

DU11

OVERLAY FOR ITEP
MD-11-DZDUO-B

EP-DZDUO-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN U.S.A.

The microfiche card contains 18 frames of technical data, arranged in a 6x3 grid. The frames contain various types of information, including:

- Tables with columns and rows of data.
- Diagrams and flowcharts.
- Textual descriptions and labels.
- Small charts and graphs.

The data is presented in a structured, technical format typical of engineering or scientific documentation.

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDU0-B-D

PRODUCT NAME: DU11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: OCTOBER 1976

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
 JOHN EGOLF
 GEORGE BAISLEY

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 DL-11 INTERFACE.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

- A. PDP-11 SYSTEM WITH 4K OF CORE.
- B. A DUII 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 202 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 DN11BB, TYPE IN DNB.

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

- 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 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 DN11B.
9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
- A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.

NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING
NEW VALUES, THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT
RESTARTS OF THE PROGRAM.

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR (INTERFACE SERVICE ROUTINE) SPECIFICATION
SMR14=SETUP DN-11B ISR
SMR13=SETUP DN-11 ISR
SMR=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 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 TYPEOUTS
SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
 IF SW12=0 AND SW04=1 END PASS IS TYPED
 AND TRANSMITTED/RECEIVED DATA IS TYPED.
SW11=1 USE PREVIOUSLY SPECIFIED DATA
SW10=1 DATA SELECT (WITH SW09)
SW09=1 DATA SELECT (WITH SW10)
 00=1 GET DATA FROM OPERATOR
 01=1 TEST MESSAGE #1 (SA QUICK BROWN FOX)
 10=1 TEST MESSAGE #2 (SB NUMERICS)
 11=1 TEST MESSAGE #3 (SC 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. ↑ 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 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 SMD9 AND SM10(SM7=0)
3. TRANSMIT THE DATA SELECTED BY SMD9 AND SM10 (SM8=0) OR
TRANSMIT THE RECEIVED DATA (SM8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SM4=1) OR
GO TO STEP 1. (SM4=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 (SM7=0)
7. RETURNS TO MONITOR FOR "END PASS". (SM4=1) OR
GO TO STEP 1(SM4=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SM7=0)
3. RETURNS TO MONITOR FOR "END PASS"(SM4=1) OR
GO TO STEP 1 (SM4=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". (SM4=1) OR
GO TO STEP 1 (SM4=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=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<OO1> 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 DU11 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: SOWI, IF "ONE WAY IN" MODE WAS SELECTED. SOWO, IF "ONE WAY OUT" MODE WAS SELECTED. SILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED. SXLB, IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 SOWI: 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 SOWO: 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 SOWO OR TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.33 SILB: 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. (SILB)

9.34 SXLB: 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 DU11

PARAM#1 (LOW BYTE) IS LOADED INTO THE TRANSMIT STATUS REGISTER (TXCSR).
BIT 3 HALF DUPLEX (1), DEFAULT= HALF DUPLEX (1)

PARAM#1 (HIGH BYTE) IS LOADED INTO THE RECEIVER STATUS REGISTER (RXCSR).
BIT 8 STRIP SYNC (1), DEFAULT= STRIP SYNC (1)

PARAM#2 IS LOADED INTO THE PARAMETER STATUS REGISTER (PARCSR).
BITS 0-7 SYNC CHARACTER, DEFAULT= 26 (26)
BIT 8 PARITY SENSE SELECT (1), DEFAULT= (0)
BIT 9 PARITY ENABLE (1), DEFAULT= (0)
BITS 10,11 WORD LENGTH SELECT, DEFAULT= 8 BITS (11)
BITS 12,13 MODE SELECT, DEFAULT= INTERNAL SYNCHRONOUS (11)

PARAM#3 IS NOT USED (177777).

```

580
581
582
583
584
585 011000 011000 000040
586 011004 160040
587 011006 000300
588 011010 000240
589 011012 000410
590 011014 036026
591 011016 177777
592 011020 000000
593 011022 000000
594 011024 000000
595 011026 030000
596 011030 000000
597 011032 000000
598 011034 000000
599 011036 011102
600 011040
601 011040 000
602 011041
603 011041 001
604 011042 000000
605 011044 177570
606 011046 177570
607
608
609
610
611 000000
612 100000
613 040000
614 020000
615 020000
616
617 011050 000000
618 011052 000000
619 011054 000000
620 011056 000000
621 011060 000000
622
623 011062 000000
624 011064 000000
625 011066 000000
626 011070 000000
627
628 011072 177560
629 011074 177562
630 011076 177564
631 011100 177566
632
633 000001

```

```

*****
DU11 INTERFACE SERVICE PARAMS
*****

```

```

DU11: .=11000
BA: .ASCIZ /DU /
RIV: 300
PRIOR: 240
PARAM1: STPSYN!HDXEN
PARAM2: SYNINT!EIGHT!NOPAR!26
PARAM3: 177777
IRDA: .WORD 0
IXDA: .WORD 0
SETTLE: .WORD 0
B2016: .WORD 0
TIME: .WORD 0
TX.TERM: .WORD START
RX.TERM: .BYTE 000
FLAG: .WORD 001
SMR: 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
SXCSR: 0
SRCSR: 0
ERCSR: 0
ERDAR: 0
DSSTAT: 0
XCC: 0
RCC: 0
RDA: 0
XDA: 0
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
FULL.DUPLEX=000001

```

```

;XMIT COMPLETE FLAG
;RCV COMPLETE FLAG
;DATA SET STATUS CHANGE FLAG
;INHIBIT PRINTOUTS
;SAVED XMIT CSR
;SAVED RCV CSR
;RCV CSR SAVED ON ERROR
;RCV DATA REG SAVED ON ERROR
;RCV CSR SAVED ON DS CHANGE
;XMIT CHAR COUNT
;RCV CHAR COUNT
;RCV DATA ADDR.
;XMIT DATA ADDR.

```

```

634
635
636
637 011102 000240
638 011104 017700 177734
639 011110 042700 177400
640 011114 013702 011006
641 011120 012722 013572
642 011124 013722 011010
643 011130 012722 013474
644 011134 013722 011010
645 011140 013704 011004
646 011144 013714 011012
647 011150 013702 011014
648 011154 042702 000001
649 011160 010264 000004
650 011164 052764 000400 000004
651 011172 113764 011012 000004
652 011200 113764 011013 000001
653 011206 013764 011014 000002
654 011214 004537 013770
655 011220 011014

```

```

*****
DUI1-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     @MRESET, @TXCSR(R4); MASTER RESET
        MOV     @PARAM1, @TXCSR(R4); SET UP VARIABLES
        MOV     @PARAM1+1, 1(R4); LOAD UPPER BYTE RXCSR
        MOV     @PARAM2, @PARCSR(R4); LOAD PARCSR
        JSR     RS,STRIP     ;GO CALC PARITY
        PARAM2

```

```

656
657
658
659
660
661
662
663 011222 005037 011032
664 011226 005037 013122
665 011232 005037 013126
666 011236 032700 000001
667 011242 001402
668 011244 000137 011420
669 011250 032700 000002
670 011254 001402
671 011256 000137 011312
672 011262 032700 000010
673 011266 001402
674 011270 000137 011516
675 011274 032700 000004
676 011300 001402
677 011302 000137 011746
678 011306 000000
679 011310 000776

```

```

*****
ROUTINE USED TO GOTO
SUBROUTINE DEPENDENT
ON MODE SELECTED.
*****
GO:    CLR     TIME
        CLR     DELAY
        CLR     STOP
        BIT     @0W0, MODE
        BEQ    1S
        JMP    @S0W0
1S:    BIT     @0W1, MODE
        BEQ    2S
        JMP    @S0W1
2S:    BIT     @1LB, MODE
        BEQ    3S
        JMP    @1LB
3S:    BIT     @XLB, MODE
        BEQ    4S
        JMP    @XLB
4S:    HALT
        BR     .-2

```

```

680
681
682
683
684
685
686
687
688
689

```

```

*****
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".
*****

```

690
691
692
693
694
695
696
697
698
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

011312 104416
011314 004737 013400
011320 032700 040000
011324 001013
011326 023727 011032 000100
011334 103771
011336 011402
011340 016403 000004
011344 104001
011346 005037 011032
011352 000762

011354 032777 000200 177462
011362 001002
011364 004737 012336
011370 042700 040000
011374 032777 000020 177442
011402 001405
011404 012737 011416 013124
011412 000137 012176
011416 000735

011420 104416
011422 004737 013130
011426 005037 011032
011432 032700 100000
011436 001013
011440 023727 011032 000100
011446 103771
011450 011402
011452 016403 000004
011456 104001
011460 005037 011032
011464 000762
011466 042700 100000
011472 032777 000020 177344
011500 001405
011502 012737 011514 013124
011510 000137 012176
011514 000741

SOWI: KBDIN
JSR PC,STARTR
1S: BIT @RFLG,STAT
BNE 2S
CMP TIME,@100
BLO 1S
MOV @RCSR,R2
MOV XCSR(R4),R3
HLT 1
CLR TIME
BR 1S

2S: BIT @NODAT,@SMR
BNE 3S
JSR PC,TESTD
3S: BIC @RFLG,STAT
BIT @LOOP,@SMR
BEQ 4S
MOV @4S,BACK
JMP EOP
4S: 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."

SOWO: KBDIN
JSR PC,STARTX
CLR TIME
1S: BIT @XFLG,STAT
BNE 2S
CMP TIME,@100
BLO 1S
MOV @RCSR,R2
MOV XCSR(R4),R3
HLT 1
CLR TIME
BR 1S
2S: BIC @XFLG,STAT
BIT @LOOP,@SMR
BEQ 3S
MOV @3S,BACK
JMP EOP
3S: BR SOWO


```

746
747
748
749
750
751
752
753
754
755
756
757
758 011516 104416
759 011520 004737 013400
760 011524 005037 011032
761 011530 032700 040000
762 011534 001013
763 011536 023727 011032 000100
764 011544 103771
765 011546 011402
766 011550 016403 000004
767 011554 104001
768 011556 005037 011032
769 011562 000762
770 011564 032777 000200 177252 2S:
771 011572 001002
772 011574 004737 012336
773 011600 042700 040000 3S:
774 011604 032777 000020 177232
775 011612 001405
776 011614 012737 011626 013124
777 011622 000137 012176
778 011626 032777 000400 177210 4S:
779 011634 001416
780 011636 013702 011020
781 011642 013703 011022
782 011646 010337 011070
783 011652 112223
784 011654 001376
785 011656 112743 000177
786 011662 005203
787 011664 112723 000177
788 011670 105023
789 011672 005037 011032 7S:
790 011676 004737 013130
791 011702 032700 100000 5S:
792 011706 001013
793 011710 023727 011032 000100
794 011716 103771
795 011720 011402
796 011722 016403 000004
797 011726 104001
798 011730 005037 011032
799 011734 000762
800 011736 042700 100000 6S:
    011742 000137 011516
  
```

```

*****
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:  KBDIN
        JSR  PC,STARTR
        CLR  TIME
1S:     BIT  #AFLG,STAT
        BNE  2S
        CMP  TIME,#100
        BLO  1S
        MOV  #ACSR,R2
        MOV  XCSR(A4),R3
        HLT  1
        CLR  TIME
        BR   1S
2S:     BIT  #NODAT,#SMR
        BNE  3S
        JSR  PC,TESTD
        BIC  #AFLG,STAT
        BIT  #LOOP,#SMR
        BEQ  4S
        MOV  #4S,BACK
        JMP  EOP
4S:     BIT  #400,#SMR
        BEQ  7S
        MOV  IRDA,R2
        MOV  IXDA,R3
        MOV  R3,XDA
        MOVB (R2)+,(R3)+
        BNE  -2
        MOVB #177,-(R3)
        INC  R3
        MOVB #177,(R3)+
        CLRB (R3)+
7S:     CLR  TIME
        JSR  PC,STARTX
5S:     BIT  #XFLG,STAT
        BNE  6S
        CMP  TIME,#100
        BLO  5S
        MOV  #ACSR,R2
        MOV  XCSR(A4),R3
        HLT  1
        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
  
```

```

801
802
803
804
805
806
807
808
809
810
811
812
813
814 011746 104416
815 011750 032737 000010 011012
816 011756 001002
817 011760 004737 013400
818 011764 004737 013130
819 011770 005037 011032
820 011774 032700 100000
821 012000 001016
822 012002 032700 040000
823 012006 001024
824 012010 023727 011032 000100
825 012016 103766
826 012020 011402
827 012122 016403 000004
828 012026 104001
829 012030 005037 011032
830 012034 000757
831 012036 032737 000010 011012
832 012044 001756
833 012046 042700 100000
834 012052 004737 013400
835 012056 000746
836 012060 032737 000010 011012
837 012066 001020
838 012070 032700 100000
839 012074 001013
840 012076 023727 011032 000100
841 012104 103765
842 012106 011402
843 012110 016403 000004
844 012114 104001
845 012116 005037 011032
846 012122 000756
847 012124 042700 100000
848 012130 042700 040000
849 012134 005037 011032
850 012140 032777 000200 176676
851 012146 001002
852 012150 004737 012336
853 012154 032777 000020 176662
854 012162 001671
855 012164 012737 011746 013124
856 012172 000137 012176

```

```

*****
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.
*****

```

```

SXLB:  KBDIN
      BIT  #HALF.DUPLEX,PARAM1
      BNE  15
      JSR  PC,STARTR
15:    JSR  PC,STARTX
      CLR  TIME
25:    BIT  #XFLG,STAT
      BNE  35
      BIT  #AFLG,STAT
      BNE  45
      CMP  TIME,#100
      BLO  25
      MOV  #ACSR,R2
      MOV  XCSR(R4),R3
      HLT  1
      CLR  TIME
      BR   25
35:    BIT  #HALF.DUPLEX,PARAM1
      BEQ  75
      BIC  #XFLG,STAT
      JSR  PC,STARTR
      BR   25
45:    BIT  #HALF.DUPLEX,PARAM1
      BNE  85
      BIT  #XFLG,STAT
      BNE  65
      CMP  TIME,#100
      BLO  45
      MOV  #ACSR,R2
      MOV  XCSR(R4),R3
      HLT  1
      CLR  TIME
      BR   45
65:    BIC  #XFLG,STAT
85:    BIC  #AFLG,STAT
      CLR  TIME
      BIT  #NODAT,#SWR
      BNE  55
      JSR  PC,TESTD
55:    BIT  #LOOP,#SWR
      BEQ  SXLB
      MOV  #SXLB,BACK
      JMP  EOP

```

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
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912

012176
012176 104414 000340
012202 016437 000004 012334
012210 042737 177677 012334
012216 042764 000100 000004
012224 012766 012264 000002
012232 010037 013106
012236 010137 013110
012242 010237 013112
012246 010337 013114
012252 010437 013116
012256 010537 013120
012262 000207

012264
012264 013700 013106
012270 013701 013110
012274 013702 013112
012300 013703 013114
012304 013704 013116
012310 013705 013120
012314 012737 177777 013122
012322 053764 012334 000004
012330 000177 000570
012334 000000

013746 011056
001413
032777 020000 176472
001007
104400 012546
004077 176444
005746
104400 012627
013701 011022
013702 011020
122122
001776
123741 011040
001453
122742 000002
001005
010237 012430
104400

ROUTINE TO RETURN
TO MONITOR FOR
END PASS.

EOP: STPS, PRTY7 ; SET PS PRIORITY TO 7
MOV XCSR(R4), @TPIE ; SAVE TX CSR
BIC @T<TIE>, @TPIE ; 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 @TPIE, XCSR(R4) ; IF ORIGINALLY SET; SET TX IE
JMP @BACK
@TPIE: 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 R0, @B2016 ; 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: CHPB (R1)+, (R2)+ ; DATA OK ?
BEQ SCAN4 ; BR IF OK
CHPB TX_TERM, -(R1) ; IS IT END OF DATA
BEQ TESTDX ; BR IF YES
CHPB @002, -(R2)
BNE 25
MOV R2, 15
TYPE

913	012430	000000			15:	WORD	0		
914	012432	000443				BR	TESTDX		
915	012434				25:				
916	012434	105712				TSTB	(R2)		
917	012436	001441				BEQ	TESTDX		::BR IF YES
918	012440	122721	000177			CMPB	#177, (R1)+		::IS IT FILL CHAR?
919	012444	001756				BEQ	SCAN4		::BR IF YES
920	012446	005301				DEC	R1		::BACKUP
921	012450	122722	000177			CMPB	#177, (R2)+		::IS IT FILL?
922	012454	001756				BEQ	SCAN4		::BR IF YES
923	012456	105744				TSTB	-(R2)		::BACK UP POINTER
924	012460	123722	011014			CMPB	PARAM2, (R2)+		
925	012464	001746				BEQ	SCAN4		::BR IF CHAR WAS SYNC
926	012466	000240			SCANS:	NOP			::DATA ERROR
927	012470	032777	020000	176346		BIT	#BIT13, 2SWR		::INHIBIT PRINTOUTS
928	012476	001016				BNE	DERR		::BR IF YES
929	012500	104400	012632			TYPE	MSG2		::<15><12>RECEIVED DATA = <15><12>
930	012504	013737	011020	012514		MOV	IRDA, RDX		::SETUP DATA ADDRESS
931	012512	104400				TYPE			::PRINT RECEIVED DATA
932	012514	000000			RDX:	0			::RECEIVED DATA ADDR.
933	012516	104400	012657			TYPE	MSG3		::<15><12>DATA SHOULD BE<15><12>
934	012522	013737	011022	012532		MOV	IXDA, .+10		::SETUP ADDR.
935	012530	104400				TYPE			::PRINT GOOD DATA
936	012532	011022				IXDA			
937	012534	111103			DERR:	MOVB	(R1), R3 ;SETUP XMIT DATA		
938	012536	114202				MOVB	-(R2), R2		::SETUP RCV DATA
939	012540	104007				HLT+7			::DATA ERROR HALT
940	012542	005726			TESTDX:	TST	(SP)+		::POP STACK
941	012544	000207				RTS	PC		::RETURN FROM SUB/ROUT
942									
943	012546	005015	044124	051105	MSG0:	.ASCIZ	<15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/		
(1)	012627	015	000012		MSG1:	.ASCIZ	<15><12>		
(1)	012632	005015	042522	042503	MSG2:	.ASCIZ	<15><12>/RECEIVED DATA = /<15><12>		
(1)	012657	015	042012	052101	MSG3:	.ASCIZ	<15><12>/DATA SHOULD BE/<15><12>		
(1)	012702	005015	046120	040505	MSG4:	.ASCII	<15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./		
(1)	012751	015	053412	042510		.ASCIZ	<15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>		
(1)	013034	005015	046120	040505	MSG5:	.ASCIZ	<15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>		
(1)						.EVEN			
(1)	013106	000000			SAVR0:	0			
944	013110	000000			SAVR1:	0			
945	013112	000000			SAVR2:	0			
946	013114	000000			SAVR3:	0			
947	013116	000000			SAVR4:	0			
948	013120	000000			SAVR5:	0			
949	013122	000000			DELAY:	0			
950	013124	000000			BACK:	0			
951	013126	000000			STOP:	0			
952									
953									
954									
955									
956									
957									
958	013130	005037	011032		STARTX:	CLR	TIME		
959	013134	005737	013122			TST	DELAY		::IF SW04=1 & SW14=0 WAIT BEFORE TURNING ON TX
960	013140	001416				BEQ	15		::NO GO AHEAD AND TURN ON TX
961	013142	005037	013470			CLR	TEMP1		::PREPARE FOR DELAY

```

;*****
;INITIALIZE TRANSMIT SUBROUTINE
;*****

```

```

961 013146 012737 000007 013472      MOV      #7,TEMP2
962 013154 062737 000001 013470      ADD      #1,TEMP1      ;INCREMENT DELAY....
963 013162 001374          BNE          -6
964 013164 005337 013472      DEC      TEMP2
965 013170 001371          BNE          -14
966 013172 005037 013122      CLR      DELAY      ;ZERO DELAY
967 013176 013737 011022 011070 1S:  MOV      IXDA,XDA      ;SETUP XMIT DATA ADDR.
968 013204 042700 100000      BIC      #XFLG,R0     ;RESET XMIT COMPLETE FLAG
969 013210 052714 000002      BIS      #DTR,#RXCSR  ;SET DTR
970 013214 005737 013126      TST     STOP
971 013220 001005          BNE          2S
972 013222 104400 012702      TYPE    ,MSG4
973 013226 000000          HALT
974 013230 005137 013126      COM
975 013234 032737 000010 011012 2S:  BIT      #HALF.DUPLEX,PARAM1
976 013242 001403          BEQ      3S
977 013244 032714 010000      4S:  BIT      #10000,#RXCSR  ;IS CARRIER UP
978 013250 001375          BNE          4S      ;BR IF YES
979 013252 052714 000004      3S:  BIS      #RTS,#RXCSR  ;SET RTS
980 013256 032714 020000      CTSW:  BIT      #CTS,#RXCSR; IS CLEAR TO SEND SET?
981 013262 001017          BNE          CTSOK   ;BR IF YES
982 013264 023727 011032 000036 CTSW:  CNP      TIME,#36  ;30 SECS ELAPSED?
983 013272 103771          BLO      CTSW      ;BR IF NO
984 013274 011402          MOV      #RXCSR,R2   ;SETUP RECEIVE CSR
985 013276 016403 000004          MOV      TXCSR(R4),R3 ;SETUP XMIT CSR
986 013302 032777 010000 175534      BIT      #SM12,#SMR  ;INHIBIT PRINTOUTS?
987 013310 001001          BNE          1S      ;BR IF YES
988 013312 104002          HLT+2          ;PRINTOUT "WAITING TO XMIT" MESSAGE
989 013314 005037 011032      1S:  CLR      TIME      ;RESET TIMER
990 013320 000756          BR       CTSW      ;WAIT SOME MORE
991 013322 005037 011032      CTSOK: CLR      TIME
992 013326 012737 000005 013376      MOV      #5,SYNCR0   ;SETUP SYNC. COUNTER
993 013334 052764 000020 000004      BIS      #SEND,TXCSR(R4); SET SEND
994 013342 113764 011014 000006 1S:  MOVB   PARAM2,TXDBUF(R4);LOAD A SYNC CHAR
995
996 013350 105764 000004          TSTB   TXCSR(R4)    ;IS XMIT READY(TXDONE=1?)
997 013354 100375          BPL          -4      ;BR IF NO
998 013356 005337 013376      DEC     SYNCR0      ;DECREMENT COUNTER
999 013362 001367          BNE          1S      ;BR IF NOT ZERO
1000
1001 013364 052764 000100 000004      BIS      #TXINTE,TXCSR(R4);SET XMIT INTERRUPT ENABLE
1002 013372 000240          NOP
1003 013374 000207          RTS      PC      ;EXIT FROM SUBROUTINE
1004 013376 000000          SYNCNO: 0
1005
1006 ;*****
1007 ;***** INITIALIZE RECEIVER SUBROUTINE *****
1008 ;*****
1009 STARTR:
1010 013400 005737 013126      TST     STOP      ;FIRST TIME HERE?
1011 013404 001006          BNE          1S      ;BR IF NO
1012 013406 052714 000002      BIS      #DTR,#RXCSR ;SET DTR
1013 013412 104400 013034      TYPE    ,MSG5      ;MAKE CONNECTION
1014 013416 005137 013126      COM
1015 013422 005037 011032 1S:  CLR      TIME
1016 013426 013737 011020 011066      MOV      IRDA,RDA   ;SETUP RCV DATA ADDR

```

1017	013434	012737	001000	011064	MOV	#1000,RCC	:SETUP RCV CHAR COUNT
1018	013442	042700	040000		BIC	#RFLG,R0	:RESET RCV COMPLETE FLAG
1019	013446	005037	011054		CLR	ERCSR	:RESET ERROR RECORDS
1020	013452	005037	011056		CLR	ERDSR	
1021	013456	005764	000002		TST	RXDBUF(R4)	:CLR ANY RXDONE
1022	013462	052714	000122		BIS	#RINTEN!SYNSCH!DTR,3RXCSR	:SET INTER ENABLE & SEARCH SYNC &DTR
1023	013466	000207			RTS	PC	:EXIT FROM SUBROUTINE
1024	013470	000000					
1025	013472	000000					

TEMP1: 0
TEMP2: 0

```

1026 ;*****
1027 ; TRANSMIT INTERRUPT SERVICE ROUTINE
1028 ;*****
1029 013474 000240 XISR: NOP
1030 013476 127737 175366 011040 CMPB 2XDA, TX.TERM ;FINISHED XMITTING?
1031 013504 001010 BNE XISR1 ;BR IF NO
1032 013506 052700 100000 BIS 2XFLG, R0 ;SET XMIT COMPLETE FLAG
1033 013512 042764 000120 000004 BIC 2TXINTE!SEND, TXCSR(R4) ;RESET XMIT INTR ENABLE
1034 013520 042714 000004 BIC 2RTS, 2RXCSR ;RESET RTS
1035 013524 000417 BR XISR2 ;
1036
1037 013526 117764 175336 000006 XISR1: MOVB 2XDA, TXDBUF(R4); XMIT NEXT CHAR.
1038 013534 032777 000100 175302 BIT 2I00, 2SWR ;MONITOR OUTPUT?
1039 013542 001406 BEQ NOXMON ;BR IF NO
1040 013544 105777 175326 TSTB 2TPS ;IS TTY AVAILABLE
1041 013550 100003 BPL NOXMON ;BR IF NO
1042 013552 117777 175312 175320 MOVB 2XDA, 2TPB; TYPE THE CHAR
1043 013560 NOXMON:
1044 013560 005237 011070 INC XDA ;INCREMENT ADDRESS
1045 013564 005037 011032 XISR2: CLR TIME ;RESET TIMER
1046 013570 000002 RTI ;RETURN FROM INTERRUPT
1047 ;*****
1048 ; RECEIVE INTERRUPT SERVICE ROUTINE
1049 ;*****
1050 013572 000240 RISR: NOP
1051 013574 011402 MOV 2RXCSR, R2 ;SAVE RXCSR CONTENTS
1052 013576 016403 000004 MOV TXCSR(R4), R3 ;SAVE TXCSR CONTENTS
1053 013602 016437 000002 013470 MOV RXDBUF(R4), TEMP1 ;STORE CHAR AND ERROR FLAGS IF ANY
1054 013610 113737 013470 013472 MOVB TEMP1, TEMP2
1055 013616 143737 014122 013472 BICB PARITY, TEMP2 ;STRIP PARITY
1056 013624 113777 013470 175234 MOVB TEMP1, 2RDA ;MOVE CHAR TO INBUF
1057 013632 032777 000040 175204 BIT 2I00, 2SWR ;MONITOR INPUT?
1058 013640 001406 BEQ NORMON ;BR IF NO
1059 013642 105777 175230 TSTB 2TPS ;IS TTY AVAILABLE?
1060 013646 100003 BPL NORMON ;BR IF NO
1061 013650 113777 013472 175222 MOVB TEMP2, 2TPB; TYPE THE CHAR
1062 013656 NORMON:
1063 013656 005237 011066 INC RDA ;BUMP POINTER
1064 013662 105077 175200 CLRB 2RDA ;CLEAR NEXT CHAR POSITION
1065 013666 005337 011064 DEC RCC ;DECREMENT CHAR. COUNTER
1066 013672 001005 BNE IS ;BR IF BUFFER NOT FULL
1067 013674 042714 000120 BIC 2RINTEN!SYNSCH, 2RXCSR ;RESET INTERRUPT ENAB
1068 013700 104006 HLT+6 ;RECEIVER BUFFER FULL
1069 013702 004737 013400 JSR PC, STARTR ;INITIALIZE RECEIVER
1070 013706 123737 011041 013472 IS: CMPB RX.TERM, TEMP2 ;IS IT LINE FEED?
1071 013714 001004 BNE RISR1 ;BR IF NO
1072 013716 042714 000120 BIC 2RINTEN!SYNSCH, 2RXCSR ;DISABLE INTERRUPT AND SEARCH SYNC
1073 013722 052700 040000 BIS 2RFLG, R0 ;SET RCVR COMPLETE FLAG
1074 013726 005737 013470 RISR1: TST TEMP1 ;IS THERE A DATA ERROR
1075 013732 100005 BPL RISR2 ;BR IF NO
1076 013734 010237 011054 MOV R2, ERCSR ;SAVE RXCSR
1077 013740 013737 013470 011056 MOV TEMP1, ERDBR ;SAVE RXDBUF
1078 013746 005702 RISR2: TST R2 ;IS THERE A DATA SET STATUS CHANGE
1079 013750 100004 BPL RISR3 ;BR IF NO
1080 013752 010237 011060 MOV R2, DSSTAT ;SAVE STATUS
1081 013756 052700 020000 BIS 2DSFLG, R0 ;SET FLAG

```

```

1082 013762
1083 013762 005037 011032
1084 013766 000002
1085
1086
1087
1088 013770 013537 013470
1089 013774 042737 170777 013470
1090 014002 022737 007000 013470
1091 014010 001005
1092 014012 012737 000400 014122
1093 014020 000137 014120
1094 014024 022737 005000 013470
1095 014032 001005
1096 014034 012737 000200 014122
1097 014042 000137 014120
1098 014046 022737 003000 013470
1099 014054 001005
1100 014056 012737 000100 014122
1101 014064 000137 014120
1102 014070 022737 001000 013470
1103 014076 001005
1104 014100 012737 000040 014122
1105 014106 000137 014120
1106 014112 012737 000200 014122
1107 014120 000205
1108 014122 000200
1109 000001

```

```

RISR3: CLR TIME ;RESET TIMER
RTI ;RETURN FROM INTERRUPT
*****
STRIP PARITY SUBROUTINE
*****
STRIP: MOV @ (R5)+,TEMP1 ;PICK UP CONTENTS
BIC @C<7000>,TEMP1 ;STRIP OUT JUNK
CMP #EIGHT,PAREN,TEMP1 ;IS IT EIGHT BITS/CHAR ?
BNE .+14 ;NO
MOV #BIT8,PARITY ;YES
JMP OUT
CMP #SEVEN,PAREN,TEMP1 ;IS IT 7 BITS/CHAR ?
BNE .+14 ;NO
MOV #BIT7,PARITY ;YES
JMP OUT
CMP #SIX,PAREN,TEMP1 ;IS IT 6 BITS/CHAR ?
BNE .+14 ;NO
MOV #BIT6,PARITY ;YES
JMP OUT
CMP #FIVE,PAREN,TEMP1 ;IS IT 5 BITS/CHAR ?
BNE .+14 ;NO
MOV #BIT5,PARITY ;YES
JMP OUT
MOV #BIT7,PARITY ;DEFAULT VALUE
OUT: RTS
PARITY:BIT7 ;DEFAULT VALUE
.END

```


BA	011004	586#	645				
BACK	013124	712*	740*	775*	855*	886	950#
BITW	= 002000	580#					
BIT0	= 000001	580#					
BIT1	= 000002	580#					
BIT10	= 002000	580#					
BIT11	= 004000	580#					
BIT12	= 010000	580#					
BIT13	= 020000	580#	615#	897	927		
BIT14	= 040000	580#					
BIT15	= 100000	580#					
BIT2	= 000004	580#					
BIT3	= 000010	580#					
BIT4	= 000020	580#					
BIT5	= 000040	580#	1104				
BIT6	= 000100	580#	1100				
BIT7	= 000200	580#	1096	1106	1108		
BIT8	= 000400	580#	1092				
BIT9	= 001000	580#					
BREAK	= 000001	580#					
B2016	011030	596#	900				
CARDET	= 010000	580#					
CLK	= 020000	580#					
CTS	= 020000	580#	980				
CTSOK	013322	981	991#				
CTSM	013256	980#	983	990			
CTSMN	013264	982#					
DELAY	013122	664#	884#	949#	958	966#	
DERR	012534	929	937#				
DISPLA	011046	606#					
DNA	= 100000	580#					
DNAINT	= 000040	580#					
DSC	= 100000	580#					
DSFLG	= 020000	580#	614#	1081			
DSINTE	= 000040	580#					
DSR	= 001000	580#					
DSSTAT	011060	621#	1080#				
DTR	= 000002	580#	969	1012	1022		
DUI1	011000	585#					
EIGHT	= 006000	580#	590	1090			
ENTER	012264	868	877#				
EOP	012176	713	741	776	856	863#	
ERCSR	011054	619#	1019#	1076#			
ERDBR	011056	620#	895	1020#	1077#		
EVEPAR	= 001400	580#					
EVPAR	= 000400	580#					
FIVE	= 000000	580#	1102				
FLAG	011042	604#					
FRMERR	= 020000	580#					
FULL.D	= 000001	633#					
GO	011222	663#					
HALF.D	= 000010	580#	815	831	836	975	
HDXEN	= 000010	580#	589				
ILB	= 000010	580#	672				
IRDA	011020	592#	779	904	930	1016	
ISYN00	= 000000	580#					

DUII ITEP OVERLAY MACY11 27(1006) 29-OCT-76 14:37 PAGE 29
OZDU08.P11 05-AUG-76 09:24 CROSS REFERENCE TABLE -- USER SYMBOLS

XFLG = 100000	612#	728	737	790	799	820	833	838	847	968	1032
XISR 013474	643	1029#									
XISR1 013526	1031	1037#									
XISR2 013564	1035	1045#									
XLB = 000004	580#	675									
XMAIT = 104412	580#										
SILB 011516	674	757#	800								
SOWI 011312	671	694#	714								
SOWO 011420	668	725#	742								
SXLB 011746	677	814#	854	855							
. = 014124	584#	679	783	934#	963	965	997	1091	1095	1099	1103

BOX	18	581	608	634	953	1006	1026	1047	1085
DCPARM	18								
DHDOC1	18								
DHPARM	18								
DJPARM	18								
DLPARM	18								
DPPARM	18								
DQDOC1	18								
DQPARM	18								
DUPARM	18	560							
DUPPAR	18								
DVDOC1	18								
DVPARM	18								
DZPARM	18								
HELLO	18								
HLT	5808	702	734	766	796	828	844	939	988
SEQUAT	18	580							
SINTF	18	580							
SITEP	18	656							
SSERV	18	623							

. ABS. 014124 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDUOB.SEG/SOL/CRF/NL:TOC=ITEP1.MAC,DZDUOB.P11
RUN-TIME: 10 13 .9 SECONDS
RUN-TIME RATIO: 44/26=1.6
CORE USED: 16K (31 PAGES)

