000000
1464  015614  000000  000     006     42100  MOP3:     .BYTE   0,0,6,0,6,0
1465  015617  000000  006     000
1466                                    42200            ;IMAGE OF PROGRAM TO BE DOWN LINE LOADED
1467                                    42300
1468  015622  005037  000006          42400            CLR     @#6             ;SET UP TIMEOUT VECTOR TO HALT
1469  015626  000005                  42500            RESET                   ;CLEAR ALL!!
1470  015630  012706  001000          42600            MOV     #1000,SP        ;SET UP STACK
1471  015634  012701  177560          42700            MOV     #177560,R1      ;SET TTY CSR
1472  015640  010700                  42800            MOV     PC,R0           ;MAKE ADDRESS PIC
```

```
1473  015642  062700  000034          42900            ADD     #<MSG-.>,R0     ;ADDRESS OF MESSAGE
1474  015646  105761  000004          43000    1$:     TSTB    4(R1)           ;READY SET?
1475  015652  100375                  43100            BPL     1$              ;BR IF NO
1476  015654  112061  000006          43200            MOVB    (R0)+,6(R1)     ;TYPE A CHARACTER
1477  015660  001372                  43300            BNE     1$              ;KEEP TYPING IF NOT ZERO
1478  015662  012737  000026  000024  43400            MOV     #26,@#24        ;SET UP POWER FAIL VECTOR
1479  015670  005037  000026          43500            CLR     @#26            ;MAKE SURE T BIT CLEAR
1480  015674  000777                  43600            BR      .               ;BR
1481  015676  006412  047502  052117  43700    MSG:    .ASCIZ  <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY - END OF TEST!!/
1482  015704  046440  051505  040523
1483  015712  042507  053440  051501
1484  015720  051040  041505  044505
1485  015726  042526  020104  052523
1486  015734  041503  051505  043123
1487  015742  046125  054514  026440
1488  015750  042440  042116  047440
1489  015756  020106  042524  052123
1490  015764  020441     000
1491  015767     006          43800            .BYTE   6
1492  015770                  43900    MOP3ED:
1493                          44000    .EVEN
1494                          44100
1495                          44200
1496                          44300    ;*****************************************************
1497                          44400    ;BINARY TO OCTAL (ASCII) AND TYPE
1498                          44500    ;$B2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1499                          44600    ;CALL:
1500                          44700    ;       MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1501                          44800    ;       JSR     R0,$B2OCT               ;CALL FOR TYPEOUT
1502                          44900    ;       .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1503                          45000    ;       .BYTE   M                       ;M=1 OR 0
1504                          45100                                            ;1=TYPE LEADING ZEROS
1505                          45200                                            ;0=SUPPRESS LEADING ZEROS
1506                          45300    ;
1507                          45400    ;$B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $BSOCT OR
1508                          45500    ;CALL:
1509                          45600    ;       MOV     NUM,-(SP)
1510                          45700    ;       JSR     R0,$B201
1511                          45800    ;
1512                          45900    ;$B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1513                          46000    ;CALL:
1514                          46100    ;       MOV     NUM,-(SP)
1515                          46200    ;       JSR     R0,$B2016
1516                          46300
1517  015770  112037  016175  46400    $B2OCT: MOVB    (R0)+,$OMODE+1          ;PICKUP THE NUMBER OF DIGITS TO TYPE
1518  015774  112037  016173  46500            MOVB    (R0)+,$OFILL            ;GET THE ZERO FILL SWITCH
1519  016000  000406          46600            BR      $B201
1520  016002  112737  000001  016173  46700    $B2016: MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
1521  016010  112737  000006  016175  46800            MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
1522  016016  112737  000005  016172  46900    $B201:  MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
1523  016024  010346          47000            MOV     R3,-(SP)                ;SAVE R3
1524  016026  010446          47100            MOV     R4,-(SP)                ;SAVE R4
1525  016030  010546          47200            MOV     R5,-(SP)                ;SAVE R5
1526  016032  113704  016175  47300            MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
1527  016036  005404          47400            NEG     R4
1528  016040  062704  000006  47500            ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
```

```
1529  016044  110437  016174          47600          MOVB    R4,$OMODE       ;SAVE IT FOR USE
1530  016050  113704  016173          47700          MOVB    $OFILL,R4       ;GET THE ZERO FILL SWITCH
1531  016054  016605  000010          47800          MOV     10(SP),R5       ;PICKUP THE INPUT NUMBER
1532  016060  005003                  47900          CLR     R3              ;CLEAR THE OUTPUT WORD
1533  016062  006105                  48000  1$:     ROL     R5              ;ROTATE MSB INTO "C"
1534  016064  000404                  48100          BR      3$              ;GO DO MSB
1535  016066  006105                  48200  2$:     ROL     R5              ;FORM THIS DIGIT
1536  016070  006105                  48300          ROL     R5
1537  016072  006105                  48400          ROL     R5
1538  016074  010503                  48500          MOV     R5,R3
1539  016076  006103                  48600  3$:     ROL     R3              ;GET LSB OF THIS DIGIT
1540  016100  105337  016174          48700          DECB    $OMODE          ;TYPE THIS DIGIT?
1541  016104  100016                  48800          BPL     7$              ;BR IF NO
1542  016106  042703  177770          48900          BIC     #177770,R3      ;GET RID OF JUNK
1543  016112  001002                  49000          BNE     4$              ;TEST FOR 0
1544  016114  005704                  49100          TST     R4              ;SUPPRESS THIS 0?
1545  016116  001403                  49200          BEQ     5$              ;BR IF YES
1546  016120  005204                  49300  4$:     INC     R4              ;DON'T SUPPRESS ANYMORE 0'S
1547  016122  052703  000060          49400          BIS     #'0,R3          ;MAKE THIS DIGIT ASCII
1548  016126  052703  000040          49500  5$:     BIS     #' ,R3          ;MAKE ASCII IF NOT ALREADY
1549  016132  110337  016170          49600          MOVB    R3,8$           ;SAVE FOR TYPING
1550  016136  104400  016170          49700          TYPE    ,8$             ;GO TYPE THIS DIGIT
1551  016142  105337  016172          49800  7$:     DECB    $OCNT           ;COUNT BY 1
1552  016146  003347                  49900          BGT     2$              ;BR IF MORE TO DO
1553  016150  002402                  50000          BLT     6$              ;BR IF DONE
1554  016152  005204                  50100          INC     R4              ;INSURE LAST DIGIT ISN'T A BLANK
1555  016154  000744                  50200          BR      2$              ;GO DO THE LAST DIGIT
1556  016156  012605                  50300  6$:     MOV     (SP)+,R5        ;RESTORE R5
1557  016160  012604                  50400          MOV     (SP)+,R4        ;RESTORE R4
1558  016162  012603                  50500          MOV     (SP)+,R3        ;RESTORE R3
1559  016164  012616                  50600          MOV     (SP)+,(SP)      ;SET THE STACK FOR RETURNING
1560  016166  000200                  50700          RTS     R0              ;RETURN
1561  016170     000                  50800  8$:     .BYTE   0               ;STORAGE FOR ASCII DIGIT
1562  016171     000                  50900          .BYTE   0               ;TERMINATOR FOR TYPE ROUTINE
1563  016172     000                  51000  $OCNT:  .BYTE   0               ;OCTAL DIGIT COUNTER
1564  016173     000                  51100  $OFILL: .BYTE   0               ;ZERO FILL SWITCH
1565  016174  000000                  51200  $OMODE: 0                       ;NUMBER OF DIGITS TO TYPE
1566                                  51300
1567                                  51400
1568  016176  012737  177777  016742  51500  CKBASE: MOV     #-1,RESUME      ;SET RESUME FLAG
1569  016204  005037  016744          51600          CLR     BASEFG          ;CLEAR BASEFG
1570  016210  052714  000146          51700          BIS     #146,(R4)       ;SHUT DOWN DMC TO UPDATE BASE TABLE
1571  016214  005737  016744          51800  1$:     TST     BASEFG          ;SHUT DOWN DONE?
1572  016220  001775                  51900          BEQ     1$              ;BR IF NO
1573  016222  012714  040000          52000          MOV     #BIT14,(R4)     ;MASTER CLEAR DMC
1574  016226  005714                  52100  2$:     TST     (R4)            ;RUN SET?
1575  016230  100376                  52200          BPL     2$              ;BR IF NO
1576  016232  005037  016744          52300          CLR     BASEFG          ;CLEAR BASEFG
1577  016236  052714  000143          52400          BIS     #143,(R4)       ;ASK FOR BASE REQUEST
1578  016242  005737  016744          52500  3$:     TST     BASEFG          ;BASE LOADED?
1579  016246  001775                  52600          BEQ     3$              ;BR IF NO
1580  016250  012764  000100  000002  52700          MOV     #100,2(R4)      ;SET INTERRUPT ENABLE
1581  016256  005037  016742          52800          CLR     RESUME          ;CLEAR RESUME FLAG
1582  016262  012702  000003          52900          MOV     #3,R2           ;LOAD BASE OFFSET TO ERROR COUNTS
1583  016266  005001                  53000          CLR     R1              ;R1 IS OFFSET INTO SOFTWARE TABLE
1584  016270  126261  016776  016766  53100  5$:     CMPB    BASE(R2),ERRCNT(R1);ANY ERRORS THIS PASS?
```

```
1585  016276  001005                  53200            BNE     6$              ;BR IF YES
1586  016300  122122                  53300            CMPB    (R1)+,(R2)+     ;INC INDEXS
1587  016302  022702  000013          53400            CMP     #13,R2          ;DONE?
1588  016306  001370                  53500            BNE     5$              ;BR IF NO
1589  016310  000207                  53600            RTS     PC              ;RETURN
1590  016312  104400  016437          53700    6$:     TYPE    ,SOFT           ;TYPE SOFT ERROR MESSAGE
1591  016316  012702  000003          53800            MOV     #3,R2           ;LOAD BASE OFFSET TO ERROR COUNTS
1592  016322  005001                  53900            CLR     R1              ;SOFTWARE TABLE OFFSET
1593  016324  116261  016776  016766  54000    7$:     MOVB    BASE(R2),ERRCNT(R1);SAVE  ERROR COUNTS
1594  016332  116246  016776          54100            MOVB    BASE(R2),-(SP)  ;PUSH ON STACK FOR TYPEOUT
1595  016336  042716  177400          54200            BIC     #^C<377>,(SP)   ;CLEAR HI-BYTE
1596  016342  004037  015770          54300            JSR     R0,$$20CT       ;TYPE IT OUT
1597  016346     003                  54400            .BYTE   3
1599  016347     001                  54500            .BYTE   1
1599  016350  104400  016562          54600            TYPE    ,SPACE3         ;INSERT 3 SPACES
1600  016354  122221                  54700            CMPB    (R2)+,(R1)+     ;INC INDEXS
1601  016356  022702  000013          54800            CMP     #13,R2          ;DONE?
1602  016362  001360                  54900            BNE     7$              ;BR IF NO
1603  016364  000207                  55000            RTS     PC              ;RETURN
1604                                  55100
1605                                  55200
1606                                  55300    .NLIST  BEX
      016366  005015  041412  047117  55400    DMCER:  .ASCII  <15><12><12>/CONTROL OUT ERROR/
      016412  005015  020040  042523  55500            .ASCII  <15><12>/  SEL4     SEL6/
      016433     015  020012     000  55600            .ASCIZ  <15><12>/ /
      016437     015  005012  047523  55700    SOFT:   .ASCII  <15><12><12>/SOFT ERROR - DOCMP ERROR COUNTS ARE NON ZERO/
      016516  005015  020040  020040  55800            .ASCII  <15><12>/            BASE+3 THRU BASE+12/<15><12><0>
      016562  020040  000040          55900    SPACE3: .ASCIZ  /   /
      016566  005015  040515  052516  56000    BOOMSG: .ASCIZ  <15><12>/MANUALLY BOOT DMC NOW (VIA M9301-YJ)/
      016635     015  047412  044522  56100    ORGOK:  .ASCIZ  <15><12>/ORIGINATING STATION HAS COMPLETED BOOT SUCCESSFULLY - E
                 016742               56200            .EVEN
                                      56300            .LIST   BEX
1607                                  56400
1608  016742  000000                  56500    RESUME: 0
1609  016744  000000                  56600    BASEFG: 0                       ;BASE LOAD FLAG
1610  016746  000000                  56700    TEMP1:  0
1611  016750  000000                  56800    TEMP2:  0
1612  016752  000000                  56900    XFLAG:  0
1613  016754  000000                  57000    RFLAG:  0
1614  016756  000000                  57100    SFLAG:  0
1615  016760  000000                  57200    MFLAG:  0
1616  016762  000000                  57300    SEL4:   0
1617  016764  000000                  57400    SEL6:   0
1618                                  57500
1619  016766  000004                  57600    ERRCNT: .BLKW   4
1620                                  57700
1621                                  57800
1622  016776  017376                  57900    BASE:   .=.+256.
1623          000001                  58000    .END
```

```
AUTBOO  014540  1224  1284#  1285  1291  1293
AUTORG  014312  1222  1234#  1240
BR      011004   665#   726
BACK    013400   827*   856*   892*   986*  1017  1078#
BASE    016776   758*   759*   760*   761*   762*  1129  1370  1584  1593  1594  1622#
BASEFG  016744   745*  1090  1092*  1112  1114*  1149*  1157*  1164*  1182*  1239  1241*  1244  1246*
                1249  1251*  1284  1286*  1302  1304*  1307  1309*  1318*  1321  1323*  1326  1328*
                1336*  1339  1341*  1344  1346*  1373*  1379*  1386*  1391*  1441  1443*  1569*  1571
                1576*  1578  1609#
BIT0  = 000001   659#  1137  1175  1375  1402
BIT1  = 000002   659#   932   957   964  1123  1368
BIT10 = 002000   659#  1145  1147  1221  1223
BIT11 = 004000   659#  1145
BIT12 = 010000   659#  1133
BIT13 = 020000   659#   694#  1028  1055
BIT14 = 040000   659#   734  1314  1333  1573
BIT15 = 100000   659#
BIT2  = 000004   659#   937  1125  1143  1151  1206  1381  1429
BIT3  = 000010   659#
BIT4  = 000020   659#
BIT5  = 000040   659#
BIT6  = 000100   659#
BIT7  = 000200   659#
BIT8  = 000400   659#   730   765
BIT9  = 001000   659#  1179  1221  1225
BOOMSG  016566  1227  1606#
BOOT    014254   767  1221#
B2016   011030   675#  1031
CKBASE  016176   824   853   889   983  1568#
DELAY   013376   778*  1015*  1077#  1257*  1260*  1448*  1451*
DERR    013010  1056  1065#
DISPLA  011046   685#
DMCER   016366  1190  1414  1606#
DMC11   011000   664#
DSFLG = 020000   659#   693#
DSSTAT  011060   700#
ENTER   012550   999  1008#
EOP     012462   828   857   893   987   994#
ERCSR   011054   698#   752*  1188*  1191  1412*  1415
ERDBR   011056   699#   753*  1026  1189*  1196  1203*  1413*  1420  1427*
ERRCNT  016766   754*   755*   756*   757*  1584  1593*  1619#
FLAG    011042   683#
FULL.D= 000001   714#   934   959   966  1139
GO      011404   777#
IISR    015126   732  1368#
ILB   = 000010   659#   786
IRDA    011020   671#   896  1035  1058  1154  1271  1355  1383
IXDA    011022   672#   735   897  1034  1062  1064  1161
KBDIN = 104416   659#   808   840   873   931
LOOP  = 000020   659#   825   854   890   984
MAINORG 014604  1226  1300#
MFLAG   016760  1287*  1290  1300*  1331  1406*  1615#
MOP1    015602  1234  1253  1459#
MOP2    015610  1270  1354  1462#
MOP3    015614  1445  1446  1464#
MOP3ED  015770  1446  1492#
```

```
MP3     015506          1278    1362    1441#   1442
MSG     015676          1473    1481#
MSG0    013022          1030    1071#
MSG1    013103          1033    1071#
MSG2    013106          1057    1071#
MSG3    013133          1061    1071#
MSG4    013156          1071#   1087
MSG5    013310          1071#   1108
NODAT = 000200           659#    820     885     980
OISR    015312           733    1402#
ORGOK   016635          1454    1606#
OWI   = 000002           659#    783
OWO   = 000001           659#    780
PARAM1  011012           668#    730     765    1221    1223    1225    1235    1236    1237    1238
PARAM2  011014           669#    932     934     937     957     959     964     966    1139    1143
PARAM3  011016           670#
PRIOR   011010           667#    723     725
PRTY0 = 000000           659#
PRTY1 = 000040           659#
PRTY2 = 000100           659#
PRTY3 = 000140           659#
PRTY4 = 000200           659#
PRTY5 = 000240           659#
PRTY6 = 000300           659#
PRTY7 = 000340           659#    995
QTPIE   012620           996*    997*   1016    1018#
RCC     011070           705#
RDA     011072           706#
RDAX    012770          1058*   1060#
RESUME  016742           764*   1131    1177    1568*   1581*   1608#
RFLAG   016754           746*   1110    1116*   1247*   1258    1288*   1292    1310*   1329*   1347*   1349    1431*   1613#
RFLG  = 040000           692#    810     823     876     888     941     948     978    1153    1208
RISR    014032           722     729    1175#
RIV     011006           666#    721     727
RWAIT = 104410           659#
RXWC    011064           703#    742*   1155
RX.TER  011041           681#
SAVR0   013362          1000*   1009    1071#
SAVR1   013364          1001*   1010    1072#
SAVR2   013366          1002*   1011    1073#
SAVR3   013370          1003*   1012    1074#
SAVR4   013372          1004*   1013    1075#
SAVR5   013374          1005*   1014    1076#
SCAN4   012666          1036#   1037    1050    1053
SCAN5   012742          1054#
SEL4    016762          1253*   1388    1445*   1616#
SEL6    016764          1242*   1254*   1305*   1324*   1342*   1377    1389    1446*   1617#
SETTLE  011024           673#
SFLAG   016756          1301*   1312    1317*   1410*   1614#
SOFT    016437          1590    1606#
SPACE3  016562          1195    1419    1599    1606#
SRCSR   011052           697#
START   011106           678     718#   1457
STARTR  013474           809     874     936     962    1106#
STARTX  013404           841     906     944    1085#
STOP    013402           779*   1079#   1085    1089*   1106    1109*
```

```
STPS  = 104414   659# 995
SWR     011044   684# 719  820  825  854  885  890  894  980  984  1028 1055 1200
                 1424
SWI2  = 010000   659#
SXCSR   011050   696#
TEMP1   016746  1093* 1095* 1256* 1262* 1610#
TEMP2   016750   763*  939  943* 1094* 1097* 1611#
TESTD   012622   822   887  982  1026#
TESTDX  013016  1039  1045 1048 1068#
TIE   = 000100   659#  997  998
TIME    011032   676#  777* 812  817* 842* 845  850* 875* 878  883* 905* 909  914#
                 945*  950  955* 970  975* 979*
TKB     011100   710#
TKS     011076   709#
TPB     011104   712#
TPS     011102   711#
TSTDAT  012656  1027  1029 1034#
TXWC    011062   702#  740* 1162
TX.TER  011040   679#  738  1038
TYPE  = 104400   659#  1030 1033 1043 1057 1059 1061 1063 1087 1108 1190 1195 1227
                1414  1419 1454 1550 1590 1599
XCC     011066   704#
XCSR  = 000002   659#  815  848  881  912  953  973  996  998* 1016*
XDA     011074   707#  898*
XFLAG   016752   747*  1160* 1212* 1252* 1264 1433* 1444* 1449 1612#
XFLG  = 100000   691#  843  852  907  916  946  961  968  977  1159 1211
XISR    013544   724   728  1123#
XLB   = 000004   659#  789
XWAIT = 104412   659#
$B20CT  015770  1192  1197 1416 1421 1517# 1596
$B201   016016  1519  1522#
$B2016  016002  1520#
$ILB    011710   788   873#  917
$OCNT   016172  1522* 1551* 1563#
$OMODE  016174  1517* 1521* 1526 1529* 1540* 1565#
$OWI    011474   785   808#  829
$OWO    011606   782   840#  858
$XLB    012144   791   931#  985  986
$OFILL  016173  1518* 1520* 1530 1564#
.     = 017376   663#  793  900  942  1062* 1096 1098 1168 1201 1228 1395 1425 1461#
                1473  1480 1606# 1619# 1622#
```

```
BOX          1#    660     687     715     1081    1102    1119    1171    1217    1230    1280    1296    1364    1398    1437
DCPARM       1#
DHDOC1       1#
DHPARM       1#
DJPARM       1#
DLPARM       1#
DMPARM       1#    555
DPPARM       1#
DQDOC1       1#
DQPARM       1#
DUPARM       1#
DUPPAR       1#
DVDOC1       1#
DVPARM       1#
DZPARM       1#
HELLO        1#
HLT        659#    816     849     882     913     954     974     1067
SEQUAT       1#    659
SINTF        1#    659
SITEP        1#    770
SSERV        1#    704

. ABS.  017376     000
ERROPS DETECTED:  0

DZDMO,DZDMO/SOL/CRF+ITEP1,DZDMO
RUN-TIME: 4 5 .3 SECONDS
RUN-TIME RATIO: 104/10=9.9
CORE USED:  17K  (33 PAGES)
```