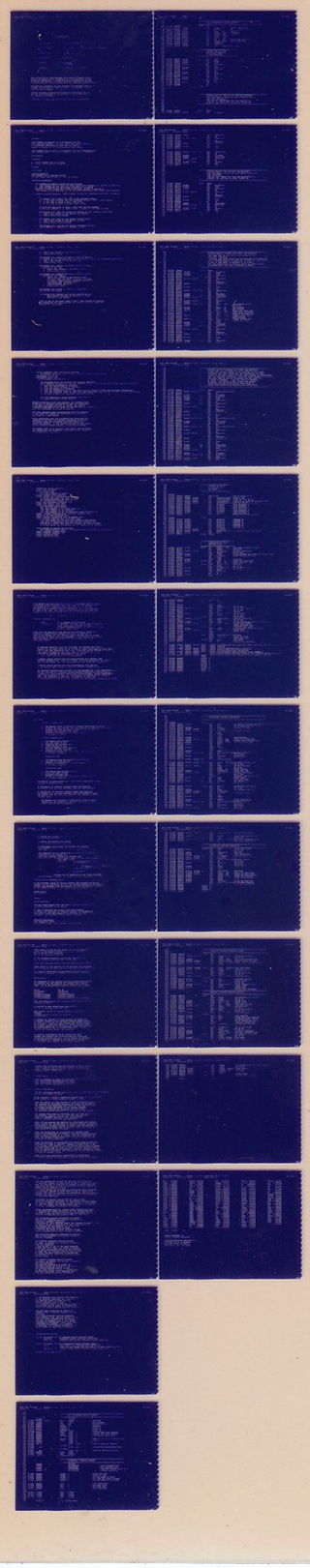
DC11

OVERLAY FOR INTERPROCESSOR
MD-11-DZDCO-C
TEST PROGRAM

EP-DZDCO-C-DL

FICHE 1 OF 1

JUN 1978
digital
MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDCO-C-D

PRODUCT NAME: DC11 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 INTEPPROCESSOR
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 DC11 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.

BL

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 (SA QUICK BROWN FOX)
        10=1 TEST MESSAGE #2 (SB NUMERICS)
        11=1 TEST MESSAGE #3 (SC 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 RRRRRP IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)
TTTTTT IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
BBB IS THE BAD DATA CHARACTER
GGG IS THE GOOD DATA CHARACTER

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

THERE WAS A RECEIVER ERROR, RECEIVER DATA REGISTER =XXXXXX

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

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

7.0 RESTRICTIONS

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

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

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

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

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

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

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

CEIVED, THIS IS BECAUSE THE TEST STATION IS STILL LOOKING
FOR THE REST OF THE INTERUPTED MESSAGE, TO AVOID THIS
ERROR , RESTART PROGRAM ONLY AT THE END OF THE MESSAGE
CURRENTLY BEING TRANSMITTED.

8.0 MISCELLANEOUS

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

9.0 PROGRAM DESCRIPTION

9.1 THE DC11 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 DC11

PARAM#1 IS LOADED INTO THE RECEIVER STATUS REGISTER (RCSR).
BITS 3,4 CHARACTER LENGTH, DEFAULT= 8 BITS (00)
BITS 9,10 RECEIVER SPEED SELECT, DEFAULT= LOWEST SPEED (00)

PARAM#2 IS LOADED INTO THE TRANSMITTER STATUS REGISTER (TSCR).
BIT 0 FULL DUPLEX (1), DEFAULT= HALF DUPLEX (0)
(BIT 0 IS NOT LOADED INTO THE TSCR,IT IS ONLY A SOFTWARE FLAG.)
BITS 3,4 TRANSMIT SPEED SELECT, DEFAULT= LOWEST SPEED (00)

PARAM#3 IS NOT USED(177777).

```
                                ;*****************************************************************
                                ;       DC11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000                  .=11000
011000  041504  000040         DC11:   .ASCIZ  /DC /           ;ISR NAME
011004  174000                 BA:     174000                  ;BUS ADDRESS
011006  000300                 RIV:    300                     ;VECTOR ADDRESS
011010  000240                 PRIOR:  240                     ;PRIORITY
011012  000001                 PARAM1: 1                       ;PARAM #1
011014  000000                 PARAM2: 0                       ;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         EPCSR:  0               ;RCV CSR SAVED ON ERROR
011056  000000         EPDBR:  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
```

```
                        ;*****************************************************************
                        ;       DC11-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  013660          MOV     #RISR,  (R2)+   ;INTERRUPT
011124  013722  011010          MOV     PRIOR,  (R2)+   ;VECTORS
011130  012722  013540          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     #0WO,MODE
011204  001402                  BEQ     1$
011206  000137  011362          JMP     $0WO
011212  032700  000002  1$:     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  013376          JSR     PC,STARTR
```

```
                                18$:    BIT     #RFLG,STAT
                                        BNE     28
                                        CMP     TIME,#100
                                        BLO     18
                                        MOV     @PCSR,R2
                                        MOV     XCSR(R4),R3
                                        HLT     1
                                        CLR     TIME
                                        BR      18

                                28$:    BIT     #NODAT,@SWR
                                        BNE     38
                                        JSR     PC,TESTD
                                38$:    BIC     #RFLG,STAT
                                        BIT     #LOOP,@SWR
                                        BEQ     48
                                        MOV     #48,BACK
                                        JMP     EOP
                                48$:    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,
                                ;*******************************************************************

                                $OWO:   KBDIN
                                        JSR     PC,STARTX
                                        CLR     TIME
                                18$:    BIT     #XFLG,STAT
                                        BNE     28
                                        CMP     TIME,#100
                                        BLO     18
                                        MOV     @RCSR,R2
                                        MOV     XCSR(R4),R3
                                        HLT     1
                                        CLR     TIME
                                        BR      18
                                28$:    BIC     #XFLG,STAT
                                        BIT     #LOOP,@SWR
                                        BEQ     38
                                        MOV     #38,BACK
                                        JMP     EOP
                                38$:    BR      $OWO
```

B1

B2

```
                                ;*****************************************************************
                                ;       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                  $ILB:   KBDIN
011462  004737  013376                  JSR     PC,STARTR
011466  005037  011032                  CLR     TIME
011472  032700  040000          1$:     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
```

C1

C2

```
                        ;*****************************************************************
                        ;       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  013376              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  013376                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     $AVR0,R0        ;RESTORE R0
012232  013701  013042                  MOV     $AVR1,R1        ;RESTORE R1
012236  013702  013044                  MOV     $AVR2,R2        ;RESTORE R2
012242  013703  013046                  MOV     $AVR3,R3        ;RESTORE R3
012246  013704  013050                  MOV     $AVR4,R4        ;RESTORE R4
012252  013705  013052                  MOV     $AVR5,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,@B2016       ;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     28
012364  010237  012372                  MOV     R2,18
012370  104400                          TYPE
```

```
904  012372  000000                  18:     .WORD   0
905  012374  000437                          BR      TESTDX
906  012376                          28:
907  012376  105712                          TSTB    (R2)            ;
908  012400  001435                          BEQ     TESTDX          ;BR IF YES
909  012402  122721  000177                  CMPB    #177,   (R1)+   ;IS IT FILL CHAR?
910  012406  001756                          BEQ     SCAN4           ;BR IF YES
911  012410  005301                          DEC     R1              ;BACKUP
912  012412  122722  000177                  CMPB    #177,   (R2)+   ;IS IT FILL?
913  012416  001752                          BEQ     SCAN4           ;BR IF YES
914  012420  000240                  SCAN5:  NOP                     ;DATA ERROR
915  012422  032777  020000  176414          BIT     #BIT13,@#SWR    ;INHIBIT PRINTOUTS
916  012430  001016                          BNE     DERR            ;BR IF YES
917  012432  104400  012564                  TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
918  012436  013737  011020  012446          MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
919  012444  104400                          TYPE                    ;PRINT RECEIVED DATA
920  012446  000000                  RDAX:   0                       ;RECEIVED DATA ADDR.
921  012450  104400  012611                  TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
922  012454  013737  011022  012464          MOV     IXDA,   .+10    ;SETUP ADDR.
923  012462  104400                          TYPE                    ;PRINT GOOD DATA
924  012464  011022                          IXDA
925  012466  111103                  DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
926  012470  114202                          MOVB    -(R2),R2        ;SETUP RCV DATA
927  012472  104007                          HLT+7                   ;DATA ERROR HALT
928  012474  005726                  TESTDX: TST     (SP)+           ;POP STACK
929  012476  000207                          RTS     PC              ;RETURN FROM SUB/ROUT
930
931  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
932  013042  000000                  SAVR1:  0
933  013044  000000                  SAVR2:  0
934  013046  000000                  SAVR3:  0
935  013050  000000                  SAVR4:  0
936  013052  000000                  SAVR5:  0
937  013054  000000                  DELAY:  0
938  013056  000000                  BACK:   0
939  013060  000000                  STOP:   0
940
```