

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

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION

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(OZITP) 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 DM118B, 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 DFFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
  - A. TYPE A CAR, RETURN TO USE DFFAULT BUS ADDRESS
  - B. TYPEIN ACTUAL BUS ADDRESS
- 3. THE PROGRAM WILL TYPE OUT THE DFFAULT VECTOR ADDRESS
  - A. TYPE A CAR, RETURN TO USE DFFAULT ADDRESS
  - B. TYPEIN ACTUAL VECTOR ADDRESS
- 4. THE PROGRAM WILL TYPE OUT THE DFFAULT INTERFACE PRIORITY  
NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.
  - A. TYPE A CAR, RETURN TO USE DFFAULT VALUE

- B. TYPEIN 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 INTRFACE
    - B. HALT FOR VECTOR ADDRESS OF INTERFACE
    - C. HALT FOR PRIORITY OF INTEPFACE
    - 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 TIMEOUTS  
SW12=1 INHIBIT ALL TIMEOUTS 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 PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176 ) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPEATOR 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 CHAPACTER(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 OPEATER, THE TERMINATING CHARACTERS ARE ADDED.

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERIFIES 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 MANUALLY 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=RRRRR  
DATA SHOULD BE TTTTTT  
DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG

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

TTTTT 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 INITIALIZE 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 CP  
AN IMMEDIATE ERROR MESSAGE FROM MAYNARD WILL BE RE-  
CEIVED, THIS IS BECAUSE THE TEST STATION IS STILL LOOKING

FOR THE REST OF THE INTERRUPTED 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 ELAPSED 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 \$XLR DEPENDING  
ON THE SWITCH SETTINGS.

- 9.4 THE RETURN TO MONITOR ROUTINE FOR END PASS AT EOP:  
LOCKS OUT INTERRUPTS AND SAVES THE TRANSMITTER INTERRUPT ENABLE  
BIT AND ALL 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,ILR,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 XISR1, 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 RISR1, 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 (MAX IN 1000 CHARACTER

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 ADDITION 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-AUTOMATIC 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 ON LINE LINK TEST FOR TWO DMC11s. DMC ITEP WILL 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.

```

659
660
661
662
663
664 A11000 000103
665 211004 100010
666 011006 000300
667 011010 000210
668 011012 000000
669 011014 000001
670 011016 177777
671 011020 000000
672 011022 000000
673 011024 000000
674 011026 000000
675 011030 000000
676 011032 000000
677 011034 000000
678 011036 011106
679 011040
680 011040 000
681 011041
682 011041 001
683 011042 000000
684 011044 177570
685 011046 177570
686
687
688
689
690 000000
691 100000
692 040000
693 020000
694 020000
695
696 011050 000000
697 011052 000000
698 011054 000000
699 011056 000000
700 011060 000000
701
702 011062 000000 01400
703 011064 000000 01500
704 011066 000000
705 011070 000000
706 011072 000000
707 011074 000000
708
709 011076 177560
710 011100 177562
711 011102 177564
712 011104 177566
713
714 000001

```

```

;*****
; DMC11 INTERFACE SERVICE PARAMS
;*****
DMC11:  =11000 /DMC/
BA: 160010 ;ISR NAME
RIV: 300 ;BUS ADDRESS
PRIOR: 240 ;VECTOR ADDRESS
PARAM1: 0 ;PRIORITY
PARAM2: 1 ;PARAM #1
PARAM3: 177777 ;PARAM #2
IRDA: .WORD 0 ;PARAM #3
IXDA: .WORD 0 ;INITIAL READ DATA ADDRESS
SETTLE: .WORD 0 ;INITIAL XMIT DATA ADDRESS
;LINE SETTLE DELAY FLAG
B2016: .WORD 0 ;
TIME: .WORD 0 ;ADDR OF BIN TO OCT TYPE ROUTINE
;TIMER
TX_TERM: .WORD START ;ADDR OF START OF PROGRAM
;BYTE 000 ;TRANSMITTER TERMINATING CHAR.
RX_TERM: .BYTE 001 ;RECEIVER TERMINATING CHAR.
FLAG: .WORD 0
SWR: 177570
DISPLAY:177570
;*****
; CONSTANTS + WORKING STORAGE
;*****
STAT=R0
XFLG=10000 ;XMIT COMPLETE FLAG
RFLG=40000 ;RCV COMPLETE FLAG
DSFLG=20000 ;DATA SET STATUS CHANGE FLAG
BIT13=20000 ;INHIBIT PRINTOUTS
SXCSR: 0 ;SAVED XMIT CSR
SRCSR: 0 ;SAVED RCV CSR
ERCSR: 0 ;RCV CSR SAVED ON ERROR
ERDBR: 0 ;RCV DATA REG SAVED ON ERROR
DSSTAT: 0 ;RCV CSR SAVED ON DS CHANGE
TXWC: 0
RXWC: 0
XCC: 0 ;XMIT CHAR COUNT
RCC: 0 ;RCV CHAR COUNT
RDA: 0 ;RCV DATA ADDR.
XDA: 0 ;XMIT DATA ADDR.
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
FULL_DUPLEX=000001

```

```

715
716
717
718 011106 000240
719 011110 017700 177730
720 011114 042700 177400
721 011120 013702 011006
722 011124 012722 014032
723 011130 013722 011010
724 011134 012722 013544
725 011140 013722 011010
726 011144 013704 011004
727 011150 013702 011006 01700
728 011154 012712 013544 01800
729 011160 012762 014032 000004 01900
730 011166 032737 000400 011012 02000
731 011174 001405 02100
732 011176 012712 015126 02200
733 011202 012762 015312 000004 02300
734 011210 012714 040000 02400 30:
735 011214 013702 011022 02500
736 011220 005003 02600
737 011222 005203 02700 10:
738 011224 123722 011040 02800
739 011230 001374 02900
740 011232 010337 011062 03000
741 011236 062703 000010 03100
742 011242 010337 011064 03200
743 011246 005714 03300 20:
744 011250 100376 03400
745 011252 005037 016744 03500
746 011256 005037 016754 03600
747 011262 005037 016752 03700
748 011266 005714 03800 40:
749 011270 100376 03900
750 011272 052764 000100 000002 04000
751 011300 052714 000143 04100
752 011304 005037 011054 04200
753 011310 005037 011056 04300
754 011314 005037 016766 04400
755 011320 005037 016770 04500
756 011324 005037 016772 04600
757 011330 005037 016774 04700
758 011334 105037 017001 04800
759 011340 005037 017002 04900
760 011344 005037 017004 05000
761 011350 005037 017006 05100
762 011354 105037 017010 05200
763 011360 005037 016750 05300
764 011364 005037 016742 05400
765 011370 032737 000400 011012 05500
766 011376 001402 05600
767 011400 000137 014254 05700
768 011404 05800 50:
769
770

```

```

;*****
; DMC11-X INTERFACE SERVICE ROUTINE
;*****
START: NOP
MOV 0SWR, R0 ;SETUP MODE IN R0
BIC #177400, R0 ;STRIP JUNK
MOV RIV, R2 ;SETUP
MOV #RISR, (R2)+ ;INTERRUPT
MOV #PRIOR, (R2)+ ;VECTORS
MOV #XISR, (R2)+ ;
MOV #PRIOR, (R2)+ ;
BA, R4 ;SETUP BUS ADDR INDEX
RIV, R2 ;ADJUST VECTORS FOR
MOV #XISR, (R2) ;INPUT SERVICE ROUTINE
MOV #RISR, 4(R2) ;OUTPUT SERVICE ROUTINE
BIT #BIT0, PARAM1 ;BOOT MODE?
BEQ 30 ;BR IF NO
MOV #IISR, (R2) ;LOAD BOOT VECTORS
MOV #OISR, 4(R2) ;
MOV #BIT14, (R4) ;MASTER CLEAR DMC11
MOV IXDA, R2 ;CALCULATE WORD COUNTS
CLR R3 ;CLEAR COUNT
INC R3 ;INC COUNT
CMPB TX_TERM, (R2)+ ;LAST CHARACTER?
BNE 10 ;BR IF NO
MOV R3, TXWC ;STORE XMIT COUNT
ADD #0, R3 ;ADD 10 TO IT
MOV R3, RXWC ;STORE REC COUNT
TST (R4) ;WAIT FOR RUN
BPL 20 ;BR IF RUN NOT SET
CLR BASEFG ;CLEAR BASE LOAD FLAG
CLR RFLAG ;CLEAR RECEIVE FLAG
CLR XFLAG ;CLEAR XMIT FLAG
TST (R4) ;RUN SET?
BPL 40 ;BR IF NO
BIS #100, 2(R4) ;SET OIE
BIS #143, (R4) ;ASK FOR BASE TRANSFER
CLR ERCSR ;CLEAR ERROR LOCATIONS
CLR ERDBR
CLR ERRCNT ;CLEAR SOFT ERROR STORAGE
CLR ERRCNT+2 ;LOCATIONS
CLR ERRCNT+4
CLR ERRCNT+6
CLRB BASE+3 ;CLEAR BASE ERROR COUNT LOCATIONS
CLRB BASE+4
CLRB BASE+6
CLRB BASE+10
CLRB BASE+12
CLR TEMP2
CLR RESUME ;CLEAR RESUME FLAG
BIT #BIT0, PARAM1 ;BOOT MODE?
BEQ 50 ;BR IF NO
JMP BOOT ;GO TO START OF BOOT CODE
50:

```

```

771 ;*****
772 ; ROUTINE USED TO GOTO
773 ; SUBROUTINE DEPENDENT
774 ; ON MODE SELECTED.
775 ;*****
776
777 011404 005037 011032 GO: CLR TIME
778 011410 005037 013376 CLR DELAY
779 011414 005037 013402 CLR STOP
780 011420 032700 000001 BIT #0W0,MODE
781 011424 001402 BEQ 10
782 011426 000137 011606 JMP #0W0
783 011432 032700 000002 10: BIT #0W1,MODE
784 011436 001402 BEQ 20
785 011440 000137 011474 JMP #0W1
786 011444 032700 000010 20: BIT #1LB,MODE
787 011450 001402 BEQ 30
788 011452 000137 011710 JMP #1LB
789 011456 032700 000004 30: BIT #XLB,MODE
790 011462 001402 BEQ 40
791 011464 000137 012144 JMP #XLB
792 011470 000000 40: HALT
793 011472 000076 BR -2
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808 011474 104416 #0W1: KBDIN
809 011476 004737 013474 JSR PC,STARTR
810 011502 032700 000000 10: BIT #RFLG,STAT
811 011506 001013 BNE 20
812 011510 023727 011032 000100 CMP TIME,#100
813 011516 103771 BLO 10
814 011520 011402 MOV #RCSR,R2
815 011522 016403 000002 MOV XCSR(R4),R3
816 011526 104001 HLT 1
817 011530 005037 011032 CLR TIME
818 011534 000762 BR 10
819
820 011536 032777 000200 177300 20: BIT #NODAT,#SWR
821 011544 001002 BNE 30
822 011546 004737 012622 JSR PC,TESTD
823 011552 042700 000000 30: BIC #RFLG,STAT
824 011556 004737 016176 JSR PC,CKBASE ;CHECK DMC SOFT ERROR COUNTERS
825 011562 032777 000020 177254 BIT #LOOP,#SWR
826 011570 001405 BEQ 40

```

```

827 011572 012737 011604 013400 MOV #40,BACK
828 011600 000137 012462 40: JMP EOP
829 011604 000733 BR #0W1
830
831
832 ;*****
833 ; ROUTINE USED IF "ONE WAY OUT" WAS SELECTED,
834 ; NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE ONLY
835 ; MODE AVAILABLE,
836 ; "ONE WAY OUT" MEANS THAT ONLY THE TRANSMITTER IS
837 ; ENABLED, THE RECEIVER IS NEVER "TURNED ON".
838 ;*****
839
840 011606 104416 #0W0: KBDIN
841 011610 004737 013404 JSR PC,STARTX
842 011614 005037 011032 CLR TIME
843 011620 032700 100000 10: BIT #XPLG,STAT
844 011624 001013 BNE 20
845 011626 023727 011032 000100 CMP TIME,#100
846 011634 103771 BLO 10
847 011636 011402 MOV #RCSR,R2
848 011640 016403 000002 MOV XCSR(R4),R3
849 011644 104001 HLT 1
850 011646 005037 011032 CLR TIME
851 011652 000762 BR 10
852 011654 042700 100000 20: BIC #XPLG,STAT
853 011660 004737 016176 JSR PC,CKBASE ;CHECK DMC SOFT ERROR COUNTERS
854 011664 032777 000020 177152 BIT #LOOP,#SWR
855 011672 001405 BEQ 30
856 011674 012737 011706 013400 MOV #30,BACK
857 011702 000137 012462 30: JMP EOP
858 011706 000737 BR #0W0
859
860
861

```

862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873 011710 104416  
874 011712 004737 013474  
875 011716 005037 011032  
876 011722 032700 040000  
877 011726 001013  
878 011730 023727 011032 000100  
879 011736 103771  
880 011740 011402  
881 011742 016403 000002  
882 011746 104001  
883 011750 005037 011032  
884 011754 000762  
885 011756 032777 000200 177060  
886 011764 001002  
887 011766 004737 012622  
888 011772 042700 040000  
889 011776 004737 016176  
890 012002 032777 000020 177034  
891 012010 001405  
892 012012 012737 012024 013400  
893 012020 000137 012462  
894 012024 032777 000400 177012  
895 012032 001416  
896 012034 013702 011020  
897 012040 013703 011022  
898 012044 010337 011074  
899 012050 112223  
900 012052 001376  
901 012054 112743 000177  
902 012060 005203  
903 012062 112723 000177  
904 012066 105023  
905 012070 005037 011032  
906 012074 004737 013404  
907 012100 032700 100000  
908 012104 001013  
909 012106 023727 011032 000100  
910 012114 103771  
911 012116 011402  
912 012120 016403 000002  
913 012124 104001  
914 012126 005037 011032  
915 012132 000762  
916 012134 042700 100000  
917 012140 000137 011714

```

;*****
; 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 COMPLETE 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 REPEATED AS ABOVE.
;*****
81LB: KBDIN
      JSR   PC,STARTR
      CLR   TIME
18:   BIT   #RFLG,STAT
      BNE   20
      CMP   TIME,#100
      BLO   18
      MOV   @RCSR,R2
      MOV   XCSR(R4),R3
      HLT   1
      CLR   TIME
      BR    18
28:   BIT   #NODAT,@SWR
      BNE   30
      JSR   PC,TESTD
      BIC   #RFLG,STAT
      JSR   PC,CKBASE ;CHECK DMC SOFT ERROR COUNTERS
      BIT   #LOOP,@SWR
      BEQ   40
      MOV   #40,BACK
      JMP   EOP
46:   BIT   #400, @SWR ;USE EXTERNAL DATA?
      BEQ   70 ;BR IF NO
      MOV   IRDA, R2 ;SET POINTER
      MOV   IXDA, R3 ;SET POINTER
      MOV   R3, .XDA ;SETUP XMIT DATA ADDR
      MOVB (R2)+, (R3)+ ;MOVE INPUT TO OUTPUT
      BNE   .-2 ;LOOP IF NOT ZERO CHAR
      MOVB #177, -(R3) ;INSERT A FILL CHAR
      INC   R3 ;BUMP ADDRESS
      MOVB #177, (R3)+ ;INSERT ANOTHER FILL
      CLRB (R3)+ ;INSERT ZERO CHAR
78:   CLR   TIME
      JSR   PC,STARTR
      BIT   #XFLG,STAT
58:   BNE   60
      CMP   TIME,#100
      BLO   58
      MOV   @RCSR,R2
      MOV   XCSR(R4),R3
      HLT   1
      CLR   TIME
      BR    58
66:   BIC   #XFLG,STAT
      JMP   61LB
```

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931 012144 104416  
932 012146 032737 000002 011014  
933 012154 001004  
934 012156 032737 000001 011014  
935 012164 001417  
936 012166 004737 013474  
937 012172 037737 000004 011014  
938 012200 001411  
939 012202 005737 016750  
940 012206 001403  
941 012210 032700 040000  
942 012214 001774  
943 012216 012737 177777 016750  
944 012224 004737 013404  
945 012230 005037 011032  
946 012234 032700 100000  
947 012240 001016  
948 012242 032700 040000  
949 012246 001030  
950 012250 023727 011032 000100  
951 012256 103766  
952 012260 011402  
953 012262 016403 000002  
954 012266 104001  
955 012270 005037 011032  
956 012274 000757  
957 012276 032737 000002 011014  
958 012304 001356  
959 012306 032737 000001 011014  
960 012314 001352  
961 012316 042700 100000  
962 012322 004737 013474  
963 012326 000742  
964 012330 032737 000002 011014  
965 012336 001004  
966 012340 032737 000001 011014  
967 012346 001420  
968 012350 032700 100000  
969 012354 001013  
970 012356 023727 011032 000100  
971 012364 103761  
972 012366 011402  
973 012370 016403 000002

```

;*****
; 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.
;*****
8XLB: KBDIN
      BIT   #BIT1,PARAM2 ;SECONDARY MODE?
      BNE   90 ;BR IF YES
      BIT   #FULL,DUPLEX,PARAM2
      BEQ   10
98:   JSR   PC,STARTR
      BIT   #BIT2,PARAM2 ;SECONDARY STATION?
      BEQ   18 ;BR IF NO
      TST   TEMP2 ;FIRST TIME HERE?
      BEQ   18 ;BR IF YES
      BIT   #RFLG,STAT ;WAIT FOR RECEIVE BEFORE
      BEQ   .-6 ;TRANSMITTING
      MOV   #-1,TEMP2 ;SET FIRST TIME FLAG
118:  JSR   PC,STARTR
18:   CLR   TIME
28:   BIT   #XFLG,STAT
      BNE   30
78:   BIT   #RFLG,STAT
      BNE   40
      CMP   TIME,#100
      BLO   28
      MOV   @RCSR,R2
      MOV   XCSR(R4),R3
      HLT   1
      CLR   TIME
      BR    28
36:   BIT   #BIT1,PARAM2 ;SECONDARY MODE?
      BNE   78 ;BR IF YES
      BIT   #FULL,DUPLEX,PARAM2
      BNE   70
      BIC   #XFLG,STAT
      JSR   PC,STARTR
      BR    28
48:   BIT   #BIT1,PARAM2 ;SECONDARY MODE?
      BNE   100 ;BR IF YES
      BIT   #FULL,DUPLEX,PARAM2
      BEQ   90
108:  BIT   #XFLG,STAT
      BNE   68
      CMP   TIME,#100
      BLO   48
      MOV   @RCSR,R2
      MOV   XCSR(R4),R3
```

```

074 #12374 104001          HLT      1
075 #12376 035037 #11032    CLR     TIME
076 #12402 #00752          BR      40
077 #12404 042700 100000    60:    BIC     #XPLG,STAT
078 #12410 042700 040000    60:    BIC     #RPLG,STAT
079 #12414 005037 011032    CLR     TIME
080 #12420 032777 000200 176416  BIT     #NODAT,#SWP
081 #12420 001002          BNE     50
082 #12430 004737 012622    JSR     PC,TESTD
083 #12434 004737 016176    JSR     PC,CKBASE ;CHECK DMC SOFT EPROR COUNTERS
084 #12440 032777 000020 176376  BIT     #LOOP,#SWR
085 #12446 001636          REQ     #XLB
086 #12450 012737 012144 013400  MOV     #XLB,BACK
087 #12456 000137 012462    JMP     EOP

```

```

988 ;*****
989 ; ROUTINE TO RETURN
990 ; TO MONITOR FOR
991 ; END PASS.
992 ;*****
993
994 #12462          EOP:
995 #12462 104414 000340          STPS,PRTY7 ;SET PS PRIORITY TO 7
996 #12466 016437 000002 012620  MOV     XCSR(R4),QTPIE ;SAVE TX CSR
997 #12474 042737 177077 012620  BIC     #*C*IE*,QTPIE ;CLEAR ALL BUT TX IE.
998 #12502 042764 000100 000002  BIC     #IE,XCSR(R4) ;CLEAR TX IE (EVEN IF IT WASN'T SET)
999 #12510 012766 012550 000002  MOV     #ENTER,2(SP) ;SET FOR RETURN IF SW 14=1
1000 #12516 010037 013302          MOV     R0,SAVR0 ;SAVE REGISTER 0
1001 #12522 010137 013304          MOV     R1,SAVR1 ;SAVE REGISTER 1
1002 #12526 010237 013306          MOV     R2,SAVR2 ;SAVE REGISTER 2
1003 #12532 010337 013370          MOV     R3,SAVR3 ;SAVE REGISTER 3
1004 #12536 010437 013372          MOV     R4,SAVR4 ;SAVE REGISTER 4
1005 #12542 010537 013374          MOV     R5,SAVR5 ;SAVE REGISTER 5
1006 #12546 000207          RTS     PC ;RETURN TO CONTROL PROGRAM
1007
1008 #12550          ENTER:
1009 #12550 013700 013302          MOV     SAVR0,R0 ;RESTORE R0
1010 #12554 013701 013304          MOV     SAVR1,R1 ;RESTORE R1
1011 #12560 013702 013306          MOV     SAVR2,R2 ;RESTORE R2
1012 #12564 013703 013370          MOV     SAVR3,R3 ;RESTORE R3
1013 #12570 013704 013372          MOV     SAVR4,R4 ;RESTORE R4
1014 #12574 013705 013374          MOV     SAVR5,R5 ;RESTORE R5
1015 #12600 012737 177777 013376  MOV     #=1,DELAY
1016 #12606 053764 012620 000002  BIS     QTPIE,XCSR(R4) ;IF ORGINALLY SET; SET TX IE
1017 #12614 000177 000560          JMP     @BACK
1018 #12620 000000          QTPIE: 000000
1019
1020
1021
1022 ;*****
1023 ; SUBROUTINE TO CHECK
1024 ; RECEIVER DATA.
1025 ;*****
1026 #12622 013746 011056          TESTD: MOV     ERDBR, -(SP) ;WAS THERE A RECEIVE ERROR?
1027 #12626 001413          BEQ     TSTDAT ;BR IF NO
1028 #12630 032777 020000 176206  BIT     #BIT1,#SWR ;INHIBIT PRINTOUTS?
1029 #12636 001007          BNE     TSTDAT ;BR IF YES
1030 #12640 104400          TYPE   ,MSG0 ;<15><12>THERE WAS A RECEIVE ERROR, RBUF=
1031 #12644 004077 176100          JSR     R0,@02016 ;PRINT CONTENTS OF RBUF
1032 #12650 005746          TST    -(SP)
1033 #12652 104400          TYPE   ,MSG1 ;<15><12>
1034 #12656 013701 011022          TSTDAT: MOV     IXDA, R1 ;SETUP XMIT DATA ADDR
1035 #12662 013702 011020          MOV     IRDA, R2 ;SETUP RCV DATA ADDR
1036 #12666 122122          SCAN4: CMPB   (R1)+,(R2)+ ;DATA OK ?
1037 #12670 001776          BEQ     SCAN4 ;BR IF OK
1038 #12672 123741 011040          CMPB   TX,TERM,-(R1) ;IS IT END OF DATA
1039 #12676 001447          BEQ     TESTDX ;BR IF YES
1040 #12700 122742 000002          CMPB   #02,-(R2)
1041 #12704 001005          BNE     24
1042 #12706 010237 012714          MOV     R2,#
1043 #12712 104400          TYPE

```

1044 012714 000000  
1045 012716 000437  
1046 012720  
1047 012720 005712  
1048 012722 001435  
1049 012724 122721 000177  
1050 012730 001756  
1051 012732 005301  
1052 012734 122722 000177  
1053 012740 001752  
1054 012742 000240  
1055 012744 032777 020000 176072  
1056 012752 001016  
1057 012754 104400 013106  
1058 012760 013737 011020 012770  
1059 012766 104400  
1060 012770 000000  
1061 012772 104400 013133  
1062 012776 013737 011022 013006  
1063 013004 104400  
1064 013006 011022  
1065 013010 111103  
1066 013012 114202  
1067 013014 104007  
1068 013016 005726  
1069 013020 000207  
1070  
1071 013022 005015 044124 051105  
(1) 013103 015 000012  
(1) 013106 005015 042522 042503  
(1) 013133 015 042012 052101  
(1) 013156 005215 046124 040505  
(1) 013225 015 053412 042510  
(1) 013310 005015 046120 040505  
(1)  
(1) 013362 000000  
1072 013364 000000  
1073 013366 000000  
1074 013370 000000  
1075 013372 000000  
1076 013374 000000  
1077 013376 000000  
1078 013400 000000  
1079 013402 000000  
1080

10: .WORD 0  
BP TFSTDX  
28: TSTB (R2) ;  
BEQ TESTDX ;BR IF YES  
CMPB #177, (R1)+ ;IS IT FILL CHAR?  
BEQ SCAN4 ;BR IF YES  
DEC R1 ;BACKUP  
CMPB #177, (R2)+ ;IS IT FILL?  
BEQ SCAN4 ;BR IF YES  
SCAN5: NOP ;DATA ERROR  
BIT #BIT13,05WR ;INHIBIT PRINTOUTS  
BNE DERR ;BR IF YES  
TYPE ,MSG2 ;<15><12>RECEIVED DATA = <15><12>  
MOV IRDA, RDAX ;SETUP DATA ADDRESS  
TYPE ;PRINT RECEIVED DATA  
RDAX: 0 ;RECEIVED DATA ADDR.  
TYPE ,MSG3 ;<15><12>DATA SHOULD BE<15><12>  
MOV IXDA, .+10 ;SETUP ADDR.  
TYPE ;PRINT GOOD DATA  
DERR: MOVB (R1),R3 ;SETUP XMIT DATA  
MOVB =(R2),R2 ;SETUP RCV DATA  
HLT+7 ;DATA ERROR HALT  
TESTDX: TST (SP)+ ;POP STACK  
RTS PC ;RETURN FROM SUB/ROUT  
MSG0: .ASCIZ <15><12>/THERE WAS A RECEIVER ERROR, REGISTER (SEL 2) =/  
MSG1: .ASCIZ <15><12>  
MSG2: .ASCIZ <15><12>/RECEIVED DATA = /<15><12>  
MSG3: .ASCIZ <15><12>/DATA SHOULD BE/<15><12>  
MSG4: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER),/  
.ASCIZ <15><12>/WHEN CONNECTION COMPLETE, HIT CONTINUE SWITCH,/<15><12>  
MSG5: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER),/<15><12>  
.EVEN  
SAVR0: 0  
SAVR1: 0  
SAVR2: 0  
SAVR3: 0  
SAVR4: 0  
SAVR5: 0  
DELAY: 0  
BACK: 0  
STOP: 0

1081  
1082  
1083  
1084  
1085 013404 005737 013402 06300  
1086 013410 001005 06400  
1087 013412 104400 013156 06500  
1088 013416 000000 06600  
1089 013420 005137 013402 06700  
1090 013424 005737 016744 06800  
1091 013430 001775 07000  
1092 013432 005037 016744 07100  
1093 013436 005037 016746 07200  
1094 013442 012737 000025 016750 07300  
1095 013450 062737 000001 016746 07400  
1096 013456 001374 07500  
1097 013460 005337 016750 07600  
1098 013464 001371 07700  
1099 013466 152714 000140 07800  
1100 013472 000207 07900  
1101 08000  
1102  
1103  
1104  
1105  
1106 013474 005737 013402 08200  
1107 013500 001004 08300  
1108 013502 104400 013310 08400  
1109 013506 005137 013402 08500  
1110 013512 005737 016754 08600  
1111 013516 001007 08700  
1112 013520 005737 016744 08800  
1113 013524 001772 08900  
1114 013526 005037 016744 09000  
1115 013532 152714 000144 09100  
1116 013536 005037 016754 09200  
1117 013542 000207 09300  
1118 09400  
1119 09500  
1120  
1121  
1122  
1123 013544 032714 000002 09700  
1124 013550 001426 09800  
1125 013552 032714 000004 09900  
1126 013556 001403 10000  
1127 013560 004737 014020 10100  
1128 013564 000002 10200  
1129 013566 012764 016776 000004 10300  
1130 013574 005064 000006 10400  
1131 013600 005737 016742 10500  
1132 013604 001403 10600  
1133 013606 012764 010000 000006 10700  
1134 013614 004737 014020 10800  
1135 013620 152714 000141 10900  
1136 013624 000002 11000

\*\*\*\*\*  
; TRANSMIT INIT ROUTINE  
\*\*\*\*\*  
STARTX: TST STOP ;FIRST TIME HERE?  
BNE 10 ;BR IF NOT  
TYPE ,MSG4 ;TYPE CONNECT MESS  
HALT  
COM STOP ;SET FIRST TIME FLAG  
10: TST BASEFG ;BASE AND CNTL IN ALL DONE?  
BEQ 10 ;BR IF NO  
CLR BASEFG ;CLEAR FLAG  
TEMP1 ;GET SET TO DELAY  
MOV #25,TEMP2 ;  
ADD #1,TEMP1 ;INC DELAY  
BNE ,=6 ;  
DEC TEMP2 ;DEC DELAY COUNTER  
BNE ,=-14 ;BR IF NOT DONE  
BISB #140,(R4) ;ASK FOR XMIT BUFFER  
RTS PC ;RETURN  
\*\*\*\*\*  
; RECEIVE INIT ROUTINE  
\*\*\*\*\*  
STARTR: TST STOP ;FIRST TIME HERE?  
BNE 10 ;BR IF NOT  
TYPE ,MSG5 ;TYPE CONNECT MESS  
COM STOP ;SET FIRST TIME FLAG  
10: TST RFLAG ;HAS AN REC BUFFER ALREADY BEEN GIVEN  
BNE 20 ;BR IF YES  
TST BASEFG ;BASE AND CNTL IN ALL DONE?  
BEQ 10 ;BR IF NO  
CLR BASEFG ;CLEAR FLAG  
BISB #144,(R4) ;ASK FOR REC BUFFER  
20: CLR RFLAG ;CLEAR FLAG  
RTS PC ;RETURN  
\*\*\*\*\*  
; INPUT SERVICE ROUTINE  
\*\*\*\*\*  
XISR: BIT #BIT1,(R4) ;BASE REQUEST?  
BEQ 10 ;BR IF NO  
BIT #BIT2,(R4) ;IS IT REALLY A SHUT DOWN?  
BEQ 90 ;BR IF NO  
JSR PC,40 ;YES, CLEAR RQI  
RTI ;AND RETURN  
90: MOV #BASE,4(R4) ;LOAD BASE ADDRESS  
CLR 6(R4) ;CLEAR SEL 6  
TST RESUME ;RESUME FLAG SET?  
BEQ 80 ;BR IF NOT  
MOV #BIT12,6(R4) ;OTHERWISE SET RESUME BIT  
80: JSR PC,40 ;CLEAR RQI  
BISB #141,(R4) ;ASK FOR CNTL I  
RTI ;RETURN

```

1137 013626 032714 000001 11200 151 BIT #BIT0,(R4) ;CNTL 1 REQUEST?
1138 013632 001430 000001 11300 151 BEQ 20 ;BR IF NO
1139 013634 032737 000001 011014 11400 151 BIT #FULL,DUPLEX,PARAM2;FULL OR HALF?
1140 013642 031403 000001 11500 151 BEQ 50 ;BR IF HALF
1141 013644 005064 000000 11600 151 CLR 6(R4) ;SELECT FULL DUPLEX
1142 013650 000413 000001 11700 151 BR 50 ;CONTINUE
1143 013652 032737 000004 011014 11800 501 BIT #BIT2,PARAM2 ;SECONDARY STATION?
1144 013660 001404 000001 11900 151 BEQ 70 ;BR IF NO
1145 013662 012764 000000 000006 12000 151 MOV #BIT10;BIT11,6(R4);SET SEC MODE AND HALF DUPLEX
1146 013670 000403 000001 12100 151 BR 60 ;CONTINUE
1147 013672 012764 002000 000006 12200 701 MOV #BIT10,6(R4) ;SELECT HALF DUPLEX
1148 013700 004737 014020 12300 601 JSR PC,40 ;CLEAR RQI
1149 013704 012737 177777 016744 12400 151 MOV #=-1,BASEFG ;SET BASE LOADED FLAG
1150 013712 000002 12500 151 RTI ;RETURN
1151 013714 032714 000004 12600 201 BIT #BIT2,(R4) ;RECEIVE BUFFER REQUEST?
1152 013720 001416 000001 12700 151 BEQ 30 ;BR IF NO
1153 013722 042700 000000 12800 151 BIC #RFLG,STAT ;CLEAR RECEIVE FLAG BIT
1154 013726 013764 011020 000004 12900 151 MOV IRDA,4(R4) ;LOAD REC BUFFER ADDRESS
1155 013734 013764 011064 000006 13000 151 MOV RXWC,6(R4) ;LOAD REC BYTE COUNT
1156 013742 004737 014020 13100 151 JSR PC,40 ;CLEAR RQI
1157 013746 012737 177777 016744 13200 151 MOV #=-1,BASEFG ;SET FLAG
1158 013754 000002 13300 151 RTI ;RETURN
1159 013756 042700 100000 13400 301 BIC #XPLG,STAT ;CLEAR XMIT FLAG BIT
1160 013762 012737 177777 016752 13500 151 MOV #=-1,XFLAG ;SET XMIT FLAG(FOR TIMEOUT ERROR)
1161 013770 013764 011022 000004 13600 151 MOV IXDA,4(R4) ;LOAD XMIT BUFFER ADDRESS
1162 013776 013764 011062 000006 13700 151 MOV TXWC,6(R4) ;LOAD XMIT BYTE COUNT
1163 014004 004737 014020 13800 151 JSR PC,40 ;CLEAR RQI
1164 014010 012737 177777 016744 13900 151 MOV #=-1,BASEFG ;SET FLAG
1165 014016 000002 14000 151 RTI ;RETURN
1166 014020 142714 000040 14100 401 BICB #40,(R4) ;CLEAR RQI
1167 014024 140714 14200 14200 151 TSTB (R4) ;WAIT FOR RQI TO DROP
1168 014026 100776 14300 151 BMI #2 ;BR IF STILL SET
1169 014030 000207 14400 151 RTS PC ;RETURN
1170
1171
1172 ;*****
1173 ; OUTPUT SERVICE ROUTINE
1174 ;*****
1175 014032 032764 000001 000002 14700 RISR: BIT #BIT0,2(R4) ;ERROR?
1176 014040 001463 14900 151 BEQ 10 ;BR IF NO
1177 014042 005737 016742 15000 151 TST RESUME ;RESUME FLAG SET?
1178 014046 001413 15100 151 BEQ 50 ;BR IF NOT
1179 014050 022764 001000 000006 15200 151 CMP #BIT9,6(R4) ;IS PROC. ERROR BIT SET?
1180 014056 001007 15300 151 BNE 50 ;BR IF NOT
1181 014060 142764 000207 000002 15400 151 BICB #207,2(R4) ;CLEAR DONE
1182 014066 012737 177777 016744 15500 151 MOV #=-1,BASEFG ;SET BASEFG
1183 014074 000002 15600 151 RTI ;RETURN
1184 014076 022764 000004 000006 15700 501 CMP #4,6(R4) ;OVERUN ERROR?
1185 014104 001003 15800 151 BNE 30 ;BR IF NO
1186 014106 152714 000144 15900 151 BISB #144,(R4) ;REQUEUE XMIT BUFFER
1187 014112 001432 16000 151 BR 40
1188 014114 016437 000004 011054 16100 301 MOV 4(R4),ERCSR ;SAVE SEL4
1189 014122 016437 000006 011056 16200 151 MOV 6(R4),ERDBR ;SAVE SEL6 (ERROR BITS)
1190 014130 104400 016366 16300 151 TYPE ,DMCSR ;ERROR MESSAGE
1191 014134 013746 011054 16400 151 MOV ERCSR,=(SP) ;PUSH SEL4 ON STACK FOR TYPEOUT
1192 014142 004037 015770 16500 151 JSR R0,#B20CT ;TYPE IT OUT

```

```

1193 014144 000 16600 .BYTE 5
1194 014146 001 16700 .BYTE 1
1195 014148 104400 016562 16800 TYPE ,SPACES3 ;INSERT 3 SPACES
1196 014152 013746 011056 16900 MOV ERDBR,=(SP) ;PUSH SEL6 ON STACK FOR TYPEOUT
1197 014156 004037 015770 17000 JSR R0,#B20CT ;TYPE IT OUT
1198 014162 000 17100 .BYTE 6
1199 014164 001 17200 .BYTE 1
1200 014166 005777 174654 17300 TST #05WR ;CHECK BIT 15
1201 014170 100001 17400 BPL #+4 ;SKIP HALT IF = 0
1202 014172 000000 17500 HALT ;HALT IF 5WR15 = 1
1203 014174 005037 011056 17600 CLR ERDBR ;CLR ERDBR LOCATION
1204 014200 142764 000207 000002 17700 401 BICB #207,2(R4) ;CLEAR DONE
1205 014206 000002 17800 151 RTI ;RETURN
1206 014210 032764 000004 000004 17900 151 BIT #BIT10,PARAM1 ;REC DONE?
1207 014216 001406 18000 151 BEQ 20 ;BR IF NO
1208 014220 052700 040000 18100 151 BIS #RFLG,STAT ;SET REC DONE FLAG
1209 014224 142764 000207 000002 18200 151 BICB #207,2(R4) ;CLEAR DONE
1210 014232 000002 18300 151 RTI ;RETURN
1211 014234 052700 100000 18400 201 BIS #XPLG,STAT ;SET XMIT DONE FLAG
1212 014240 005037 016752 18500 151 CLR XFLAG ;CLEAR XFLAG
1213 014244 142764 000207 000002 18600 151 BICB #207,2(R4) ;CLEAR DONE
1214 014252 000002 18700 151 RTI ;RETURN
1215
1216
1217
1218 ;*****
1219 ; ENTER HERE IF BOOT MODE WAS SELECTED
1220 ;*****
1221 014254 032737 000000 011012 19100 BOOT: BIT #BIT10;BIT9,PARAM1;DETERMINE WHICH BOOT MODE
1222 014262 001413 19200 BEQ AUTOORG ;BR IF AUTO MODE, ORIGINATING STATION
1223 014264 032737 002000 011012 19300 BIT #BIT10,PARAM1
1224 014272 001522 19400 BEQ AUTOORG ;BR IF AUTO MODE, BOOT STATION
1225 014276 032737 001000 011012 19500 BIT #BIT9,PARAM1
1226 014302 001540 19600 BEQ MANORG ;BR IF MANUAL MODE, ORIGINATING STATION
1227 014304 104400 016566 19700 TYPE ,ROOMSG ;MANUAL MODE, BOOT STATION
1228 014310 000777 19800 BR , ;WAIT FOR MANUAL BOOT
1229
1230
1231 ;*****
1232 ; AUTOMATIC MODE ORIGINATING STATION
1233 ;*****
1234 014312 012701 015603 20200 AUTOORG: MOV #MOP1+1,R1 ;LOAD MOP1 MESSAGE WITH
1235 014316 113721 011012 20400 MOVB PARAM1,(R1)+ ;PASSWORD FOUR TIMES
1236 014322 113721 011012 20500 MOVB PARAM1,(R1)+
1237 014326 113721 011012 20600 MOVB PARAM1,(R1)+
1238 014332 113721 011012 20700 MOVB PARAM1,(R1)+
1239 014336 005737 016744 20800 TST BASEFG ;IS BASE REQUEST DONE?
1240 014342 001765 20900 BEQ AUTOORG ;BR IF NO
1241 014344 005037 016744 21000 CLR BASEFG ;CLEAR LOCK FLAG
1242 014350 012737 002000 016744 21100 MOV #2400,SEL5 ;MAINT. MODE BIT(MOP)
1243 014356 052714 000144 21200 DIS #141,(R6) ;ASK FOR CNTL IN
1244 014362 005737 016744 21300 101 TST BASEFG ;IS CNTL IN DONE?
1245 014366 001775 21400 BEQ 10 ;BR IF NO
1246 014372 005037 016744 21500 CLR BASEFG ;CLEAR LOCK FLAG
1247 014374 005037 016744 21600 CLR RFLG ;CLEAR RECEIVE FLAG
1248 014400 052714 000144 21700 RTS #400,(R4) ;BACK FOR REC BA/CC

```

```

1249 014404 005737 016744 21800 281 TST BASEFG ;IS REQUEST DONE?
1250 014410 001775 21900 REQ 20 ;BR IF NO
1251 014412 005037 016744 22000 CLR BASEFG ;CLEAR LOCK FLAG
1252 014416 005037 016752 CLR XFLAG ;CLEAR XMIT FLAG
1253 014422 012737 015600 016762 22200 MOV #MOP1,SEL4 ;SET FOR DMC LOAD
1254 014430 012737 000005 016764 22300 MOV #5,SEL6 ;SET FOR DMC LOAD
1255 014438 052714 000140 22400 BIS #14,(R4) ;ASK FOR XMIT BA/CC
1256 014442 012737 000005 016746 22500 MOV #5,TEHP1 ;SET UP DELAY COUNT
1257 014450 005037 013376 22600 CLR DELAY
1258 014454 005737 016754 22700 381 TST RFLAG ;RECEIVED ANYTHING?
1259 014460 001112 22800 BNE 44 ;BR IF YES
1260 014462 005337 013376 22900 DEC DELAY ;DEC DELAY COUNTER
1261 014466 001372 23000 BNE 38 ;WAIT TO RECEIVE
1262 014470 005337 016746 23100 DEC TEHP1 ;DEC SECOND COUNT
1263 014474 001367 23200 BNE 38 ;BR IF NOT DONE DELAY
1264 014476 005737 016752 23300 TST XFLAG ;WAS XMIT COMPLETED?
1265 014502 001140 23400 BNE 28 ;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 481 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 581 CMPB (R1)+,(R2)+ ;COMPARE DATA
1274 014522 001317 24300 BNE 10 ;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 58 ;BR IF NO
1278 014534 004737 015506 24700 JSR PC,MP3 ;IT WAS MOP2, SO SEND MOP3
1279 24800
1280 ;
1281 ; *****
1282 ; AUTOMATIC MODE BOOT STATION
1283 ; *****
1284 25000
1285 014540 005737 016744 25100 AUTO001 TST BASEFG ;BASE COMPLETED?
1286 014544 001775 25200 BEQ AUTO000 ;BR IF NO
1287 014546 005037 016744 25300 CLR BASEFG ;CLEAR LOCK FLAG
1288 014552 005037 016760 25400 CLR MFLAG ;CLEAR MAINT FLAG
1289 014556 005037 016754 25500 CLR RFLAG ;CLEAR REC FLAG
1290 014562 052714 000144 25600 BIS #14,(R4) ;ASK FOR REC BA/CC
1291 014572 001362 25700 181 TST MFLAG ;ARE WE IN MAINT MODE?
1292 014574 005737 016754 25800 BNE AUTO000 ;BR IF YES
1293 014600 001357 25900 281 TST RFLAG ;DID WE RECEIVE ANYTHING?
1294 014602 000771 26000 BNE AUTO000 ;YES REQUEST RECEIVE BUFFER
1295 26100 ;KEEP ON TRUCKIN'
1296 ;
1297 ; *****
1298 ; MANUAL MODE ORIGINATING STATION
1299 ; *****
1300 26400
1301 014604 005037 016760 26500 MANORG1 CLR MFLAG ;CLEAR MAINT FLAG
1302 014610 005037 016756 26600 CLR SFLAG ;CLEAR DDCMP START RECEIVED ERROR FLAG
1303 014614 005737 016744 26700 781 TST BASEFG ;BASE LOADED?
1304 014620 001775 26800 BEQ 78 ;BR IF NO
1305 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 #14,(R4) ;ASK FOR CNTLI
1307 014636 005737 016744 27200 881 TST BASEFG ;CNTLI DONE?
1308 014642 001775 27300 BEQ 88 ;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 #14,(R4) ;ASK FOR REC BA/CC
1312 014660 005737 016756 27700 181 TST SFLAG ;DDCMP START ERROR?
1313 014664 001437 27800 BEQ 44 ;BR IF NO
1314 014666 012714 000000 27900 MOV #BIT14,(R4) ;INITIALIZE DMC
1315 014672 005714 28000 281 TST (R4) ;RUN SET?
1316 014674 100376 28100 BPL 28 ;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 381 TST BASEFG ;BASE DONE?
1322 014724 001775 28700 BEQ 38 ;BR IF NO
1323 014726 005037 016744 28800 CLR BASEFG ;CLEAR FLAG
1324 014732 005037 016744 28900 CLR SFL6 ;SET UP FOR FULL-DUPLEX
1325 014736 052714 000141 29000 BIS #141,(R4) ;ASK FOR CNTLI
1326 014742 005737 016744 29100 981 TST BASEFG ;CNTLI FINISHED?
1327 014746 001775 29200 BEQ 98 ;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 481 TST MFLAG ;ARE WE IN MAINT MODE?
1332 014770 001733 29700 BEQ 10 ;BR IF NO
1333 014772 012714 000000 29800 MOV #BIT14,(R4) ;INITIALIZE DMC
1334 014776 005714 29900 1081 TST (R4) ;RUN SET?
1335 015000 100376 30000 BPL 108 ;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 1181 TST BASEFG ;BASE DONE?
1340 015024 001775 30500 BEQ 118 ;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 1281 TST BASEFG ;CNTLI FINISHED?
1345 015050 001775 31000 BEQ 128 ;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 581 TST RFLAG ;HAVE WE RECEIVED ANYTHING?
1350 015072 001775 31500 BEQ 58 ;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 ;RECEIVE BUFFER ADDRESS
1356 015104 005003 32100 CLR R3 ;CLEAR COUNT
1357 015106 122122 32200 681 CMPB (R1)+,(R2)+ ;COMPARE DATA
1358 015110 001343 32300 BNE 118 ;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      68          ;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 (ROOT MODE)
1366          ;*****
1367          33000
1368 015126 032714 000002 33100 IISR:  BIT      #BIT1,(R4) ;IS IT A BASE REQUEST?
1369 015132 001413          33200      BEQ      18          ;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,46       ;CLEAR RQI
1373 015152 012737 177777 016744 33600      MOV      #-1,BASEFG  ;SET FLAG
1374 015160 000002 33700      RTI          ;RETURN
1375 015162 032714 000001 33800 18:  BIT      #BIT0,(R4) ;IS IT CNTL IN?
1376 015166 001411          33900      BEQ      28          ;BR IF NO
1377 015170 013764 016764 000006 34000      MOV      SEL6,6(R4)  ;LOAD SEL6 FOR CNTLI
1378 015176 004737 015300 34100      JSR      PC,46       ;CLEAR RQI
1379 015202 012737 177777 016744 34200      MOV      #-1,BASEFG  ;SET FLAG
1380 015210 000002 34300      RTI          ;RETURN
1381 015212 012714 000004 34400 28:  BIT      #BIT2,(R4) ;IS IT RECEIVE REQUEST?
1382 015216 001414          34500      BEQ      38          ;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,46       ;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 38:  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,46       ;CLEAR RQI
1391 015270 012737 177777 016744 35400      MOV      #-1,BASEFG  ;SET FLAG
1392 015276 000002 35500      RTI          ;RETURN
1393 015300 142714 000040 35600 48:  BICB    #40,(R4)   ;CLEAR RQI
1394 015304 105714          35700      TSTB    (R4)        ;RQI CLEAR?
1395 015306 102776          35800      BMI     -2          ;NO
1396 015310 002707          35900      RTS     PC          ;RETURN
1397          36000
1398          ;*****
1399          ; OUTPUT INTERRUPT SERVICE ROUTINE (BOOT MODE)
1400          ;*****
1401          36200
1402 015312 012764 000001 000002 36300 OISR:  BIT      #BIT0,2(R4) ;ERROR?
1403 015320 001453          36400      BEQ      38          ;NO
1404 015322 022764 000010 000006 36500      CMP      #10,6(R4)  ;YES, MAINT MODE ENTERED?
1405 015330 001004          36600      BNE     18          ;NO
1406 015332 012737 177777 016760 36700      MOV      #-1,MFLAG   ;YES, SET MFLAG
1407 015340 000456          36800      BR      58          ;RETURN
1408 015342 022764 000200 000006 36900 18:  CMP      #200,6(R4) ;DDCMP START RECEIVED ERROR?
1409 015350 001004          37000      BNE     28          ;NO
1410 015352 012737 177777 016756 37100      MOV      #-1,SFLAG   ;YES, SET SFLAG
1411 015360 000446          37200      BR      58          ;RETURN
1412 015362 016437 000004 011054 37300 28:  MOV      4(R4),ERCSR ;SAVE SEL4
1413 015370 010437 000006 011056 37400      MOV      6(R4),ERDBR ;SAVE SEL6 (ERROR BITS)
1414 015376 104000 016366 37500      TYPE    ,DMCSR      ;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          37800      ,BYTE  6
1418 015413 001          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,#B2OCT  ;TYPE IT OUT
1422 015430 006          38300      ,BYTE  6
1423 015431 001          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      58          ;RETURN
1429 015450 032764 000004 000002 39000 38:  BIT      #BIT2,2(R4) ;RECEIVE DONE?
1430 015456 001424 39100      BEQ      48          ;BR IF NO
1431 015460 012737 177777 016754 39200      MOV      #-1,MFLAG   ;SET RECEIVE FLAG
1432 015466 000403 39300      BR      58          ;RETURN
1433 015470 012737 177777 016752 39400 48:  MOV      #-1,XFLAG   ;XMIT DONE, SET XMIT FLAG
1434 015476 142764 000207 000002 39500 58:  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 MOP3:  TST     BASEFG    ;IS IT OK TO REQUEST
1442 015512 001775 40100      BFO     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 0052714 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 18:  TST     XFLAG      ;XMIT DONE?
1450 015554 001004 40900      BNE     28          ;BR IF YES
1451 015556 005337 013376 41000      DEC     DELAY      ;DEC DELAY COUNT
1452 015562 001372 41100      BNE     18          ;BR IF NO DONE
1453 015564 000000 41200      HALT          ;ERROR, MOP3 SEND NOT DONE
1454 015566 104400 016635 41300 28:  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 006 000 000 41800 MOP1:  ,BYTE  6,0,0,0,0
1460 015605 000 000
1461          015610 41900      ,EVEN
1462 015610 010 014 001 42000 MOP2:  ,BYTE  10,12,,1,0
1463 015613 000
1464 015614 000 000 42100 MOP3:  ,BYTE  0,0,6,0,6,0
1465 015617 000 006 000
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 #15642 #62700 000034 42900 ADD #<MSG-,>,R0 ;ADDRESS OF MMESSAGE
1474 #15646 #05761 #000034 43000 18: TSTB 4(R1) ;READY SET?
1475 #15652 100375 43100 BPL 18 ;BR IF NO
1476 #15651 112001 #000006 43200 MOV# (R0)+,0(R1) ;TYPE A CHARACTER
1477 #15660 #01372 43300 RNE 18 ;KEEP TYPING IF NOT ZERO
1478 #15662 #12737 #000026 #000024 43400 MOV #26,#*24 ;SET UP POWER FAIL VECTOR
1479 #15670 #05737 #000026 43500 CLR #*26 ;MAKE SURE T BIT CLEAR
1480 #15674 #00777 43600 BR ;BR
1481 #15676 #00112 047502 052117 43700 MSG: .ASCII <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY = END OF TEST1//
1482 #15704 #04640 #051505 040523
1483 #15712 042507 #053440 #051501
1484 #15720 #051040 #041505 044505
1485 #15726 042526 #020104 052523
1486 #15734 #041503 #051505 043123
1487 #15742 046125 #054514 026440
1488 #15750 #04240 #042116 047440
1489 #15756 #020106 042524 052123
1490 #15764 #000 000
1491 #15767 006 43800 .BYTE 6
1492 #15770 43900 MOP3ED:
1493 44000 .EVEN
1494 44100
1495 44200
1496 44300
1497 44400 ;*****
1498 44500 ;BINARY TO OCTAL (ASCII) AND TYPE
1499 44600 ;#B20CT==ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1500 ;CALL:
1501 ; MOV NUM,=(SP) ;NUMBER TO BE TYPED
1502 ; JSR R0,#B20CT ;CALL FOR TYPEOUT
1503 ; .BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1504 ; .BYTE M ;M=1 OR 0
1505 ; ;1=TYPE LEADING ZEROS
1506 ; ;0=SUPPRESS LEADING ZEROS
1507 45300
1508 ; ;#B201===ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST #BSOCT OR
1509 45500 ;CALL:
1510 ; MOV NUM,=(SP)
1511 ; JSR R0,#B201
1512 45800
1513 ; ;#B2016===ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1514 ;CALL:
1515 ; MOV NUM,=(SP)
1516 ; JSR R0,#B2016
1517 #15770 112037 #016175 46300 #B20CT: MOV# (R0)+,#OMODE+1 ;PICKUP THE NUMBER OF DIGITS TO TYPE
1518 #15774 112037 #016173 46400 MOV# (R0)+,#0FILL ;GET THE ZERO FILL SWITCH
1519 #16000 #00406 46500 BR #B201
1520 #16002 112737 000001 #016173 46700 #B2016: MOV# #1,#0FILL ;SET THE ZERO FILL SWITCH
1521 #16010 112737 000006 #016175 46800 MOV# #6,#OMODE+1 ;SET FOR SIX(6) DIGITS
1522 #16016 112737 000005 #016172 46900 #B201: MOV# #5,#OCNT ;SET THE ITERATION COUNT
1523 #16024 #010346 47000 MOV R3,-(SP) ;SAVE R3
1524 #16026 #010446 47100 MOV R4,=(SP) ;SAVE R4
1525 #16030 #010546 47200 MOV R5,=(SP) ;SAVE R5
1526 #16032 113704 #016175 47300 MOV# #OMODE+1,R4 ;GET THE NUMBER OF DIGITS TO TYPE
1527 #16036 #045404 47400 NEG R4
1528 #16040 #02704 000006 47500 ADD #6,R4 ;SUBTRACT IT FOR MAX. ALLOWED

```

```

1529 #16044 110437 #016174 47600 MOV# R4,#OMODE ;SAVE IT FOR USE
1530 #16050 113704 #016173 47700 MOV# #0FILL,R4 ;GET THE ZERO FILL SWITCH
1531 #16054 010605 #000010 47800 MOV 10(SP),R5 ;PICKUP THE INPUT NUMBER
1532 #16060 #005003 47900 CLP R3 ;CLEAR THE OUTPUT WORD
1533 #16062 #006105 48000 18: ROL R5 ;ROTATE MSB INTO "C"
1534 #16064 #004004 48100 BR 30 ;GO DO MSB
1535 #16066 #006105 48200 28: ROL R5 ;FORM THIS DIGIT
1536 #16070 #006105 48300 ROL R5
1537 #16072 #006105 48400 ROL R5
1538 #16074 #010503 48500 MOV R5,R3
1539 #16076 #006103 48600 38: ROL R3 ;GET LSB OF THIS DIGIT
1540 #16100 105337 #016174 48700 DECB #OMODE ;TYPE THIS DIGIT?
1541 #16104 100010 48800 BPL 78 ;BR IF NO
1542 #16106 #042703 177770 48900 BIC #177770,R3 ;GET RID OF JUNK
1543 #16112 #001002 49000 BNE 48 ;TEST FOR 0
1544 #16114 #005704 49100 TST R4 ;SUPPRESS THIS 0?
1545 #16116 #011003 49200 BEQ 58 ;BR IF YES
1546 #16120 #005204 49300 48: INC R4 ;DON'T SUPPRESS ANYMORE 0'S
1547 #16122 #052703 000006 49400 BIS #'0,R3 ;MAKE THIS DIGIT ASCII
1548 #16126 #052703 000040 49500 58: BIS #' ,R3 ;MAKE ASCII IF NOT ALREADY
1549 #16132 110337 #016170 49600 MOV# R3,#0 ;SAVE FOR TYPING
1550 #16136 104400 #016170 49700 TYPE #0 ;GO TYPE THIS DIGIT
1551 #16142 105337 #016172 49800 78: DECB #OCNT ;COUNT BY 1
1552 #16146 #003347 49900 BGT 28 ;BR IF MORE TO DO
1553 #16150 #002402 50000 BLT 68 ;BR IF DONE
1554 #16152 #005204 50100 INC R4 ;INSURE LAST DIGIT ISN'T A BLANK
1555 #16154 #000744 50200 BR 28 ;GO DO THE LAST DIGIT
1556 #16156 #012605 50300 68: MOV (SP)+,R5 ;RESTORE R5
1557 #16160 #012604 50400 MOV (SP)+,R4 ;RESTORE R4
1558 #16162 #012603 50500 MOV (SP)+,R3 ;RESTORE R3
1559 #16164 #012616 50600 MOV (SP)+,(SP) ;SET THE STACK FOR RETURNING
1560 #16166 #002000 50700 RTS R0 ;RETURN
1561 #16170 #000 50800 88: .BYTE 0 ;STORAGE FOR ASCII DIGIT
1562 #16171 #000 50900 .BYTE 0 ;TERMINATOR FOR TYPE ROUTINE
1563 #16172 #000 51000 #OCNT: .BYTE 0 ;OCTAL DIGIT COUNTER
1564 #16173 #000 51100 #0FILL: .BYTE 0 ;ZERO FILL SWITCH
1565 #16174 #000000 51200 #OMODE: 0 ;NUMBER OF DIGITS TO TYPE
1566 51300
1567 51400
1568 #16176 #12737 177777 #016742 51500 CKBASE: MOV #-1,RESUME ;SET RESUME FLAG
1569 #16204 #005037 #016744 51600 CLR BASEFG ;CLEAR BASEFG
1570 #16210 #052714 #000146 51700 BIS #146,(R4) ;SHUT DOWN DMC TO UPDATE BASE TABLE
1571 #16214 #005737 #016744 51800 18: TST BASEFG ;SHUT DOWN DONE?
1572 #16220 #011775 51900 BEQ 18 ;BR IF NO
1573 #16222 #12714 040000 52000 MOV #BIT14,(R4) ;MASTER CLEAR DMC
1574 #16226 #005714 52100 28: TST (R4) ;RUN SET?
1575 #16230 #006376 52200 BPL 28 ;BR IF NO
1576 #16232 #005037 #016744 52300 CLR BASEFG ;CLEAR BASEFG
1577 #16236 #052714 #000143 52400 BIS #143,(R4) ;ASK FOR BASE REQUEST
1578 #16242 #005737 #016744 52500 38: TST BASEFG ;BASE LOADED?
1579 #16246 #011775 52600 BEQ 38 ;BR IF NO
1580 #16250 #12764 #000100 #000002 52700 MOV #100,2(R4) ;SET INTERRUPT ENABLE
1581 #16256 #005037 #016742 52800 CLR RESUME ;CLEAR RESUME FLAG
1582 #16262 #12702 #000003 52900 MOV #3,R2 ;LOAD BASE OFFSET TO ERROR COUNTS
1583 #16266 #005001 53000 CLR R1 ;R1 IS OFFSET INTO SOFTWARE TABLE
1584 #16270 #126261 #016776 #016766 53100 58: CMPB BASE(R2),ERRCNT(R1);ANY ERRORS THIS PASS?

```





BOX	1#	660	697	715	1081	1102	1119	1171	1217	1230	1280	1296	1304	1390	1437
DCPARM	1#														
DHDOC1	1#														
DHPARM	1#														
DJPARM	1#														
DLPARM	1#														
DMPARM	1#	555													
DPPARM	1#														
DQDOC1	1#														
DOPARM	1#														
DUPARM	1#														
DUPPAR	1#														
DVDOC1	1#														
DVPARM	1#														
DZPARM	1#														
HELLO	1#														
HLT	659#	816	849	882	913	954	974	1067							
\$EQUAT	1#	659													
\$INTF	1#	659													
\$ITEP	1#	770													
\$SERV	1#	704													

, ABS, 017376 000

ERRORS 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)