IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDJD-B-D
PRODUCT NAME:	DJ11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM
PROGRAM DATE:	OCTOBER 1976
MAINTAINER:	DIAGNOSTICS
AUTHORS:	R A JONES JOHN EGOLF FRY BROWN

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 DJ11 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 PARMS FROM CONSOLE TTY, PROCEED TO SECTION B.
2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE STARTING TO ENTER PARMS 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 CARRAGE RETURN
B. IF YOU WISH TO SETUP A DH11, TYPE IN DH.
C. IF YOU WISH TO SETUP A DH11&8, TYPE IN DH8.

IF DH OR DH8 WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS, VECTOR ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DH11 OR DH8.

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 DH11BB.

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 DH-11B ISR
   SWR13=SETUP DH-11 ISR
   SWR=000000=SETUP VARIABLE ISR

   2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
      SETUP SEQUENCE IS: DH11,DH11-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 (DH11 AND DMB8 PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
      F. GO BACK TO STEP A IF THIS SETUP WAS FOR DH 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
ASCERTAIN 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 DH-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 (SB QUICK BROWN FOX)
        10=1 TEST MESSAGE #2 (SB NUMERICS)
        11=1 TEST MESSAGE #3 (SC CONTEST/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. 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 (SW7=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW4=1) OR
   GO TO STEP 1(SW4=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SW7=0)
3. RETURNS TO MONITOR FOR "END PASS"(SW4=1) OR
   GO TO STEP 1 (SW4=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". (SW4=1) OR
   GO TO STEP 1 (SW4=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 PARMS 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:

#WOOOOOO	TO PRINTOUT THE 8 WORDS AT LOC XXXXXX.
#BOOOOOO	TO PRINTOUT THE 16 BYTES AFTER LOC XXXXXX.
#C	TO CONTINUE

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

5.0 PROGRAM AND/OR OPERATOR ACTION

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

5.1 NORMAL HALTS
SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

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

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

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

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

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

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

7.0 RESTRICTIONS

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

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

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

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

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

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

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

CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING
FOR THE REST OF THE 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 DJ11 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: SOWI ,IF "ONE WAY IN" MODE
WAS SELECTED. SOWO,IF "ONE WAY OUT" MODE WAS SELECTED.
SILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED.
SXLB,IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 SOWI: 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 SOWO: 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 SOWO OR TYPES END PASS
DEPENDING ON SWITCH SETTINGS.

9.33 SILB: 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. (SILB)

9.34 SXLB: 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 SXLB.
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 SXLB 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(OWD,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 DJ11

                PARAM#1 IS THE LINE NUMBER IN OCTAL, DEFAULT= LINE 0

                PARAM#2 BIT 0 IS FULL DUPLEX (1), DEFAULT= FULL DUPLEX (1)

                PARAM#3 IS NOT USED (177777)
```

```
                                ;*****************************************************************
                                ;       DJ11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000          .=11000
011000 046104 000040    DJ11:   .ASCIZ  /DJ /           ;ISR NAME
011004 160010           BA:     160010                  ;BUS ADDRESS
011006 000300           RIV:    300                     ;VECTOR ADDRESS
011010 000240           PRIOR:  240                     ;PRIORITY
011012 000000           PARAM1: 0                       ;PARAM #1
011014 000001           PARAM2: 1                       ;PARAM #2
011016 177777           PARAM3: 177777                  ;PARAM #3
011020 000000           IRDA:   .WORD   0               ;INITIAL READ DATA ADDRESS
011022 000000           IXDA:   .WORD   0               ;INITIAL XMIT DATA ADDRESS
011024 000000           SETTLE: .WORD   0               ;LINE SETTLE DELAY FLAG
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
                                ;*****************************************************************
                                        STAT=R0
000000                                  XFLG=100000             ;XMIT COMPLETE FLAG
100000                                  RFLG=40000              ;RCV COMPLETE FLAG
040000                                  DSFLG=20000             ;DATA SET STATUS CHANGE FLAG
020000                                  BIT13=20000             ;INHIBIT PRINTOUTS
020000

011050 000000           SXCSR:  0                       ;SAVED XMIT CSR
011052 000000           SRCSR:  0                       ;SAVED RCV CSR
011054 000000           ERCSR:  0                       ;RCV CSR SAVED ON ERROR
011056 000000           ERDBR:  0                       ;RCV DATA REG SAVED ON ERROR
011060 000000           DSSTAT: 0                       ;RCV CSR SAVED ON DS CHANGE

011062 000000           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
```

```
                        ;*****************************************************************
                        ;       DJ11-X INTERFACE SERVICE ROUTINE
                        ;*****************************************************************
                        START:  NOP
                                MOV     @SAR,   R0      ;SETUP MODE IN R0
                                BIC     #177400,R0      ;STRIP JUNK
                                MOV     R1Y,    R2      ;SETUP
                                MOV     @R1SR,  (R2)+   ;INTERRUPT
                                MOV     @PRIOR, (R2)+   ;VECTORS
                                MOV     @XCSR,  (R2)+   ;
                                MOV     @PRIOR, (R2)+   ;
                                MOV     R0,     R4      ;SETUP BUS ADDR INDEX
                                MOV     PARAM1, @RCSR   ;SETUP VARIABLES
                                MOV     PARAM2, R2      ;
                                BIC     #0001,R2        ;
                                MOV     R2,     XCSR(R4);IN CSR'S
                                CLR     @RCSR           ;
                                CLR     TCR(R4)         ;
                                BIC     #RCEN,@RCSR     ;CLEAR RECEIVER ENABLE
                                BIS     #MSC,@RCSR      ;CLEAR SILO+UARTS
                        1$:     BIT     #BSY,@RCSR      ;CLEAR PULSE DONE?
                                BNE     1$              ;BR IF NO
                                BIT     #FULL.DUPLEX,PARAM2     ;FULL DUPLEX?
                                BNE     2$              ;BR IF YES
                                BIS     #HD,@RCSR       ;SET HALF DUPLEX
                        2$:     CMPB    PARAM1,#17
                                BLOS    3$
                                HALT    0               ;ERROR HLT
                                HALT                    ;NUMBER IN PARAM1 IS TOO LARGE!
                                SUB     #2              ;MUST BE 17 OR LESS (LINE# IN OCTAL)
                        3$:     MOV     R0,-(SP)        ;SAVE R0
                                MOV     #1,R0
                                MOV     PARAM1,R1
                        4$:     TSTB    R1
                                BEQ     5$              ;CALCULATE TCR BIT
                                ASL     R0
                                DEC     R1
                                BR      4$
                        5$:     MOV     R0,TCRTMP
                                MOV     (SP)+,R0
                                CLR     TIME            ;RESET TIME
                                CLR     DELAY           ;RESET DELAY

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

                        90:     CLR     TIME
                                CLR     DELAY
                                CLR     STOP
                                BIT     #DMD,MODE
                                BEQ     1$
                                JMP     @DMD
                        1$:     BIT     #DUI,MODE
```

```
011336  001402                          BEQ     2$
011340  000137  011374                  JMP     $OWI
011344  032700  000010          2$:     BIT     #ILB,MODE
011350  001402                          BEQ     3$
011352  000137  011600                  JMP     $ILB
011356  032700  000004          3$:     BIT     #XLB,MODE
011362  001402                          BEQ     4$
011364  000137  012030                  JMP     $XLB
011370  000000                  4$:     HALT
011372  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".
                                ;*******************************************************



011374  104416                  $OWI:   KBDIN
011376  004737  012556                  JSR     PC,STARTR
011402  032700  040000          1$:     BIT     #FLG,STAT
011406  001012                          BNE     2$
011410  022737  011032  000100          CMP     TIME,#100
011416  103771                          BLO     1$
011420  011402                          MOV     @RCSR,R2
011422  016403  000004                  MOV     XCSR(R4),R3
011426  104001                          HLT     1
011430  005037  011032                  CLR     TIME
011434  000762                          BR      1$

011436  032777  000200  177400  2$:     BIT     #MODAT,@SWR
011444  001002                          BNE     3$
011446  004737  012420                  JSR     PC,TESTD
011452  042700  040000          3$:     BIC     #FLG,STAT
011456  032777  000020  177360          BIT     #LOOP,@SWR
011464  001405                          BEQ     4$
011466  012737  011500  013176          MOV     #4$,BACK
011474  000137  012250                  JMP     EOP
011500  000736                  4$:     BR      $OWI



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

011502  104416                  $OWO:   KBDIN
```

```
011504  004737  013202                  JSR     PC,STARTX
011510  005037  011032                  CLR     TIME
011514  032700  100000          1$:     BIT     #XFLG,STAT
011520  001013                          BNE     2$
011522  023727  011032  000100          CMP     TIME,#100
011530  103771                          BLO     1$
011532  011402                          MOV     @RCSR,R2
011534  016403  000004                  MOV     XCSR(R4),R3
011540  104001                          HLT     1
011542  005037  011032                  CLR     TIME
011546  000762                          BR      1$
011550  042700  100000          2$:     BIC     #XFLG,STAT
011554  032777  000020  177262          BIT     #LOOP,@SWR
011562  001405                          BEQ     3$
011564  012737  011576  013176          MOV     #3$,BACK
011572  000137  012260                  JMP     EOP
011576  000741                  3$:     BR      SOWO
```

E02


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

011600  104416                     SILB:   KBDIN
011602  004737  013566                     JSR     PC,STARTR
011606  005037  011032                     CLR     TIME
011612  032700  040000             1$:     BIT     #RFLG,STAT
011616  001013                             BNE     2$
011620  023727  011032  000100             CMP     TIME,#100
011626  103771                             BLO     1$
011630  011402                             MOV     @RCSR,R2
011632  016403  000004                     MOV     XCSR(R4),R3
011636  104001                             HLT     1
011640  005037  011032                     CLR     TIME
011644  000762                             BR      1$
011646  032777  000200  177170     2$:     BIT     #NODAT,@SWR
011654  001002                             BNE     3$
011656  004737  012420                     JSR     PC,TESTD
011662  042700  040000             3$:     BIC     #RFLG,STAT
011666  032777  000020  177150             BIT     #LOOP,@SWR
011674  001405                             BEQ     4$
011676  012737  011710  013176             MOV     #4$,BACK
011704  000137  012260                     JMP     EOP
011710  032777  000400  177126     4$:     BIT     #400,   @SWR        ;USE EXTERNAL DATA?
011716  001416                             BEQ     7$                  ;BR IF NO
011720  013702  011020                     MOV     IRDA,   R2          ;SET POINTER
011724  013703  011022                     MOV     IXDA,   R3          ;SET POINTER
011730  010337  011070                     MOV     R3,     XDA         ;SETUP XMIT DATA ADDR
011734  112223                             MOVB    (R2)+,  (R3)+       ;MOVE INPUT TO OUTPUT
011736  001376                             BNE     .-2                 ;LOOP IF NOT ZERO CHAR
011740  112743  000177                     MOVB    #177,   -(R3)       ;INSERT A FILL CHAR
011744  005203                             INC     R3                  ;BUMP ADDRESS
011746  112723  000177                     MOVB    #177,   (R3)+       ;INSERT ANOTHER FILL
011752  105023                             CLRB    (R3)+               ;INSERT ZERO CHAR
011754  005037  011032             7$:     CLR     TIME
011760  004737  013202                     JSR     PC,STARTX
011764  032700  100000             5$:     BIT     #XFLG,STAT
011770  001013                             BNE     6$
011772  023727  011032  000100             CMP     TIME,#100
012000  103771                             BLO     5$
012002  011402                             MOV     @RCSR,R2
012004  016403  000004                     MOV     XCSR(R4),R3
012010  104001                             HLT     1
012012  005037  011032                     CLR     TIME
012016  000762                             BR      5$
012020  042700  100000             6$:     BIC     #XFLG,STAT
012024  000137  011600                     JMP     SILB
```

```
                        ;*****************************************************************
                        ;       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.
                        ;*****************************************************************
012020  104416          SXLB:   KBDIN
012022  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
012030  001402                  BEQ     1$
012032  004737  013566          JSR     PC,STARTR
012036  004737  012202  1$:     JSR     PC,STARTX
012042  005037  011032          CLR     TIME
012046  032700  100000  2$:     BIT     #XFLG,STAT
012052  001016                  BNE     3$
012054  032700  040000  7$:     BIT     #RFLG,STAT
012060  001004                  BNE     4$
012062  023727  011032  000100          CMP     TIME,#100
012070  103766                  BLO     2$
012072  011402                  MOV     @RCSR,R2
012074  016403  000004          MOV     XCSR(R4),R3
012100  104001                  WAIT
012102  005037  011032          CLR     TIME
012106  000757                  BR      2$
012110  032737  000001  011014  3$:     BIT     #FULL.DUPLEX,PARAM2
012116  001366                  BNE     7$
012120  032700  100000          BIT     #XFLG,STAT
012124  004737  013566          JSR     PC,STARTR
012130  000746                  BR      2$
012132  032737  000001  011014  4$:     BIT     #FULL.DUPLEX,PARAM2
012140  001420                  BEQ     6$
012142  032700  100000          BIT     #XFLG,STAT
012146  001013                  BNE     6$
012150  023727  011032  000100          CMP     TIME,#100
012156  103766                  BLO     4$
012160  011402                  MOV     @RCSR,R2
012162  016403  000004          MOV     XCSR(R4),R3
012166  104001                  WAIT
012170  005037  011032          CLR     TIME
012174  000746                  BR      4$
012176  032700  100000  6$:     BIT     #XFLG,STAT
012202  040000  8$:     BIC     #RFLG,STAT
012206  005037  011032          CLR     TIME
012212  032777  000200  176614          BIT     #MODAT,@SAR
012220  001002                  BNE     9$
012222  004737  012420          JSR     PC,TESTD
012226  032777  000020  176600  9$:     BIT     #LOOP,@SAR
012234  001671                  BEQ     SXLB
012236  012737  012030  013176          MOV     #SXLB,BACK
012244  000137  012260          JMP     EOP
```

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

012260                          EOP:
012260  104414  000340                  STPS,PRTY7              ;SET PS PRIORITY TO 7
012264  016437  000004  012416          MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012272  042737  177777  012416          BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012300  042764  040000  000004          BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012306  012766  012346  000002          MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012314  010037  013160                  MOV     R0,SAVR0        ;SAVE REGISTER 0
012320  010137  013162                  MOV     R1,SAVR1        ;SAVE REGISTER 1
012324  010237  013164                  MOV     R2,SAVR2        ;SAVE REGISTER 2
012330  010337  013166                  MOV     R3,SAVR3        ;SAVE REGISTER 3
012334  010437  013170                  MOV     R4,SAVR4        ;SAVE REGISTER 4
012340  010537  013172                  MOV     R5,SAVR5        ;SAVE REGISTER 5
012344  000207                          RTS     PC              ;RETURN TO CONTROL PROGRAM

012346                          ENTER:
012346  013700  013160                  MOV     SAVR0,R0        ;RESTORE R0
012352  013701  013162                  MOV     SAVR1,R1        ;RESTORE R1
012356  013702  013164                  MOV     SAVR2,R2        ;RESTORE R2
012362  013703  013166                  MOV     SAVR3,R3        ;RESTORE R3
012366  013704  013170                  MOV     SAVR4,R4        ;RESTORE R4
012372  013705  013172                  MOV     SAVR5,R5        ;RESTORE R5
012376  012737  177777  013174          MOV     #-1,DELAY
012404  052764  012416  000004          BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012412  000177  000560                  JMP     @BACK
012416  000000                  QTPIE:  000000



                                ;*********************************
                                ;       SUBROUTINE TO CHECK
                                ;       RECEIVER DATA.
                                ;*********************************
012420  013746  011086          TESTD:  MOV     ERRBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012424  001413                          BEQ     TSTDAT          ;BR IF NO
012426  032777  020000  176410          BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012434  001002                          BNE     TSTDAT          ;BR IF YES
012436  104400  012620                  TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012442  004077  176362                  JSR     R0,@#2016       ;PRINT CONTENTS OF RBUF
012446  005746                          TST     -(SP)
012450  104400  012701                  TYPE    ,MSG1           ;<15><12>
012454  013701  011088          TSTDAT: MOV     1XDA,   R1      ;SETUP XMIT DATA ADDR
012460  013702  011020                  MOV     1RDA,   R2      ;SETUP RCV DATA ADDR
012464  122122                  SCANH:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012466  001776                          BEQ     SCANH           ;BR IF OK
012470  123741  011040                  CMPB    TX,TERM,-(R1)   ;IS IT END OF DATA
012474  001447                          BEQ     TESTDX          ;BR IF YES
012476  122742  000002                  CMPB    #002,-(R2)
012502  001005                          BNE     2$
012504  010237  012512                  MOV     R2,1$
012510  104400                          TYPE
```

```
012512  000000                     1$:     .WORD   0
012514  000437                             BR      TESTDX
012516                             2$:
012516  105712                             TSTB    (R2)                   ;
012520  001435                             BEQ     TESTDX                 ;BR IF YES
012522  122721  000177                     CMPB    #177,   (R1)+          ;IS IT FILL CHAR?
012526  001756                             BEQ     SCANA                  ;BR IF YES
012530  005301                             DEC     R1                     ;BACKUP
012532  122722  000177                     CMPB    #177,   (R2)+          ;IS IT FILL?
012536  001752                             BEQ     SCANA                  ;BR IF YES
012540  000240                     SCAN5:  NOP                            ;DATA ERROR
012542  032777  020000  176274             BIT     #BIT13,@SWR            ;INHIBIT PRINTOUTS
012550  001016                             BNE     DERR                   ;BR IF YES
012552  104400  012704                     TYPE    ,MSG2                  ;<15><12>RECEIVED DATA = <15><12>
012556  013737  011020  012566             MOV     IRDA,   RDAX           ;SETUP DATA ADRESS
012564  104400                             TYPE                           ;PRINT RECEIVED DATA
012566  000000                     RDAX:   0                              ;RECEIVED DATA ADDR.
012570  104400  012731                     TYPE    ,MSG3                  ;<15><12>DATA SHOULD BE<15><12>
012574  013737  011022  012604             MOV     IXDA,   .+10           ;SETUP ADDR.
012602  104400                             TYPE                           ;PRINT GOOD DATA
012604  011022                             IXDA
012606  111103                     DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
012610  114202                             MOVB    -(R2),R2       -       ;SETUP RCV DATA
012612  104007                             HLT+7                          ;DATA ERROR HALT
012614  005726                     TESTDX: TST     (SP)+                  ;POP STACK
012616  000207                             RTS     PC                     ;RETURN FROM SUB/ROUT

012620  005015  044124  051105     MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
012701     015  000012             MSG1:   .ASCIZ  <15><12>
012704  005015  042522  042503     MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
012731     015  042012  052101     MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
012754  005015  046120  040505     MSG4:   .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
013023     015  063412  042510             .ASCIZ <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
013106  005015  046120  040505     MSG5:   .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
                                   .EVEN
013160  000000                     SAVR0:  0
013162  000000                     SAVR1:  0
013164  000000                     SAVR2:  0
013166  000000                     SAVR3:  0
013170  000000                     SAVR4:  0
013172  000000                     SAVR5:  0
013174  000000                     DELAY:  0
013176  000000                     BACK:   0
013200  000000                     STOP:   0
```

1

```
                        ;*****************************************************************
                        ;       TRANSMITTER INITIALIZATION SUBROUTINE
                        ;*****************************************************************

013172  005737  013174          STARTX: TST     DELAY           ;IF SW04=1 & SW14=0 DELAY
013176  001416                          BEQ     1$              ;NO DELAY START TRANSMITTER
013200  005037  013554                  CLR     TEMP1           ;PREPARE FOR DELAY
013204  012737  000007  013556          MOV     #7,TEMP2
013212  062737  000001  013554          ADD     #1,TEMP1        ;INCREMENT DELAY
013220  001374                          BNE     .-6
013222  005337  013556                  DEC     TEMP2
013226  001371                          BNE     .-14
013230  005037  013174                  CLR     DELAY
013234  042700  100000          1$:     BIC     #XFLG,STAT
013240  013737  011022  011070          MOV     IXDA,XDA        ;SET UP XMIT DATA ADD
013246  005737  013200                  TST     STOP            ;FIRST TIME HERE?
013252  001021                          BNE     2$              ;NO
013254  104400  012754                  TYPE    ,MSG4           ;MAKE CONNECTION
013260  000000                          HALT
013262  005137  013200                  COM     STOP            ;COMPLEMENT STOP
013266  005037  013554                  CLR     TEMP1           ;YES PREPARE FOR DELAY
013272  012737  000030  013556          MOV     #14*2,TEMP2
013300  062737  000001  013554          ADD     #1,TEMP1        ;INCREMENT DELAY
013306  001374                          BNE     .-6
013310  005337  013556                  DEC     TEMP2
013314  001371                          BNE     .-14
013316  052714  002000          2$:     BIS     #BANK,@RCSR     ;SELECT TCR
013322  013764  013560  000004          MOV     TCRTMP,TCR(R4)  ;SET LINE # IN TCR
013330  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
013336  001414                          BEQ     3$              ;BR IF NO
013340  032700  000004                  BIT     #XLB,MODE       ;XLB MODE?
013344  001411                          BEQ     3$              ;BR IF NO
013346  012737  177777  013564          MOV     #-1,TRNFLG      ;SET FLAG
013354  052714  040400                  BIS     #TIE+MTSE,@RCSR ;SET INTERUPT ENABLE
013360  000001                          WAIT
013362  005737  013562                  TST     SNCFLG          ;FIRST CHAR RECEIVED YET?
013366  001375                          BNE     .-4             ;BR IF NO
013370  052714  040400          3$:     BIS     #TIE+MTSE,@RCSR ;SET INTERUPT ENABLE,SCAN ENABLE
013374  000207                  4$:     RTS     PC

013376  127737  175456  011040  XISR:   CMPB    @XDA,TX.TERM    ;IS CHAR TRANSMITTER TERMINATION CHAR
013404  001005                          BNE     XISR1           ;BR IF NO
013406  052737  100000  011040          BIS     #XFLG,STAT      ;SET XMIT DONE FLAG
013414  042714  040400                  BIC     #TIE+MTSE,@RCSR ;CLEAR ENABLES
013420  000444                          BR      XISR3
013422  005737  011012  XISR1:  TST     PARAM1          ;IS IT LINE 0
013426  001015                          BNE     XISR2           ;BR IF NO
013430  105764  000007                  TSTB    XBUF+1(R4)      ;DID SCAN STOP AT LINE 0
013434  001412                          BEQ     XISR2           ;BR IF YES
013436  016437  000006  013552          MOV     XBUF(R4),ERROR1 ;SAVE XBUF
013444  011402                          MOV     @RCSR,R2        ;SET UP R2 WITH SCR CONTENTS
013446  005003                          CLR     R3
013450  104010                          MLT     10              ;ERROR WRONG LINE
013452  104400  014136                  TYPE    ,SCANE          ;TYPE ERROR MESSAGE
013456  000000                          HALT                    ;HALT
013460  000776                          BR      .-2
```

```
1017  013470  117764  175374  000006  XISR2:  MOVB    2XDA,XBUF(R4)   ;TRANSMIT DATA
1018  013476  032777  000100  175340          BIT     #100,@RCSR      ;MONITOR TX DATA?
1019  013504  001406                          BEQ     NOXMON          ;BR IF NO
1020  013506  105777  175364                  TSTB    @TPS            ;TTY READY?
1021  013512  100003                          BPL     NOXMON          ;BR IF NO
1022  013514  117777  175350  175356          MOVB    @XDA,@TPB       ;TYPE CHAR
1023  013522  005267  011070          NOXMON: INC     XDA             ;INC TXBUF POINTER
1024  013526  005737  013564                  TST     TRNFLG
1025  013532  001402                          BEQ     XISR3
1026  013534  042714  040000                  BIC     #TIE+MTSE,@RCSR
1027  013540  005037  011062          XISR3:  CLR     TIME
1030  013544  005037  013564                  CLR     TRNFLG
1031  013550  000002                          RTI
1032  013552  000000          ERROR1: 0
1033  013554  000000          TEMP1:  0
1034  013556  000000          TEMP2:  0
1035  013560  000000          TCRTMP: 0
1036  013562  000000          SNCFLG: 0
1036  013564  000000          TRNFLG: 0
```

```
                        ;*****************************************************************
                        ;       RECEIVER INITIALIZATION SUBROUTINE
                        ;*****************************************************************

013564  005737  013200       STARTR: TST     STOP            ;FIRST TIME HERE?
013570  001005                       BNE     1$              ;BR IF NO
013572  104400  012754               TYPE    ,MSG4           ;TYPE"MAKE CONNECTION"
013576  005137  013200               COM     STOP            ;COMPLEMENT STOP
013602  000000                       HALT
013604  032737  000001  011014 1$:   BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
013614  001410                       BEQ     2$              ;BR IF NO
013616  032700  000004               BIT     #XLB,MODE       ;XLB MODE?
013624  001406                       BEQ     2$              ;BR IF NO
013626  005037  012654               CLR     TEMP1           ;START DELAY
013632  005237  012654               INC     TEMP1
013636  001375                       BNE     .-4
013640  042700  040000       2$:     BIC     #RFLG,STAT
013644  013737  011020  011066       MOV     IRDA,NDA        ;SET UP RECEIVER DATA ADD
013652  012737  001000  011064       MOV     #1000,RCC       ;SET UP BUFFER LIMIT
013660  012737  177777  013662       MOV     #-1,SNCFLG
013666  005037  011054               CLR     ERCSR           ;CLEAR ERROR RECORDS
013672  005037  011056               CLR     ERDBR
013676  005764  000002       2$:     TST     RBUF(R4)        ;CLEAR SILO
013702  100775                       BMI     2$              ;KEEP CLEARING UNTIL BIT 15 CLEAR
013704  052714  000101               BIS     #RCEN+RIE,@RCSR ;SET INTERUPT ENABLE,RECEIVER ENABLE
013710  000207                       RTS     PC

013712  000240              RISR:    NOP
013714  105714                       TSTB    @RCSR           ;DID RECEIVER DONE SET?
013716  100403                       BMI     1$              ;BR IF YES
013720  011402                       MOV     @RCSR,R2        ;SAVE CSR
013722  005003                       CLR     R3
013724  104010                       MLT     10              ;ERROR RECEIVER INTERUPTED BUT DONE NOT SET
013726  016401  000002       1$:     MOV     RBUF(R4),R1     ;GET CHAR
013732  042701  000200               BIC     #200,R1         ;STRIP A BIT
013736  005701                       TST     R1              ;IS CHAR PRESENT SET
013740  100403                       BMI     2$              ;BR IF YES
013742  011402                       MOV     @RCSR,R2        ;SAVE CSR
013744  005003                       CLR     R3
013746  104010                       MLT     10              ;ERROR CHAR PRESENT NOT SET
013750  032701  070000       2$:     BIT     #ORUN+FRAME+PARE,R1     ;CHECK FOR RECEIVER ERRORS
013754  001404                       BEQ     3$              ;BR IF NO ERRORS
013756  011437  011054               MOV     @RCSR,ERCSR     ;SAVE CSR
013762  010137  011056               MOV     R1,ERDBR        ;SAVE RBUF
013766  110177  175076       3$:     MOVB    R1,@TPB         ;STORE CHAR
013772  032777  000040  175046       BIT     #BITS,@SSR      ;MONITOR RXDATA?
014000  001405                       BEQ     NORMON          ;BR IF NO
014002  105777  175072               TSTB    @TPS            ;IS TTY READY?
014006  100002                       BPL     NORMON          ;BR IF NO
014010  110177  175066               MOVB    R1,@TPB         ;TYPE CHAR
014014  005237  011060       NORMON: INC     RDA             ;INC RBUF POINTER
014020  105077  175064               CLRB    @RDA            ;CLEAR NEXT POSITION
014024  005337  011064               DEC     RCC             ;DEC CHAR COUNT
014030  001007                       BNE     1$              ;BUFFER FULL YET?
014032  000005                       RESET                   ;STOP THE SHOW,BUFFER OVERFLOWED
014034  005002                       CLR     R2
```

```
1092  014034  005003                          CLR     R3
1093  014036  104000                          HLT     0
1094  014040  104006                          HLT+6                   ;RECEIVER BUFFER FULL
1095  014042  000000                          HALT
1096  014044  000776                          BR      .-2
1097  014046  123701  011041          1$:     CMPB    RX.TERM,R1      ;IS CHAR 001?
1098  014052  001004                          BNE     RISR1           ;BR IF NO
1099  014054  042714  000100                  BIC     #RIE,@RCSR      ;CLEAR RECEIVER INTERUPT ENABLE
1100  014060  052700  040000                  BIS     #RFLG,STAT      ;SET R DONE FLAG
1101  014064  005037  011032          RISR1:  CLR     TIME
1102  014070  005037  013662                  CLR     SNCFLG
1103  014074  000002                          RTI                     ;GO HOME
1104  014076  005015  042522  042503  NFULL:  .ASCIZ<15><12>/RECEIVER BUFFER FULL ERROR!!/
      014136     015  042412  051122  SCRNE:  .ASCIZ<15><12>/ERROR!TRANSMITTER SCAN STOPPED ON WRONG LINE/
                                      .EVEN
                  000001                  .END
```

```
BA          011004      577#    636
BACK        013176      723#    751#    786#    866#    897     958#
BIT0     =  000001      571#
BIT1     =  000002      571#
BIT10    =  002000      571#
BIT11    =  004000      571#
BIT12    =  010000      571#
BIT13    =  020000      571#    606#    908     935
BIT14    =  040000      571#
BIT15    =  100000      571#
BIT2     =  000004      571#
BIT3     =  000010      571#
BIT4     =  000020      571#
BIT5     =  000040      571#    1081
BIT6     =  000100      571#
BIT7     =  000200      571#
BIT8     =  000400      571#
BIT9     =  001000      571#
BSY      =  000020      571#    645
B2016       011030      587#    911
DELAY       013174      666#    675#    896#    957#    965     973#
DERR        012606      936     945#
DISPLA      011046      597#
DJ11        011000      576#
DSFLG    =  020000      571#    605#
DSSTAT      011060      612#
ENTER       012346      879     888#
EOP         012260      724     752     787     867     874#
ERCSR       011054      610#    1056#   1078#
ERDBR       011056      611#    906     1057#   1079#
ERROR1      013552      1010#   1030#
FLAG        011042      595#
FRAME    =  020000      571#    1076
FULL.D   =  000001      624#    647     826     842     847     989     1045
GO          011304      674#
HO       =  000002      571#    649
ILB      =  000010      571#    683
IRDA        011020      583#    790     915     938     1053
IXDA        011022      584#    791     914     942     974     975
KBDIN    =  104416      571#    705     736     758     825
LOOP     =  000020      571#    721     749     784     864
MFULL       014076      1104#
MOSC     =  000010      571#    644
MSG0        012620      910     951#
MSG1        012701      913     951#
MSG2        012704      937     951#
MSG3        012731      941     951#
MSG4        012754      951#    978     1042
MSG5        013106      951#
MTSE     =  000400      571#    994     998     1004    1026
NODAT    =  000200      571#    717     780     861
NORMON      014012      1082    1084    1086#
NOXMON      013522      1019    1021    1023#
ONLN     =  040000      571#    1076
OW1      =  000002      571#    680
OW0      =  000001      571#    677
```

N02


```
PARAM1    011012      580#   637    650    657    1006
PARAM2    011014      581#   638    647    826    842    847    989    1045
PARAM3    011016      582#
PARE    = 010000      571#   1076
PRIOR     011010      579#   633    635
PRTY0   = 000000      571#
PRTY1   = 000040      571#
PRTY2   = 000100      571#
PRTY3   = 000140      571#
PRTY4   = 000200      571#
PRTY5   = 000240      571#
PRTY6   = 000300      571#
PRTY7   = 000340      571#   875
QTPIE     012416      876#   877#   896    898#
RBUF    = 000002      571#   1058   1069
RCC       011064      615#   1054#  1080#
RCEN    = 000001      571#   643    1060
RDA       011066      616#   1053#  1080#  1086#  1087#
RDAX      012???      938#   940#
RFLG    = 0?0100      604#   707    720    771    783    833    859    1052   1100
RIE     = 000100      571#   1060   1099
RISR      013710      632    1063#
RISR1     014064      1098   1101#
RIV       011006      578#   631
RWAIT   = 104410      571#
RWBK    = 002000      571#   987
RX.TER    011041      593#   1097
SAVR0     013160      880#   889    951#
SAVR1     013162      881#   890    952#
SAVR2     013164      882#   891    953#
SAVR3     013166      883#   892    954#
SAVR4     013170      884#   893    955#
SAVR5     013172      885#   894    956#
SCANE     014135      1014   1104#
SCAN4     012464      916#   917    930    933
SCAN5     012540      934#
SETTLE    011084      595#
SNCFLG    013??2      [illegible]   1034#  1055#  1102#
SRCSR     011052      609#
START     011102      590    628#
STARTR    013???      705    769    828    845    1040#
STARTX    013202      737    800    829    9?5#
STOP      013200      676#   959#   976    980#   1040   1043#
STPS    = 104414      571#   875
SWR       011044      596#   629    717    721    749    780    784    788    861    864    908    926    1018
                      1081
SW12    = 010000      571#
SXCSR     011050      608#
TCR     = 000004      571#   642#   988#
TCRTMP    013560      663#   988    1033#
TEMP1     013554      967#   969#   981#   983#   1031#  1049#  1060#
TEMP2     013556      968#   971#   982#   985#   1032#
TESTD     012420      719    782    863    906#
TESTDX    012614      919    925    929    948#
TIE     = 040000      571#   877    878    994    998    1004   1026
TIME      011032      588#   665#   674#   709    714#   730#   741    746#   770#   773    778#   799#   803
```

```
                            808#   820#   835    840#   851    856#   860#   1027#  1101#
TKB          011074         520#
TKS          011072         519#
TPB          011100         522#   1022#  1085#
TPS          011076         521#   1080   1082
TRNFLG       01[illegible]  952#   1025   1026#  1035#
TSTOBT       01[illegible]  907    909    914#
TX.TER       011040         911#   910    1001
TYPE       = 100000         571#   910    912    922    927    939    941    943    978    1014   1042
XBUF       = 000006         571#   1008   1010   1017#
XCC          011062         614#
XCSR       = 00000[illegible] 571# 640#   712    744    776    806    838    854    876    878#   896#
XDA          011070         617#   752#   975#   1001   1017   1022   1023#
XFLG       = 100000         602#   775    748    801    810    821    844    849    868    974    1003
XTIB         01[illegible]  521    1001#
XTIB1        01[illegible]  1002   1006#
XTIB2        01[illegible]  1002   1009   1017#
XTIB3        01[illegible]  1006   1025   1027#
XLB        = 00000[illegible] 571# 886    951    1047
XSRITT     = 104012         571#
[illegible]  011640         [illegible] 710#  811
[illegible]  01[illegible]  [illegible] 705#  725
[illegible]  01[illegible]  [illegible] 720#  752
[illegible]  01[illegible]  [illegible] 880#  862    866
.          = 014214         575#   884    890    794    942#   970    972    984    986    997    1016   1051   1096
```

```
BOX          18    572    599    625    961    1036
DCPARM       18
DHDOC1       18
DHPARM       18
DJPARM       18    960
DLPARM       18
DPPARM       18
DQDOC1       18
DQPARM       18
DUPARM       18
DUPPAR       18
DVDOC1       18
DVPARM       18
DZPARM       18
HELLO        18
HLT        57 18    662    713    746    777    807    839    855    947    1013    1068    1075    1093    1094
SEQUAT       18    571
SINTF        18    571
SITEP        18    647
SSERV        18    614

. ABS.  014214     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 ,DZDJD8.SEQ/SOL/CRF/ML:TOC=ITEP1.MAC,DZDJD8.P11
 RUN-TIME: 10 13 .8 SECONDS
 RUN-TIME RATIO: 49/26=1.8
 CORE USED:  16K  (31 PAGES)
```