

DC11

OVERLAY FOR ITEP
MD-11-DZDCO-B

EP-DZDCO-B-DL-A

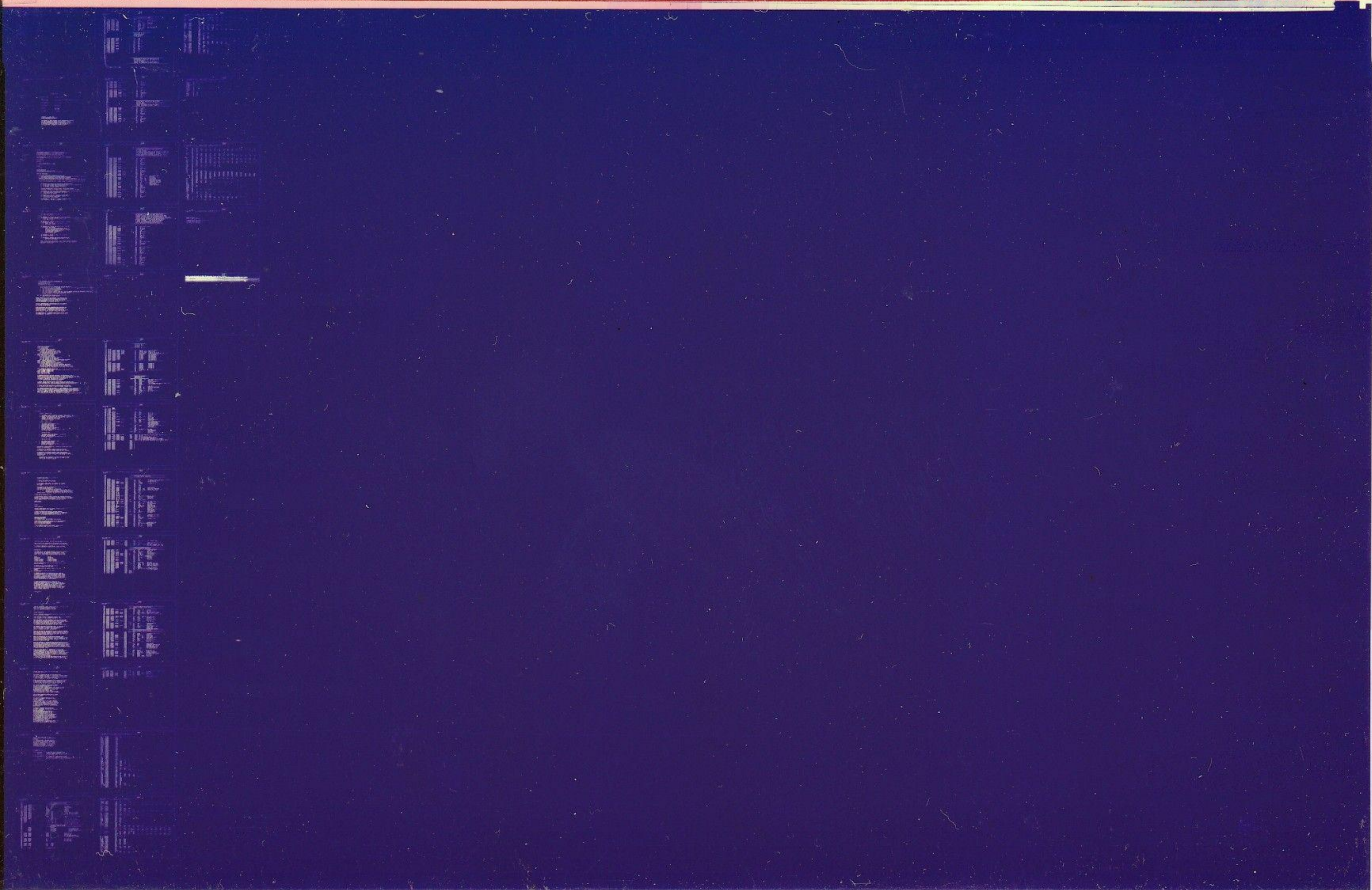
NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN U.S.A



B01

DC11 ITER OVERLAY
ITER V4.P11

MACY11 27(732) 23-SEP-76 08:44 PAGE 36

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDCO-B-D
PRODUCT NAME: DC11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM
DATE CREATED: JULY 21, 1974
MAINTAINER: DIAGNOSTICS
AUTHORS: R A JONES
JOHN EGOLF
REVISED BY: FAY BASHAW 1/21/75

COPYRIGHT (C) JUNE 1973
DIGITAL EQUIPMENT CORPORATION
MAYNARD MASSACHUSETTS 10754

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATIONAL PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY IT. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS WHICH MAY APPEAR IN THIS DOCUMENT.

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 DC11 COMMUNICATION INTERFACE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE

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

4.0 OPERATING PROCEDURES.

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

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

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

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

- 2. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
 - A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
 - B. TYPEIN ACTUAL BUS ADDRESS
- 3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
 - A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS
 - B. TYPEIN ACTUAL VECTOR ADDRESS

- 4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.

- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
 - B. 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 DM11BB.
9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
- A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.

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

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR (INTERFACE SERVICE ROUTINE) SPECIFICATION
SWR14=SETUP DM-11B ISR
SWR13=SETUP DN-11 ISR
SWR=000000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11, DM11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWITCHES THEN HIT CONTINUE)
 - A. HALT FOR BUS ADDRESS OF INTERFACE
 - B. HALT FOR VECTOR ADDRESS OF INTERFACE
 - C. HALT FOR PRIORITY OF INTERFACE
 - D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - E. HALT FOR INTERFACE PARAM #2 (DN11 AND DM11-BB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MON)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DM11.
 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 (\$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

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE. TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN ↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377) SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).

I.E. ABCD↑ 000 123 377↑ EFG (CAR.RETURN)

A TYPICAL SWITCH SETTING FOR HALF-DUPLEX=003150 THIS SETTING USES INTERNAL LOOPBACK MODE, LOOPS IN OVERLAY, MONITORS TRANSMITTED AND RECEIVED DATA ON THE CONSOLE TTY, AND TESTS RECEIVED DATA USING TEST MESSAGE #3.

A TYPICAL SWITCH SETTING FOR FULL-DUPLEX=003144 THIS SETTING IS THE SAME AS ABOVE EXCEPT IT USES THE EXTERNAL LOOPBACK MODE.

ALL STANDARD MESSAGES (TEST MESSAGES 1-3) ARE PRECEDED BY 2 FILL CHARACTERS(177), AND ARE FOLLOWED BY A CR(015), LF(012), RECEIVE TERMINATING CHARACTER(001), 4 FILLS(177), AND A TRANSMIT TERMINATING CHARACTER(000). DURING TRANSMISSION, WHEN A 000 CHARACTER IS SEEN THE TRANSMISSION IS STOPPED. DURING RECEPTION, WHEN A 001 CHARACTER IS RECEIVED, THE RECEIVER IS SHUT OFF. IF THE MESSAGE WAS INPUTED BY THE OPERATER, THE TERMINATING CHARACTERS ARE ADDED.

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10(SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR
TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR
GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
GO TO STEP 1(SW04=0)

ONE-WAY-IN MODE

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

ONE-WAY-OUT MODE

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

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

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

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

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

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

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

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

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING CHARACTERS ON THE CONSOLE TTY.

LINE FEED = RESTART PROGRAM AT LOCATION 200.

QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER. (ASCII)

SET SWITCH 15 AND PRESS CONTINUE FOR NEXT 8 WORDS.

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 10020 AND EXAMINE. TO FIND THE STARTING ADDRESS OF THE TRANSMIT BUFFER, LOAD ADDRESS 10022 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 TTTTTT
DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG

WHERE RRRRR 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 DC11 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 DC11

PARAM#1 IS LOADED INTO THE RECEIVER STATUS REGISTER (RCSR).
BITS 3,4 CHARACTER LENGTH, DEFAULT= 8 BITS (00)
BITS 9,10 RECEIVER SPEED SELECT, DEFAULT= LOWEST SPEED (00)

PARAM#2 IS LOADED INTO THE TRANSMITTER STATUS REGISTER (TSCR).
BIT 0 FULL DUPLEX (1), DEFAULT= HALF DUPLEX (0)
(BIT 0 IS NOT LOADED INTO THE TSCR, IT IS ONLY A SOFTWARE FLAG.)
BITS 3,4 TRANSMIT SPEED SELECT, DEFAULT= LOWEST SPEED (00)

PARAM#3 IS NOT USED(177777).

540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590

010000 041504 000040
010004 174000
010006 000300
010010 000240
010012 000001
010014 000000
010016 177777
010020 000000
010022 000000
010024 000000
010026 000000
010030 000000
010032 000000
010034 000000
010036 010074
010040
010040 000
010041
010041 001

000000
100000
040000
020000
020000

000000
000000
000000
000000
000000

010054 000000
010056 000000
010060 000000
010062 000000

010064 177560
010066 177562
010070 177564
010072 177566

000001
177776

```
*****
: DC11 INTERFACE SERVICE PARAMS
*****
:
:   =10000
DC11:  .ASCIZ  /DC /           ;ISR NAME
BA:    174000                 ;BUS ADDRESS
RIV:   300                    ;VECTOR ADDRESS
PRIOR: 240                    ;PRIORITY
PARAM1: 1                     ;PARAM #1
PARAM2: 0                     ;PARAM #2
PARAM3: 177777                ;PARAM #3
IRDA:  .WORD 0                ;INITIAL READ DATA ADDRESS
IXDA:  .WORD 0                ;INITIAL XMIT DATA ADDRESS
SETTLE: .WORD 0               ;LINE SETTLE DELAY FLAG
B2016: .WORD 0               ;ADDR OF BIN TO OCT TYPE ROUTINE
TIME:  .WORD 0               ;TIMER
:
:   .WORD START              ;ADDR OF START OF PROGRAM
TX. TERM: .BYTE 000          ;TRANSMITTER TERMINATING CHAR.
RX. TERM: .BYTE 001         ;RECEIVER TERMINATING CHAR.
:
*****
:   CONSTANTS + WORKING STORAGE
*****
:   STAT=R0
:   XFLG=100000                ;XMIT COMPLETE FLAG
:   RFLG=40000                 ;RCV COMPLETE FLAG
:   DSFLG=20000                ;DATA SET STATUS CHANGE FLAG
:   BIT13=20000                ;INHIBIT PRINTOUTS
:
SXCSR: 0                       ;SAVED XMIT CSR
SRCSR: 0                       ;SAVED RCV CSR
ERCSR: 0                       ;RCV CSR SAVED ON ERROR
ERDR: 0                        ;RCV DATA REG SAVED ON ERROR
DSSTAT: 0                      ;RCV CSR SAVED ON DS CHANGE
:
XCC: 0                          ;XMIT CHAR COUNT
RCC: 0                          ;RCV CHAR COUNT
RDA: 0                          ;RCV DATA ADDR.
XDA: 0                          ;XMIT DATA ADDR.
:
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
:
FULL DUPLEX=000001
PS=177776
```

600	010074	000240	
601	010076	013700	177570
602	010102	042700	177400
603	010106	013702	010006
604	010112	012722	012644
605	010116	013722	010010
606	010122	012722	012524
607	010126	013722	010010
608	010132	013704	010004
609	010136	013714	010012
610	010142	013702	010014
611	010146	042702	000001
612	010152	010264	000004
613			
614	010156	005037	010032
615	010162	005037	012040
616	010166	005037	012044
617	010172	032700	000001
618	010176	001402	
619	010200	000137	010352
620	010204	032700	000002
621	010210	001402	
622	010212	000137	010246
623	010216	032700	000010
624	010222	001402	
625	010224	000137	010446
626	010230	032700	000004
627	010234	001402	
628	010236	000137	010674
629	010242	000000	
630	010244	000776	
631			
632			
633			
634			
635			
636			
637			
638			
639			
640			
641			
642			
643			
644			
645	010246	004737	012362
646	010252	032700	040000

```

*****
: DC11-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, RCSR   ; SETUP VARIABLES
        MOV     PARAM2, R2
        BIC     #0001, R2
        MOV     R2,     XCSR(R4); IN CSR'S

```

```

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

```

```

GO:     CLR     TIME
        CLR     DELAY
        CLR     STOP
        BIT     #OWO, MODE
        BEQ    1$
        JMP    $OWO
1$:     BIT     #OWI, MODE
        BEQ    2$
        JMP    $OWI
2$:     BIT     #ILB, MODE
        BEQ    3$
        JMP    $ILB
3$:     BIT     #XLB, MODE
        BEQ    4$
        JMP    $XLB
4$:     HALT
        BR     .-2

```

```

*****
: 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".
*****

```

```

$OWI:   JSR     PC, STARTR
1$:     BIT     #RFLG, STAT

```

647	010256	001013		
648	010260	023727	010032	000100
649	010266	103771		
650	010270	011402		
651	010272	016403	000004	
652	010276	104001		
653	010300	005037	010032	
654	010304	000762		
655				
656	010306	032737	000200	177570
657	010314	001002		
658	010316	004737	011264	
659	010322	042700	040000	
660	010326	032737	000020	177570
661	010334	001405		
662	010336	012737	010350	012042
663	010344	000137	011122	
664	010350	000736		
665				
666				
667				
668				
669				
670				
671				
672				
673				
674				
675	010352	004737	012046	
676	010356	005037	010032	
677	010362	032700	100000	
678	010366	001013		
679	010370	023727	010032	000100
680	010376	103771		
681	010400	011402		
682	010402	016403	000004	
683	010406	104001		
684	010410	005037	010032	
685	010414	000762		
686	010416	042700	100000	
687	010422	032737	000020	177570
688	010430	001405		
689	010432	012737	010444	012042
690	010440	000137	011122	
691	010444	000742		
692				
693				
694				

BNE	2\$
CMP	TIME, #100
BLO	1\$
MOV	2RCSR, R2
MOV	XCSR(R4), R3
HLT	1
CLR	TIME
BR	1\$
2\$:	BIT #NODAT, SWR
	BNE 3\$
	JSR PC, TESTD
3\$:	BIC #RFLG, STAT
	BIT #LOOP, SWR
	BEQ 4\$
	MOV #4\$, BACK
	JMP EOP
4\$:	BR \$OWI

```

*****
: 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."
*****

```

\$OWO:	JSR PC, STARTX
	CLR TIME
1\$:	BIT #XFLG, STAT
	BNE 2\$
	CMP TIME, #100
	BLO 1\$
	MOV 2RCSR, R2
	MOV XCSR(R4), R3
	HLT 1
	CLR TIME
	BR 1\$
2\$:	BIC #XFLG, STAT
	BIT #LOOP, SWR
	BEQ 3\$
	MOV #3\$, BACK
	JMP EOP
3\$:	BR \$OWO

```

695
696
697
698
699
700
701
702
703
704
705
706 010446 004737 012362
707 010452 005037 010032
708 010456 032700 040000
709 010462 001013
710 010464 023727 010032 000100
711 010472 103771
712 010474 011402
713 010476 016403 000004
714 010502 104001
715 010504 005037 010032
716 010510 000762
717 010512 032737 000200 177570
718 010520 001002
719 010522 004737 011264
720 010526 042700 040000
721 010532 032737 000020 177570
722 010540 001405
723 010542 012737 010554 012042
724 010550 000137 011122
725 010554 032737 000400 177570
726 010562 001416
727 010564 013702 010020
728 010570 013703 010022
729 010574 010337 010062
730 010600 112223
731 010602 001376
732 010604 112743 000177
733 010610 005203
734 010612 112723 000177
735 010616 105023
736 010620 005037 010032
737 010624 004737 012046
738 010630 032700 100000
739 010634 001013
740 010636 023727 010032 000100
741 010644 103771
742 010646 011402
743 010650 016403 000004
744 010654 104001
745 010656 005037 010032
746 010662 000762
747 010664 042700 100000
748 010670 000137 010446

```

```

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

```

```

$ILB: JSR PC,STARTR
      CLR TIME
1$: BIT #RFLG,STAT
      GNE 2$
      CMP TIME,#100
      BLO 1$
      MOV @RCSR,R2
      MOV XCSR(A4),R3
      HLT 1
      CLR TIME
2$: BR 1$
      BIT #NOOAT,SWR
      BNE 3$
      JSR PC,TESTD
3$: BIC #RFLG,STAT
      BIT #LOOP,SWR
      BEQ 4$
      MOV #4$,BACK
      JMP EOP
4$: BIT #400, @#SWR ;USE EXTERNAL DATA?
      BEQ 7$ ;BR IF NO
      MOV IRDA, R2 ;SET POINTER
      MOV IXDA, R3 ;SET POINTER
      MOV R3, XDA ;SETUP XMIT DATA ADDR
      MOVB (R2)+, (R3)+ ;MOVE INPUT TO OUTPUT
      BNE #-2 ;LOOP IF NOT ZERO CHAR
      MOVB #177, -(R3) ;INSERT A FILL CHAR
      INC R3 ;BUMP ADDRESS
      MOVB #177, (R3)+ ;INSERT ANOTHER FILL
      CLRB (R3)+ ;INSERT ZERO CHAR
7$: CLR TIME
      JSR PC,STARTX
5$: BIT #XFLG,STAT
      BNE 6$
      CMP TIME,#100
      BLO 5$
      MOV @RCSR,R2
      MOV XCSR(A4),R3
      HLT 1
      CLR TIME
6$: BR 5$
      BIC #XFLG,STAT
      JMP $ILB

```

```

749
750
751
752
753
754
755
756
757
758
759
760
761
762 010674
763 010674 032737 000001 010014
764 010702 001402
765 010704 004737 012362
766 010710 004737 012046
767 010714 005037 010032
768 010720 032700 100000
769 010724 001016
770 010726 032700 040000
771 010732 001024
772 010734 023727 010032 000100
773 010742 103766
774 010744 011402
775 010746 016403 000004
776 010752 104001
777 010754 005037 010032
778 010760 000757
779 010762 032737 000001 010014
780 010770 001356
781 010772 042700 100000
782 010776 004737 012362
783 011002 000746
784 011004
785 011004 032737 000001 010014
786 011012 001420
787 011014 032700 100000
788 011020 001013
789 011022 023727 010032 000100
790 011030 103765
791 011032 011402
792 011034 016403 000004
793 011040 104001
794 011042 005037 010032
795 011046 000756
796 011050 042700 100000
797 011054 042700 040000
798 011060 005037 010032
799 011064 032737 000200 177570
800 011072 001002
801 011074 004737 011264
802 011100 032737 000020 177570
803 011106 001672
804 011110 012737 010674 012042

```

```

*****
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.
*****

```

```

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

```

F02

DC11 ITEP OVERLAY
DZDCOB.P11

MACY11 27(732) 23-SEP-76 08:44 PAGE 54

805 011116 000137 011122

JMP EOP

```

806
807
808
809
810
811
812 011122
813 011122 012737 000340 177776
814 011130 016437 000004 011262
815 011136 042737 177677 011262
816 011144 042764 000100 000004
817 011152 012766 011212 000002
818 011160 010037 012024
819 011164 010137 012026
820 011170 010237 012030
821 011174 010337 012032
822 011200 010437 012034
823 011204 010537 012036
824 011210 000207
825
826 011212
827 011212 013700 012024
828 011216 013701 012026
829 011222 013702 012030
830 011226 013703 012032
831 011232 013704 012034
832 011236 013705 012036
833 011242 012737 177777 012040
834 011250 053764 011262 000004
835 011256 000177 000560
836 011262 000000
837
838
839
840
841
842
843
844 011264 013746 010050
845 011270 001413
846 011272 032737 020000 177570
847 011300 001007
848 011302 104400 011464
849 011306 004077 176516
850 011312 005746
851 011314 104400 011545
852 011320 013701 010022
853 011324 013702 010020
854 011330 122122
855 011332 001776
856 011334 123741 010040
857 011340 001447
858 011342 122742 000002
859 011346 001005
860 011350 010237 011356
861 011354 104400

```

```

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

```

```

EOP:
MOV #340,PS ;LOCK OUT INTERRUPTS.
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 ;IF ORGINALLY SET; SET TX IE
BIS QTPIE,XCSR(R4)
JMP @BACK
QTPIE: 000000

```

```

*****
SUBROUTINE TO CHECK
RECEIVER DATA.
*****
TESTD: MOV ERDBR,-(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: 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 2$
MOV R2,1$
TYPE

```

```

862 011356 000000
863 011360 000437
864 011362
865 011362 105712
866 011364 001435
867 011366 122721 000177
868 011372 001756
869 011374 005301
870 011376 122722 000177
871 011402 001752
872 011404 000240
873 011406 032737 020000 177570
874 011414 001016
875 011416 104400 011550
876 011422 013737 010020 011432
877 011430 104400
878 011432 000000
879 011434 104400 011575
880 011440 013737 010022 011450
881 011446 104400
882 011450 010022
883 011452 111103
884 011454 114202
885 011456 104007
886 011460 005726
887 011462 000207
888
889 011464 005015 044124 051105
(1) 011545 015 000012
(1) 011550 005015 042522 042503
(1) 011575 015 042012 052101
(1) 011620 005015 046120 040505
(1) 011667 015 053412 042510
(1) 011752 005015 046120 040505
(1)
(1) 012024 000000
890 012026 000000
891 012030 000000
892 012032 000000
893 012034 000000
894 012036 000000
895 012040 000000
896 012042 000000
897 012044 000000
898

```

```

1$: .WORD 0
BR TESTDX
2$: TSTB (R2) ;
BEQ TESTDX ;BR IF YES
CMPB #177, (R1)+ ;IS IT FILL CHAR?
BEQ SCAN4 ;BR IF YES
DEC R1 ;BACKUP
CMPB #177, (R2)+ ;IS IT FILL?
BEQ SCAN4 ;BR IF YES
NOP ;DATA ERROR
BIT #BIT13, @#SWR ;INHIBIT PRINTOUTS
BNE DERR ;BR IF YES
TYPE MSG2 ;<15><12>RECEIVED DATA = <15><12>
MOV IRDA, RDAX ;SETUP DATA ADDRESS
TYPE ;PRINT RECEIVED DATA
RDAX: 0 ;RECEIVED DATA ADDR.
TYPE MSG3 ;<15><12>DATA SHOULD BE<15><12>
MOV IXDA, .+10 ;SETUP ADDR.
TYPE ;PRINT GOOD DATA
IXDA
DERR: MOVB (R1),R3 ;SETUP XMIT DATA
MOVB -(R2),R2 ;SETUP RCV DATA
HLT+7 ;DATA ERROR HALT
TESTDX: TST (SP)+ ;POP STACK
RTS PC ;RETURN FROM SUB/ROUT
MSG0: .ASCIZ <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
MSG1: .ASCIZ <15><12>
MSG2: .ASCIZ <15><12>/RECEIVED DATA = /<15><12>
MSG3: .ASCIZ <15><12>/DATA SHOULD BE/<15><12>
MSG4: .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
.ASCIZ <15><12>/WHEN CONNECTION COMPLETE: HIT CONTINUE SWITCH./<15><12>
MSG5: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
.EVEN
SAVR0: 0
SAVR1: 0
SAVR2: 0
SAVR3: 0
SAVR4: 0
SAVR5: 0
DELAY: 0
BACK: 0
STOP: 0

```

```

899
900
901
902      04100
903      012046      04200
904      012046      005737      012040      04300
905      012052      001416      04400
906      012054      005037      012514      04500
907      012060      012737      000007      012516      04600
908      012066      062737      000001      012514      04700
909      012074      001374      04800
910      012076      005337      012516      04900
911      012102      001371      05000
912      012104      005037      012040      05100
913      012110      013737      010022      010062      05200
914      012116      042700      100000      05300
915      012122      052714      000001      05400
916      012126      005737      012044      05500
917      012132      001005      05600
918      012134      104400      011620      05700
919      012140      000000      05800
920      012142      005137      012044      05900
921      012146      032737      000001      010014      06000
922      012154      001003      06100
923      012156      032714      000004      06200
924      012162      001375      06300
925      012164      052764      000001      000004      06400
926      06500
927      012172      032764      000002      000004      06600
928      012200      001017      06700
929      012202      023727      010032      000036      06800
930      012210      103770      06900
931      012212      011402      07000
932      012214      016403      000004      07100
933      012220      032737      010000      177570      07200
934      012226      001001      07300
935      012230      104002      07400
936      012232      005037      010032      07500
937      012236      000755      07600
938      012240      032714      000004      07700
939      012244      001775      07800
940      07900
941      012246      005737      010024      08000
942      012252      001416      08100
943      012254      005037      012514      08200
944      012260      012737      000030      012516      08300
945      012266      062737      000001      012514      08400
946      012274      001374      08500
947      012276      005337      012516      08600
948      012302      001371      08700
949      012304      005037      010024      08800
950      012310      032737      000001      010014      08900
951      012316      001415      09000
952      012320      032700      000004      09100
953      012324      001412      09200
954      012326      012737      177777      012522      09300

```

```

*****
INITIALIZE TRANSMIT SUBROUTINE
*****
STARTX:
TST      DELAY      ;IF SW04=1 & SW14=0 WAIT BEFORE TURNING ON TX
BEQ      1$         ;NO GO AHEAD AND TURN ON TX
CLR      TEMP1      ;PREPARE FOR DELAY
MOV      #7,TEMP2
ADD      #1,TEMP1   ;INCREMENT DELAY.....
BNE      .-6
DEC      TEMP2
BNE      .-14
CLR      DELAY      ;ZERO POINTER.
MOV      IXDA, XDA  ;SETUP XMIT DATA ADDR.
BIC      #XFLG, STAT ;RESET XMIT COMPLETE FLAG
BIS      #DTR,  DRCR ;SET DATA TERM READY
TST      STOP
BNE      2$
TYPE    ,MSG4
HALT
COM      STOP
1$:      BIT      #FULL.DUPLEX,PARAM2;FULL DUPLEX?
BNE      3$         ;BR IF YES
BIT      #4,DRCR    ;CARRIER UP?
BNE      .-4        ;BR IF YES
3$:      BIS      #RQTS,XCSR(R4)
CTSW:    BIT      #CTS, XCSR(R4); IS CLEAR TO SEND SET?
BNE      2$         ;BR IF YES
CMP      TIME, #36  ;30 SECS ELAPSED?
BLO      CTSW      ;BR IF NO
MOV      DRCR, R2   ;SETUP RCV CSR
MOV      XCSR(R4),R3 ;SETUP XMIT CSR
BIT      #SW12,SWR  ;INHIBIT PRINTOUTS?
BNE      1$         ;BR IF YES
HLT+2    ;PRINTOUT 'WAITING TO XMIT'
1$:      CLR      TIME
BR       CTSW      ;RESET TIMER
2$:      BIT      #4,DRCR ;WAIT SOME MORE
BEQ      2$        ;IS CARRIER UP?
          ;BR IF NO
CTSOX:   TST      SETTLE ;CONNECTION JUST MADE?
BEQ      2$        ;BR IF NO
CLR      TEMP1      ;YES PREPARE FOR DELAY
MOV      #14*2,TEMP2
ADD      #1,TEMP1   ;INCREMENT DELAY
BNE      .-6
DEC      TEMP2
BNE      .-14
CLR      SETTLE     ;CLEAR DELAY FLAG
2$:      BIT      #FULL.DUPLEX,PARAM2;FULL DUPLEX?
BEQ      1$         ;BR IF NO
BIT      #XLB,MODE  ;XLB MODE?
BEQ      1$         ;BR IF NO
MOV      #-1,TRNFLG ;SET FLAG

```

```

955 012334 052764 000100 000004 09400
956 012342 000001          09500
957 012344 005737 012520          09600
958 012350 001375          09700
959 012352 052764 000100 000004 09800
960 012360 000207          09900
961          10000
962
963
964
965 012362 005737 012044          10200
966 012366 001012          10300
967 012370 052714 000001          10400
968 012374 104400 011752          10500
969 012400 012737 177777 010024 10600
970 012406 012737 177777 012044 10700
971 012414 032737 000001 010014 10800
972 012422 001410          10900
973 012424 032700 000004          11000
974 012430 001405          11100
975 012432 005037 012514          11200
976 012436 005237 012514          11300
977 012442 001375          11400
978 012444 012737 177777 012520 11500
979 012452 013737 010020 010060 11600
980 012460 012737 001000 010056 11700
981 012466 042700 040000          11800
982 012472 005037 010046          11900
983 012476 005037 010050          12000
984 012502 005764 000002          12100
985 012506 052714 000101          12200
986 012512 000207          12300
987 012514 000000          12400
988 012516 000000          12500
989 012520 000000          12600
990 012522 000000          12700

```

```

          BIS      #TIE,XCSR(R4) ;SET INTERRUPT ENABLE
          WAIT
          TST      SNCFLG        ;FIRST CHAR RECEIVED YET?
          BNE      .-4           ;BR IF NO
1$:      BIS      #TIE, XCSR(R4);SET XMIT INTERRUPT ENABLE
          RTS      PC           ;EXIT FROM SUBROUTINE

;*****
;      INITIALIZE RECEIVER SUBROUTINE
;*****
STARTR:  TST      STOP          ;FIRST TIME HERE?
          BNE      1$           ;BR IF NO
          BIS      #DTR,DRCSR   ;SET DTR
          TYPE     MSGS         ;MAKE CONNECTION
2$:      MOV      #-1,SETTLE   ;YES SET DELAY FLAG
          MOV      #-1,STOP
          BIT      #FULL.DUPLEX,PARAM2;FULL DUPLEX?
          BEQ      3$           ;BR IF NO
          BIT      #XLB,MODE    ;XLB MODE?
          BEQ      3$           ;BR IF NO
          CLR      TEMP1        ;START DELAY
          INC      TEMP1
          BNE      .-4
          MOV      #-1,SNCFLG   ;SET FLAG
          MOV      IRDA, RDA     ;SETUP RCV DATA ADDR
          MOV      #1000, RCC    ;SETUP RCV CHAR COUNT
          BIC      #RFLG, STAT   ;RESET RCV COMPLETE FLAG
          CLR      ERCSR        ;RESET ERROR RECORDS
          CLR      ERDBR
          TST      RBUF(R4)     ;CLEAR ANY DONE BIT.
          BIS      #RIE+DTR,DRCSR;SET INTERRUPT ENABLES
          RTS      PC           ;EXIT FROM SUBROUTINE

TEMP1:  0
TEMP2:  0
SNCFLG: 0
TRNFLG: 0

```

```

991
992
993
994 012524 000240 13000
995 012526 127737 175330 010040 13100
996 012534 001011 13200
997 012536 052700 100000 13300
998 012542 042764 000001 000004 13400
999 012550 042764 000100 000004 13500
1000 012556 000417 13600
1001 13700
1002 012560 117764 175276 000006 13800
1003 012566 032737 000100 177570 13900
1004 012574 001406 14000
1005 012576 105777 175266 14100
1006 012602 100003 14200
1007 012604 117777 175252 175260 14300
1008 012612 14400
1009 012612 005237 010062 14500
1010 012616 005737 012522 14600
1011 012622 001403 14700
1012 012624 042764 000100 000004 14800
1013 012632 005037 010032 14900
1014 012636 005037 012522 15000
1015 012642 000002 15100
1016
1017
1018
1019 012644 000240 15300
1020 012646 105714 15400
1021 012650 100050 15500
1022 012652 116401 000002 15600
1023 012656 142701 000200 15700
1024 012662 110177 175172 15800
1025 012666 032737 000040 177570 15900
1026 012674 001405 16000
1027 012676 105777 175166 16100
1028 012702 100002 16200
1029 012704 110177 175162 16300
1030 012710 16400
1031 012710 005237 010060 16500
1032 012714 105077 175140 16600
1033 012720 005337 010056 16700
1034 012724 001003 16800
1035 012726 104006 16900
1036 012730 004737 012362 17000
1037 17100
1038 012734 123701 010041 17200
1039 012740 001004 17300
1040 012742 042714 000100 17400
1041 012746 052700 040000 17500
1042 012752 005764 000002 17600
1043 012756 100005 17700
1044 012760 011437 010046 17800
1045 012764 016437 000002 010050 17900
1046 18000

```

```

*****
; TRANSMIT INTERRUPT SERVICE ROUTINE
*****

```

```

XISR:  NOP
      CMPB  @XDA,TX.TERM  ;FINISHED XMITTING?
      BNE  XISR1          ;BR IF NO
      BIS  #XFLG, STAT   ;SET XMIT COMPLETE FLAG
      BIC  #RQTS,XCSR(R4);RESET REQUEST TO SEND
      BIC  #TIE, XCSR(R4);RESET XMIT INTERRUPT ENABLE
      BR   XISR2
;
XISR1: MOVB  @XDA, XBUF(R4);XMIT NEXT CHAR.
      BIT  #100, SWR     ;MONITOR OUTPUT?
      BEQ  NOXMON        ;BR IF NO
      TSTB @TPS          ;IS TTY AVAILABLE
      BPL  NOXMON        ;BR IF NO
      MOVB @XDA, @TPB   ;TYPE THE CHAR
;
NOXMON:
XISR2: INC  XDA          ;INCRMENT ADDRESS
      TST  TRNFLG       ;FIRST CHAR?
      BEQ  1$           ;BR IF NO
      BIC  #TIE,XCSR(R4);CLEAR INTERUPT ENABLE
1$:   CLR  TIME         ;RESET TIMER
      CLR  TRNFLG      ;CLEAR FLAG
      RTI              ;RETURN FROM INTERRUPT

```

```

*****
; RECEIVE INTERRUPT SERVICE ROUTINE
*****

```

```

RISR:  NOP
      TSTB @RCSR        ;IS RECEIVER DONE BIT SET
      BPL  RISR2        ;BR IF NO
      MOVB RBUF(R4),R1  ;STORE CHAR.
      BICB #200, R1     ;STRIP A BIT
      MOVB R1, @RDA     ;MOVE CHAR TO INBUF
      BIT  #40, SWR     ;MONITOR INPUT?
      BEQ  NORMON       ;BR IF NO
      TSTB @TPS        ;IS TTY AVAILABLE?
      BPL  NORMON       ;BR IF NO
      MOVB R1, @TPB    ;TYPE THE CHAR
;
NORMON:
      INC  RDA          ;BUMP POINTER
      CLRB @RDA        ;CLEAR NEXT CHAR POSITION
      DEC  RCC          ;DECREMENT CHAR. COUNTER
      BNE  1$          ;BR IF BUFFER NOT FULL
      HLT+6            ;RECEIVER BUFFER FULL
      JSR  PC,STARTR   ;INITIALIZE RECEIVER
;
1$:   CMPB  RX.TERM,R1  ;IS IT LINE FEED?
      BNE  RISR1        ;BR IF NO
      BIC  #RIE,@RCSR   ;DISABLE INTERRUPTS
      BIS  #RFLG, STAT  ;SET RCVR COMPLETE FLAG
RISR1: TST  RBUF(R4)    ;IS THERE A DATA ERROR
      BPL  RISR2        ;BR IF NO
      MOV  @RCSR, ERCSR ;SAVE RCSR
      MOV  RBUF(R4),ERDBR;SAVE RDBR

```

1047	012772	005714		18100	RISR2:	TST	@RCSR		: IS THERE A DATA SET STATUS CHANGE
1048	012774	100004		18200		BPL	RISR3		: BR IF NO
1049	012776	011437	010052	18300		MOV	@RCSR,	DSSTAT	: SAVE STATUS
1050	013002	052700	020000	18400		BIS	#DSFLG,	STAT	: SET FLAG
1051	013006	005037	012520	18500	RISR3:	CLR	SNCFLG		
1052	013012	005037	010032	18600		CLR	TIME		: RESET TIMER
1053	013016	000002		18700		RTI			: RETURN FROM INTERRUPT
1054				18800					
1055		000001		18900		.END			

BA	010004	545#	602						
BACK	012042	662*	689*	723*	804*	835	896#		
BIT0	= 000001	540#							
BIT1	= 000002	540#							
BIT10	= 002000	540#							
BIT11	= 004000	540#							
BIT12	= 010000	540#							
BIT13	= 020000	540#	571#	846	873				
BIT14	= 040000	540#							
BIT15	= 100000	540#							
BIT2	= 000004	540#							
BIT3	= 000010	540#							
BIT4	= 000020	540#							
BIT5	= 000040	540#							
BIT6	= 000100	540#							
BIT7	= 000200	540#							
BIT8	= 000400	540#							
BIT9	= 001000	540#							
BRK	= 000002	540#							
B2016	010030	555#	849						
CD	= 000004	540#							
CT	= 040000	540#							
CTS	= 000002	540#	927						
CTSOK	012246	941#							
CTSW	012172	927#	930	937					
DC11	010000	544#							
DELAY	012040	615*	833*	895#	904	912*			
DERR	011452	874	883#						
DSFLG	= 020000	540#	570#	1050					
DSSTAT	010052	577#	1049*						
DTR	= 000001	540#	915	967	985				
ENTER	011212	817	826#						
EOP	011122	663	690	724	805	812#			
ER	= 100000	540#							
ERCSR	010046	575#	982*	1044*					
ERDBR	010050	576#	844	983*	1045*				
FULL.D	= 000001	589#	763	779	785	921	950	971	
GO	010156	614#							
HD	= 004000	540#							
ILB	= 000010	540#	623						
IRDA	010020	551#	727	853	876	979			
IXDA	010022	552#	728	852	880	882	913		
LOOP	= 000020	540#	660	687	721	802			
MODE	= %000000	540#	617	620	623	626	952	973	
MSG0	011464	848	889#						
MSG1	011545	851	889#						
MSG2	011550	875	889#						
MSG3	011575	879	889#						
MSG4	011620	889#	918						
MSG5	011752	889#	968						
NODAT	= 000200	540#	656	717	799				
NORMON	012710	1026	1028	1030#					
NOXMON	012612	1004	1006	1008#					
OV	= 010000	540#							
OWI	= 000002	540#	620						
OWO	= 000001	540#	617						

STAT = X000000	567#	646	659*	677	686*	708	720*	738	747*	768	770	781*	787
STD = 000400	795*	797*	914*	981*	997*	1041*	1050*						
STOP = 012044	540#												
SWR = 177570	516*	897#	916	920*	965	970*							
SW12 = 010000	540#	595	656	660	687	717	721	725	799	802	846	873	933
SXCSR = 010042	1003	1025											
TEMP1 = 012514	540#	933											
TEMP2 = 012516	573#												
TESTD = 011264	906*	908*	943*	945*	975*	976*	987#						
TESTDX = 011460	907*	910*	944*	947*	988#								
TIF = 000100	658	719	801	844#									
TIME = 010032	657	853	866	886#									
TKB = 010066	540#	815	816	955#	959	999	1012						
TKS = 010064	536#	614*	648	653*	676*	679	684*	707*	710	715*	736*	740	745*
TPB = 010072	767*	772	777*	789	794*	798*	929	936*	1013*	1052*			
TPS = 010070	585#												
TR = 000200	584#												
TRNFLG = 012522	587#	1007*	1029*										
TSTDAT = 011320	586#	1005	1027										
TX.TER = 010040	540#												
TYPE = 104400	954*												
XBUF = 000006	954*	990#	1010	1014*									
XCC = 010054	845	847	852#										
XCSR = 000004	540#	855	859										
XDA = 010062	540#	848	851	861	875	877	879	881	918	968			
XFLG = 100000	579#	1002*											
XISR = 012524	540#	606*	651	682	713	743	775	792	814	816*	834*	925*	927
XISR1 = 012560	932	955*	959*	998*	999*	1012*							
XISR2 = 012616	582#	729*	913*	995	1002	1007	1009*						
XLB = 000004	568#	677	686	738	747	768	781	787	796	914	997		
XWAIT = 104412	600	994#											
XLB = 010446	996	1002#											
SILB = 010446	1000	1010#											
SOWI = 010246	540#	626	952	973									
SOWO = 010352	540#												
SXLB = 010674	625	706#	748										
.	622	645#	664										
	619	675#	691										
	628	762#	803	804									
	542#	630	731	880*	909	911	924	946	948	958	977		

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DZDCOB.SEG/SOL/CRF/PAGNUM-ITEPV4.DZDCOB
RUN-TIME: 10 12 1 SECONDS
RUN-TIME RATIO: 228/24=9.3
CORE USED: 16K (31 PAGES)

