IDENTIFICATION

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

PRODUCT NAME: DVZ11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: MAY 1977

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
          JOHN EGOLF

REvised BY: FAY BASHAW 1/21/75

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

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

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

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION
1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR
FIELD SERVICE PERSONEL. IT WILL VERIFY THE PROPER OPERATION
OF A COMPLETE COMMUNICATION LINK FROM ONE PDP-11
SYSTEM TO ANOTHER OR TO A COMMUNICATION TEST CENTER.

THIS PROGRAM MUST BE USED IN CONJUNCTION WITH THE INTERPROCESSOR
TEST PROGRAM(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 DZVI11 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 TYPE OUT THE NAME OF THE VARIABLE OVERLAY.
   A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A GARAGE RETURN
   B. IF YOU WISH TO SETUP A DNI1 TYPE IN DN.
   C. IF YOU WISH TO SETUP A DNI188, TYPE IN DMB.

   IF DN OR DMB WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS
   VECTOR ETC. REFER TO IN STEPS 2 THRU 7, PERTAIN TO THE DNI1 OR DMB.

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. TYPE IN ACTUAL BUS ADDRESS

3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
   A. TYPE A CAR, RETURN TO USE DEFAULT ADDRESS
   B. TYPE IN ACTUAL VECTOR ADDRESS

4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITIES
   NOTE: 200=PRI0 4, 240=PRI0 5, 300=PRI0 6, ETC.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPE IN ACTUAL VALUE

5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
   IF REQUIRED BY THE ISR. (SEE Sect. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
   A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
   B. TYPE IN 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 DM-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 DN11B.

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 DM-11 ISR
   SWR000000=SETUP VARIABLE ISR

2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
   SETUP SEQUENCE IS: DM11, 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.C IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
   E. HALT FOR INTERFACE PARAM #2 (DM11 AND DM1B PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
   F. GO BACK TO STEP A IF THIS SETUP WAS FOR DM OR DB.

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 PROCEDURES
TO BE USED, INCLUDING THE TYPE OF MODEMS, THE TYPE OF
INTERFACE BEING USED AT THE OTHER CPU AND THE MDES OF OPERATION,
DATA AND PARAMETERS TO BE USED AT EACH CPU.

THIS WILL REQUIRE 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 DM-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 (#Q QUICK BROWN FOX)
      10=1 TEST MESSAGE #2 ($B NUMERICS)
      11=1 TEST MESSAGE #3 ($C 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) \text{SWR=\texttt{xxxxxx}} \text{ NEW=}

POSSIBLE RESPONSES ARE:

1. \text{<CR>} IF NO CHANGES ARE TO BE MADE
2. \text{6 DIGITS 0-7} TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER
   VALUE (LAST DIGIT FOLLOWED BY \text{<CR>}).
3. \text{\texttt{\textarrow}} 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 \text{16G}
\text{<CTRL 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
\text{1(UP ARROW)} FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377)
SEPARATED BY SPACES AND TERMINATED BY \text{1(UP ARROW)}.
I.E. \text{ABCD\ 000 123 377\ FFG (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 CRO(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 INPUT BY THE OPERATOR, THE TERMINATING CHARACTERS ARE ADDED.
TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY \(\text{DO}1\))
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10 (SW7=0)
3. TRANSMITS THE DATA SELECTED BY SW09 AND SW10 (SWB=0) OR
   TRANSMITS THE RECEIVED DATA (SWB=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 SWD=0 THE OVERLAY WILL CONTINUE TO
   TRANSMIT/RECEIVE DATA.

   IF SWD=1 THE OVERLAY WILL RETURN
   TO THE MONITOR AND TYPE "END PASS".

   IF BOTH SWD=1 AND SW4=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:
   *Wxxxxxxxx TO PRINTOUT THE 8 WORDS
     AT LOC xxxxxx.
   *Bxxxxxxxx TO PRINTOUT THE 16 BYTES
     AFTER LOC xxxxxx.
   *C TO CONTINUE

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

5.0 PROGRAM AND/OR OPERATOR ACTION

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

5.1 NORMAL HALTS
   SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

   RECEIVED DATA=RRRRRR
   DATA SHOULD BE TTTTT
   DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG
WHERE RRRRRR IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)
ITTTTT IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
BBBB IS THE BAD DATA CHARACTER
GGGG 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 (D01) 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:

<table>
<thead>
<tr>
<th>CPU #1</th>
<th>CPU #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE-WAY-OUT</td>
<td>ONE-WAY-IN</td>
</tr>
<tr>
<td>ONE-WAY-IN</td>
<td>ONE-WAY-OUT</td>
</tr>
<tr>
<td>EXTERNAL-LOOPBACK</td>
<td>INTERNAL-LOOPBACK</td>
</tr>
<tr>
<td>INTERNAL-LOOPBACK</td>
<td>EXTERNAL-LOOPBACK</td>
</tr>
<tr>
<td>EXTERNAL-LOOPBACK</td>
<td>EXTERNAL-LOOPBACK (FULL DUPLEX)</td>
</tr>
</tbody>
</table>

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 OCCUR DURING A TYPEOUT THE PROGRAM SHOULD BE RESTARTED.

IF USING AN ASYNCRONOUS DEVICE, MODEMS AND THE MAYNARD TEST STATION AND INITIALIZE DOES NOT CLEAR THE CONNECTION (EXAMPLE THE DJI) IF THE PROGRAM IS RESTARTED IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING CR AN IMMEDIATE ERROR MESSAGE FROM MAYNARD WILL BE RE-
CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING
FOR THE REST OF THE INTERRUPTED MESSAGE. TO AVOID THIS
ERROR, RESTART PROGRAM ONLY AT THE END OF THE MESSAGE
CURRENTLY BEING TRANSMITTED.

8.0 MISCELLANEOUS

ITFP 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 DZV11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR,
BY THE ITFP CONTROL PROGRAM.
TIME: PROVIDES A MEANS OF MEASURING ELASPED TIME. IT IS INCREMENTED
EVERY SECOND BY A CLOCK INTERRUPT ROUTINE IN ITFP.

9.2 WHEN THE OVERLAY IS FIRST ENTERED BY ITFP 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: $O1, IF “ONE WAY IN” MODE
WAS SELECTED, $040, IF “ONE WAY OUT” MODE WAS SELECTED.
$1LB, IF “INTERNAL LOOP BACK” MODE WAS SELECTED.
$XLB, IF “EXTERNAL LOOP BACK” WAS SELECTED.

9.31 $O1: 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 $040: 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 $040 OR TYPES END PASS
DEPENDING ON SWITCH SETTINGS.

9.33 $1LB: 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. ($1LB)

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 INTERRUPTS AND SAVES THE TRANSMITTER INTERRUPT ENABLE
BIT AND ALL GENERAL REGISTERS. IT THEN RETURNS TO THE MONITOR
TO TYPE "END PASS". THE MONITOR CHECKS SW14 IF 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($WO, $WI, $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 PARAM 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 START R:
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 XISR: 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 TESTC: 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 DZV11

PARAM#1 IS LOADED INTO THE LINE PARAMETER REGISTER(DZVLPR)

BIT 0-1 LINE NUMBER BEING USED, DEFAULT = LINE 0

BIT 3,4 CHARACTER LENGTH, DEFAULT = EIGHT BITS

BIT 5 STOP BIT COUNT, DEFAULT IS TWO STOP BITS

BIT 6,7 PARITY ENABLE AND SELECT, DEFAULT IS NO PARITY

BIT 8-11 BAUD RATE SELECT, DEFAULT IS 110 BAUD

BIT 12 RECEIVER ON (THIS SHOULD ALWAYS BE SET)

THE FOLLOWING ARE EXAMPLES FOR VARIOUS LEGAL BAUD RATES:

10070 : 50 BAUD
10470 : 75 BAUD
11070 : 110 BAUD
11470 : 134.5 BAUD
12070 : 150 BAUD
12470 : 300 BAUD
13070 : 600 BAUD
13470 : 1200 BAUD
14070 : 1800 BAUD
14470 : 2400 BAUD
15070 : 4800 BAUD
15470 : 9600 BAUD
16070 : 7200 BAUD
17070 : 9600 BAUD

THE PREVIOUS EXAMPLES SET THE RECEIVER ON FOR LINE 0, DESIGNATE THE BAUD RATE, AND SET THE LINE FOR 8 BITS/CHARACTER WITH 2 STOP BITS AND NO PARITY.

PARAM#2 IS NOT USED AT THIS POINT IN TIME

PARAM#3 IS NOT USED(177777).
DZV11 INTERFACE SERVICE PARAMS

:01000 000126
DZV11: ASCIZ /DZV/ ;ISR NAME
BA: 160010 ;BUS ADDRESS
RV: 300 ;VECTOR ADDRESS
PRIOR: 200 ;PRIORITY
PARAM1: 11070 ;PARAM #1
PARAM2: 177777 ;PARAM #2
PARAM3: 177777 ;PARAM #3
IDX: 0 ;INITIAL READ DATA ADDRESS
IDX: 0 ;INITIAL XMIT DATA ADDRESS
SETTL: 0 ;LINE SETTLE DELAY FLAG
B016: 0 ;ADDR OF BIN TO OCT TYPE ROUTINE
TIME: 0 ;TIMER
B04: 0 ;WORD START ;ADDR OF START OF PROGRAM
TX.TERM: 000 ;TRANSMITTER TERMINATING CHAR.
RX.TERM: 001 ;RECEIVER TERMINATING CHAR.
FLAG: 177570 ;DATA DUMMY
DISPLAY: 177570 ;DATA DUMMY

:CONSTANTS + WORKING STORAGE

STAT=RO
XPLG=0
PLG=0
PCFG=0
PLF=0
XMIT COMPLETE FLAG
RCV COMPLETE FLAG
DATA SET STATUS CHANGE FLAG
INHIBIT PRINTOUTS

SXCSR: 0 ;SAVED XMIT CSR
SRC: 0 ;SAVED RCV CSR
RCSR: 0 ;RCV CSR SAVED ON ERROR
ERCSR: 0 ;RCV DATA REG SAVED ON ERROR
DSSTAT: 0 ;RCV CSR SAVED ON DS CHANGE

XCC: 0 ;XMIT CHAR COUNT
RCC: 0 ;RCV CHAR COUNT
RA: 0 ;RCV DATA ADDR.
XDA: 0 ;XMIT DATA ADDR.

TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566

FULL.DUPLEX=000001
C02

**DZVII-X INTERFACE SERVICE ROUTINE**

**START:**

```
DEC R1

MOV RVR, R2

MOV @RISR, (R2)+

MOV PRIOR, (R2)+

MOV @RISR, (R2)+

MOV PRIOR, (R2)+

MOV BA, R4

SETUP BUS ADDR INDEX

BIS @CLR, @RCRSR

CLEAR SILO=VARTS

1%

BIT @CLR, @RCRSR

CLEAR PULSE DONE?

BNE 1%

BR IF NO

2%

MOV PARA1, TEMP1

DON'T TURN ON RECEIVER YET

MOV TEMP1, LPR(RH)

LOAD LINE NUMBER AND PARAMETERS

3%

MOV RO, -(SP)

SAVE RO

MOV @I, R0

MOV PARA1, R1

4%

MOV @C<3>, R1

ISOLATE THE LINE NUMBER

BEQ 5%

;CALCULATE TCR BIT

ASL R0

DEC R1

BR 4%

5%

MOV RO, TCRTMP

SAVE THE ACTIVE TCR BIT

MOV (SP)+, RO

MOV TCRTMP, TCR+1(R4)

SET DATA TERMINAL READY
```

**ROUTINE USED TO GOTO**

**SUBROUTINE DEPENDENT**

**ON MODE SELECTED.**

```
GO: CLR TIME

CLR DELAY

CLR STOP

BIT @HOW, MODE

BEQ 1%

JMP @HOW

1%

BIT @H1, MODE

BEQ 2%

JMP @H1

2%

BIT @L1, MODE

BEQ 3%

JMP @L1

3%

BIT @X1, MODE

BEQ 4%

JMP @X1

4%

HALT

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".

******************************************************************************************

$SOWI:  KBDIN
JMP PC.START
BNE 3$
CPL TIME, #100
BLO 1$
MOV QCRS,R2
MOV XCRS(R4),R3
HLT 1
CLR TIME
BR 1$

$SOWO:  KBDIN
JMP PC.STARTX
CLR TIME
BNE 2$
CPL TIME, #100
BLO 1$
MOV QCRS,R2
MOV XCRS(R4),R3
ALT 1
CLR TIME
BR 1$

******************************************************************************

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".

******************************************************************************
765  011520  001405  BEQ  3$
766  011522  012737  011534  013134  MOV  #3, BACK
767  011530  000137  012216  JMP  EOP
768  011534  000741  3$:  BR  $040
 啊，这是一段程序代码。它看起来像是用汇编语言写的。程序描述了一个特殊的处理过程，可能是用于通信的。具体来说，它提到了内部循环回路的使用，如果选择了该选项。程序还提到了半双工模式的可用性，并且“内部循环回路”意味着接收器是“已激活的”，并且完整的消息被接收到后，如果数据需要被检查，它就会被检查。如果“终止标志”被设置，它就会被给出。然后，发送器被激活，整个消息被传输，接着这个过程就重复了。
**ROUTINE**

**USE**

**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**. **THEM** **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**.

```
840 011766 104416 $XLB: KBDIN  
841 011770 027376 BIT #FULL.DUPLEX,PARAM2  
842 011774 001402 BEQ 1%  
843 011200 004737 013504 JSR PC.STARTR  
844 012004 004737 013140 JSR PC.STARTX  
845 012010 005037 011032 1%: CLR TIME  
846 012014 027000 100000 2%: BIT #XFLG,STAT  
847 012020 001016 BNE 3%  
848 012022 032700 040000 7%: BIT #XFLG,STAT  
849 012026 001024 BNE 4%  
850 012030 027376 011032 000100 CMP TIME,#100  
851 012034 103766 BLO 2%  
852 012040 011402 MOV &RCSR,R2  
853 012042 016033 MOV &XCSR(R4),R3  
854 012046 104001 HLT 1  
855 012050 005037 011032 CLR TIME  
856 012054 007572 BR 2%  
857 012056 027376 000001 011014 3%: BIT #FULL.DUPLEX,PARAM2  
858 012064 001356 BNE 7%  
859 012066 027000 100000 BIT #XFLG,STAT  
860 012072 004737 013504 JSR PC.STARTR  
861 012076 002746 BR 2%  
862 012100 027376 000001 011014 4%: BIT #FULL.DUPLEX,PARAM2  
863 012106 001420 BEQ 8%  
864 012110 032700 100000 BIT #XFLG,STAT  
865 012114 001013 BNE 6%  
866 012116 027376 011032 000100 CMP TIME,#100  
867 012124 103766 BLO 4%  
868 012126 011402 MOV &RCSR,R2  
869 012130 016033 MOV &XCSR(R4),R3  
870 012134 104001 HLT 1  
871 012136 005037 011032 CLR TIME  
872 012140 000756 BR 4%  
873 012144 027000 100000 6%: BIT #XFLG,STAT  
874 012150 027000 040000 8%: BIT #XFLG,STAT  
875 012154 005037 011032 CLR TIME  
876 012160 032777 000200 176556 BIT #MODAT,ASWR  
877 012166 001002 BNE 5%  
878 012170 004737 012356 JSR PC.TESTD  
879 012174 032777 000020 17642 5%: BIT #LOOP,ASWR  
880 012202 001671 BEQ #XLB  
881 012204 012737 011766 013134 MOV #XLB,BACK  
882 012212 000137 012216 JMP E0P
```
;******************************************************
; ROUTINE TO RETURN TO MONITOR FOR END PASS.
;******************************************************

EOP:

STPS,PRTY7

; SET PS PRIORITY TO 7

MOV XCSR(R4),QTPIE

; SAVE TX CSR

MOV RD,SAVR0

; SAVE REGISTER 0

MOV R1,SAVR1

; SAVE REGISTER 1

MOV R2,SAVR2

; SAVE REGISTER 2

MOV R3,SAVR3

; SAVE REGISTER 3

MOV R4,SAVR4

; SAVE REGISTER 4

MOV R5,SAVR5

; SAVE REGISTER 5

RTS PC

; RETURN TO CONTROL PROGRAM

ENTER:

MOV SAVRO,RO

; RESTORE RO

MOV SAVR1,R1

; RESTORE R1

MOV SAVR2,R2

; RESTORE R2

MOV SAVR3,R3

; RESTORE R3

MOV SAVR4,R4

; RESTORE R4

MOV SAVR5,R5

; RESTORE R5

MOV #1,DELAY

; DELAY 1

BIS QTPIE,XCSR(R4)

; IF ORIGINAL SET; SET TX IE

JMP QTPIE

; QBACK

QTPIE: 000000

;******************************************************
; SUBROUTINE TO CHECK RECEIVER DATA.
;******************************************************

TEST:

MOV ERROR,-(SP)

; HAS THERE A RECEIVE ERROR?

BEQ TSTDATE

; BR IF NO

BIT #BIT13,3SWR

; INHIBIT PRINTOUT?

BNE TSTDAT

; BR IF YES

TYPE MSG0

; (15)<(12) THERE WAS A RECEIVE ERROR. RBUF =

176424

; PRINT CONTENTS OF RBUF

LDI RO,32016

TST - (SP)

; (15)<(12)

TYPE MSG1

; (15)<(12)

TSTDAT:

MOV IXDA, R1

; SETUP XMIT DATA ADDR

MOV IRDA, R2

; SETUP RCV DATA ADDR

SCAN4:

CMPB (R1)

; DATA OK?

BEQ SCAN4

; BR IF OK

CMPB TXTERM,-(R1)

; IS IT END OF DATA

BEQ TESTDX

; BR IF YES

BNE 242

; RX data

BNE 23

; TX data

TYPE R2.1$
I02

939 012450 000000 1%: .WORD 0
940 012452 000437 BR TESTDX
941 012454 0105712 2%: TSIB (R2)
942 012456 0101356 BEQ TESTDX ;BR IF YES
943 012458 0122721 CMPB #177 (R1)+ ;IS IT FILL CHAR?
944 01245a 0001756 BEQ SCAN4 ;BR IF YES
945 01245c 005301 DEC R1 ;BACKUP
946 01245e 0001757 CMPB #177 (R2)+ ;IS IT FILL?
947 012460 0001759 BEQ SCAN4 ;BR IF YES
948 012462 000240 SCANS: NOP ;DATA ERROR
949 012464 02777 BIT #BIT13,35SR ;INHIBIT PRINTOUTS
950 012466 020000 176336 BNE DERR ;BR IF YES
951 012468 001016 MOV IXDA, R0 ;SETUP DATA ADDRESS
952 01246a 104000 TYPE MSG2 ;PRINT RECEIVED DATA
953 01246c 012524 010400 ;RECEIVED DATA ADDR.
954 01246e 012526 000000 RXD: 0 ;DATA SHOULD BE 15/12
955 012470 012528 010400 TYPE MSG3 ;SETUP ADDR.
956 012472 01252a 000000 ;PRINT GOOD DATA
957 012474 013737 011022 TYPE MSG4 ;RETURN FROM SUB/RUT
958 012476 012478 010400 TYPE IXDA
959 01247a 012542 011022 DERR: MOVB (R1),R3 ;SETUP XMIT DATA
960 012544 111103 MOVB -(R2),R2 ;SETUP RCV DATA
961 012546 114202 HLT+7 ;DATA ERROR HALT
962 012548 104000 TESTDX: TST (SP)+ ;POPC STACK
963 01254a 005726 RTS PC
964 01254c 000207 RETURN FROM SUB/RUT
965 01254e 005015 044124 051105 MSG0: .ASCIZ (15)(12)/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =
966 (1) 012637 015 000012 MSG1: .ASCIZ (15)(12)/
967 (1) 012649 005015 042526 042503 MSG2: .ASCIZ (15)(12)/RECEIVED DATA = (15)(12)
968 (1) 012667 018 042019 062101 MSG3: .ASCIZ (15)(12)/DATA SHOULD BE (15)(12)
969 (1) 012712 005015 046120 040505 MSG4: .ASCIZ (15)(12)/PLEASE MAKE CONNECTION (DIAL NUMBER)/
970 (1) 012761 015 053412 042510 .ASCIZ (15)(12)/WHEN CONNECTION COMPLETE, HIT CONTINUE SWITCH. (15)(12)
971 (1) 013044 005015 046120 040505 .ASCIZ (15)(12)/PLEASE MAKE CONNECTION (DIAL NUMBER). (15)(12)
972 EVEN
973 013116 000000 SAVRD: 0
974 013118 000000 SAVR1: 0
975 013120 000000 SAVR2: 0
976 013122 000000 SAVR3: 0
977 013124 000000 SAVR4: 0
978 013126 000000 SAVRS: 0
979 013130 000000 SAVR5: 0
980 013132 000000 DELAY: 0
981 013134 000000 BACK: 0
982 013136 000000 STOP: 0
983
**TRANSMITTER INITIALIZATION SUBROUTINE**

**STARTX:**

```assembly
TST         DELAY
```

**IF SMOD=1 & SH4=0 DELAY**

```assembly
BEG         1%   ;NO DELAY START TRANSMITTER
```

**MAKE TEMP TIMING DELAY**

```assembly
CLR          TEMP1   ;PREPARE FOR DELAY
```

**INCREMENT DELAY**

```assembly
MOV         #7, TEMP2
INC          TEMP1
```

**BNE -4**

```assembly
DEC          TEMP2
```

**BNE -12**

```assembly
CLR          DELAY
```

**BIC XFLG,STAT**

```assembly
MOV         IXDA, XDA
```

**STOP**

```assembly
FIRST TIME HERE?
```

**NO**

```assembly
BNE 2%
```

**MAKE CONNECTION**

```assembly
TYPE MSG4
```

**HALT**

```assembly
STOP
```

**COMPLEMENT STOP**

```assembly
COM          TEMP1
```

**YES PREPARE FOR DELAY**

```assembly
BNE 2%
```

**INCREMENT DELAY**

```assembly
INC          TEMP2
```

**MOV #142, TEMP2**

```assembly
INC          TEMP1
```

**BNE -4**

```assembly
DEC          TEMP2
```

**BNE -12**

```assembly
CLR          DELAY
```

**MOV #13764**

```assembly
MOV         #00000004
```

**TIC TEMP, TCR(R4)**

```assembly
MOV         TIC TEMP, TCR+R4
```

**SET DATA TERMINAL READY**

```assembly
BIC XFLG,STAT
```

**SET FLAG**

```assembly
BIT XLR, MODE
```

**BR IF NO**

```assembly
BNE 3%
```

**SET EIE**

```assembly
MOV IXDA +MSENAB, ARC
```

**SET INTERRUPT ENABLE**

```assembly
WAIT 00001
```

**SET EIE**

```assembly
NOP 00200
```

**FIRST Char RECEIVED YET?**

```assembly
TST SNCFLG
```

**BR IF NO**

```assembly
BNE .4
```

**SET EIE**

```assembly
NOP 00240
```

**SET INTERRUPT ENABLE, SCAN ENABLE PC**

```assembly
RTS 00204
```

**IS CHAR TRANSMITTER TERMINATION CAR**

```assembly
CMPB IXDA, TX.TERM
```

**BR IF NO**

```assembly
BNE XISR1
```

**SET XIR2**

```assembly
BIC #1033, R5
```

**ISOLATE THE LINE NUMBER**

```assembly
MOV 1(R4), R5
```

**GET THE EXPECTED LINE NUMBER**

```assembly
BIC #1033, R1
```

**ARE THEY EQUAL?**

```assembly
CMPB R5, R1
```

**IF SO, GO TRANSMIT A CHARACTER**

```assembly
BR XISR2
```

**SET UP R2 WITH CSR CONTENTS**

```assembly
MOV @ARCsr, R2
```

**CLR R3**

```assembly
WHT 10
```

**ERROR WRONG LINE**

```assembly
BAR .4
```

**TYPE ERROR MESSAGE**

```assembly
HALT
```

```

Br -.2
```
DZVIITEP OVERLAY  MACYII 30(1046)  05-JUL-77  10:59  PAGE 24

DVOZDA.P11  05-JUL-77  10:54

1032 013420 117764 175444 000006 XISR2: MOVB $XDA, TDR(R4) ; TRANSMIT DATA
1033 013426 032777 000100 175410 BIT #100 @SPR ; MONITOR TX DATA?
1034 013434 001406 BEQ NOXMON ; BR IF NO
1035 013436 115777 175434 BS1B JTPS ; ITY READY?
1036 013442 100003 BPL NOXMON ; BR IF NO
1037 013449 117777 175420 175426 MOVB $XDA, JTPB ; TYPE CHAR
1038 013455 005237 011070 NOXMON: INC XDA ; INC TDR POINTER
1039 013456 005037 011032 XISR3: CLR TIME
1040 013462 005037 013602 CLRT  TRNFLG
1041 013465 000002 RTI
1042 013470 000000 ERROR1: 0
1043 013472 000000 TEMPI: 0
1044 013474 000000 TEMP2: 0
1045 013476 000000 TCRMP: 0
1046 013480 000000 SRCFLG: 0
1047 013502 000000 TRNFLG: 0
L02

RECEIVER INITIALIZATION SUBROUTINE

STARTI: TST STOP ;FIRST TIME HERE?
BNE 10 IS ;BR IF NO
TYPE MSG4 ;TYPE MAKE CONNECTION
COM STOP ;COMPLEMENT STOP
HALT

1$: BIT XLB,MODE ;XLB MODE?
BEQ 2$ ;BR IF NO
CLR TEMPI ;START DELAY
INC TEMPI

2$: BIC #FLG,STAT
MOV IRDA, RDA ;SET UP RECEIVER DATA ADD
MOV #1000, RCC ;SET UP BUFFER LIMIT
MOV #1, SMFLG
CLR ERCSR ;CLEAR ERROR RECORDS
CLR ERDBR

3$: TST RBUF(R4) ;CLEAR SILO
BMI 3$ ;KEEP CLEARING UNTIL BIT IS CLEAR

4$: MOV PARAM, TEMPI ;GET READY TO LOAD PARAMETERS
BIS #RCVON, TEMPI ;BE SURE TO TURN RECEIVER ON
MOV TEMPI, LPR(R4) ;LOAD PARAMETERS, ENABLE RECEIVER
MOV TOTAP, TCR1+1(R4) ;SET DATA TERMINAL READY

5$: BIS #RDSERENAB, ARCSR ;SET INTERRUPT ENABLE, RECEIVER ENABLE

RISR: TSTB ARCSR ;DID RECEIVER DONE SET?
BMI 1$ ;BR IF YES
MOV ARCSR, R2 ;SAVE CSR

6$: CLR R3

7$: HLT 10 ;ERROR RECEIVER INTERRUPTED BUT DONE SET NOT

8$: MOV RBUF(R4), R1 ;GET CHAR
BMI 2$ ;BR IF YES

9$: MOV ARCSR, R2 ;SAVE CSR

10$: CLR R3

11$: HLT 10 ;ERROR CHAR PRESENT NOT SET

12$: BIC #200, R1 ;STRIP A BIT

13$: BIT #RUN,FRAME+PARE, R1 ;CHECK FOR RECEIVER ERRORS
BEG 3$ ;BR IF NO ERRORS

14$: MOV ARCSR, ERCR ;SAVE CSR

15$: MOV R1, ERCR ;SAVE RBUF

16$: MOV R1, RDA ;STORE CHAR

17$: BIT #BITS, JSWR ;MONITOR RXDATA?
BEG NORMON ;BR IF NO

18$: TSB APS ;IS IT READY?

19$: MOV R1, JTPB ;TYPE CHAR

20$: NORMON: INC RDA ;INC RBUF POINTER

21$: CLR JDA ;CLEAR NEXT POSITION

22$: DEC RCC ;DEC CHAR COUNT

23$: BNE 1$ ;BUFFER FULL YET?

24$: RESER

25$: CLR R2
CLR R3
HLT 0
HLT 6
; RECEIVER BUFFER FULL
BR -2
CMPB RX_TERM,R1 ; IS CHAR 001?
BNE RISR1 ; BR IF NO
BIC WRIE,WRCSR ; CLEAR RECEIVER INTERRUPT ENABLE
BIS #RFLG,STAT ; SET A DONE FLAG
RISR1: CLR TIME
CLR SNCFLG
; GO HOME
MFULL: .ASCIZ<15><12>/RECEIVER BUFFER FULL ERROR!!/
042503 .ASCIZ<15><12>/ERROR! TRANSMITTER SCAN STOPPED ON WRONG LINE/
042522 .EVEN
042412 051122 .END
04150
000001
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>A$</td>
<td>011004</td>
<td>605$</td>
<td>664</td>
<td>801$</td>
<td>801$</td>
<td>912</td>
</tr>
<tr>
<td>BIT0</td>
<td>000003</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT1</td>
<td>000002</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT10</td>
<td>000002</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT2</td>
<td>000004</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT3</td>
<td>000010</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT4</td>
<td>000020</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT5</td>
<td>000040</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT6</td>
<td>000040</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT7</td>
<td>000020</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2016</td>
<td>01030</td>
<td>615$</td>
<td>926</td>
<td>665</td>
<td>666</td>
<td>923</td>
</tr>
<tr>
<td>DCLR</td>
<td>000020</td>
<td>599$</td>
<td>926</td>
<td>926</td>
<td>926</td>
<td>926</td>
</tr>
<tr>
<td>DELAY</td>
<td>013132</td>
<td>690$</td>
<td>910$</td>
<td>972$</td>
<td>980</td>
<td>988$</td>
</tr>
<tr>
<td>DERR</td>
<td>012544</td>
<td>951</td>
<td>960$</td>
<td>988</td>
<td>988</td>
<td>988</td>
</tr>
<tr>
<td>DISPLA</td>
<td>011046</td>
<td>625$</td>
<td>633$</td>
<td>921</td>
<td>1067</td>
<td>1091</td>
</tr>
<tr>
<td>DSMFLG</td>
<td>001000</td>
<td>599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSTAT</td>
<td>011060</td>
<td>640$</td>
<td>604$</td>
<td>894</td>
<td>903$</td>
<td></td>
</tr>
<tr>
<td>DZV11</td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENTER</td>
<td>012304</td>
<td>690$</td>
<td>766$</td>
<td>802</td>
<td>882</td>
<td>889$</td>
</tr>
<tr>
<td>EOP</td>
<td>012216</td>
<td>689$</td>
<td>921</td>
<td>1067</td>
<td>1091</td>
<td>1091</td>
</tr>
<tr>
<td>ERCSR</td>
<td>011054</td>
<td>638$</td>
<td>1066$</td>
<td>1090$</td>
<td>1090$</td>
<td>1090$</td>
</tr>
<tr>
<td>EREDR</td>
<td>011056</td>
<td>639$</td>
<td>921</td>
<td>1067$</td>
<td>1091$</td>
<td>1091$</td>
</tr>
<tr>
<td>ERROR1</td>
<td>013470</td>
<td>739</td>
<td>767</td>
<td>802</td>
<td>882</td>
<td>889$</td>
</tr>
<tr>
<td>FRME</td>
<td>023000</td>
<td>652$</td>
<td>894</td>
<td>957</td>
<td>962</td>
<td>962</td>
</tr>
<tr>
<td>FULL.D.</td>
<td>000020</td>
<td>652$</td>
<td>894</td>
<td>957</td>
<td>962</td>
<td>962</td>
</tr>
<tr>
<td>GO</td>
<td>011242</td>
<td>669$</td>
<td>698</td>
<td>698</td>
<td>698</td>
<td>698</td>
</tr>
<tr>
<td>ILB</td>
<td>000010</td>
<td>599$</td>
<td>699</td>
<td>699</td>
<td>699</td>
<td>699</td>
</tr>
<tr>
<td>IIDA</td>
<td>011020</td>
<td>611$</td>
<td>805</td>
<td>938</td>
<td>1063</td>
<td>990</td>
</tr>
<tr>
<td>IXDA</td>
<td>011022</td>
<td>612$</td>
<td>806</td>
<td>929</td>
<td>957</td>
<td>957</td>
</tr>
<tr>
<td>KBDIN</td>
<td>014416</td>
<td>599$</td>
<td>720</td>
<td>751</td>
<td>783</td>
<td>840</td>
</tr>
<tr>
<td>LOOP</td>
<td>000020</td>
<td>599$</td>
<td>736</td>
<td>764</td>
<td>799</td>
<td>879</td>
</tr>
<tr>
<td>LPR</td>
<td>000020</td>
<td>599$</td>
<td>670$</td>
<td>1072$</td>
<td>1072$</td>
<td>1072$</td>
</tr>
<tr>
<td>MFULL</td>
<td>014030</td>
<td>1116$</td>
<td>1116$</td>
<td>1116$</td>
<td>1116$</td>
<td>1116$</td>
</tr>
<tr>
<td>MSERAB</td>
<td>000040</td>
<td>599$</td>
<td>1007</td>
<td>1012</td>
<td>1074</td>
<td></td>
</tr>
<tr>
<td>MSGO</td>
<td>012556</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSG0</td>
<td>012556</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSG1</td>
<td>012542</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSG2</td>
<td>012542</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSG3</td>
<td>012667</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSG4</td>
<td>012712</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSG5</td>
<td>013044</td>
<td>925$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
<td>966$</td>
</tr>
<tr>
<td>MSDA</td>
<td>002000</td>
<td>599$</td>
<td>732</td>
<td>795</td>
<td>876</td>
<td></td>
</tr>
<tr>
<td>NORM</td>
<td>013714</td>
<td>1094</td>
<td>1096</td>
<td>1098$</td>
<td>1098$</td>
<td></td>
</tr>
<tr>
<td>NORMX</td>
<td>013452</td>
<td>1094</td>
<td>1096</td>
<td>1098$</td>
<td>1098$</td>
<td></td>
</tr>
<tr>
<td>OPM</td>
<td>014000</td>
<td>599$</td>
<td>1088</td>
<td>1088</td>
<td>1088</td>
<td></td>
</tr>
<tr>
<td>OWI</td>
<td>000002</td>
<td>599$</td>
<td>695</td>
<td>695</td>
<td>695</td>
<td></td>
</tr>
<tr>
<td>OWC</td>
<td>000001</td>
<td>599$</td>
<td>695</td>
<td>695</td>
<td>695</td>
<td></td>
</tr>
<tr>
<td>PARAM1</td>
<td>011012</td>
<td>608$</td>
<td>668</td>
<td>673</td>
<td>1022</td>
<td>1070</td>
</tr>
<tr>
<td>Symbol</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TKB</td>
<td>011074</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TKS</td>
<td>011072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP8</td>
<td>011100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPS</td>
<td>011076</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRNFLG</td>
<td>013508</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSTDAT</td>
<td>012412</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX_TER</td>
<td>011040</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td>104400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCC</td>
<td>011062</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCSR</td>
<td>000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XDA</td>
<td>011070</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XFLG</td>
<td>100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XISR</td>
<td>013334</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XISR1</td>
<td>013356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XISR2</td>
<td>013420</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XISR3</td>
<td>013456</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XLB</td>
<td>000004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XWAIT</td>
<td>104412</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#LB</td>
<td>011536</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#WJ</td>
<td>011332</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#WO</td>
<td>011440</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#XLB</td>
<td>011766</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Remarks:
DO3

DZY11 ITEP OVERLAY MACY11 30(1046) 05-IUL-77 10:59 PAGE 32
DYZDA.P11 05-JUL-77 10:54 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0029

BOX 10 600 627 653 976 1048
DCPAM 10
DPMCI 10
DMPARM 10
DPARAR 10
DLPAM 10
DPPARM 10
DQDQCI 10
DQPARM 10
DUPARM 10
DUPPAR 10
DVPDCI 10
DVPARM 10
DZPARM 10
DZVPAR 561
HELLO 10
HLT 599 728 760 792 822 854 870 962 1028 1081 1086 1105 1106

$EQUAT 10 599
$INTF 10 599
$ITEP 10 602
$SSERV 10 642

ABS. 014150 000

ERRORS DETECTED: 0

DYZDA DYZDA SEQ=ITEP1 MAC,DYZDA.P11
RUN-TIME: 3.4 .2 SECONDS
RUN-TIME RATIO: 194/8=23.9
CORE USED: 16K (31 PAGES)