

# DZV-11

OVERLAY FOR INTERPROCESSOR  
MD-11-DVDZD-A  
TEST PROGRAM

EP-DVDZD-A-DL-A

COPYRIGHT 1977

FICHE 1 OF 1

OCT 1977

**digital**

MADE IN USA

The image shows a grid of 15 small, illegible technical diagrams or data tables arranged in two columns on the left side of the page. The diagrams appear to be technical drawings or data tables, but the text is too small to read. The grid is approximately 10 rows by 2 columns, with the last row containing only one diagram.

B01

EOF10VDZDASB0411  
DVDZDA.P11

05-JUL-77 10:54

00010000EP OVERZ0920 MACY11 30(10R091005+JUL-77 1005RDR000BZBASEQ

00010000

770920  
SEQ 0001

IDENTIFICATION  
-----

PRODUCT CODE: MAINDEC-11-DVDZD-A-D  
PRODUCT NAME: DZV11 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 DZV11 COMMUNICATION INTERFACE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE

THIS PROGRAM IS IN ABSOLUTE FORMAT.  
THE ABS LOADER MUST BE USED TO LOAD THE PROGRAM.

4.0 OPERATING PROCEDURES.

- A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED
  - 1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
  - 2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.

\*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)
- B. CONSOLE DIALOGUE PARAMETER INPUT (CURRENT VALUES FOR PARAMETERS ARE FOUND IN OVERLAY)

- 1. THE PROGRAM WILL TYPEOUT THE NAME OF THE VARIABLE OVERLAY.
  - A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A CARAGE RETURN
  - B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
  - C. IF YOU WISH TO SETUP A DM11BB, TYPE IN DMB.

IF DN OR DMB WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS VECTOR ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DMBB.

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

- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
  - B. TYPEIN ACTUAL VALUE
5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1  
IF REQUIRED BY THE ISR. (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
  - B. TYPEIN ACTUAL VALUE
6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2  
IF REQUIRED BY THE ISR.
- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
  - B. ENTER ACTUAL VALUE
7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3  
IF REQUIRED BY THE OVERLAY.
- A. TYPE F 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 DMI1BB.
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 SWITCHES 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 DM11-BB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.)
    - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DM.
  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 MODES OF OPERATION, DATA AND PARAMETERS TO BE USED AT EACH CPU.

THIS WILL REQUIRED VOCAL COMMUNICATION WITH THE OPERATOR AT THE OTHER CPU UNLESS ITS CONFIGURATION AND OPERATION ARE FIXED AS A TEST CENTER.

AFTER DETERMINING THAT THE EQUIPMENTS ARE COMPATIBLE AND AGREEING ON THE MODE AND VARIABLE PARAMETERS TO BE USED, THE SYSTEM WHICH IS TO RECEIVE DATA FIRST SHOULD BE LOADED AND STARTED. IF THE MODEM BEING USED ON THIS SYSTEM HAS AN AUTOMATIC ANSWER FEATURE, IT SHOULD BE ENABLED.

THE SYSTEM WHICH IS TO TRANSMIT FIRST SHOULD THEN BE LOADED AND STARTED AND THE CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY (VIA DN-11).

## D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR  
SW14=1 SINGLE PASS  
SW14 HAS NO EFFECT IF SW04=0  
SW13=1 INHIBIT ERROR TIMEOUTS  
SW12=1 INHIBIT ALL TIMEOUTS EXCEPT ERRORS  
IF SW12=0 AND SW04=1 END PASS IS TYPED  
AND TRANSMITTED/RECEIVED DATA IS TYPED.  
SW11=1 USE PREVIOUSLY SPECIFIED DATA  
SW10=1 DATA SELECT (WITH SW09)  
SW09=1 DATA SELECT (WITH SW10)  
00=1 GET DATA FROM OPERATOR  
01=1 TEST MESSAGE #1 (\$A QUICK BROWN FOX)  
10=1 TEST MESSAGE #2 (\$B NUMERICS)  
11=1 TEST MESSAGE #3 (\$C COMTEST/QUICK BROWN FOX/NUMERICS)  
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)  
SW07=1 DO NOT TEST RECEIVED DATA  
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.\*  
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.\*  
\* IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE  
TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS  
RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL  
OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.  
  
SW04=1 RETURN TO MONITOR FOR END PASS  
WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.  
SW03=1 INTERNAL LOOPBACK MODE  
SW02=1 EXTERNAL LOOPBACK MODE  
SW01=1 ONE-WAY-IN MODE  
SW00=1 ONE-WAY-OUT MODE

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176 ) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE)      SWR=XXXXXX    NEW=

POSSIBLE RESPONSES ARE:

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

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (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 (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR GO TO STEP 1(SW04=0)

## ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SW07=0)
3. RETURNS TO MONITOR FOR "END PASS"(SW04=1) OR GO TO STEP 1 (SW04=0)

## ONE-WAY-OUT MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAITS FOR CLEAR TO SEND
3. TRANSMITS SELECTED DATA
4. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR GO TO STEP 1 (SW04=0)

E. THE OVERLAY IS THEN ENTERED AND A CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY.

IF ONE-WAY-IN OR INTERNAL LOOPBACK MODES ARE SELECTED.  
THE OVERLAY WILL SET DATA TERMINAL READY AND WAIT FOR DATA.

IF ONE-WAY-OUT OR EXTERNAL LOOPBACK MODES WERE SELECTED.  
THE OVERLAY WILL SET DATA TERMINAL READY AND REQUEST TO SEND.  
THE OVERLAY WILL THEN WAIT FOR CLEAR TO SEND BEFORE ATTEMPTING TO TRANSMIT DATA.

THE PROGRAM WILL PRINTOUT A "WAITING FOR CLEAR TO SEND"  
MESSAGE AND THE CONTENTS OF THE XMIT CSR EVERY 60 SECS.  
UNTIL CLEAR TO SEND IS ASSERTED.

F. IF SW04=0 THE OVERLAY WILL CONTINUE TO TRANSMIT/RECEIVE DATA.

IF SW04=1 THE OVERLAY WILL RETURN TO THE MONITOR AND TYPE "END PASS".

IF BOTH SW04=1 AND SW14=1, THE PROGRAM WILL REQUEST NEW INTERFACE PARAMS AFTER ONE PASS OF THE SELECTED TEST MODE.

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING CHARACTERS ON THE CONSOLE TTY.  
LINE FEED = RESTART PROGRAM AT LOCATION 200.  
QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)

THEN TYPE EITHER:  
\*WXXXXXX TO PRINTOUT THE 8 WORDS AT LOC XXXXXX.

\*BXXXXXX TO PRINTOUT THE 16 BYTES AFTER LOC XXXXXX.

\*C TO CONTINUE

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

## 5.0 PROGRAM AND/OR OPERATOR ACTION

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

### 5.1 NORMAL HALTS SEE SECTION 4.

## 6.0 ERRORS

### 6.1 ERROR REPORTING

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

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

RECEIVED DATA=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 ASYNCHRONOUS DEVICE, MODEMS AND THE  
MAYNARD TEST STATION AND INITIALIZE DOES NOT CLEAR THE  
CONNECTION (EXAMPLE THE DJ11) IF THE PROGRAM IS RESTARTED  
IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING 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

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 DZV11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR, BY THE ITEP CONTROL PROGRAM.

TIME: PROVIDES A MEANS OF MEASURING ELAPSED TIME. IT IS INCREMENTED EVERY SECOND BY A CLOCK INTERRUPT ROUTINE IN ITEP.

9.2 WHEN THE OVERLAY IS FIRST ENTERED BY ITEP AT LOCATION START:, THE CONTENTS OF THE SWITCH REGISTER ARE STORED IN REGISTER 0. THE MODE AND DATA SELECTIONS ARE FIXED AT THIS TIME AND CANNOT BE ALTERED WITHOUT RETURNING TO THE CONTROL PROGRAM. THE INTERRUPT VECTORS AND VARIABLES ARE THEN SETUP. THE SELECTED ROUTINE DETERMINED BY THE MODE IS THEN ENTERED

9.3 THE OVERLAY THEN LOOPS IN ROUTINES: \$OWI, IF "ONE WAY IN" MODE WAS SELECTED. \$OWO, IF "ONE WAY OUT" MODE WAS SELECTED. \$ILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED. \$XLB, IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 \$OWI: IN THIS ROUTINE THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE RECEIVER TO FINISH. IF NOTHING IS RECEIVED FOR 60 SECS A "WAITING" MESSAGE IS TYPED. WHEN THE RECEIVER IS DONE, THE PROGRAM CHECKS DATA IF SWITCHES PERMIT, AND TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.32 \$OWO: THE TRANSMITTER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR TRANSMITTER TO FINISH. A "WAITING" MESSAGE IS TYPED EVERY 60 SECS IF THERE IS NO ACTION. WHEN THE TRANSMITTER IS DONE, THE PROGRAM EITHER LOOPS BACK TO \$OWO OR TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.33 \$ILB: THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR RECEIVER TO FINISH, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF NO ACTION. WHEN RECEIVER IS DONE PROGRAM CHECKS DATA IF SWITCH SETTINGS PERMIT, AND END PASS IS TYPED IF SWITCH SETTINGS PERMIT. THEN THE TRANSMITTER IS INITIALIZED, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF NO ACTION. WHEN TRANSMITTER IS DONE PROGRAM RETURNS TO START OF ROUTINE. (\$ILB)

9.34 \$XLB: IF IN HALF DUPLEX THE TRANSMITTER IS INITIALIZED, A "WAITING MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION

WHEN THE TRANSMITTER IS DONE THE RECEIVER IS INITIALIZED  
A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION.  
WHEN THE RECEIVER IS DONE DATA IS CHECKED IF SWITCH SETTINGS  
PERMIT AND END PASS IS TYPED IF SWITCHES ALLOW. THE PROGRAM NOW  
REPEATS CYCLE STARTING AT \$XLB.  
IF IN FULL DUPLEX THE RECEIVER AND TRANSMITTER ARE INITIALIZED  
A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO  
ACTION. WHEN BOTH THE RECEIVER AND TRANSMITTER ARE DONE DATA IS  
CHECKED, END PASS IS TYPED AND PROGRAM LOOPS TO \$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 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 DZV11

PARAM#1 IS LOADED INTO THE LINE PARAMETER REGISTER(DZVLPR)  
BITS 0-1 LINE NUMBER BEING USED, DEFAULT = LINE 0  
BITS 3,4 CHARACTER LENGTH, DEFAULT = EIGHT BITS  
BIT 5 STOP BIT COUNT, DEFAULT IS TWO STOP BITS  
BITS 6,7 PARITY ENABLE AND SELECT, DEFAULT IS NO PARITY  
BITS 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 ;2000 BAUD  
15070 ;2400 BAUD  
15470 ;3600 BAUD  
16070 ;4800 BAUD  
16470 ;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).

599  
600  
601  
602  
603 011000 011000  
604 011000 055104 000126  
605 011004 160010  
606 011006 000300  
607 011010 000200  
608 011012 011070  
609 011014 177777  
610 011016 177777  
611 011020 000000  
612 011022 000000  
613 011024 000000  
614 011026 000000  
615 011030 000000  
616 011032 000000  
617 011034 000000  
618 011036 011102  
619 011040  
620 011040 000  
621 011041  
622 011041 001  
623 011042 000000  
624 011044 177570  
625 011046 177570  
626  
627  
628  
629  
630 000000  
631 100000  
632 040000  
633 020000  
634 020000  
635  
636 011050 000000  
637 011052 000000  
638 011054 000000  
639 011056 000000  
640 011060 000000  
641  
642 011062 000000  
643 011064 000000  
644 011066 000300  
645 011070 000000  
646  
647 011072 177560  
648 011074 177562  
649 011076 177564  
650 011100 177566  
651  
652 000001

```

;*****
; DZV11 INTERFACE SERVICE PARAMS
;*****

```

```

DZV11: .ASCIZ /DZV/ ;ISR NAME
BA: 160010 ;BUS ADDRESS
RIV: 300 ;VECTOR ADDRESS
PRIOR: 200 ;PRIORITY
PARAM1: 11070 ;PARAM #1
PARAM2: 177777 ;PARAM #2
PARAM3: 177777 ;PARAM #3
IRDA: .WORD 0 ;INITIAL READ DATA ADDRESS
IXDA: .WORD 0 ;INITIAL XMIT DATA ADDRESS
SETTLE: .WORD 0 ;LINE SETTLE DELAY FLAG
B2016: .WORD 0 ;ADDR OF BIN TO OCT TYPE ROUTINE
TIME: .WORD 0 ;TIMER
TX.TERM: .WORD START ;ADDR OF START OF PROGRAM
RX.TERM: .BYTE 000 ;TRANSMITTER TERMINATING CHAR.
FLAG: .BYTE 001 ;RECEIVER TERMINATING CHAR.
SWR: .WORD 0
DISPLAY: 177570

```

```

;*****
; CONSTANTS + WORKING STORAGE
;*****

```

```

STAT=RD
XFLG=100000 ;XMIT COMPLETE FLAG
RFLG=40000 ;RCV COMPLETE FLAG
DSFLG=20000 ;DATA SET STATUS CHANGE FLAG
BIT13=20000 ;INHIBIT PRINTOUTS
SXCSR: 0 ;SAVED XMIT CSR
SRCSR: 0 ;SAVED RCV CSR
ERCSR: 0 ;RCV CSR SAVED ON ERROR
ERDBR: 0 ;RCV DATA REG SAVED ON ERROR
DSSTAT: 0 ;RCV CSR SAVED ON DS CHANGE
XCC: 0 ;XMIT CHAR COUNT
RCC: 0 ;RCV CHAR COUNT
RDA: 0 ;RCV DATA ADDR.
XDA: 0 ;XMIT DATA ADDR.
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
FULL.DUPLEX=000001

```

653  
654  
655  
656 011102 000240  
657 011104 017700 177734  
658 011110 042700 177400  
659 011114 013702 011006  
660 011120 012722 013646  
661 011124 013722 011010  
662 011130 012722 013334  
663 011134 013722 011010  
664 011140 013704 011004  
665 011144 052714 000020  
666 011150 032714 000020  
667 011154 001375  
668 011156 013737 011012 013472  
669 011164 042737 010000 013472  
670 011172 013754 013472 000002  
671 011200 010046  
672 011202 012700 000001  
673 011206 013701 011012  
674 011212 042701 177774  
675 011216 001403  
676 011220 006300  
677 011222 005301  
678 011224 000774  
679 011226 010037 013476  
680 011232 012600  
681 011234 113764 013476 000005  
682  
683  
684  
685  
686  
687  
688  
689 011242 005037 011032  
690 011246 005037 013132  
691 011252 005037 013136  
692 011256 032700 000001  
693 011262 001402  
694 011264 000137 011440  
695 011270 032700 000002  
696 011274 001402  
697 011276 000137 011332  
698 011302 032700 000010  
699 011306 001402  
700 011310 000137 011536  
701 011314 032700 000004  
702 011320 001402  
703 011322 000137 011766  
704 011326 000000  
705 011330 000776  
706  
707  
708

```

*****
DZV11-X INTERFACE SERVICE ROUTINE
*****
START:  NOP
        MOV     @SWR,  RO      ; SETUP MODE IN RO
        BIC     #177400, RO    ; STRIP JUNK
        MOV     RIV,   R2      ; SETUP
        MOV     #RISR, (R2)+   ; INTERRUPT
        MOV     PRIOR, (R2)+   ; VECTORS
        MOV     #XISR, (R2)+
        MOV     PRIOR, (R2)+
        MOV     BA,    R4      ; SETUP BUS ADDR INDEX
        BIS     #DCLR,@RCSR    ; CLEAR SILO+UARTS
1$:     BIT     #DCLR,@RCSR    ; CLEAR PULSE DONE?
        BNE     1$            ; BR IF NO
2$:     MOV     PARAM1,TEMP1
        BIC     #RCVON,TEMP1   ; DON'T TURN ON RECEIVER YET
        MOV     TEMP1,(R4)     ; LOAD LINE NUMBER AND PARAMETERS
3$:     MOV     RO,-(SP)       ; SAVE RO
        MOV     #1,RO
        MOV     PARAM1,R1
        BIC     #1<3>,R1       ; ISOLATE THE LINE NUMBER
4$:     BEQ     5$            ; CALCULATE TCR BIT
        ASL     RO
        DEC     R1
        BR     4$
5$:     MOV     RO,TCRTMP      ; SAVE THE ACTIVE TCR BIT
        MOV     (SP)+,RO
        MOVB   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     #OWO,MODE
        BEQ     1$
        JMP     $OWO
1$:     BIT     #OWI,MODE
        BEQ     2$
        JMP     $OWI
2$:     BIT     #ILB,MODE
        BEQ     3$
        JMP     $ILB
3$:     BIT     #XLB,MODE
        BEQ     4$
        JMP     $XLB
4$:     HALT
        BR     .-2

```



765	011520	001405			BEQ	3\$
766	011522	012737	011534	013134	MOV	#3\$,BACK
767	011530	000137	012216		JMP	EOP
768	011534	000741			BR	\$OWO
769						
770						
771						

```

772
773
774
775
776
777
778
779
780
781
782
783 011536 104416
784 011540 004737 013504
785 011544 005037 011032
786 011550 032700 040000
787 011554 001013
788 011556 023727 011032 000100
789 011564 103771
790 011566 011402
791 011570 016403 000000
792 011574 104001
793 011576 005037 011032
794 011602 000762
795 011604 032777 000200 177232
796 011612 001002
797 011614 004737 012356
798 011620 042700 040000
799 011624 032777 000020 177212
800 011632 001405
801 011634 012737 011646 013134
802 011642 000137 012216
803 011646 032777 000400 177170
804 011654 001416
805 011656 013702 011020
806 011662 013703 011022
807 011666 010337 011070
808 011672 112223
809 011674 001376
810 011676 112743 000177
811 011702 005203
812 011704 112723 000177
813 011710 105023
814 011712 005037 011032
815 011716 004737 013140
816 011722 032700 100000
817 011726 001013
818 011730 023727 011032 000100
819 011736 103771
820 011740 011402
821 011742 016403 000000
822 011746 104001
823 011750 005037 011032
824 011754 000762
825 011756 042700 100000
826 011762 000137 011536

```

```

*****
ROUTINE USED IF INTERNAL LOOP BACK" WAS SELECTED.
NOTE THAT WHEN IN THIS MODE; HALF DUPLEX IS THE
ONLY MODE AVAILABLE.
"INTERNAL LOOP BACK" MEANS THAT THE RECEIVER IS "TURNED ON"
AND A COMPLETE MESSAGE IS RECEIVED. IF DATA IS TO BE CHECKED
IT IS; IF "END PASS" IS DESIRED; IT IS GIVEN.
THEN THE TRANSMITTER IS ENABLED. AFTER THE WHOLE MESSAGE
IS TRANSMITTED; THE CYCLE IS REPETED AS ABOVE.
*****

```

```

$ILB: KBDIN
JSR PC, STARTR
CLR TIME
1$: BIT #RFLG, STAT
BNE 2$
CMP TIME, #100
BLO 1$
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 1$
2$: BIT #NODAT, @SWR
BNE 3$
JSR PC, TESTD
BIC #RFLG, STAT
3$: BIT #LOOP, @SWR
BEQ 4$
MOV #4$, BACK
4$: BIT #400, @SWR
;USE EXTERNAL DATA?
;BR IF NO
;SET POINTER
;SET POINTER
;SETUP XMIT DATA ADDR
;MOVE INPUT TO OUTPUT
;LOOP IF NOT ZERO CHAR
;INSERT A FILL CHAR
;BUMP ADDRESS
;INSERT ANOTHER FILL
;INSERT ZERO CHAR
7$: CLR TIME
JSR PC, STARTX
5$: BIT #XFLG, STAT
BNE 6$
CMP TIME, #100
BLO 5$
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 5$
6$: BIC #XFLG, STAT
JMP $ILB

```

827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882

011766 104416  
011770 032737 000001 011014  
011776 001402  
012000 004737 013504  
012004 004737 013140  
012010 005037 011032  
012014 032700 100000  
012020 001016  
012022 032700 040000  
012026 001024  
012030 023727 011032 000100  
012036 103766  
012040 011402  
012042 016403 000000  
012046 104001  
012050 005037 011032  
012054 000757  
012056 032737 000001 011014  
012064 001356  
012066 042700 100000  
012072 004737 013504  
012076 000746  
012100 032737 000001 011014  
012106 001420  
012110 032700 100000  
012114 001013  
012116 023727 011032 000100  
012124 103765  
012126 011402  
012130 016403 000000  
012134 104001  
012136 005037 011032  
012142 000756  
012144 042700 100000  
012150 042700 040000  
012154 005037 011032  
012160 032777 000200 176656  
012166 001002  
012170 004737 012356  
012174 032777 000020 176642  
012202 001671  
012204 012737 011766 013134  
012212 000137 012216

```
*****
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.
*****
$XLB: KBDIN
      BIT      #FULL.DUPLEX,PARAM2
      BEQ      1$
      JSR      PC,STARTR
1$:   JSR      PC,STARTX
      CLR      TIME
2$:   BIT      #XFLG,STAT
      BNE      3$
      BIT      #RFLG,STAT
7$:   BIT      #RFLG,STAT
      BNE      4$
      CMP      TIME,#100
      BLO      2$
      MOV      @RCSR,R2
      MOV      XCSR(R4),R3
      HLT      1
      CLR      TIME
      BR       2$
3$:   BIT      #FULL.DUPLEX,PARAM2
      BNE      7$
      BIC      #XFLG,STAT
      JSR      PC,STARTR
      BR       2$
4$:   BIT      #FULL.DUPLEX,PARAM2
      BEQ      8$
      BIT      #XFLG,STAT
      BNE      6$
      CMP      TIME,#100
      BLO      4$
      MOV      @RCSR,R2
      MOV      XCSR(R4),R3
      HLT      1
      CLR      TIME
      BR       4$
6$:   BIC      #XFLG,STAT
8$:   BIC      #RFLG,STAT
      CLR      TIME
      BIT      #NODAT,@SWR
      BNE      5$
      JSR      PC,TESTD
5$:   BIT      #LOOP,@SWR
      BEQ      $XLB
      MOV      #$XLB,BACK
      JMP      EOP
```



```

939 012450 000000 1$: .WORD 0
940 012452 000437 BR TESTDX
941 012454 2$: TSTB (R2)
942 012454 105712 BEQ TESTDX ; BR IF YES
943 012456 001435 CMPB #177, (R1)+ ; IS IT FILL CHAR?
944 012460 122721 000177 BEQ SCAN4 ; BR IF YES
945 012464 001756 DEC R1 ; BACKUP
946 012466 005301 CMPB #177, (R2)+ ; IS IT FILL?
947 012470 122722 000177 BEQ SCAN4 ; BR IF YES
948 012474 001752 SCANS: NOP ; DATA ERROR
949 012476 000240 BIT #BIT13, RSWR ; INHIBIT PRINTOUTS
950 012500 032777 020000 176336 BNE DERR ; BR IF YES
951 012506 001016 TYPE MSG2 ; <15><12>RECEIVED DATA = <15><12>
952 012510 104400 012642 MOV IRDA, RDAX ; SETUP DATA ADDRESS
953 012514 013737 011020 012524 TYPE ; PRINT RECEIVED DATA
954 012522 104400 RDAX: 0 ; RECEIVED DATA ADDR.
955 012524 000000 TYPE MSG3 ; <15><12>DATA SHOULD BE<15><12>
956 012526 104400 012667 012542 MOV IXDA, .+10 ; SETUP ADDR.
957 012532 013737 C11022 012542 TYPE ; PRINT GOOD DATA
958 012540 104400 IXDA
959 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 012550 104007 TESTDX: TST (SP)+ ; POP STACK
963 012552 005726 RTS ; RETURN FROM SUB/ROUT
964 012554 000207
965
966 012556 005015 044124 051105 MSG0: .ASCIZ <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1) 012637 015 000012 MSG1: .ASCIZ <15><12>
(1) 012642 005015 042522 042503 MSG2: .ASCIZ <15><12>/RECEIVED DATA = /<15><12>
(1) 012667 015 042012 052101 MSG3: .ASCIZ <15><12>/DATA SHOULD BE/<15><12>
(1) 012712 005015 046120 040505 MSG4: .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1) 012761 015 053412 042510 .ASCIZ <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
(1) 013044 005015 046120 040505 MSG5: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1) .EVEN
(1) 013116 000000 SAVR0: 0
967 013120 000000 SAVR1: 0
968 013122 000000 SAVR2: 0
969 013124 000000 SAVR3: 0
970 013126 000000 SAVR4: 0
971 013130 000000 SAVR5: 0
972 013132 000000 DELAY: 0
973 013134 000000 BACK: 0
974 013136 000000 STOP: 0
975

```

```

976 ;*****
977 ; TRANSMITTER INITIALIZATION SUBROUTINE
978 ;*****
979
980 013140 005737 013132 STARTX: TST DELAY ;IF SW04=1 & SW14=0 DELAY
981 013144 001415 BEQ 1$ ;NO DELAY START TRANSMITTER
982 013146 005037 013472 CLR TEMP1 ;PREPARE FOR DELAY
983 013152 012737 000007 013474 MOV #7,TEMP2
984 013160 005237 013472 INC TEMP1 ;INCREMENT DELAY
985 013164 001375 BNE -4
986 013166 005337 013474 DEC TEMP2
987 013172 001372 BNE -12
988 013174 005037 013132 CLR DELAY
989 013200 042700 100000 1$: BIC #XFLG,STAT
990 013204 013737 011022 011070 MOV IXDA,XDA ;SET UP XMIT DATA ADD
991 013212 005737 013136 TST STOP ;FIRST TIME HERE?
992 013216 001020 BNE 2$ ;NO
993 013220 104400 012712 TYPE ,MSG4 ;MAKE CONNECTION
994 013224 000000 HALT
995 013226 005137 013136 COM STOP ;COMPLEMENT STOP
996 013232 005037 013472 CLR TEMP1 ;YES PREPARE FOR DELAY
997 013236 012737 000030 013474 MOV #14*2,TEMP2
998 013244 005237 013472 INC TEMP1 ;INCREMENT DELAY
999 013250 001375 BNE -4
1000 013252 005337 013474 DEC TEMP2
1001 013256 001372 BNE -12
1002 013260 013764 013476 000004 2$: MOV TCRTMP,TCR(R4) ;SET LINE # IN TCR
1003 013266 113764 013476 000005 MOVB TCRTMP,TCR+1(R4) ;SET DATA TERMINAL READY
1004 013274 032700 000004 BIT #XLB,MODE ;XLB MODE?
1005 013300 001412 BEQ 3$ ;BR IF NO
1006 013302 012737 177777 013502 MOV #-1,TRNFLG ;SET FLAG
1007 013310 052714 040040 BIS #TIE+MSENAB,ARCSR ;SET INTERRUPT ENABLE
1008 013314 000001 WAIT
1009 013316 000240 NOP
1010 013320 005737 013500 TST SNCFLG ;FIRST CHAR RECEIVED YET?
1011 013324 001375 BNE -4 ;BR IF NO
1012 013326 052714 040040 3$: BIS #TIE+MSENAB,ARCSR ;SET INTERRUPT ENABLE,SCAN ENABLE
1013 013332 000207 RTS PC
1014
1015 013334 127737 175530 011040 XISR: CMPB @XDA,TX.TERM ;IS CHAR TRANSMITTER TERMINATION CHAR
1016 013342 001005 BNE XISR1 ;BR IF NO
1017 013344 052700 100000 BIS #XFLG,STAT ;SET XMIT DONE FLAG
1018 013350 042714 040000 BIC #TIE,ARCSR ;CLEAR ENABLES
1019 013354 000440 BR XISR3
1020 013356 116405 000001 XISR1: MOVB 1(R4),R5 ;GET LINE NUMBER OF READY LINE
1021 013362 042705 177774 BIC #1<3>,R5 ;ISOLATE THE LINE NUMBER
1022 013366 013701 011012 MOV PARAM1,R1 ;GET THE EXPECTED LINE NUMBER
1023 013372 042701 177774 BIC #1<3>,R1 ;ISOLATE IT
1024 013376 120501 CMPB R5,R1 ;ARE THEY EQUAL?
1025 013400 001407 BEQ XISR2 ;IF SO, GO TRANSMIT A CHARACTER
1026 013402 011402 MOV ARCSR,R2 ;SET UP R2 WITH CSR CONTENTS
1027 013404 005003 CLR R3
1028 013406 104010 HLT 10 ;ERROR WRONG LINE
1029 013410 104400 014067 TYPE ,SCANE ;TYPE ERROR MESSAGE
1030 013414 000000 HALT
1031 013416 000776 BR -2

```

1032	013420	117764	175444	000006	XISR2:	MOVB	@XDA,TDR(R4)	; TRANSMIT DATA
1033	013426	032777	000100	175410		BIT	#100,@SWR	; MONITOR TX DATA?
1034	013434	001406				BEQ	NOXMON	; BR IF NO
1035	013436	105777	175434			TSIB	@TPS	; TTY READY?
1036	013442	100003				BPL	NOXMON	; BR IF NO
1037	013444	117777	175420	175426		MOVB	@XDA,@TPB	; TYPE CHAR
1038	013452	005237	011070		NOXMON:	INC	XDA	; INC TTDR POINTER
1039	013456	005037	011032		XISR3:	CLR	TIME	
1040	013462	005037	013502			CLR	TRNFLG	
1041	013466	000002				RTI		
1042	013470	000000			ERROR1:	0		
1043	013472	000000			TEMP1:	0		
1044	013474	000000			TEMP2:	0		
1045	013476	000000			TCRTMP:	0		
1046	013500	000000			SNCFLG:	0		
1047	013502	000000			TRNFLG:	0		

```

1048 ;*****
1049 ; RECEIVER INITIALIZATION SUBROUTINE
1050 ;*****
1051
1052 013504 005737 013136 STARTR: TST STOP ;FIRST TIME HERE?
1053 013510 001005 BNE 1$ ;BR IF NO
1054 013512 104400 012712 TYPE MSG4 ;TYPE"MAKE CONNECTION"
1055 013516 005137 013136 COM STOP ;COMPLEMENT STOP
1056 013522 000000 HALT
1057 013524 032700 000004 1$: BIT #XLB,MODE ;XLB MODE?
1058 013530 001405 BEQ 2$ ;BR IF NO
1059 013532 005037 013472 CLR TEMP1 ;START DELAY
1060 013536 005237 013472 INC TEMP1
1061 013542 001375 BNE -4
1062 013544 042700 040000 2$: BIC #RFLG,STAT
1063 013550 013737 011020 011066 MOV IRDA,RDA ;SET UP RECEIVER DATA ADD
1064 013556 012737 001000 011064 MOV #1000,RCC ;SET UP BUFFER LIMIT
1065 013564 012737 177777 013500 MOV #-1,SNCFLG
1066 013572 005037 011054 CLR ERCSR ;CLEAR ERROR RECORDS
1067 013576 005037 011056 CLR ERDBR
1068 013602 005764 000002 3$: TST RBUF(R4) ;CLEAR SILO
1069 013606 100775 BMI 3$ ;KEEP CLEARING UNTIL BIT 15 CLEAR
1070 013610 013737 011012 013472 MOV PARAM1,TEMP1 ;GET READY TO LOAD PARAMETERS
1071 013616 052737 010000 013472 BIS #RCVON,TEMP1 ;BE SURE TO TURN RECEIVER ON
1072 013624 013764 013472 000002 MOV TEMP1,(R4) ;LOAD PARAMETERS, ENABLE RECEIVER
1073 013632 113764 013476 000005 MOVB TCRTP,TCR+1(R4) ;SET DATA TERMINAL READY
1074 013640 052714 000140 BIS #RIE!MSENAB,RCRCSR ;SET INTERRUPT ENABLE,RECEIVER ENABLE
1075 013644 000207 RTS PC
1076
1077 013646 105714 RISR: TSTB RCRCSR ;DID RECEIVER DONE SET?
1078 013650 100403 BMI 1$ ;BR IF YES
1079 013652 011402 MOV RCRCSR,R2 ;SAVE CSR
1080 013654 005003 CLR R3
1081 013656 104010 HLT 10 ;ERROR RECEIVER INTERRUPTED BUT DONE NOT SET
1082 013660 016401 000002 1$: MOV RBUF(R4),R1 ;GET CHAR
1083 013664 100403 BMI 2$ ;BR IF YES
1084 013666 011402 MOV RCRCSR,R2 ;SAVE CSR
1085 013670 005003 CLR R3
1086 013672 104010 HLT 10 ;ERROR CHAR PRESENT NOT SET
1087 013674 042701 000200 2$: BIC #200,R1 ;STRIP A BIT
1088 013700 032701 070000 BIT #ORUN+FRME+PARE,R1 ;CHECK FOR RECEIVER ERRORS
1089 013704 001404 BEQ 3$ ;BR IF NO ERRORS
1090 013706 011437 011054 MOV RCRCSR,ERCSR ;SAVE CSR
1091 013712 010137 011056 MOV R1,ERDBR ;SAVE RBUF
1092 013716 110177 175144 3$: MOVB R1,RDA ;STORE CHAR
1093 013722 032777 090040 175114 BIT #BIT5,RSWR ;MONITOR RXDATA?
1094 013730 001405 BEQ NORMON ;BR IF NO
1095 013732 105777 175140 TSTB JTPS ;IS TTY READY?
1096 013736 100002 BPL NORMON ;BR IF NO
1097 013740 110177 175134 MOVB R1,JTPB ;TYPE CHAR
1098 013744 005237 011066 NORMON: INC RDA ;INC RBUF POINTER
1099 013750 105077 175112 CLRB RDA ;CLEAR NEXT POSITION
1100 013754 005337 011064 DEC RCC ;DEC CHAR COUNT
1101 013760 001007 BNE 1$ ;BUFFER FULL YET?
1102 013762 000005 RESET ;STOP THE SHOW,BUFFER OVERFLOWED
1103 013764 005002 CLR R2

```

```

1104 013766 005003          CLR      R3
1105 013770 104000          HLT      0
1106 013772 104006          HLT      6          ;RECEIVER BUFFER FULL
1107 013774 000000          HALT
1108 013776 000776          BR      -2
1109 014000 123701 011041    15:      CMPB     RX,TERM,R1      ;IS CHAR 001?
1110 014004 001004          BNE     RISR1          ;BR IF NO
1111 014006 042714 000100          BIC     #R1,#RCSR     ;CLEAR RECEIVER INTERRUPT ENABLE
1112 014012 052700 040000          BIS     #RFLG,STAT    ;SET R DONE FLAG
1113 014016 005037 011032    RISR1:   CLR      TIME
1114 014022 005037 013500          CLR     SNCFLG
1115 014026 000002          RTI
1116 014030 005015 042522 042503 MFULL:   .ASCIZ<15><12>/RECEIVER ;GO HOME
      014067 015 042412 051122 SCANE:   .ASCIZ<15><12>/ERROR!  BUFFER FULL ERROR!!/
      014150          .EVEN      TRANSMITTER SCAN STOPPED ON WRONG LINE/
      000001          .END

```

BA	011004	605#	664					
BACK	013134	738#	766*	801*	881*	912	973#	
BIT0	= 000001	599#						
BIT1	= 000002	599#						
BIT10	= 002000	599#						
BIT11	= 004000	599#						
BIT12	= 010000	599#						
BIT13	= 020000	599#	634#	923	950			
BIT14	= 040000	599#						
BIT15	= 100000	599#						
BIT2	= 000004	599#						
BIT3	= 000010	599#						
BIT4	= 000020	599#						
BIT5	= 000040	599#	1093					
BIT6	= 000100	599#						
BIT7	= 000200	599#						
BIT8	= 000400	599#						
BIT9	= 001000	599#						
B2016	011030	615#	926					
OCLR	= 000020	599#	665	666				
DELAY	013132	690*	910*	972#	980	988*		
DERR	012544	951	960#					
DISPLA	011046	625#						
DSFLG	= 020000	599#	633#					
DSSTAT	011060	640#						
DZV11	11000	604#						
ENTER	012304	894	903#					
EOP	012216	739	767	802	882	889#		
ERCSR	011054	638#	1066*	1090*				
EROBR	011056	639#	921	1067*	1091*			
ERROR1	013470	1042#						
FLAG	011042	623#						
FRME	= 020000	599#	1088					
FULL.D	= 000001	652#	841	857	862			
GO	011242	689#						
ILB	= 000010	599#	698					
IRDA	011020	611#	805	930	953	1063		
IXDA	011022	612#	806	929	957	959	990	
KBDIN	= 104416	599#	720	751	783	840		
LOOP	= 000020	599#	736	764	799	879		
LPR	= 000002	599#	670*	1072*				
MFULL	014030	1116#						
MSENA8	= 000040	599#	1007	1012	1074			
MSG0	012556	925	966#					
MSG1	012637	928	966#					
MSG2	012642	952	966#					
MSG3	012667	956	966#					
MSG4	012712	966#	993	1054				
MSG5	013044	966#						
NODAT	= 000200	599#	732	795	876			
NORMON	013744	1094	1096	1098#				
NOXMON	013452	1034	1036	1038#				
ORJN	= 040000	599#	1088					
OWI	= 000002	599#	695					
OWO	= 000001	599#	692					
PARAM1	011012	608#	668	673	1022	1070		





BOX	1#	600	627	653	976	1048								
DCPAM	1#													
DHDOC1	1#													
DHPAM	1#													
DJPAM	1#													
DLPAM	1#													
DPPAM	1#													
DQDOC1	1#													
DQPAM	1#													
DUPAM	1#													
DUPPAR	1#													
DVDOC1	1#													
DVPAM	1#													
DZPAM	1#													
DZVPAR	561#													
HELLO	1#													
HLT	599#	728	760	792	822	854	870	962	1028	1081	1086	1105	1106	
SEQUAT	1#	599												
SINTF	1#	599												
SITEP	1#	682												
SSERV	1#	642												

. ABS. 014150 000

ERRORS DETECTED: 0

DVDZDA, DVDZDA.SEQ=ITEP1.MAC, DVDZDA.P11  
RUN-TIME: 3 4 .2 SECONDS  
RUN-TIME RATIO: 194/8=23.9  
CORE USED: 16K (31 PAGES)

E03