DL11

OVERLAY FOR ITEP

MD-11-DZDLO-C

EP-DZDLO-C-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDLO-C-D	
PRODUCT NAME:	DL11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM	
PROGRAM DATE:	OCTOBER 1976	
MAINTAINER:	DIAGNOSTICS	
AUTHORS:	R A JONES JOHN EGOLF	
REVISED BY:	FAY BASHAW	1/21/75

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 DL11 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. 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 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 MONITOR.
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 PROGRAM TESTS
THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A
SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO.
IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE
DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

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<DC1> 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 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 DL11 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 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,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 DL11

PARAM#1 MUST BE ALL ZEROS.

PARAM#2 BIT 0 OF THIS PARAMETER IS CHECKED BY THE SOFTWARE TO RUN
EITHER FULL-DUPLEX OR HALF-DUPLEX. BIT0=1 SELECTS FULL-DUPLEX,
BIT0=0 SELECTS HALF-DUPLEX. DEFAULT IS HALF-DUPLEX, ALL OTHER BITS MUST BE ZEROS.

PARAM#3 IS NOT USED(177777).

```
                                ;*****************************************************************
                                ;       DL11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000          .=11000
011000  046104  000040        DL11:   .ASCIZ  /DL /           ;ISR NAME
011004  175610                BA:     175610                  ;BUS ADDRESS
011006  000300                RIV:    300                     ;VECTOR ADDRESS
011010  000200                PRIOR:  200                     ;PRIORITY
011012  000000                PARAM1: 000000                  ;PARAM #1
011014  000000                PARAM2: 000000                  ;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
011026  000000                        .WORD   0               ;
011030  000000                B2016:  .WORD   0               ;ADDR OF BIN TO OCT TYPE ROUTINE
011032  000000                TIME:   .WORD   0               ;TIMER
011034  000000                        .WORD   0               ;
011036  011102                        .WORD   START           ;ADDR OF START OF PROGRAM
011040                        TX.TERM:
011040     000                        .BYTE   000             ;TRANSMITTER TERMINATING CHAR.
011041                        RX.TERM:
011041     001                        .BYTE   001             ;RECEIVER TERMINATING CHAR.
011042  000000                FLAG:   .WORD   0
011044  177570                SWR:    177570
011046  177570                DISPLAY:177570

                                ;*****************************************************************
                                ;       CONSTANTS + WORKING STORAGE
                                ;*****************************************************************
                        000000          STAT=R0
                        100000          XFLG=100000             ;XMIT COMPLETE FLAG
                        040000          RFLG=40000              ;RCV COMPLETE FLAG
                        020000          DSFLG=20000             ;DATA SET STATUS CHANGE FLAG
                        020000          BIT13=20000             ;INHIBIT PRINTOUTS

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        XCC:    0               ;XMIT CHAR COUNT
011064  000000        RCC:    0               ;RCV CHAR COUNT
011066  000000        RDA:    0               ;RCV DATA ADDR.
011070  000000        XDA:    0               ;XMIT DATA ADDR.

011072  177560        TKS:    177560
011074  177562        TKB:    177562
011076  177564        TPS:    177564
011100  177566        TPB:    177566

                        000001          FULL.DUPLEX=000001
```

```
                        ;*****************************************************************
                        ;       DL11-X INTERFACE SERVICE ROUTINE
                        ;*****************************************************************
011102  000240  START:  NOP
011104  017700  177734        MOV     @SWR,   R0      ;SETUP MODE IN R0
011110  042700  177400        BIC     #177400,R0      ;STRIP JUNK
011114  013702  011006        MOV     RIV,    R2      ;SETUP
011120  012722  013654        MOV     #RISR,  (R2)+   ;INTERRUPT
011124  013722  011010        MOV     PRIOR,  (R2)+   ;VECTORS
011130  012722  013536        MOV     #XISR,  (R2)+   ;
011134  013722  011010        MOV     PRIOR,  (R2)+   ;
011140  013704  011004        MOV     BA,     R4      ;SETUP BUS ADDR INDEX
011144  013714  011012        MOV     PARAM1, @RCSR   ;SETUP VARIABLES
011150  013702  011014        MOV     PARAM2,R2       ;
011154  042702  000001        BIC     #0001,R2        ;
011160  010264  000004        MOV     R2,     XCSR(R4);IN CSR'S

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

011164  005037  011032  GO:     CLR     TIME
011170  005037  013054          CLR     DELAY
011174  005037  013060          CLR     STOP
011200  032700  000001          BIT     #0W0,MODE
011204  001402                  BEQ     1$
011206  000137  011362          JMP     $0W0
011212  032700  000002  1$:     BIT     #0WI,MODE
011216  001402                  BEQ     2$
011220  000137  011254          JMP     $0WI
011224  032700  000010  2$:     BIT     #ILB,MODE
011230  001402                  BEQ     3$
011232  000137  011460          JMP     $ILB
011236  032700  000004  3$:     BIT     #XLB,MODE
011242  001402                  BEQ     4$
011244  000137  011710          JMP     $XLB
011250  000000  4$:     HALT
011252  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".
                        ;*****************************************************************


011254  104416  $0WI:   KBDIN
011256  004737  013374          JSR     PC,STARTR
```

```
                                                    ;*****************************************************************
                                                    ;       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.
                                                    ;*****************************************************************
```

685	011262	032700	040000		1$:	BIT	#RFLG,STAT	
686	011266	001013				BNE	2$	
687	011270	023727	011032	000100		CMP	TIME,#100	
688	011276	103771				BLO	1$	
689	011300	011402				MOV	@RCSR,R2	
690	011302	016403	000004			MOV	XCSR(R4),R3	
691	011306	104001				HLT	1	
692	011310	005037	011032			CLR	TIME	
693	011314	000762				BR	1$	
694								
695	011316	032777	000200	177520	2$:	BIT	#NODAT,@SWR	
696	011324	001002				BNE	3$	
697	011326	004737	012300			JSR	PC,TESTD	
698	011332	042700	040000		3$:	BIC	#RFLG,STAT	
699	011336	032777	000020	177500		BIT	#LOOP,@SWR	
700	011344	001405				BEQ	4$	
701	011346	012737	011360	013056		MOV	#4$,BACK	
702	011354	000137	012140			JMP	EOP	
703	011360	000735			4$:	BR	$OWI	
704								
705								
706								
707								
708								
709								
710								
711								
712								
713								
714	011362	104416			$OWO:	KBDIN		
715	011364	004737	013062			JSR	PC,STARTX	
716	011370	005037	011032			CLR	TIME	
717	011374	032700	100000		1$:	BIT	#XFLG,STAT	
718	011400	001013				BNE	2$	
719	011402	023727	011032	000100		CMP	TIME,#100	
720	011410	103771				BLO	1$	
721	011412	011402				MOV	@RCSR,R2	
722	011414	016403	000004			MOV	XCSR(R4),R3	
723	011420	104001				HLT	1	
724	011422	005037	011032			CLR	TIME	
725	011426	000762				BR	1$	
726	011430	042700	100000		2$:	BIC	#XFLG,STAT	
727	011434	032777	000020	177402		BIT	#LOOP,@SWR	
728	011442	001405				BEQ	3$	
729	011444	012737	011456	013056		MOV	#3$,BACK	
730	011452	000137	012140			JMP	EOP	
731	011456	000741			3$:	BR	$OWO	
732								
733								
734								

```
                                        ;*****************************************************************************
                                        ;       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.
                                        ;*****************************************************************************

011460  104416                  $ILB:   KBDIN
011462  004737  013374                  JSR     PC,STARTR
011466  005037  011032                  CLR     TIME
011472  032700  040000          1$:     BIT     #RFLG,STAT
011476  001013                          BNE     2$
011500  023727  011032  000100          CMP     TIME,#100
011506  103771                          BLO     1$
011510  011402                          MOV     @RCSR,R2
011512  016403  000004                  MOV     XCSR(R4),R3
011516  104001                          HLT     1
011520  005037  011032                  CLR     TIME
011524  000762                          BR      1$
011526  032777  000200  177310  2$:     BIT     #NODAT,@SWR
011534  001002                          BNE     3$
011536  004737  012300                  JSR     PC,TESTD
011542  042700  040000          3$:     BIC     #RFLG,STAT
011546  032777  000020  177270          BIT     #LOOP,@SWR
011554  001405                          BEQ     4$
011556  012737  011570  013056          MOV     #4$,BACK
011564  000137  012140                  JMP     EOP
011570  032777  000400  177246  4$:     BIT     #400,   @SWR    ;USE EXTERNAL DATA?
011576  001416                          BEQ     7$              ;BR IF NO
011600  013702  011020                  MOV     IRDA,   R2      ;SET POINTER
011604  013703  011022                  MOV     IXDA,   R3      ;SET POINTER
011610  010337  011070                  MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
011614  112223                          MOVB    (R2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
011616  001376                          BNE     .-2             ;LOOP IF NOT ZERO CHAR
011620  112743  000177                  MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
011624  005203                          INC     R3              ;BUMP ADDRESS
011626  112723  000177                  MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
011632  105023                          CLRB    (R3)+           ;INSERT ZERO CHAR
011634  005037  011032          7$:     CLR     TIME
011640  004737  013062                  JSR     PC,STARTX
011644  032700  100000          5$:     BIT     #XFLG,STAT
011650  001013                          BNE     6$
011652  023727  011032  000100          CMP     TIME,#100
011660  103771                          BLO     5$
011662  011402                          MOV     @RCSR,R2
011664  016403  000004                  MOV     XCSR(R4),R3
011670  104001                          HLT     1
011672  005037  011032                  CLR     TIME
011676  000762                          BR      5$
011700  042700  100000          6$:     BIC     #XFLG,STAT
011704  000137  011460                  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.
                                ;*************************************************************************

011710  104416                  $XLB:   KBDIN
011712  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
011720  001402                          BEQ     1$
011722  004737  013374                  JSR     PC,STARTR
011726  004737  013062          1$:     JSR     PC,STARTX
011732  005037  011032                  CLR     TIME
011736  032700  100000          2$:     BIT     #XFLG,STAT
011742  001016                          BNE     3$
011744  032700  040000          7$:     BIT     #RFLG,STAT
011750  001024                          BNE     4$
011752  023727  011032  000100          CMP     TIME,#100
011760  103766                          BLO     2$
011762  011402                          MOV     @RCSR,R2
011764  016403  000004                  MOV     XCSR(R4),R3
011770  104001                          HLT     1
011772  005037  011032                  CLR     TIME
011776  000757                          BR      2$
012000  032737  000001  011014  3$:     BIT     #FULL.DUPLEX,PARAM2
012006  001356                          BNE     7$
012010  042700  100000                  BIC     #XFLG,STAT
012014  004737  013374                  JSR     PC,STARTR
012020  000746                          BR      2$
012022  032737  000001  011014  4$:     BIT     #FULL.DUPLEX,PARAM2
012030  001420                          BEQ     8$
012032  032700  100000                  BIT     #XFLG,STAT
012036  001013                          BNE     6$
012040  023727  011032  000100          CMP     TIME,#100
012046  103765                          BLO     4$
012050  011402                          MOV     @RCSR,R2
012052  016403  000004                  MOV     XCSR(R4),R3
012056  104001                          HLT     1
012060  005037  011032                  CLR     TIME
012064  000756                          BR      4$
012066  042700  100000          6$:     BIC     #XFLG,STAT
012072  042700  040000          8$:     BIC     #RFLG,STAT
012076  005037  011032                  CLR     TIME
012102  032777  000200  176734          BIT     #NODAT,@SWR
012110  001002                          BNE     5$
012112  004737  012300                  JSR     PC,TESTD
012116  032777  000020  176720  5$:     BIT     #LOOP,@SWR
012124  001671                          BEQ     $XLB
012126  012737  011710  013056          MOV     #$XLB,BACK
012134  000137  012140                  JMP     EOP
```

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

012140                                  EOP:
012140  104414  000340                          STPS,PRTY7              ;SET PS PRIORITY TO 7
012144  016437  000004  012276                  MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012152  042737  177677  012276                  BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012160  042764  000100  000004                  BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012166  012766  012226  000002                  MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012174  010037  013040                          MOV     R0,SAVR0        ;SAVE REGISTER 0
012200  010137  013042                          MOV     R1,SAVR1        ;SAVE REGISTER 1
012204  010237  013044                          MOV     R2,SAVR2        ;SAVE REGISTER 2
012210  010337  013046                          MOV     R3,SAVR3        ;SAVE REGISTER 3
012214  010437  013050                          MOV     R4,SAVR4        ;SAVE REGISTER 4
012220  010537  013052                          MOV     R5,SAVR5        ;SAVE REGISTER 5
012224  000207                                  RTS     PC              ;RETURN TO CONTROL PROGRAM

012226                                  ENTER:
012226  013700  013040                          MOV     SAVR0,R0        ;RESTORE R0
012232  013701  013042                          MOV     SAVR1,R1        ;RESTORE R1
012236  013702  013044                          MOV     SAVR2,R2        ;RESTORE R2
012242  013703  013046                          MOV     SAVR3,R3        ;RESTORE R3
012246  013704  013050                          MOV     SAVR4,R4        ;RESTORE R4
012252  013705  013052                          MOV     SAVR5,R5        ;RESTORE R5
012256  012737  177777  013054                  MOV     #-1,DELAY
012264  053764  012276  000004                  BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012272  000177  000560                          JMP     @BACK
012276  000000                          QTPIE:  000000

                                        ;*********************************
                                        ;       SUBROUTINE TO CHECK
                                        ;       RECEIVER DATA.
                                        ;*********************************
012300  013746  011056                  TESTD:  MOV     ERDBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012304  001413                                  BEQ     TSTDAT          ;BR IF NO
012306  032777  020000  176530                  BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012314  001007                                  BNE     TSTDAT          ;BR IF YES
012316  104400  012500                          TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012322  004077  176502                          JSR     R0,@82016       ;PRINT CONTENTS OF RBUF
012326  005746                                  TST     -(SP)
012330  104400  012561                          TYPE    ,MSG1           ;<15><12>
012334  013701  011022                  TSTDAT: MOV     IXDA,   R1      ;SETUP XMIT DATA ADDR
012340  013702  011020                          MOV     IRDA,   R2      ;SETUP RCV DATA ADDR
012344  122122                          SCAN4:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012346  001776                                  BEQ     SCAN4           ;BR IF OK
012350  123741  011040                          CMPB    TX.TERM,-(R1)   ;IS IT END OF DATA
012354  001447                                  BEQ     TESTDX          ;BR IF YES
012356  122742  000002                          CMPB    #002,-(R2)
012362  001005                                  BNE     2$
012364  010237  012372                          MOV     R2,1$
012370  104400                                  TYPE
```

```
 902  012372  000000                    1$:     .WORD   0
 903  012374  000437                            BR      TESTDX
 904  012376                            2$:
 905  012376  105712                            TSTB    (R2)            ;
 906  012400  001435                            BEQ     TESTDX          ;BR IF YES
 907  012402  122721  000177                    CMPB    #177,   (R1)+   ;IS IT FILL CHAR?
 908  012406  001756                            BEQ     SCAN4           ;BR IF YES
 909  012410  005301                            DEC     R1              ;BACKUP
 910  012412  122722  000177                    CMPB    #177,   (R2)+   ;IS IT FILL?
 911  012416  001752                            BEQ     SCAN4           ;BR IF YES
 912  012420  000240                    SCAN5:  NOP                     ;DATA ERROR
 913  012422  032777  020000  176414            BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS
 914  012430  001016                            BNE     DERR            ;BR IF YES
 915  012432  104400  012564                    TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
 916  012436  013737  011020  012446            MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
 917  012444  104400                            TYPE                    ;PRINT RECEIVED DATA
 918  012446  000000                    RDAX:   0                       ;RECEIVED DATA ADDR.
 919  012450  104400  012611                    TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
 920  012454  013737  011022  012464            MOV     IXDA,   .+10    ;SETUP ADDR.
 921  012462  104400                            TYPE                    ;PRINT GOOD DATA
 922  012464  011022                            IXDA
 923  012466  111103                    DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
 924  012470  114202                            MOVB    -(R2),R2        ;SETUP RCV DATA
 925  012472  104007                            HLT+7                   ;DATA ERROR HALT
 926  012474  005726                    TESTDX: TST     (SP)+           ;POP STACK
 927  012476  000207                            RTS     PC              ;RETURN FROM SUB/ROUT
 928
 929  012500  005015  044124  051105    MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1)   012561     015  000012            MSG1:   .ASCIZ  <15><12>
(1)   012564  005015  042522  042503    MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
(1)   012611     015  042012  052101    MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
(1)   012634  005015  046120  040505    MSG4:   .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1)   012703     015  053412  042510            .ASCIZ <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
(1)   012766  005015  046120  040505    MSG5:   .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)                                     .EVEN
(1)   013040  000000                    SAVR0:  0
 930  013042  000000                    SAVR1:  0
 931  013044  000000                    SAVR2:  0
 932  013046  000000                    SAVR3:  0
 933  013050  000000                    SAVR4:  0
 934  013052  000000                    SAVR5:  0
 935  013054  000000                    DELAY:  0
 936  013056  000000                    BACK:   0
 937  013060  000000                    STOP:   0
 938
```

```
                                ;*****************************************************************
                                ;       INITIALIZE TRANSMIT SUBROUTINE
                                ;*****************************************************************

013062                          STARTX:
013062  005737  013054                  TST     DELAY           ;IF SW04=1 & SW14=0 WAIT BEFORE TURINING ON TX
013066  001416                          BEQ     1$              ;NO GO AHEAD AND TURN ON TX
013070  005037  013526                  CLR     TEMP1           ;PREPARE FOR DELAY
013074  012737  000007  013530          MOV     #7,TEMP2
013102  062737  000001  013526          ADD     #1,TEMP1        ;INCREMENT DELAY.......
013110  001374                          BNE     .-6
013112  005337  013530                  DEC     TEMP2
013116  001371                          BNE     .-14
013120  005037  013054                  CLR     DELAY           ;ZERO POINTER.
013124  013737  011022  011070  1$:     MOV     IXDA,   XDA     ;SETUP XMIT DATA ADDR.
013132  052714  000002                  BIS     #DTR,@RCSR      ;SET REQUEST TO SEND
013136  005737  013060                  TST     STOP
013142  001005                          BNE     2$
013144  104400  012634                  TYPE    ,MSG4
013150  000000                          HALT
013152  005137  013060                  COM     STOP
013156  032737  000001  011014  2$:     BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
013164  001003                          BNE     3$              ;BR IF YES
013166  032714  010000                  BIT     #10000,@RCSR    ;CARRIER UP?
013172  001375                          BNE     .-4             ;BR IF YES
013174  052714  000004          3$:     BIS     #RQTS,@RCSR
013200  032714  000002                  BIT     #DTR,   @RCSR   ;IS THIS A DL11-E?
013204  001425                          BEQ     CTSOK           ;BR IF NO
013206
013206  032714  020000          CTSW:   BIT     #CTS,   @RCSR   ;IS CLEAR TO SEND SET?
013212  001017                          BNE     2$              ;BR IF YES
013214  023727  011032  000036          CMP     TIME,   #36     ;30 SECS ELASPED?
013222  103771                          BLO     CTSW            ;BR IF NO
013224  011402                          MOV     @RCSR,  R2      ;SETUP RCV CSR
013226  016403  000004                  MOV     XCSR(R4),R3     ;SETUP XMIT CSR
013232  032777  010000  175604          BIT     #SW12,@SWR      ;INHIBIT PRINTOUTS?
013240  001001                          BNE     1$              ;BR IF YES
013242  104002                          HLT+2                   ;PRINTOUT 'WAITING TO XMIT'
013244  005037  011032          1$:     CLR     TIME            ;RESET TIMER
013250  000756                          BR      CTSW            ;WAIT SOME MORE
013252  032714  010000          2$:     BIT     #10000,@RCSR    ;IS CARRIER UP?
013256  001775                          BEQ     2$              ;BR IF NO
013260
013260  005737  011024          CTSOK:  TST     SETTLE          ;CONNECTION JUST MADE?
013264  001416                          BEQ     2$              ;BR IF NO
013266  005037  013526                  CLR     TEMP1           ;YES PREPARE FOR DELAY
013272  012737  000030  013530          MOV     #14*2,TEMP2
013300  062737  000001  013526          ADD     #1,TEMP1        ;INCREMENT DELAY
013306  001374                          BNE     .-6
013310  005337  013530                  DEC     TEMP2
013314  001371                          BNE     .-14
013316  005037  011024                  CLR     SETTLE          ;CLEAR DELAY FLAG
013322  032737  000001  011014  2$:     BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
013330  001415                          BEQ     1$              ;BR IF NO
013332  032700  000004                  BIT     #XLB,MODE       ;XLB MODE?
013336  001412                          BEQ     1$              ;BR IF NO
```

```
  995  013340  012737  177777  013534          MOV     #-1,TRNFLG      ;SET FLAG
  996  013346  052764  000100  000004          BIS     #TIE,XCSR(R4)   ;SET INTERUPT ENABLE
  997  013354  000001                          WAIT
  998  013356  005737  013532                  TST     SNCFLG          ;FIRST CHAR RECEIVED YET?
  999  013362  001375                          BNE     .-4             ;BR IF NO
 1000  013364  052764  000100  000004  1$:     BIS     #TIE,   XCSR(R4);SET XMIT INTERRUPT ENABLE
 1001  013372  000207                          RTS     PC              ;EXIT FROM SUBROUTINE
 1002
 1003                                  ;*****************************************************************
 1004                                  ;       INITIALIZE RECEIVER SUBROUTINE
 1005                                  ;*****************************************************************
 1006  013374  005737  013060          STARTR: TST     STOP            ;FIRST TIME HERE?
 1007  013400  001012                          BNE     1$              ;BR IF NO
 1008  013402  052714  000002                  BIS     #DTR,@RCSR      ;SET DTR
 1009  013406  104400  012766                  TYPE    ,MSG$           ;MAKE CONNECTION
 1010  013412  012737  177777  011024  2$:     MOV     #-1,SETTLE      ;YES SET DELAY FLAG
 1011  013420  012737  177777  013060          MOV     #-1,STOP
 1012  013426  032737  000001  011014  1$:     BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
 1013  013434  001410                          BEQ     3$              ;BR IF NO
 1014  013436  032700  000004                  BIT     #XLB,MODE       ;XLB MODE?
 1015  013442  001405                          BEQ     3$              ;BR IF NO
 1016  013444  005037  013526                  CLR     TEMP1           ;START DELAY
 1017  013450  005237  013526                  INC     TEMP1
 1018  013454  001375                          BNE     .-4
 1019  013456  012737  177777  013532  3$:     MOV     #-1,SNCFLG      ;SET FLAG
 1020  013464  013737  011020  011066          MOV     IRDA,   RDA     ;SETUP RCV DATA ADDR
 1021  013472  012737  001000  011064          MOV     #1000,  RCC     ;SETUP RCV CHAR COUNT
 1022  013500  042700  040000                  BIC     #RFLG,STAT      ;CLEAR RFLG
 1023  013504  005037  011054                  CLR     ERCSR           ;RESET ERROR RECORDS
 1024  013510  005037  011056                  CLR     ERDBR
 1025  013514  005764  000002                  TST     RBUF(R4)
 1026  013520  052714  000143                  BIS     #RIE+DTR+DIE+RE,@RCSR;SET INTERRUPT ENABLES
 1027  013524  000207                          RTS     PC              ;EXIT FROM SUBROUTINE
 1028
 1029  013526  000000          TEMP1:  0
 1030  013530  000000          TEMP2:  0
 1031  013532  000000          SNCFLG: 0
 1032  013534  000000          TRNFLG: 0
```

```
                                        ;*****************************************************************
                                        ;       TRANSMIT INTERRUPT SERVICE ROUTNE
                                        ;*****************************************************************
013536  000240                  XISR:   NOP
013540  127737  175324  011040          CMPB    @XDA,TX.TERM    ;FINISHED XMITTING?
013546  001010                          BNE     XISR1           ;BR IF NO
013550  052700  100000                  BIS     #XFLG,  STAT    ;SET XMIT COMPLETE FLAG
013554  042714  000004                  BIC     #RQTS,  @RCSR   ;RESET REQUEST TO SEND
013560  042764  000100  000004          BIC     #TIE,   XCSR(R4);RESET XMIT INTERRUPT ENABLE
013566  000417                          BR      XISR2           ;
013570  117764  175274  000006  XISR1:  MOVB    @XDA,   XBUF(R4);XMIT NEXT CHAR.
013576  032777  000100  175240          BIT     #100,   @SWR    ;MONITOR OUTPUT?
013604  001406                          BEQ     NOXMON          ;BR IF NO
013606  105777  175264                  TSTB    @TPS            ;IS TTY AVAILABLE
013612  100003                          BPL     NOXMON          ;BR IF NO
013614  117777  175250  175256          MOVB    @XDA,   @TPB    ;TYPE THE CHAR
013622                          NOXMON:
013622  005237  011070                  INC     XDA             ;INCRMENT ADDRESS
013626  005737  013534          XISR2:  TST     TRNFLG          ;FIRST CHAR?
013632  001403                          BEQ     1$              ;BR IF NO
013634  042764  000100  000004          BIC     #TIE,XCSR(R4)   ;CLEAR INTERUPT ENABLE
013642  005037  011032          1$:     CLR     TIME            ;RESET TIMER
013646  005037  013534                  CLR     TRNFLG          ;CLEAR FLAG
013652  000002                          RTI                     ;RETURN FROM INTERRUPT
                                        ;*****************************************************************
                                        ;       RECEIVE INTERRUPT SERVICE ROUTINE
                                        ;*****************************************************************
013654  000240                  RISR:   NOP
013656  105714                          TSTB    @RCSR           ;IS RECEIVER DONE BIT SET
013660  100055                          BPL     RISR2           ;BR IF NO
013662  116401  000002                  MOVB    RBUF(R4),R1     ;STORE CHAR.
013666  042701  000200                  BICB    #200,   R1      ;STRIP A BIT
013672  110177  175170                  MOVB    R1,     @RDA    ;MOVE CHAR TO INBUF
013676  032777  000040  175140          BIT     #40,    @SWR    ;MONITOR INPUT?
013704  001405                          BEQ     NORMON          ;BR IF NO
013706  105777  175164                  TSTB    @TPS            ;IS TTY AVAILABLE?
013712  100002                          BPL     NORMON          ;BR IF NO
013714  110177  175160                  MOVB    R1,     @TPB    ;TYPE THE CHAR
013720                          NORMON:
013720  005237  011066                  INC     RDA             ;BUMP POINTER
013724  105077  175136                  CLRB    @RDA            ;CLEAR NEXT CHAR POSITION
013730  005337  011064                  DEC     RCC             ;DECREMENT CHAR. COUNTER
013734  001010                          BNE     1$              ;BR IF BUFFER NOT FULL
013736  042714  000100                  BIC     #RIE,   @RCSR   ;RESET INTERRUPT ENAB
013742  011402                          MOV     @RCSR,  R2      ;SETUP RCV CSR
013744  016403  000004                  MOV     XCSR(R4),R3     ;SETUP XMIT CSR
013750  104006                          HLT+6                   ;RECEIVER BUFFER FULL
013752  004737  013374                  JSR     PC,STARTR       ;INITIALIZE RECEIVER
013756  123701  011041          1$:     CMPB    RX.TERM,R1      ;IS IT LINE FEED?
013762  001004                          BNE     RISR1           ;BR IF NO
013764  042714  000140                  BIC     #RIE+DIE,@RCSR  ;DISABLE INTERRUPTS
013770  052700  040000                  BIS     #RFLG,  STAT    ;SET RCVR COMPLETE FLAG
013774  005764  000002          RISR1:  TST     RBUF(R4)        ;IS THERE A DATA ERROR
014000  100005                          BPL     RISR2           ;BR IF NO
```

```
1089  014002  011437  011054                MOV     @RCSR, ERCSR    ;SAVE RCSR
1090  014006  016437  000002  011056        MOV     RBUF(R4),ERDBR  ;SAVE RDBR
1091  014014  005714                RISR2:  TST     @RCSR           ;IS THERE A DATA SET STATUS CHANGE
1092  014016  100004                        BPL     RISR3           ;BR IF NO
1093  014020  011437  011060                MOV     @RCSR, DSSTAT   ;SAVE STATUS
1094  014024  052700  020000                BIS     #DSFLG, STAT    ;SET FLAG
1095  014030  005037  013532        RISR3:  CLR     SNCFLG          ;CLEAR FLAG
1096  014034  005037  011032                CLR     TIME            ;RESET TIMER
1097  014040  000002                        RTI                     ;RETURN FROM INTERRUPT
1098
1099          000001                        .END
```

```
BA      011004          581#    640
BACK    013056          701*    729*    764*    844*    875     936#
BIT0  = 000001          575#
BIT1  = 000002          575#
BIT10 = 002000          575#
BIT11 = 004000          575#
BIT12 = 010000          575#
BIT13 = 020000          575#    610#    886     913
BIT14 = 040000          575#
BIT15 = 100000          575#
BIT2  = 000004          575#
BIT3  = 000010          575#
BIT4  = 000020          575#
BIT5  = 000040          575#
BIT6  = 000100          575#
BIT7  = 000200          575#
BIT8  = 000400          575#
BIT9  = 001000          575#
B2016   011030          591#    889
CD    = 010000          575#
CTS   = 020000          575#    968
CTSOK   013260          966     982#
CTSW    013206          968#    971     978
DELAY   013054          653*    873*    935#    944     952*
DERR    012466          914     923#
DIE   = 000040          575#   1026    1085
DISPLA  011046          601#
DL11    011000          580#
DSC   = 100000          575#
DSFLG = 020000          575#    609#   1094
DSSTAT  011060          616#   1093*
DTR   = 000002          575#    954     965    1008    1026
ENTER   012226          857     866#
EOP     012140          702     730     765     845     852#
ER    = 100000          575#
ERCSR   011054          614#   1023*   1089*
ERDBR   011056          615#    884    1024*   1090*
FE    = 020000          575#
FLAG    011042          599#
FULL.D= 000001          628#    804     820     825     960     991    1012
GO      011164          652#
ILB   = 000010          575#    661
IRDA    011020          587#    768     893     916    1020
IXDA    011022          588#    769     892     920     922     953
KBDIN = 104416          575#    683     714     746     803
LOOP  = 000020          575#    699     727     762     842
MSG0    012500          888     929#
MSG1    012561          891     929#
MSG2    012564          915     929#
MSG3    012611          919     929#
MSG4    012634          929#    957
MSG5    012766          929#   1009
NODAT = 000200          575#    695     758     839
NORMON  013720         1068    1070    1072#
NOXMON  013622         1046    1048    1050#
OR    = 040000          575#
```

```
OWI    = 000002            575#    658
OWO    = 000001            575#    655
PARAM1   011012            584#    641
PARAM2   011014            585#    642    804    820    825    960    991   1012
PARAM3   011016            586#
PE     = 010000            575#
PRIOR    011010            583#    637    639
PRTY0  = 000000            575#
PRTY1  = 000040            575#
PRTY2  = 000100            575#
PRTY3  = 000140            575#
PRTY4  = 000200            575#
PRTY5  = 000240            575#
PRTY6  = 000300            575#
PRTY7  = 000340            575#    853
QTPIE    012276            854*   855*    874    876#
RA     = 004000            575#
RBUF   = 000002            575#   1025   1064   1087   1090
RCC      011064            619#   1021*  1075*
RD     = 000200            575#
RDA      011066            620#   1020*  1066*  1073*  1074*
RDAX     012446            916*    918#
RE     = 000001            575#   1026
RFLG   = 040000            608#    685    698    749    761    811    837   1022   1086
RI     = 040000            575#
RIE    = 000100            575#   1026   1077   1085
RISR     013654             636   1061#
RISR1    013774            1084   1087#
RISR2    014014            1063   1088   1091#
RISR3    014030            1092   1095#
RIV      011006             582#    635
RQTS   = 000004            575#    964   1040
RWAIT  = 104410            575#
RX.TER   011041            597#   1083
SAVR0    013040            858*    867    929#
SAVR1    013042            859*    868    930#
SAVR2    013044            860*    869    931#
SAVR3    013046            861*    870    932#
SAVR4    013050            862*    871    933#
SAVR5    013052            863*    872    934#
SCAN4    012344            894#    895    908    911
SCAN5    012420            912#
SETTLE   011024            589#    982    990*  1010*
SNCFLG   013532            998   1019*  1031#  1095*
SRCSR    011052            613#
SRD    = 002000            575#
START    011102            594    632#
STARTR   013374            684    747    806    823   1006#  1081
STARTX   013062            715    778    807    943#
STD    = 000010            575#
STOP     013060            654*   937#    955    959*  1006   1011*
STPS   = 104414            575#    853
SWR      011044            600#    633    695    699    727    758    762    766    839    842    886    913    974
                          1045   1067
SW12   = 010000             575#    974
SXCSR    011050             612#
```

```
TEMP1   013526          946*   948*   984*   986*  1016*  1017*  1029#
TEMP2   013530          947*   950*   985*   988*  1030#
TESTD   012300          697    760    841    884#
TESTDX  012474          897    903    906    926#
TIE   = 000100          575#   855    856    996   1000   1041   1054
TIME    011032          592#   652*   687    692*   716*   719    724*   748*   751    756*   777*   781    786*
                        808*   813    818*   829    834*   838*   970    977*  1055*  1096*
TKB     011074          624#
TKS     011072          623#
TPB     011100          626#  1049*  1071*
TPS     011076          625#  1047   1069
TR    = 000200          575#
TRNFLG  013534          995*  1032#  1052   1056*
TSTDAT  012334          885    887    892#
TX.TER  011040          595#   896   1037
TYPE  = 104400          575#   888    891    901    915    917    919    921    957   1009
XBUF  = 000006          575#  1044*
XCC     011062          618#
XCSR  = 000004          575#   644*   690    722    754    784    816    832    854    856*   874*   973    996*
                       1000*  1041*  1054*  1079
XDA     011070          621#   770*   953*  1037   1044   1049  1051*
XFLG  = 100000          607#   717    726    779    788    809    822    827    836   1039
XISR    013536          638   1036#
XISR1   013570         1038   1044#
XISR2   013626         1042   1052#
XLB   = 000004          575#   664    993   1014
XWAIT = 104412          575#
$ILB    011460          663    746#   789
$OWI    011254          660    683#   703
$OWO    011362          657    714#   731
$XLB    011710          666    803#   843   844
.     = 014042          579#   668    772   920*   949    951    963    987    989    999   1018
```

```
DL11 ITEP OVERLAY       MACY11 27(1006)  29-OCT-76  14:36  PAGE 30
DZDLOC.P11    29-OCT-76 10:28           CROSS REFERENCE TABLE -- MACRO NAMES

BOX          1#    576     603     629     939    1003    1033    1058
DCPARM       1#
DHDOC1       1#
DHPARM       1#
DJPARM       1#
DLPARM       1#    561
DPPARM       1#
DQDOC1       1#
DQPARM       1#
DUPARM       1#
DUPPAR       1#
DVDOC1       1#
DVPARM       1#
DZPARM       1#
HELLO        1#
HLT        575#    691     723     755     785     817     833     925     976    1080
SEQUAT       1#    575
SINTF        1#    575
SITEP        1#    645
SSERV        1#    618

. ABS.  014042     000

 ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

,DZDLOC.SEQ/SOL/CRF/NL:TOC=ITEP1.MAC,DZDLOC.P11
RUN-TIME:  10 13  .8 SECONDS
RUN-TIME RATIO: 75/26=2.8
CORE USED:  16K  (31 PAGES)
```

Spooler runtime 3 Seconds, 20 KCS, 9% disk reads, 3 disk writes, 27 pages