

DP11

OVERLAY FOR ITEP
MD-11-DZDPO-C

EP-DZDPO-C-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

This microfiche card contains a grid of frames. The first column contains 14 frames of text, likely a table of contents or index. The second column contains 14 frames of diagrams or data tables. The third column contains 14 frames of text, possibly descriptions or additional data. The frames are arranged in a 3x14 grid, with the top row having three frames and the subsequent two rows having four frames each.

IDENTIFICATION

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

PRODUCT NAME: DP11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: OCTOBER 1976

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
 JIM 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) 1973, 1976. 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 GL-11 INTERFACE.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

- A. PDP-11 SYSTEM WITH 4K OF CORE.
- B. A DP11 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 DN.1 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. 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 B1 IF THIS SETUP
WAS FOR DN11 OR DN11BB.
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=00000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
 SETUP SEQUENCE IS: DN11, DN11-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 DN11B PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DN11.
 3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
 - A. PRESS CONTINUE TO START TESTING

BEFORE ATTEMPTING TO RUN THIS PROGRAM, THE OPERATOR MUST ASCERTAIN THE COMPLETE COMMUNICATION LOOP AND 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 CONTEST/QUICK BROWN FOX/NUMERICS)
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
SW07=1 DO NOT TEST RECEIVED DATA
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
* IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.
SW04=1 RETURN TO MONITOR FOR END PASS
WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.
SW03=1 INTERNAL LOOPBACK MODE
SW02=1 EXTERNAL LOOPBACK MODE
SW01=1 ONE-WAY-IN MODE
SW00=1 ONE-WAY-OUT MODE

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTRL 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 OPERATOR, 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 (SW9=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:
- #XXXXXX 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=RRRRR
DATA SHOULD BE TTTTT
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 DP11 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 DP11

PARAM#1 IS LOADED INTO THE RECEIVER STATUS REGISTER(RCSR)
BIT 1 HALF DUPLEX (1), DEFAULT= HALF DUPLEX (1)
BITS 8,9,10 CHARACTER LENGTH, DEFAULT= 8 BITS (000)

PARAM#2 IS LOADED INTO THE TRANSMITTER STATUS REGISTER (XCSR).
BIT 0 FULL DUPLEX (1), DEFAULT= HALF DUPLEX (0)
(BIT 0) IS NOT LOADED INTO THE XCSR, IT IS ONLY A SOFTWARE FLAG)

PARAM#2(HIGH BYTE) IS LOADED INTO THE SYNC REGISTER, DEFAULT=26 (26)

PARAM#3 IS NOT USED (177777).

```

578
579
580
581
582      011000
583 011000 050104 000040
584 011004 174770
585 011006 000300
586 011010 000240
587 011012 000002
588 011014 013000
589 011016 177777
590 011020 000000
591 011022 000000
592 011024 000000
593 011026 000000
594 011030 000000
595 011032 000000
596 011034 000000
597 011036 011102
598 011040
599 011040      000
600 011041
601 011041      001
602 011042 000000
603 011044 177570
604 011046 177570
605
606
607
608
609      000000
610      100000
611      040000
612      020000
613      020000
614
615 011050 000000
616 011052 000000
617 011054 000000
618 011056 000000
619 011060 000000
620
621 011062 000000
622 011064 000000
623 011066 000000
624 011070 000000
625
626 011072 177560
627 011074 177562
628 011076 177564
629 011100 177566
630
631      000001

```

```

;*****
; DP11 INTERFACE SERVICE PARAMS
;*****

```

```

DP11:      =11000
BA:        .ASCIZ /DP /
RIV:       300
PRIOR:     240
PARAM1:    2
PARAM2:    13000
PARAM3:    177777
IRDA:      .WORD 0
IXDA:      .WORD 0
SETTLE:    .WORD 0
B2016:     .WORD 0
TIME:      .WORD 0
           .WORD 0
           .WORD START
TX. TERM:  .BYTE 000
RX. TERM:  .BYTE 001
FLAG:      .WORD 0
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
EPDSR: 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
TKB: 177562
TPS: 177564
TPB: 177566

```

```

FULL.DUPLEX=000001

```

```

632
633
634
635 011102 000240
636 011104 017700 177734
637 011110 042700 177400
638 011114 013702 011006
639 011120 012722 013606
640 011124 013722 011010
641 011130 012722 013514
642 011134 013722 011010
643 011140 013704 011004
644 011144 013714 011012
645 011150 013702 011014
646 011154 042702 000001
647 011160 010264 000004
648 011164 113764 011015 000003
649 011172 005037 011032
650 011176 005037 013102
651
652
653
654
655
656
657
658 011202 005037 011032
659 011206 005037 013102
660 011212 005037 013106
661 011216 032700 000001
662 011222 001402
663 011224 000137 011400
664 011230 032700 030002
665 011234 001402
666 011238 000137 011272
667 011242 032700 000010
668 011246 001402
669 011250 000137 011476
670 011254 032700 000004
671 011260 001402
672 011262 000137 011726
673 011266 000006
674 011270 000776
675
676
677
678
679
680
681
682
683
684
685
686
687

```

```

*****
DP11-X INTERFACE SERVICE ROUTINE
*****

```

```

START:  NOP
        MOV     JSWR,   R0      ; SETUP MODE IN R0
        BIC     #177400, R0    ; STRIP JUNK
        MOV     RIV,   R2      ; SETUP
        MOV     BRISR, (R2)+   ; INTERRUPT
        MOV     PRIOR, (R2)+  ; VECTORS
        MOV     BXISR, (R2)+
        MOV     PRIOR, (R2)+
        MOV     BR,    R4      ; SETUP BUS ADDR INDEX
        MOV     PARAM1, 2RCSR  ; SETUP VARIABLES
        MOV     PARAM2, R2
        BIC     #0001, R2
        MOV     R2,     XCSR(R4); IN CSR'S
        MOVB   PARAM2+1, SYNC(4); SETUP SYNC CHAR.
        CLR    TIME          ; RESET TIMER
        CLR    D.LAY        ; SET FOR NO TIME DELAY.

```

```

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

```

```

GO:     CLR    TIME
        CLR    DELAY
        CLR    STOP
        BIT    #0W0, MODE
        BEQ   IS
        JMP   $0W0
IS:     BIT    #0W1, MODE
        BEQ   2S
        JMP   $0W1
2S:     BIT    #1LB, MODE
        BEQ   3S
        JMP   $1LB
3S:     BIT    #XLB, MODE
        BEQ   4S
        JMP   $XLB
4S:     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".
*****

```


741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
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

011476 104416
011500 004737 013404
011504 005037 011032
011510 032700 040000
011514 001013
011516 023727 011032 000100
011524 103771
011526 011402
011530 016403 000004
011534 104001
011536 005037 011032
011542 000762
011544 032777 000200 177272 2S:
011552 001002 3S:
011554 004737 012316
011560 042700 040000 3S:
011564 032777 000020 177252
011572 001405
011574 012737 011606 013104
011602 000137 012156
011606 032777 000400 177230 4S:
011614 001416
011616 013702 011020
011622 013703 011022
011626 010337 011070
011632 112223
011634 001376
011636 112743 000177
011642 005203
011644 112723 000177
011650 105023
011652 005037 011032 7S:
011656 004737 013110
011662 032700 100000 5S:
011666 001013
011670 023727 011032 000100
011676 103771
011700 011402
011702 016403 000004
011706 104001
011710 000337 011032
011714 000762
011716 042700 100000 6S:
011722 000137 011476

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.

SILB: KBOIN
JSR PC, STARTR
CLR TIME
1S: BIT #RFLG, STAT
BNE 2S
CMP TIME, #100
BLO 1S
MOV @RCSR, R2
MOV XCSR(A4), R3
HLT
CLR TIME
BR 1S
2S: BIT #MODAT, @SWR
BNE 3S
JSR PC, TESTD
3S: BIC #RFLG, STAT
BIT #LOOP, @SWR
BEQ 4S
MOV #4S, BACK
4S: JMP EOP
BIT #400, @SWR
BEQ 7S
MOV IR0A, R2
MOV IX0A, R3
MOV R3, X0A
MOVSB (R2)+, (R3)+
BNE -2
MOVSB #177, -(R3)
INC R3
MOVSB #177, (R3)+
CLRB (R3)+
7S: CLR TIME
JSR PC, STARTX
5S: BIT #XFLG, STAT
BNE 6S
CMP TIME, #100
BLO 5S
MOV @RCSR, R2
MOV XCSR(A4), R3
HLT
CLR TIME
BR 5S
6S: 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

```

796
797
798
799
800
801
802
803
804
805
806
807
808
809 011726 104416
810 011730 032737 000001 011014
811 011736 001402
812 011740 004737 013404
813 011744 004737 013110
814 011750 005037 011032
815 011754 032700 100000
816 011760 001016
817 011762 032700 040000
818 011766 001024
819 011770 023727 011032 000100
820 011776 103766
821 012000 011402
822 012002 016403 000004
823 012006 104001
824 012010 005037 011032
825 012014 000757
826 012016 032737 000001 011014
827 012024 001356
828 012026 042700 100000
829 012032 004737 013404
830 012036 000746
831 012040 032737 000001 011014
832 012046 001420
833 012050 032700 100000
834 012054 001013
835 012056 023727 011032 000100
836 012064 103765
837 012066 011402
838 012070 016403 000004
839 012074 104001
840 012076 005037 011032
841 012102 000756
842 012104 042700 100000
843 012110 042700 040000
844 012114 005037 011032
845 012120 032777 000200 176716
846 012126 001002
847 012130 004737 012316
848 012134 032777 000020 176702
849 012142 001671
850 012144 012737 011726 013104
851 012152 000137 012156

```

```

*****
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      #NOOAT,@SWR
      BNE      5$
      JSR      PC,TESTD
5$:    BIT      #LOOP,@SWR
      BEQ      $XLB
      MOV      #XLB,BACK
      JMP      EOP

```

```

852
853
854
855
856
857
858 012156
859 012156 104414 000340
860 012162 016437 000004 012314
861 012170 042737 177677 012314
862 012176 042764 000100 000004
863 012204 012766 012244 000002
864 012212 010037 013056
865 012216 010137 013070
866 012222 010237 013072
867 012226 010337 013074
868 012232 010437 013076
869 012236 010537 013100
870 012242 000207
871
872 012244
873 012244 013700 013066
874 012250 013701 013070
875 012254 013702 013072
876 012260 013703 013074
877 012264 013704 013076
878 012270 013705 013100
879 012274 012737 177777 013102
880 012302 053764 012314 000004
881 012310 000177 000570
882 012314 000000
883
884
885
886
887
888
889
890 012316 013746 011056
891 012332 001413
892 012334 032777 020000 176512
893 012332 001007
894 012334 104400 012526
895 012340 004077 176464
896 012344 005746
897 012346 104400 012607
898 012352 013701 011022
899 012356 013702 011020
900 012362 122122
901 012364 001776
902 012366 123741 011040
903 012372 001453
904 012374 122742 000002
905 012400 001005
906 012402 010237 012410
907 012406 104400

```

```

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

```

```

EOP:
STPS,PRTY7 ;SET PS PRIORITY TO 7
MOV XCSR(R4),QPIE ;SAVE TX CSR
BIC #1<TIE>,QPIE ;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 ;-1 DELAY
BIS QPIE,XCSR(R4) ;IF ORIGINALLY SET; SET TX IE
JMP @BACK
QPIE: 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. REUF=
JSR R0,@R2016 ;PRINT CONTENTS OF REUF
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 SCAN4 ;BR IF OK
CMPB TX_TERM,-(R1) ;IS IT END OF DATA
BEQ TESTDX ;BR IF YES
CMPB #002,-(R2)
BNE ZS
MOV R2,IS
TYPE

```


H02

DP11 ITEP OVERLAY MACY11 27(1006) 29-OCT-76 14:41 PAGE 21
D.TDPOC.P11 05-AUG-76 09:23

```
948 ;*****
949 ;      INITIALIZE TRANSMIT SUBROUTINE
950 ;*****
951
952 013110 005037 011032      STARTX: CLR      TIME
953 013114 005737 013102            TST      DELAY
954 013120 001416            REQ      15            ; IF SW04=1 & SW14=0 WAIT BEFORE TURNING ON TX
955 013122 005037 013510            CLR      TEMP1            ; NO GO AHEAD AND TURN ON TX
956 013126 012737 000007 013512      MOV      #7,TEMP2            ; PREPARE FOR DELAY
957 013134 062737 000001 013510      ADD      #1,TEMP1            ; INCREMENT DELAY.....
958 013142 001374            BNE      -6
959 013144 005337 013512            DEC      TEMP2
960 013150 001371            BNE      -14
961 013152 005037 013102            CLR      DELAY            ; ZERO POINTER.
962 013156 013737 011070 15:      MOV      IXDA, XDA            ; SETUP XMIT DATA ADDR.
963 013164 052700 100000            BIC      #XFLG, STAT            ; RESET XMIT COMPLETE FLAG
964 013170 052764 000001 000004      BIS      #DTR,XCSR(R4)            ; SET DATA TERMINAL READY.
965 013176 005737 013106            TST      STOP
966 013202 001004            BNE      CTSW
967 013204 104400 013014            TYPE    MSGS
968 013210 005137 013106            COM      STOP
969
970 013214 032764 010000 000004      CTSW:    BIT      #MRDY, XCSR(R4)            ; IS MODEM READY SET?
971 013222 001037            BNE      CTSOK            ; BR IF YES
972 013224 032764 010000 000004      CTSXYZ: BIT      #MRDY,XCSR(R4)            ; KEEP CHECKING MODEM READY.
973 013232 001017            BNE      CTSABC            ; BR IF READY NOW.
974 013234 023727 011032 000036      CTSWLN: CMP      TIME, #36            ; 30 SECS ELAPSED?
975 013242 103770            BLO      CTSXYZ            ; BR IF NO
976 013244 011402            MOV      #RCSR, R2            ; SETUP RECEIVE CSR
977 013246 016403 000004            MOV      XCSR(R4),R3            ; SETUP XMIT CSR
978 013252 032777 010000 175564      BIT      #SW12,#SWR            ; INHIBIT PRINTOUTS?
979 013260 001001            BNE      15            ; BR IF YES
980 013262 104002            HLT+2            ; PRINTOUT "WAITING TO XMIT" MESSAGE
981 013264 005037 011032      15:      CLR      TIME            ; RESET TIMER
982 013270 000755            BR      CTSXYZ            ; WAIT SOME MORE
983 013272 005037 013510      CTSABC: CLR      TEMP1            ; PREPARE FOR DELAY
984 013276 012737 000005 013512      MOV      #5,TEMP2
985 013304 062737 000001 013510      ADD      #1,TEMP1            ; INCREMENT DELAY.....
986 013312 001374            BNE      -6
987 013314 005337 013512            DEC      TEMP2
988 013320 001371            BNE      -14
989 013322 032737 000001 011014      CTSOK: BIT      #FULL.DUPLEX,PARAM2
990 013330 001004            BNE      25
991 013332 032764 004000 000004      BIT      #4000,XCSR(R4)
992 013340 001374            BNE      -6
993 013342 052764 001000 000004      25:    BIS      #RQTS,XCSR(R4)
994 013350 012702 000005            MOV      #5,R2            ; SETUP SYNC. COUNTER
995
996 013354 113764 011015 000006      15:    MOV8    PARAM2+1,XBUF(R4);LOAD A SYNC CHAR
997
998 013362 105764 000004            TSTB    XCSR(R4)            ; IS XMIT READY
999 013366 100375            BPL      -4            ; BR IF NO
1000 013370 005302            DEC      R2            ; DECREMENT COUNTER
1001 013372 001370            BNE      15            ; BR IF NOT ZERO
1002
1003 013374 052764 000100 000004      BIS      #TIE, XCSR(R4);SET XMIT INTERRUPT ENABLE
```

```

1004 013402 000207          RTS      PC          ;EXIT FROM SUBROUTINE
1005
1006          ;*****
1007          ;***** INITIALIZE RECEIVER SUBROUTINE *****
1008          ;*****
1009 013404 005737 013106  STARTR: TST      STOP
1010 013410 001007          BNE      IS
1011 013412 052764 000001 000004  BIS      @DTR, XCSR(R4) ;SET DTR
1012 013420 104400 013014          TYPE     MSGS ;MAKE CONNECTION
1013 013424 005137 013106          COM      STOP
1014 013430 005037 011032  IS:     CLR      TIME
1015 013434 013737 011020 011066  MOV      IRDA, RDA
1016 013442 012737 001000 011064  MOV      @1000, RCC ;SETUP RCV CHAR COUNT
1017 013450 042700 040000          BIC      @RFLG, STAT ;RESET RCV COMPLETE FLAG
1018 013454 005037 011054          CLR      ERCSR ;RESET ERROR RECORDS
1019 013460 005037 011056          CLR      ERDBR
1020 013464 042714 004000          BIC      @4000, @RCSR ;CLEAR ACTIVE IF SET
1021 013470 005764 000002          TST     REUF(R4) ;CLEAR ANY RXDONE
1022 013474 052764 000001 000004  BIS      @DTR, XCSR(4) ;SET DATA TERM READY
1023 013502 052714 000100          BIS      @RIE, @RCSR ;SET INTERRUPT ENABLES
1024 013506 000207          RTS      PC          ;EXIT FROM SUBROUTINE
1025 013510 000000          TEMP1: 0
1026 013512 000000          TEMP2: 0

```

```

1027
1028
1029
1030 013514 000240
1031 013516 127737 175346 011040
1032 013524 001006
1033 013526 052700 100000
1034 013532 042764 001100 000004
1035 013540 000417
1036
1037 013542 117764 175322 000006 XISR1: MOV B @XDA, XBUF(R4); XMIT NEXT CHAR.
1038 013550 032777 000100 175266 BIT #100 @SWR ; MONITOR OUTPUT?
1039 013556 001406 BEQ NOXMON ; BR IF NO
1040 013560 105777 175312 TSTB @TPS ; IS TTY AVAILABLE
1041 013564 100003 BPL NOXMON ; BR IF NO
1042 013566 117777 175276 175304 MOV B @XDA, @TPB ; TYPE THE CHAR
1043 013574 NOXMON:
1044 013574 005237 011070 INC XDA ; INCREMENT ADDRESS
1045 013600 005037 011032 XISR2: CLR TIME ; RESET TIMER
1046 013604 000002 RTI ; RETURN FROM INTERRUPT
1047
1048
1049
1050 013606 000240
1051 013610 116401 000002
1052 013614 110177 175246
1053 013620 032777 000040 175216
1054 013626 001405
1055 013630 105777 175242
1056 013634 100002
1057 013636 110177 175236
1058 013642
1059 013642 005237 011066 NORMON: INC RDA ; BUMP POINTER
1060 013646 105077 175214 CLR B @RDA ; CLEAR NEXT CHAR POSITION
1061 013652 005337 011064 DEC RCC ; DECREMENT CHAR. COUNTER
1062 013656 001005 BNE IS ; BR IF BUFFER NOT FULL
1063 013660 042714 000100 BIC #RIE, @RCSR ; RESET INTERRUPT ENAB
1064 013664 104006 MLT+6 ; RECEIVER BUFFER FULL
1065 013666 004737 013404 JSR PC, STARTR ; INITIALIZE RECEIVER
1066
1067 013672 142701 000200 IS: BIC B #200, R1 ; STRIP PARITY
1068 013676 123701 011041 CMP B RX.TERM, R1 ; IS IT LINE FEED?
1069 013702 001004 BNE RISR1 ; BR IF NO
1070 013704 042714 004100 BIC #RIE+RA, @RCSR ; DISABLE INTERRUPTS
1071 013710 052700 040000 BIS #RFLG, STAT ; SET RCVR COMPLETE FLAG
1072 013714
1073
1074 013714
1075 013714
1076 013714 005037 011032 RISR2: CLR TIME ; RESET TIMER
1077 013720 000002 RISR3: RTI ; RETURN FROM INTERRUPT
1078 000001 .END

```

BA	011004	584#	643					
BACK	013104	707*	735*	770*	850*	881	945#	
BIT0	= 000001	578#						
BIT1	= 000002	578#						
BIT10	= 002000	578#						
BIT11	= 004000	578#						
BIT12	= 010000	578#						
BIT13	= 020000	578#	613#	892	922			
BIT14	= 040000	578#						
BIT15	= 100000	578#						
BIT2	= 000004	578#						
BIT3	= 000010	578#						
BIT4	= 000020	578#						
BIT5	= 000040	578#						
BIT6	= 000100	578#						
BIT7	= 000200	578#						
BIT8	= 000400	578#						
BIT9	= 001000	578#						
B2016	011030	594#	895					
CD	= 004000	578#						
CT	= 100000	578#						
CTS	= 002000	578#						
CTSABC	013272	973	983#					
CTSJK	013322	971	989#					
CTSM	013214	966	970#					
CTSMN	013234	974#						
CTSXYZ	013224	972#	975	982				
DELAY	013102	650*	659*	879*	944#	953	961#	
DERR	012514	923	932#					
DIE	= 000040	578#						
DISPLA	011046	604#						
DP11	011090	583#						
DSFLG	= 020000	578#	612#					
DSSTAT	011060	619#						
DTR	= 000001	578#	964	1011	1022			
ENTER	012244	863	872#					
ECP	012156	708	736	771	851	858#		
ERCSR	011054	617#	1018#					
ERGER	011056	618#	890	1019#				
FLAG	011042	602#						
FULL.D=	000001	631#	810	826	831	989		
GO	011202	658#						
HO	= 000002	578#						
ILB	= 000010	578#	667					
IRDA	011020	590#	774	899	925	1015		
IS	= 000002	578#						
IXDA	011022	591#	775	898	929	931	962	
KBOIN	= 104416	578#	689	720	752	809		
LOOP	= 000020	578#	705	733	768	848		
MIS	= 000020	578#						
MR	= 000010	578#						
MROY	= 010000	578#	970	972				
MSG0	012526	894	938#					
MSG1	012607	897	938#					
MSG2	012612	924	938#					
MSG3	012637	928	938#					

DP11 ITEP OVERLAY MACY11 27(1006) 29-OCT-76 14:41 PAGE 29
DZDPOC.P11 05-AUG-76 09:23 CROSS REFERENCE TABLE -- MACRO NAMES

BOX	10	579	606	632	948	1006	1027	1047		
DCPAM	10									
DHDOC1	10									
DHPAM	10									
DJPAM	10									
DLPAM	10									
DPPAM	10	561								
DQDOC1	10									
DQPAM	10									
DUPAM	10									
DUPPAR	10									
DVDOC1	10									
DVPAM	10									
DZPAM	10									
HELLO	10									
HLT	578	697	729	761	791	823	839	934	980	1064
SEQUAT	10	578								
SINTF	10	578								
SITEP	10	651								
SSERV	10	621								

. ABS. 013722 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDPOC.SEG/SOL/CRF/NL:TOC=ITEP1.MAC,DZDPOC.P11
RUN-TIME: 10 13 .8 SECONDS
RUN-TIME RATIO: 64/25=2.4
CORE USED: 16K (31 PAGES)

