

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDZB-B-D
PRODUCT NAME: DZ11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM
PROGRAM DATE: OCTOBER 1976
MAINTAINER: DIAGNOSTICS
AUTHORS: R A JONES
JOHN EGOLF
JERRYL PAYNE

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) 1973, 1976, BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR FIELD SERVICE PERSONNEL. 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 DZ11 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 E.
 - 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 CARRAGE RETURN
 - B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
 - C. IF YOU WISH TO SETUP A DM11BB, TYPE IN DM.

IF DN OR DM WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS, VECTOR ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DM11BB.

- 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 PRIORITY
NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 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 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 91 IF THIS SETUP
WAS FOR DN11 OR DN11B8.
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 DN-11B ISR
SWR13=SETUP DN-11 ISR
SWR=00000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11, DN1:-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 DNBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DNBB.
 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 <OO!>)
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:

*WXXX:XXX TO PRINTOUT THE 8 WORDS AT LOC XXXXX.

*BXXXXXX TO PRINTOUT THE 16 BYTES AFTER LOC XXXXX.

*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 XXXXX 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 DZ11 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 "M" UPTS 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 JBR 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 DZ11

PARAM#1 IS LOADED INTO THE LINE PARAMETER REGISTER(DZLPR)
 BITS 0-2 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)

PARAM#2 IS NOT USED AT THIS POINT IN TIME

PARAM#3 IS NOT USED(177777).

DZ11 RESTRICTIONS

THE RTS MODEM SIGNAL ON THE D:11 IS JUMPER SELECTABLE
AT THE TERMINATION PANEL. IT IS EITHER ALWAYS ASSERTED OR
ASSERTED WHEN DATA TERMINAL READY (DTR) IS SET.
CONSEQUENTLY, AT THIS POINT IN TIME, DZ11 ITEP CAN NOT BE USED
WITH SERIES 200 AND OTHER HALF DUPLEX MODEMS. ALL ITEP
MODES ARE VALID WITH FULL DUPLEX MODEMS, AND ALL MODES
MAY BE USED TO TERMINALS (ONLY ONE WAY OUT AND IN ARE
RECOMMENDED HERE, HOWEVER).

```

588
589
590
591
592
593
594 011020 011000
595 011024 055104 000040
596 011024 160010
597 011026 000300
598 011010 000240
599 011012 011070
600 011014 177777
601 011016 177777
602 011022 000000
603 011024 000000
604 011026 000000
605 011030 000000
606 011032 000000
607 011034 000000
608 011036 011102
609 011040 000
610 011041 001
611 011041 000000
612 011042 000000
613 011044 177570
614 011046 177570
615
616
617
618
619 000000
620 100000
621 040000
622 020000
623 020000
624
625 011050 000000
626 011052 000000
627 011054 000000
628 011056 000000
629 011060 000000
630
631 011062 000000
632 011064 000000
633 011066 000000
634 011070 000000
635
636 011072 177560
637 011074 177562
638 011076 177564
639 011100 177566
640
641 000001

```

```

;*****
; DZ11 INTERFACE SERVICE PARAMS
;*****

```

```

DZ11: .ASCIZ /DZ /
BA: 160010
RIV: 300
PRIOR: 240
PARAM1: 11070
PARAM2: 177777
PARAM3: 177777
IRDA: .WORD 0
IXDA: .WORD 0
SETTLE: .WORD 0
B2016: .WORD 0
TIME: .WORD 0
TX.TERM: .WORD START
RX.TERM: .WORD 000
FLAG: .WORD 001
SWR: 177570
DISPLAY: 177570

```

```

;ISR NAME
;BUS ADDRESS
;VECTOR ADDRESS
;PRIORITY
;PARAM #1
;PARAM #2
;PARAM #3
;INITIAL READ DATA ADDRESS
;INITIAL XMIT DATA ADDRESS
;LINE SETTLE DELAY FLAG
;ADDR OF BIN TO OCT TYPE ROUTINE
;TIMER
;ADDR OF START OF PROGRAM
;TRANSMITTER TERMINATING CHAR.
;RECEIVER TERMINATING CHAR.

```

```

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

```

```

STAT=R0
XFLG=100000
RFLG=40000
DSFLG=20000
BIT13=20000

```

```

;XMIT COMPLETE FLAG
;RCV COMPLETE FLAG
;DATA SET STATUS CHANGE FLAG
;INHIBIT PRINTOUTS

```

```

SXCSR: 0
SRCSR: 0
ERCSR: 0
ERDBR: 0
DSSTAT: 0

```

```

;SAVED XMIT CSR
;SAVED RCV CSR
;RCV CSR SAVED ON ERROR
;RCV DATA REG SAVED ON ERROR
;RCV CSR SAVED ON DS CHANGE

```

```

XCC: 0
RCC: 0
RDA: 0
XDA: 0

```

```

;XMIT CHAR COUNT
;RCV CHAR COUNT
;RCV DATA ADDR.
;XMIT DATA ADDR.

```

```

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

```

```

FULL.DUPLX=000001

```

```

642
643
644
645 011102 000240
646 011104 017730 177734
647 011110 042730 177400
648 011114 013702 011006
649 011120 012722 013666
650 011124 013722 011010
651 011130 012722 013354
652 011134 013722 011010
653 011140 013704 011004
654 011144 013714 011012
655 011150 013702 011014
656 011154 042702 000001
657 011160 010264 000000
658 011164 052714 000020
659 011170 032714 000020
660 011174 001375
661 011176 013737 011012 013512
662 011204 042737 010000 013512
663 011212 013764 013512 000002
664 011220 010046
665 011222 012700 000001
666 011226 013701 011012
667 011232 042701 177770
668 011236 001403
669 011240 006300
670 011242 005301
671 011244 000774
672 011246 010037 012516
673 011252 012600
674 011254 113764 013516 000005
675
676
677
678
679
680
681
682 011262 005037 011032
683 011266 005037 013152
684 011272 005037 013156
685 011276 032700 000001
686 011302 001402
687 011304 000137 011460
688 011310 032700 000002
689 011314 001402
690 011316 000137 011352
691 011322 032700 000010
692 011326 001402
693 011330 000137 011556
694 011334 032700 000004
695 011340 001402
696 011342 000137 012006
697 011346 000000

```

```

*****
: DZ11-X INTERFACE SERVICE ROUTINE
*****
START: NOP
MOV @SWR, R0 ; SETUP MODE IN R0
BIC @177400, R0 ; 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
MOV PARAM1, @RCR ; SETUP VARIABLES
MOV PARAM2, R2
BIC @0001, R2
MOV R2, XCSR(R4) ; IN CSR'S
BIS @DCLR, @RCR ; CLEAR SILO+UARTS
BIT @DCLR, @RCR ; CLEAR PULSE DONE?
BNE IS ; BR IF NO
MOV PARAM1, TEMP1 ; DON'T TURN ON RECEIVER YET
BIC @RCVON, TEMP1 ; LOAD LINE NUMBER AND PARAMETERS
MOV TEMP1, (R4) ; SAVE R0
MOV R0, -(SP)
MOV @1, R0
MOV PARAM1, R1 ; ISOLATE THE LINE NUMBER
BIC @1<7>, R1 ; CALCULATE TCR BIT
BEQ 5$
ASL R0
DEC R1
BR 4$
MOV R0, TCRMP ; SAVE THE ACTIVE TCR BIT
MOV (SP)+, R0
MOV@ TCRMP, TCR+1(R4) ; SET DATA TERMINAL READY

```

```

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

```

```

60: CLR TIME
CLR DELAY
CLR STOP
BIT @OWO, MODE
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

```

698 011350 000776 BR -2

699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753

011352 104416
011354 004737 013524
011360 032700 040000
011364 001013
011366 023727 011032 000100
011374 103771
011376 011462
011400 016403 000000
011404 104001
011406 005037 011032
011412 000762

011414 032777 000200 177422
011422 001002
011424 004737 012376
011430 042700 040000
011434 032777 000020 177402
011442 001405
011444 012737 011456 013154
011452 000137 012236
011456 000735

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: KBOIN
JSR PC_STARTR
15: BIT #RFLG,STAT
BNE 25
CMP TIME,#100
BLO 15
MOV @RCSR,R2
MOV XCSR(A4),R3
HLT 1
CLR TIME
BR 15

25: BIT #NODAT,@SWR
BNE 35
JSR PC_TESTD
35: BIC #RFLG,STAT
BIT #LOOP,@SWR
BEQ 45
MOV #45,BACK
JMP EOP
45: BR SOWI

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".

011460 104416
011462 004737 013160
011466 005037 011032
011472 032700 100000
011476 001013
011500 023727 011032 000100
011506 103771
011510 011402
011512 016403 000000
011516 104001

SOWO: KBOIN
JSR PC_STARTX
CLR TIME
15: BIT #XFLG,STAT
BNE 25
CMP TIME,#100
BLO 15
MOV @RCSR,R2
MOV XCSR(A4),R3
HLT 1

754	011520	005037	011032			CLR	TIME
755	011524	000762				BR	IS
756	011526	042700	100000		2\$:	BIC	#XFLG, STAT
757	011532	032777	000020	177304		BIT	#LOOP, @SWR
758	011540	001405				BEQ	3\$
759	011542	012737	011554	013154		MOV	#3\$, BACK
760	011550	000137	012236			JMP	EOP
761	011554	000741			3\$:	BR	\$OWO
762							
763							
764							

765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819

011556 104416
011560 004737 013524
011564 005037 011032
011570 032700 040000
011574 001013
011576 023727 011032 000100
011604 103771
011606 011402
011610 016403 000000
011614 104001
011616 005037 011032
011622 000762
011624 032777 000200 177212 25:
011632 001002
011634 004737 012376
011640 042700 040000 35:
011644 032777 000020 177172
011652 001405
011654 012737 011666 013154
011662 000137 012236
011666 032777 000400 177150 45:
011674 001416
011676 013702 011020
011702 013703 011022
011706 010337 011070
011712 112223
011714 001376
011716 112743 000177
011722 005203
011724 112723 000177
011730 105023
011732 005037 011032 75:
011736 004737 013160
011742 032700 100000 55:
011746 001013
011750 023727 011032 000100
011756 103771
011760 011402
011762 016403 000000
011766 104001
011770 005037 011032
011774 000762
011776 042700 100000 65:
012002 000137 011556

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 REPEATED AS ABOVE.

SILB: KBD IN
JSR PC, STARTR
CLR TIME
15: BIT #RFLG, STAT
BNE 25
CMP TIME, #100
BLO 15
MOV @RCSR, R2
MOV XCSR(A4), R3
HLT
CLR TIME
BR 15
25: BIT #NODAT, @SWR
BNE 35
JSR PC, TESTD
35: BIC #RFLG, STAT
BIT #LOOP, @SWR
BEQ 45
MOV #45, BACK
JMP EOP
45: BIT #400, @SWR
BEQ 75
MOV IRDA, R2
MOV IXDA, R3
MOV R3, XDA
MOVVB (R2)+, (R3)+
BNE -2
MOVVB @177, -(R3)
INC R3
MOVVB @177, (R3)+
CLRB (R3)+
75: CLR TIME
JSR PC, STARTX
55: BIT #XFLG, STAT
BNE 65
CMP TIME, #100
BLO 55
MOV @RCSR, R2
MOV XCSR(A4), R3
HLT 1
CLR TIME
BR 55
65: BIC #XFLG, STAT
JMP SILB

: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

820
821
822
823
824
825
826
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

012006 104416
012010 032737 000001 011014
012016 001402
012020 004737 013524
012024 004737 013160
012030 005037 011032
012034 032700 100000
012040 001016
012042 032700 040000
012046 001024
012050 023727 011032 000100
012056 103766
012060 011402
012062 016403 000000
012066 104001
012070 005037 011032
012074 000757
012076 032737 000001 011014
012104 001356
012106 042700 100000
012112 004737 013524
012116 000746
012120 032737 000001 011014
012126 001420
012130 032700 100000
012134 001013
012136 023727 011032 000100
012144 103765
012146 011402
012150 016403 000000
012154 104001
012156 005037 011032
012162 000756
012164 042700 100000
012170 042700 040000
012174 005037 011032
012200 032777 000200 175636
012206 001012
012210 004737 012376
012214 032777 000020 176622
012222 001671
012224 012737 012006 013154
012232 000137 012236

```
*****
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.
*****
```

```
5XLB: 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
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$: BIT #FULL.DUPLEX,PARAM2
BEQ 9$
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 5XLB
MOV #5XLB,BACK
JMP EOP
```

```

876
877
878
879
880
881
882 012236
883 012236 104414 000340
884 012242 016437 000000 012374
885 012250 042737 137777 012374
886 012256 042764 040000 000000
887 012264 012766 012324 000002
888 012272 010037 013136
889 012276 010137 013140
890 012302 010237 013142
891 012306 010337 013144
892 012312 010437 013146
893 012316 010537 013150
894 012322 000207
895
896 012324
897 012324 013700 013136
898 012330 013701 013140
899 012334 013702 013142
900 012340 013703 013144
901 012344 013704 013146
902 012350 013705 013150
903 012354 012737 177777 013152
904 012362 053764 012374 000000
905 012370 000177 000560
906 012374 000000
907
908
909
910
911
912
913
914 012376 013746 011056
915 012402 001413
916 012404 032777 020000 176432
917 012412 001007
918 012414 104400 012576
919 012420 004077 176404
920 012424 005746
921 012426 104400 012657
922 012432 013701 011022
923 012436 013702 011020
924 012442 122122
925 012444 001776
926 012446 123741 011040
927 012452 001447
928 012454 122742 000002
929 012460 001005
930 012462 010237 012470
931 012466 104400

```

```

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

```

```

EOP:
STPS, PRTY7 ; SET PS PRIORITY TO 7
MOV XCSR(R4), QTPIE ; SAVE TX CSR
BIC #1<TIE>, QTPIE ; CLEAR ALL BUT TX IE.
BIC #TIE, XCSR(R4) ; CLEAR TX IE (EVEN IF IT WASN'T SET)
MOV #ENTER, 2(SP) ; SET FOR RETURN IF SW 14=1
MOV R0, 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 SAVR0, R0 ; RESTORE R0
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
BIS QTPIE, XCSR(R4) ; IF ORIGINALLY SET; SET TX IE
JMP @BACK
QTPIE: 000000

```

```

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

```

```

TESTD: MOV ERDR, -(SP) ; WAS THERE A RECEIVE ERROR?
BEQ TSTDAT ; BR IF NO
BIT #BIT13, @SWR ; INHIBIT PRINTOUTS?
BNE TSTDAT ; BR IF YES
TYPE MSG0 ; (15)<12> THERE WAS A RECEIVE ERROR RBUF=
JSR @J282016 ; PRINT CONTENTS OF RBUF
TST -(SP)
TYPE MSG1 ; (15)<12>
TSTDAT: MOV IXDA, R1 ; SETUP XMIT DATA ADDR
MOV IRDA, R2 ; SETUP RCV DATA ADDR
SCAN4: CMPB (R1)+, (R2)+ ; DATA OK?
BEQ ; BR IF OK
CMPB TX.TERM, -(R1) ; IS IT END OF DATA
BEQ TESTDX ; BR IF YES
CMPB #002, -(R2)
SNE 2$
MOV R2, 1$
TYPE

```



```

969
970
971
972
973 013160 005737 013152 STARTX: TST DELAY ;IF SW04=1 & SW14=0 DELAY
974 013164 001415 BEQ 15 ;NO DELAY START TRANSMITTER
975 013166 005037 013512 CLR TEMP1 ;PREPARE FOR DELAY
976 013172 012737 000007 013514 MOV #7,TEMP2
977 013200 005237 013512 INC TEMP1 ;INCREMENT DELAY
978 013204 001375 BNE -4
979 013206 005337 013514 DEC TEMP2
980 013212 001372 BNE -12
981 013214 005037 013152 CLR DELAY
982 013220 042700 100000 15: BIC #XFLG,STAT
983 013224 013737 011022 011070 MOV IXDA,XDA ;SET UP XMIT DATA ADDR
984 013232 005737 013156 TST STOP ;FIRST TIME HERE?
985 013236 001020 BNE 25 ;NO
986 013240 104400 012732 TYPE ,MSG4 ;MAKE CONNECTION
987 013244 000000 HALT
988 013246 005137 013156 COM STOP ;COMPLEMENT STOP
989 013252 005037 013512 CLR TEMP1 ;YES PREPARE FOR DELAY
990 013256 012737 000030 013514 MOV #14*2,TEMP2
991 013264 005237 013512 INC TEMP1 ;INCREMENT DELAY
992 013270 001375 BNE -4
993 013272 005337 013514 DEC TEMP2
994 013276 001372 BNE -12
995 013300 013764 013516 000004 25: MOV TCRTEMP,TCR(R4) ;SET LINE # IN TCR
996 013306 113764 013516 000005 MOVB TCRTEMP,TCR+1(R4) ;SET DATA TERMINAL READY
997 013314 032700 000004 BIT #XLB,MODE ;XLB MODE?
998 013320 001412 BEQ 35 ;BR IF NO
999 013322 012737 177777 013522 MOV #-1,TRNFLG ;SET FLAG
1000 013330 052714 040040 BIS #TIE+MSENAB,DRCSR ;SET INTERRUPT ENABLE
1001 013334 000001 WAIT
1002 013336 000240 NOP
1003 013340 005737 013520 TST SNCF LG ;FIRST CHAR RECEIVED YET?
1004 013344 001375 BNE -4 ;BR IF NO
1005 013346 052714 040040 35: BIS #TIE+MSENAB,DRCSR ;SET INTERRUPT ENABLE,SCAN ENABLE
1006 013352 000207 RTS PC
1007
1008 013354 127737 175510 011040 XISR: CMPB @XDA,TX.TERM ;IS CHAR TRANSMITTER TERMINATION CHAR
1009 013362 001005 BNE XISR1 ;BR IF NO
1010 013364 052700 100000 BIS #XFLG,STAT ;SET XMIT DONE FLAG
1011 013370 042714 340000 BIC #TIE,DRCSR ;CLEAR ENABLES
1012 013374 000440 BR XISR3
1013 013376 116405 000001 XISR1: MOVB 1(R4),R5 ;GET LINE NUMBER OF READY LINE
1014 013402 042705 177770 BIC #C(7),R5 ;ISOLATE THE LINE NUMBER
1015 013406 013701 011012 MOV PARAM1,R1 ;GET THE EXPECTED LINE NUMBER
1016 013412 042701 177770 BIC #C(7),R1 ;ISOLATE IT
1017 013416 120501 CMPB R5,R1 ;ARE THEY EQUAL?
1018 013420 001407 BEQ XISR2 ;IF SO, GO TRANSMIT W CHARACTER
1019 013422 011402 MOV DRCSR,R2 ;SET UP R2 WITH CSR CONTENTS
1020 013424 005003 CLR R3
1021 013426 104010 HALT 10 ;ERROR WRONG LINE
1022 013430 104400 014107 TYPE ,SCANE ;TYPE ERROR MESSAGE
1023 013434 000000 HALT
1024 013436 000776 BR -2
    
```

1025	013440	117764	175424	000006	XISR2:	MOV8	2XDA, TDR(R4)	: TRANSMIT DATA
1026	013446	032777	000100	175370		BIT	#100, 25WR	: MONITOR TX DATA?
1027	013454	001406				BEQ	NOXMON	: BR IF NO
1028	013456	105777	175414			TSTB	2TPS	: TTY READY?
1029	013462	100003				BPL	NOXMON	: BR IF NO
1030	013464	117777	175400	175406		MOV8	2XDA, 2TPB	: TYPE CHAR
1031	013472	005237	011070		NOXMON:	INC	XDA	: INC TDR POINTER
1032	013476	005037	011032		XISR3:	CLR	TIME	
1033	013502	005037	013522			CLR	TRNFLG	
1034	013506	000002				RTI		
1035	013510	000000			ERROR1:	0		
1036	013512	000000			TEMP1:	0		
1037	013514	000000			TEMP2:	0		
1038	013516	000000			TCRTMP:	0		
1039	013520	000000			SNCFLG:	0		
1040	013522	000000			TRNFLG:	0		

```

1041 ;*****
1042 ; RECEIVER INITIALIZATION SUBROUTINE
1043 ;*****
1044
1045 013524 005737 013156 STARTR: TST STOP ;FIRST TIME HERE?
1046 013530 001005 BNE IS ;BR IF NO
1047 013532 104400 012732 TYPE MSG4 ;TYPE MAKE CONNECTION
1048 013536 005137 013156 COM STOP ;COMPLEMENT STOP
1049 013542 000000 HALT
1050 013544 032700 000004 1S: BIT #XLB,MODE ;XLB MODE?
1051 013550 001405 BEQ 2S ;BR IF NO
1052 013552 005037 013512 CLR TEMP1 ;START DELAY
1053 013556 005237 013512 INC TEMP1
1054 013562 001375 BNE -4
1055 013564 042700 040000 2S: BIC #RFLG,STAT
1056 013570 013737 011020 011066 MOV IRDA,RDA ;SET UP RECEIVER DATA ADD
1057 013576 012737 001000 011064 MOV #1000,RCC ;SET UP BUFFER LIMIT
1058 013604 012737 177777 013520 MOV #-1,SNCFLG
1059 013612 005037 011054 CLR ERCSR ;CLEAR ERROR RECORDS
1060 013616 005037 011056 CLR ERDBR
1061 013622 005764 000002 3S: TST RBUF(R4) ;CLEAR SILO
1062 013626 100775 BMI 3S ;KEEP CLEARING UNTIL BIT IS CLEAR
1063 013630 013737 011012 013512 MOV PARAM1,TEMP1 ;GET READY TO LOAD PARAMETERS
1064 013636 052737 010000 013512 BIS #RCVON,TEMP1 ;BE SURE TO TURN RECEIVER ON
1065 013644 013764 013512 000002 MOV TEMP1,LPR(R4) ;LOAD PARAMETERS, ENABLE RECEIVER
1066 013652 113764 013516 000005 MOVB TCRTMP,TCR+1(R4) ;SET DATA TERMINAL READY
1067 013660 052714 000140 BIS #RIE!MSENAB,IRCSR ;SET INTERRUPT ENABLE,RECEIVER ENABLE
1068 013664 000207 RTS PC
1069
1070 013666 105714 RISR: TSTB IRCSR ;DID RECEIVER DONE SET?
1071 013670 100403 BMI IS ;BR IF YES
1072 013672 011402 MOV IRCSR,R2 ;SAVE CSR
1073 013674 005003 CLR R3
1074 013676 104010 HLT 10 ;ERROR RECEIVER INTERRUPTED BUT DONE NOT SET
1075 013700 016401 000002 1S: MOV RBUF(R4),R1 ;GET CHAR
1076 013704 100403 BMI 2S ;BR IF YES
1077 013706 011402 MOV IRCSR,R2 ;SAVE CSR
1078 013710 005003 CLR R3
1079 013712 104010 HLT 10 ;ERROR CHAR PRESENT NOT SET
1080 013714 042701 000200 2S: BIC #200,R1 ;STRIP A BIT
1081 013720 032701 070000 BIT #ORUN+FRME+PARE,R1 ;CHECK FOR RECEIVER ERRORS
1082 013724 001404 BEQ 3S ;BR IF NO ERRORS
1083 013726 011437 011054 MOV IRCSR,ERCSR ;SAVE CSR
1084 013732 010137 011056 MOV R1,ERDBR ;SAVE RBUF
1085 013736 110177 175124 3S: MOVB R1,IRDA ;STORE CHAR
1086 013742 032777 000040 175074 BIT #BITS,ISWR ;MONITOR RXDATA?
1087 013750 001405 BEQ NORMON ;BR IF NO
1088 013752 105777 175120 TSTB IRTPS ;IS ITTY READY?
1089 013756 100002 BPL NORMON ;BR IF NO
1090 013760 110177 175114 MOVB R1,IRTPB ;TYPE CHAR
1091 013764 005237 011066 NORMON: INC RDA ;INC RBUF POINTER
1092 013770 105077 175072 CLRB IRDA ;CLEAR NEXT POSITION
1093 013774 005337 011064 DEC RCC ;DEC CHAR COUNT
1094 014000 001007 BNE IS ;BUFFER FULL YET?
1095 014002 000005 RESET ;STOP THE SHOW,BUFFER OVERFLOWED
1096 014004 005002 CLR R2

```



```

1097 014006 005003          CLR      R3
1098 014010 104000          HLT      0
1099 014012 104006          HLT      6          ;RECEIVER BUFFER FULL
1100 014014 000000          HALT
1101 014016 J00776          BR
1102 014020 123701 01104!        IS:  CMPB   -2          ; IS CHAR 001?
1103 014024 01004          BNE      RISRI      ; BR IF NO
1104 014026 042714 000100        BIC      #RIE, JRCSR ; CLEAR RECEIVER INTERRUPT ENABLE
1105 014032 052700 040000        BIS      #RFLG, STAT ; SET R DONE FLAG
1106 014036 005037 011032        RISRI:  CLR      TIME
1107 014042 005037 013520        CLR      SNCFLG
1108 014046 000002          RTI
1109 014050 005015 042522 042503 MFULL:  .ASCIZ<15><12>/RECEIVER ; GO HOME
      015 042412 051122 SCANE:  .ASCIZ<15><12>/ERROR! ; BUFFER FULL ERROR!!/
      014170          .EVEN ; TRANSMITTER SCAN STOPPED ON WRONG LINE/
      000001          .END

```

BA	011004	594#	653				
BACK	013154	731*	759*	794*	874*	905	966#
BIT0	= 000001	588#					
BIT1	= 000002	588#					
BIT10	= 002000	588#					
BIT11	= 004000	588#					
BIT12	= 010000	588#					
BIT13	= 020000	588#	623#	916	943		
BIT14	= 040060	588#					
BIT15	= 100000	588#					
BIT2	= 000004	588#					
BIT3	= 000010	588#					
BIT4	= 000020	588#					
BIT5	= 000040	588#	1086				
BIT6	= 000100	588#					
BIT7	= 000200	588#					
BIT8	= 000400	588#					
BIT9	= 001000	588#					
B2016	011030	604#	919				
DCLR	= 000020	588#	658	659			
DELAY	013152	683*	903*	965#	973	981*	
DERR	012564	944	953#				
DISPLA	011046	614#					
DSFLG	= 020000	588#	622#				
DSSTAT	011060	629#					
DZ11	011000	593#					
ENTER	012324	867	896#				
EOP	012236	732	760	795	875	882#	
ERCSR	011054	627#	1059*	1083*			
ERDBR	011056	628#	914	1060*	1084*		
ERROR1	013510	1035#					
FLAG	011042	612#					
FRME	= 020000	588#	1081				
FULL.D	= 000001	641#	834	850	855		
GO	011262	682#					
ILB	= 000010	588#	691				
IRDA	011020	600#	798	923	946	1056	
IXCA	011022	601#	799	922	950	952	983
KBDIN	= 104416	588#	713	744	776	833	
LOOP	= 000020	588#	729	757	792	872	
LPR	= 000002	588#	663*	1065*			
MFULL	014050	1109#					
MSENAB	= 000040	588#	1000	1005	1067		
MSG0	012576	918	959#				
MSG1	012657	921	959#				
MSG2	012662	945	959#				
MSG3	012707	949	959#				
MSG4	012732	959#	986	1047			
MSG5	013064	959#					
NODAT	= 000200	588#	725	788	869		
NORMON	013764	1087	1089	1091#			
XCXMON	013472	1027	1029	1031#			
ORUN	= 040000	588#	1081				
OWI	= 000002	588#	688				
OWO	= 000001	588#	685				
PARAM1	011012	597#	654	661	666	1015	1063

TKB	011074	637#																		
TKS	011072	636#																		
TPB	011100	639#	1030*	1090*																
TPS	011076	638#	1029	1088																
TRNFLG	013522	999#	1033*	1040#																
TSTDAY	012432	915	917	922#																
TX. TER	011040	608#	926	1008																
TYPE =	104400	588#	918	921	931	945	947	949	951	986	1022	1047								
XCC	011062	631#																		
XCSR =	000000	588#	657*	720	752	784	814	846	852	884	886*	904*								
XOR	011070	654#	800*	983	1008	1025	1030	1031*												
XFLG =	100000	620#	747	756	809	818	839	852	857	866	982	1010								
XISR	013354	651	1008#																	
XISR1	013376	1009	1013#																	
XISR2	013440	1018	1025#																	
XISR3	013476	1012	1032#																	
XLB =	000004	588#	694	997	1050															
XWAIT =	104412	588#																		
\$ILB	011556	693	776#	819																
\$CHI	011352	690	713#	733																
\$OMO	011460	687	744#	761																
\$XLB	012006	696	833#	873	874															
.	= 014170	592#	698	802	950*	978	980	992	994	1004	1024	1054	1101	1109#						

BOX	18	559	616	642	969	1041								
DCPARM	18													
DHDOC1	18													
DHPARM	18													
DJPARM	18													
DLPARM	18													
DPPARM	18													
DQDOC1	18													
DQPARM	18													
DUPARM	18													
DUPPAR	18													
DVDOC1	18													
DVPARM	18													
DZPARM	18	560												
HELLO	18													
HLT	588	721	753	785	815	847	863	955	1021	1074	1079	1098	1099	
SEQUAT	18	588												
SINTF	18	588												
SITEP	18	675												
SSERV	18	631												

. ABS. 014170 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDZBB.SEQ/SOL/CRF/NL:TOC=ITEP1.MAC,DZDZBB.P11
RUN-TIME: 10 13 .7 SECONDS
RUN-TIME RATIO: 57/26=2.2
CORE USED: 16K (31 PAGES)

