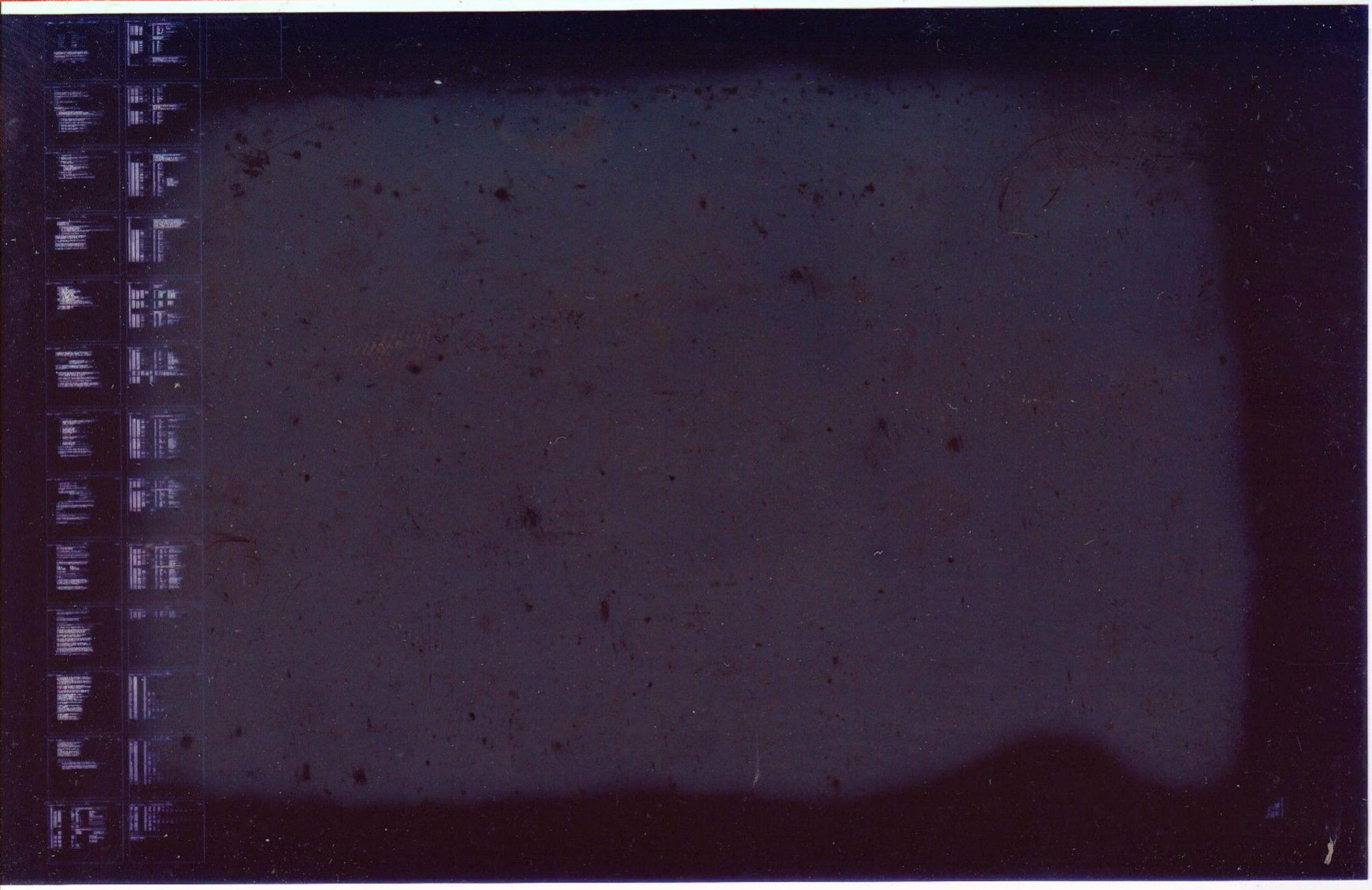
DL11

OVERLAY FOR ITEP
CZDL0D0

AH-8533D-MC

FICHE 1 OF 1

JUN 1978
digital
MADE IN USA



IDENTIFICATION

PRODUCT CODE:	AC-8532D-MC
PRODUCT NAME:	CZDLODO DL11 OVRLY FOR ITEP
PROGRAM DATE:	MARCH 1978
MAINTAINER:	DIAGNOSTICS
AUTHORS:	R A JONES JOHN EGOLF
REVISED BY:	FAY BASHAW 1/21/75

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 CZDLODO DL11 COMMUNICATION INTERFACE.

2.2 STORAGE.

4K OF CORE

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

4.0 OPERATING PROCEDURES.

A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED
1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE
STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.
*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

B. CONSOLE DIALOGUE PARAMETER INPUT (CURRENT VALUES FOR PARAMETERS ARE FOUND IN OVERLAY)

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

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

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

3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS
B. TYPEIN ACTUAL VECTOR ADDRESS

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

A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPEIN ACTUAL VALUE

5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE ISR.(SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPEIN ACTUAL VALUE

6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE ISR.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. ENTER ACTUAL VALUE

7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
A. TYPE 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 SWICHES 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 DMBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DMB.

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 PROCEEDURES
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
```

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE
CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G
(CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE
THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM
CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER
APPLICABLE AREAS.

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A
REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE.
TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN
↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377)
SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).
I.E. ABCD↑ 000 123 377↑ EFG (CAR.RETURN)

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

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

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

TEST MODES

INTERNAL LOOPBACK MODE

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

EXTERNAL LOOPBACK MODE

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

ONE-WAY-IN MODE

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

ONE-WAY-OUT MODE

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

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

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

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

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

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

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

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

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

*WXXXXXX	TO PRINTOUT THE 8 WORDS AT LOC XXXXXX.
*BXXXXXX	TO PRINTOUT THE 16 BYTES AFTER LOC XXXXXX.
*C	TO CONTINUE

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

5.0 PROGRAM AND/OR OPERATOR ACTION

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

5.1 NORMAL HALTS

SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

```
RECEIVED DATA=RRRRRR
DATA SHOULD BE TTTTTT
DATA COMPARE ERROR;  BAD DATA=BBB  GOOD DATA=GGG
```

WHERE RRRRRR IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)
TTTTTT IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
BBB IS THE BAD DATA CHARACTER
GGG IS THE GOOD DATA CHARACTER

IF THE INTERFACE DETECTS A DATA ERROR, THE FOLLOWING
WILL BE PRINTED BEFORE THE DATA IS COMPARED:

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

WHERE XXXXXX IS THE CONTENTS OF THE RECEIVER DATA REGISTER
THE LOW BYTE IS THE DATA, AND THE HIGH BYTE IS THE ERROR BITS.

IF A RECEIVE TERMINATING CHARACTER<DD1> 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 ASYNCRONOUS DEVICE, MODEMS AND THE
MAYNARD TEST STATION AND INITALIZE 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 IMMEDEATE ERROR MESSAGE FROM MAYNARD WILL BE RE-

CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING
FOR THE REST OF THE INTERUPTED 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 CZDLODO DL11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR,
BY THE ITEP CONTROL PROGRAM.

TIME: PROVIDES A MEANS OF MEASURING ELASPED 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 INTERUPTS AND SAVES THE TRANSMITTER INTERUPT 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 DL11

PARAM#1 MUST BE ALL ZEROS.

PARAM#2 BIT 0 OF THIS PARAMETER IS CHECKED BY THE SOFTWARE TO RUN
EITHER FULL-DUPLEX OR HALF-DUPLEX. BIT0=1 SELECTS FULL-DUPLEX,
BIT0=0 SELECTS HALF-DUPLEX. DEFAULT IS HALF-DUPLEX, ALL OTHER BITS MUST BE ZEROS.

PARAM#3 IS USED BY SOFTWARE TO DETERMINE IF TEST IS TO BE RUN A MODEM ;;++D
OTHER THAN A STANDARD AMERICAN MODEM. FOR EXAMPLE,EUORPEAN MODEMS
EMPLOY A CLAMP TO INHIBIT CARRIER DETECT(BIT12) WHEN REQUEST TO
SEND IS ASSERTED. IF THIS DIAGNOSTIC WILL BE RUN ON A EUROPEAN
MODEM, MANUALLY SET PARAM3 TO ZERO(0). OTHERWISE LEAVE VALUE AT
MINUS ONE(177777).

```
                                ;*****************************************************************
                                ;       DL11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000          .=11000
011000  046104  000040         DL11:   .ASCIZ  /DL /           ;ISR NAME
011004  175610                 BA:     175610                  ;BUS ADDRESS
011006  000300                 RIV:    300                     ;VECTOR ADDRESS
011010  000200                 PRIOR:  200                     ;PRIORITY
011012  000000                 PARAM1: 000000                  ;PARAM #1
011014  000000                 PARAM2: 000000                  ;PARAM #2
011016  177777                 PARAM3: 177777                  ;PARAM #3
011020  000000                 IRDA:   .WORD   0               ;INITIAL READ DATA ADDRESS
011022  000000                 IXDA:   .WORD   0               ;INITIAL XMIT DATA ADDRESS
011024  000000                 SETTLE: .WORD   0               ;LINE SETTLE DELAY FLAG
011026  000000                         .WORD   0               ;
011030  000000                 B2016:  .WORD   0               ;ADDR OF BIN TO OCT TYPE ROUTINE
011032  000000                 TIME:   .WORD   0               ;TIMER
011034  000000                         .WORD   0               ;
011036  011102                         .WORD   START           ;ADDR OF START OF PROGRAM
011040                          TX.TERM:
011040      000                         .BYTE   000             ;TRANSMITTER TERMINATING CHAR.
011041                          RX.TERM:
011041      001                         .BYTE   001             ;RECEIVER TERMINATING CHAR.
011042  000000                 FLAG:   .WORD   0
011044  177570                 SWR:    177570
011046  177570                 DISPLAY:177570

                                ;*****************************************************************
                                ;       CONSTANTS + WORKING STORAGE
                                ;*****************************************************************
                        000000          STAT=R0
                        100000          XFLG=100000             ;XMIT COMPLETE FLAG
                        040000          RFLG=40000              ;RCV COMPLETE FLAG
                        020000          DSFLG=20000             ;DATA SET STATUS CHANGE FLAG
                        020000          BIT13=20000             ;INHIBIT PRINTOUTS

011050  000000         SXCSR:  0                               ;SAVED XMIT CSR
011052  000000         SRCSR:  0                               ;SAVED RCV CSR
011054  000000         ERCSR:  0                               ;RCV CSR SAVED ON ERROR
011056  000000         ERDBR:  0                               ;RCV DATA REG SAVED ON ERROR
011060  000000         DSSTAT: 0                               ;RCV CSR SAVED ON DS CHANGE

011062  000000         XCC:    0                               ;XMIT CHAR COUNT
011064  000000         RCC:    0                               ;RCV CHAR COUNT
011066  000000         RDA:    0                               ;RCV DATA ADDR.
011070  000000         XDA:    0                               ;XMIT DATA ADDR.

011072  177560         TKS:    177560
011074  177562         TKB:    177562
011076  177564         TPS:    177564
011100  177566         TPB:    177566

                        000001  FULL.DUPLEX=000001
```

```
                                ;*****************************************************************
                                ;       DL11-X INTERFACE SERVICE ROUTINE
                                ;*****************************************************************
011102  000240                  START:  NOP
011104  017700  177734                  MOV     @SWR,   R0      ;SETUP MODE IN R0
011110  042700  177400                  BIC     #177400,R0      ;STRIP JUNK
011114  013702  011006                  MOV     RIV,    R2      ;SETUP
011120  012722  013662                  MOV     #RISR,  (R2)+   ;INTERRUPT
011124  013722  011010                  MOV     PRIOR,  (R2)+   ;VECTORS
011130  012722  013644                  MOV     #XISR,  (R2)+   ;
011134  013722  011010                  MOV     PRIOR,  (R2)+   ;
011140  013704  011004                  MOV     BA,     R4      ;SETUP BUS ADDR INDEX
011144  013714  011012                  MOV     PARAM1, @RCSR   ;SETUP VARIABLES
011150  013702  011014                  MOV     PARAM2,R2       ;
011154  042702  000001                  BIC     #0001,R2        ;
011160  010264  000004                  MOV     R2,     XCSR(R4);IN CSR'S

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

011164  005037  011032          GO:     CLR     TIME
011170  005037  013054                  CLR     DELAY
011174  005037  013060                  CLR     STOP
011200  032700  000001                  BIT     #OWO,MODE
011204  001402                          BEQ     1$
011206  000137  011362                  JMP     $OWO
011212  032700  000002          1$:     BIT     #OWI,MODE
011216  001402                          BEQ     2$
011220  000137  011254                  JMP     $OWI
011224  032700  000010          2$:     BIT     #ILB,MODE
011230  001402                          BEQ     3$
011232  000137  011460                  JMP     $ILB
011236  032700  000004          3$:     BIT     #XLB,MODE
011242  001402                          BEQ     4$
011244  000137  011710                  JMP     $XLB
011250  000000                  4$:     HALT
011252  000776                          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".
                                ;*************************************************************************


011254  104416                  $OWI:   KBDIN
011256  004737  013402                  JSR     PC,STARTR
```

```
   691  011262  032700  040000          1$:     BIT     #RFLG,STAT
   692  011266  001013                          BNE     2$
   693  011270  023727  011032  000100          CMP     TIME,#100
   694  011276  103771                          BLO     1$
   695  011300  011402                          MOV     @RCSR,R2
   696  011302  016403  000004                  MOV     XCSR(R4),R3
   697  011306  104001                          HLT     1
   698  011310  005037  011032                  CLR     TIME
   699  011314  000762                          BR      1$
   700
   701  011316  032777  000200  177520  2$:     BIT     #NODAT,@SWR
   702  011324  001002                          BNE     3$
   703  011326  004737  012300                  JSR     PC,TESTD
   704  011332  042700  040000          3$:     BIC     #RFLG,STAT
   705  011336  032777  000020  177500          BIT     #LOOP,@SWR
   706  011344  001405                          BEQ     4$
   707  011346  012737  011360  013056          MOV     #4$,BACK
   708  011354  000137  012140                  JMP     EOP
   709  011360  000735                  4$:     BR      $OWI
   710
   711
   712                                  ;*************************************************************************
   713                                  ;       ROUTINE USED IF "ONE WAY OUT" WAS SELECTED.
   714                                  ;       NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE ONLY
   715                                  ;       MODE AVAILABLE.
   716                                  ;       "ONE WAY OUT" MEANS THAT ONLY THE TRANSMITTER IS
   717                                  ;       ENABLED. THE RECEIVER IS NEVER "TURNED ON.
   718                                  ;*************************************************************************
   719
   720  011362  104416                  $OWO:   KBDIN
   721  011364  004737  013062                  JSR     PC,STARTX
   722  011370  005037  011032                  CLR     TIME
   723  011374  032700  100000          1$:     BIT     #XFLG,STAT
   724  011400  001013                          BNE     2$
   725  011402  023727  011032  000100          CMP     TIME,#100
   726  011410  103771                          BLO     1$
   727  011412  011402                          MOV     @RCSR,R2
   728  011414  016403  000004                  MOV     XCSR(R4),R3
   729  011420  104001                          HLT     1
   730  011422  005037  011032                  CLR     TIME
   731  011426  000762                          BR      1$
   732  011430  042700  100000          2$:     BIC     #XFLG,STAT
   733  011434  032777  000020  177402          BIT     #LOOP,@SWR
   734  011442  001405                          BEQ     3$
   735  011444  012737  011456  013056          MOV     #3$,BACK
   736  011452  000137  012140                  JMP     EOP
   737  011456  000741                  3$:     BR      $OWO
   738
   739
   740
```

```
                        ;*************************************************************************
                        ;       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 COMLPETE 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.
                        ;*************************************************************************

011460  104416          $ILB:   KBDIN
011462  004737  013402          JSR     PC,STARTR
011466  005037  011032          CLR     TIME
011472  032700  040000  1$:     BIT     #RFLG,STAT
011476  001013                  BNE     2$
011500  023727  011032  000100  CMP     TIME,#100
011506  103771                  BLO     1$
011510  011402                  MOV     @RCSR,R2
011512  016403  000004          MOV     XCSR(R4),R3
011516  104001                  HLT     1
011520  005037  011032          CLR     TIME
011524  000762                  BR      1$
011526  032777  000200  177310  2$:     BIT     #NODAT,@SWR
011534  001002                  BNE     3$
011536  004737  012300          JSR     PC,TESTD
011542  042700  040000  3$:     BIC     #RFLG,STAT
011546  032777  000020  177270  BIT     #LOOP,@SWR
011554  001405                  BEQ     4$
011556  012737  011570  013056  MOV     #4$,BACK
011564  000137  012140          JMP     EOP
011570  032777  000400  177246  4$:     BIT     #400,   @SWR    ;USE EXTERNAL DATA?
011576  001416                  BEQ     7$              ;BR IF NO
011600  013702  011020          MOV     IRDA,   R2      ;SET POINTER
011604  013703  011022          MOV     IXDA,   R3      ;SET POINTER
011610  010337  011070          MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
011614  112223                  MOVB    (R2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
011616  001376                  BNE     .-2             ;LOOP IF NOT ZERO CHAR
011620  112743  000177          MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
011624  005203                  INC     R3              ;BUMP ADDRESS
011626  112723  000177          MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
011632  105023                  CLRB    (R3)+           ;INSERT ZERO CHAR
011634  005037  011032  7$:     CLR     TIME
011640  004737  013062          JSR     PC,STARTX
011644  032700  100000  5$:     BIT     #XFLG,STAT
011650  001013                  BNE     6$
011652  023727  011032  000100  CMP     TIME,#100
011660  103771                  BLO     5$
011662  011402                  MOV     @RCSR,R2
011664  016403  000004          MOV     XCSR(R4),R3
011670  104001                  HLT     1
011672  005037  011032          CLR     TIME
011676  000762                  BR      5$
011700  042700  100000  6$:     BIC     #XFLG,STAT
011704  000137  011460          JMP     $ILB
```

```
                        ;*************************************************************************
                        ;       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.
                        ;*************************************************************************

011710  104416                  $XLB:   KBDIN
011712  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
011720  001402                          BEQ     1$
011722  004737  013402                  JSR     PC,STARTR
011726  004737  013062          1$:     JSR     PC,STARTX
011732  005037  011032                  CLR     TIME
011736  032700  100000          2$:     BIT     #XFLG,STAT
011742  001016                          BNE     3$
011744  032700  040000          7$:     BIT     #RFLG,STAT
011750  001024                          BNE     4$
011752  023727  011032  000100          CMP     TIME,#100
011760  103766                          BLO     2$
011762  011402                          MOV     @RCSR,R2
011764  016403  000004                  MOV     XCSR(R4),R3
011770  104001                          HLT     1
011772  005037  011032                  CLR     TIME
011776  000757                          BR      2$
012000  032737  000001  011014  3$:     BIT     #FULL.DUPLEX,PARAM2
012006  001356                          BNE     7$
012010  042700  100000                  BIC     #XFLG,STAT
012014  004737  013402                  JSR     PC,STARTR
012020  000746                          BR      2$
012022  032737  000001  011014  4$:     BIT     #FULL.DUPLEX,PARAM2
012030  001420                          BEQ     8$
012032  032700  100000                  BIT     #XFLG,STAT
012036  001013                          BNE     6$
012040  023727  011032  000100          CMP     TIME,#100
012046  103765                          BLO     4$
012050  011402                          MOV     @RCSR,R2
012052  016403  000004                  MOV     XCSR(R4),R3
012056  104001                          HLT     1
012060  005037  011032                  CLR     TIME
012064  000756                          BR      4$
012066  042700  100000          6$:     BIC     #XFLG,STAT
012072  042700  040000          8$:     BIC     #RFLG,STAT
012076  005037  011032                  CLR     TIME
012102  032777  000200  176734          BIT     #NODAT,@SWR
012110  001002                          BNE     5$
012112  004737  012300                  JSR     PC,TESTD
012116  032777  000020  176720  5$:     BIT     #LOOP,@SWR
012124  001671                          BEQ     $XLB
012126  012737  011710  013056          MOV     #$XLB,BACK
012134  000137  012140                  JMP     EOP
```

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

012140                                  EOP:
012140  104414  000340                          STPS,PRTY7              ;SET PS PRIORITY TO 7
012144  016437  000004  012276                  MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012152  042737  177677  012276                  BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012160  042764  000100  000004                  BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012166  012766  012226  000002                  MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012174  010037  013040                          MOV     R0,SAVR0        ;SAVE REGISTER 0
012200  010137  013042                          MOV     R1,SAVR1        ;SAVE REGISTER 1
012204  010237  013044                          MOV     R2,SAVR2        ;SAVE REGISTER 2
012210  010337  013046                          MOV     R3,SAVR3        ;SAVE REGISTER 3
012214  010437  013050                          MOV     R4,SAVR4        ;SAVE REGISTER 4
012220  010537  013052                          MOV     R5,SAVR5        ;SAVE REGISTER 5
012224  000207                                  RTS     PC              ;RETURN TO CONTROL PROGRAM

012226                                  ENTER:
012226  013700  013040                          MOV     SAVR0,R0        ;RESTORE R0
012232  013701  013042                          MOV     SAVR1,R1        ;RESTORE R1
012236  013702  013044                          MOV     SAVR2,R2        ;RESTORE R2
012242  013703  013046                          MOV     SAVR3,R3        ;RESTORE R3
012246  013704  013050                          MOV     SAVR4,R4        ;RESTORE R4
012252  013705  013052                          MOV     SAVR5,R5        ;RESTORE R5
012256  012737  177777  013054                  MOV     #-1,DELAY
012264  053764  012276  000004                  BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012272  000177  000560                          JMP     @BACK
012276  000000                          QTPIE:  000000


                                        ;*************************************
                                        ;       SUBROUTINE TO CHECK
                                        ;       RECEIVER DATA.
                                        ;*************************************
012300  013746  011056                  TESTD:  MOV     ERCBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012304  001413                                  BEQ     TSTDAT          ;BR IF NO
012306  032777  020000  176530                  BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012314  001007                                  BNE     TSTDAT          ;BR IF YES
012316  104400  012500                          TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012322  004077  176502                          JSR     R0,@B2016       ;PRINT CONTENTS OF RBUF
012326  005746                                  TST     -(SP)
012330  104400  012561                          TYPE    ,MSG1           ;<15><12>
012334  013701  011022                  TSTDAT: MOV     IXDA,   R1      ;SETUP XMIT DATA ADDR
012340  013702  011020                          MOV     IRDA,   R2      ;SETUP RCV DATA ADDR
012344  122122                          SCAN4:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012346  001776                                  BEQ     SCAN4           ;BR IF OK
012350  123741  011040                          CMPB    TX.TERM,-(R1)   ;IS IT END OF DATA
012354  001447                                  BEQ     TESTDX          ;BR IF YES
012356  122742  000002                          CMPB    #002,-(R2)
012362  001005                                  BNE     2$
012364  010237  012372                          MOV     R2,1$
012370  104400                                  TYPE
```

```
 908  012372  000000                    1$:     .WORD   0
 909  012374  000437                            BR      TESTDX
 910  012376                            2$:
 911  012376  105712                            TSTB    (R2)            ;
 912  012400  001435                            BEQ     TESTDX          ;BR IF YES
 913  012402  122721  000177                    CMPB    #177,   (R1)+   ;IS IT FILL CHAR?
 914  012406  001756                            BEQ     SCAN4           ;BR IF YES
 915  012410  005301                            DEC     R1              ;BACKUP
 916  012412  122722  000177                    CMPB    #177,   (R2)+   ;IS IT FILL?
 917  012416  001752                            BEQ     SCAN4           ;BR IF YES
 918  012420  000240                    SCAN5:  NOP                     ;DATA ERROR
 919  012422  032777  020000  176414            BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS
 920  012430  001016                            BNE     DERR            ;BR IF YES
 921  012432  104400  012564                    TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
 922  012436  013737  011020  012446            MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
 923  012444  104400                            TYPE                    ;PRINT RECEIVED DATA
 924  012446  000000                    RDAX:   0                       ;RECEIVED DATA ADDR.
 925  012450  104400  012611                    TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
 926  012454  013737  011022  012464            MOV     IXDA,   .+10    ;SETUP ADDR.
 927  012462  104400                            TYPE                    ;PRINT GOOD DATA
 928  012464  011022                            IXDA
 929  012466  111103                    DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
 930  012470  114202                            MOVB    -(R2),R2        ;SETUP RCV DATA
 931  012472  104007                            HLT+7                   ;DATA ERROR HALT
 932  012474  005726                    TESTDX: TST     (SP)+           ;POP STACK
 933  012476  000207                            RTS     PC              ;RETURN FROM SUB/ROUT
 934
 935  012500  005015  044124  051105    MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1)   012561     015  000012            MSG1:   .ASCIZ  <15><12>
(1)   012564  005015  042522  042503    MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
(1)   012611     015  042012  052101    MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
(1)   012634  005015  046120  040505    MSG4:   .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1)   012703     015  053412  042510            .ASCIZ <15><12>/WHEN CONNECTION COMPLETE, HIT CONTINUE SWITCH./<15><12>
(1)   012766  005015  046120  040505    MSG5:   .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)                               .EVEN
(1)   013040  000000                    SAVR0:  0
 936  013042  000000                    SAVR1:  0
 937  013044  000000                    SAVR2:  0
 938  013046  000000                    SAVR3:  0
 939  013050  000000                    SAVR4:  0
 940  013052  000000                    SAVR5:  0
 941  013054  000000                    DELAY:  0
 942  013056  000000                    BACK:   0
 943  013060  000000                    STOP:   0
 944
```

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

013062                  STARTX:
013062  005737  013054          TST     DELAY           ;IF SW04=1 & SW14=0 WAIT BEFORE TURINING ON TX
013066  001416                  BEQ     1$              ;NO GO AHEAD AND TURN ON TX
013070  005037  013534          CLR     TEMP1           ;PREPARE FOR DELAY
013074  012737  000007  013536  MOV     #7,TEMP2
013102  062737  000001  013534  ADD     #1,TEMP1        ;INCREMENT DELAY.......
013110  001374                  BNE     .-6
013112  005337  013536          DEC     TEMP2
013116  001371                  BNE     .-14
013120  005037  013054          CLR     DELAY           ;ZERO POINTER.
013124  013737  011022  011070  1$:     MOV     IXDA,   XDA     ;SETUP XMIT DATA ADDR.
013132  052714  000002          BIS     #DTR,@RCSR      ;SET REQUEST TO SEND
013136  005737  013060          TST     STOP
013142  001005                  BNE     2$
013144  104400  012634          TYPE    ,MSG4
013150  000000                  HALT
013152  005137  013060          COM     STOP
013156  032737  000001  011014  2$:     BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
013164  001003                  BNE     3$              ;BR IF YES
013166  032714  010000          BIT     #10000,@RCSR    ;CARRIER UP?
013172  001375                  BNE     .-4             ;BR IF YES
013174  052714  000004          3$:     BIS     #RQTS,@RCSR
013200  032714  000002          BIT     #DTR,   @RCSR   ;IS THIS A DL11-E?
013204  001430                  BEQ     CTSOK           ;BR IF NO

013206  032714  020000          CTSW:   BIT     #CTS,   @RCSR   ;IS CLEAR TO SEND SET?
013212  001017                  BNE     2$              ;BR IF YES
013214  023727  011032  000036          CMP     TIME,   #36     ;30 SECS ELASPED?
013222  103771                  BLO     CTSW            ;BR IF NO
013224  011402                  MOV     @RCSR,  R2      ;SETUP RCV CSR
013226  016403  000004          MOV     XCSR(R4),R3     ;SETUP XMIT CSR
013232  032777  010000  175604  BIT     #SW12,@SWR      ;INHIBIT PRINTOUTS?
013240  001001                  BNE     1$              ;BR IF YES
013242  104002                  HLT+2                   ;PRINTOUT 'WAITING TO XMIT'
013244  005037  011032          1$:     CLR     TIME            ;RESET TIMER
013250  000756                  BR      CTSW            ;WAIT SOME MORE
013252  005737  011016          2$:     TST     PARAM3          ;STANDARD MODEM?                ;;++D
013256  001403                  BEQ     CTSOK           ;IF NO, BR.                     ;;++D
013260  032714  010000          3$:     BIT     #10000,@RCSR    ;IS CARRIER UP?                 ;;++D
013264  001775                  BEQ     3$              ;BR IF NO                       ;;++D

013266  005737  011024          CTSOK:  TST     SETTLE          ;CONNECTION JUST MADE?
013272  001416                  BEQ     2$              ;BR IF NO
013274  005037  013534          CLR     TEMP1           ;YES PREPARE FOR DELAY
013300  012737  000030  013536  MOV     #14*2,TEMP2
013306  062737  000001  013534  ADD     #1,TEMP1        ;INCREMENT DELAY
013314  001374                  BNE     .-6
013316  005337  013536          DEC     TEMP2
013322  001371                  BNE     .-14
013324  005037  011024          CLR     SETTLE          ;CLEAR DELAY FLAG
013330  032737  000001  011014  2$:     BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
013336  001415                  BEQ     1$              ;BR IF NO
```

```
1001  013340  032700  000004            BIT     #XLB,MODE          ;XLB MODE?
1002  013344  001412                    BEQ     1$                 ;BR IF NO
1003  013346  012737  177777  013542    MOV     #-1,TRNFLG         ;SET FLAG
1004  013354  052764  000100  000004    BIS     #TIE,XCSR(R4)      ;SET INTERUPT ENABLE
1005  013362  000001                    WAIT
1006  013364  005737  013540            TST     SNCFLG             ;FIRST CHAR RECEIVED YET?
1007  013370  001375                    BNE     .-4                ;BR IF NO
1008  013372  052764  000100  000004  1$:BIS     #TIE,   XCSR(R4);SET XMIT INTERRUPT ENABLE
1009  013400  000207                    RTS     PC                 ;EXIT FROM SUBROUTINE
1010
1011                                  ;*************************************************************
1012                                  ;       INITIALIZE RECEIVER SUBROUTINE
1013                                  ;*************************************************************
1014  013402  005737  013060          STARTR: TST     STOP               ;FIRST TIME HERE?
1015  013406  001012                    BNE     1$                 ;BR IF NO
1016  013410  052714  000002            BIS     #DTR,@RCSR         ;SET DTR
1017  013414  104400  012766            TYPE    ,MSG$              ;MAKE CONNECTION
1018  013420  012737  177777  011024  2$:MOV     #-1,SETTLE         ;YES SET DELAY FLAG
1019  013426  012737  177777  013060      MOV     #-1,STOP
1020  013434  032737  000001  011014  1$:BIT     #FULL.DUPLEX,PARAM2;FULL DUPLEX?
1021  013442  001410                    BEQ     3$                 ;BR IF NO
1022  013444  032700  000004            BIT     #XLB,MODE          ;XLB MODE?
1023  013450  001405                    BEQ     3$                 ;BR IF NO
1024  013452  005037  013534            CLR     TEMP1              ;START DELAY
1025  013456  005237  013534            INC     TEMP1
1026  013462  001375                    BNE     .-4
1027  013464  012737  177777  013540  3$:MOV     #-1,SNCFLG         ;SET FLAG
1028  013472  013737  011020  011066      MOV     IRDA,   RDA        ;SETUP RCV DATA ADDR
1029  013500  012737  001000  011064      MOV     #1000,  RCC        ;SETUP RCV CHAR COUNT
1030  013506  042700  040000            BIC     #RFLG,STAT         ;CLEAR RFLG
1031  013512  005037  011054            CLR     ERCSR              ;RESET ERROR RECORDS
1032  013516  005037  011056            CLR     ERDBR
1033  013522  005764  000002            TST     RBUF(R4)
1034  013526  052714  000143            BIS     #RIE+DTR+DIE+RE,@RCSR;SET INTERRUPT ENABLES
1035  013532  000207                    RTS     PC                 ;EXIT FROM SUBROUTINE
1036
1037  013534  000000                  TEMP1:  0
1038  013536  000000                  TEMP2:  0
1039  013540  000000                  SNCFLG: 0
1040  013542  000000                  TRNFLG: 0
```

```
1041                                    ;*****************************************************************
1042                                    ;       TRANSMIT INTERRUPT SERVICE ROUTNE
1043                                    ;*****************************************************************
1044  013544  000240                    XISR:   NOP
1045  013546  127737  175316  011040            CMPB    @XDA,TX.TERM    ;FINISHED XMITTING?
1046  013554  001010                            BNE     XISR1           ;BR IF NO
1047  013556  052700  100000                    BIS     #XFLG,  STAT    ;SET XMIT COMPLETE FLAG
1048  013562  042714  000004                    BIC     #RQTS,  @RCSR   ;RESET REQUEST TO SEND
1049  013566  042764  000100  000004            BIC     #TIE,   XCSR(R4);RESET XMIT INTERRUPT ENABLE
1050  013574  000417                            BR      XISR2           ;
1051
1052  013576  117764  175266  000006  XISR1:   MOVB    @XDA,   XBUF(R4);XMIT NEXT CHAR.
1053  013604  032777  000100  175232            BIT     #100,   @SWR    ;MONITOR OUTPUT?
1054  013612  001406                            BEQ     NOXMON          ;BR IF NO
1055  013614  105777  175256                    TSTB    @TPS            ;IS TTY AVAILABLE
1056  013620  100003                            BPL     NOXMON          ;BR IF NO
1057  013622  117777  175242  175250            MOVB    @XDA,   @TPB    ;TYPE THE CHAR
1058  013630                          NOXMON:
1059  013630  005237  011070                    INC     XDA             ;INCRMENT ADDRESS
1060  013634  005737  013542          XISR2:    TST     TRNFLG          ;FIRST CHAR?
1061  013640  001403                            BEQ     1$              ;BR IF NO
1062  013642  042764  000100  000004            BIC     #TIE,XCSR(R4)   ;CLEAR INTERUPT ENABLE
1063  013650  005037  011032          1$:       CLR     TIME            ;RESET TIMER
1064  013654  005037  013542                    CLR     TRNFLG          ;CLEAR FLAG
1065  013660  000002                            RTI                     ;RETURN FROM INTERRUPT
1066                                    ;*****************************************************************
1067                                    ;       RECEIVE INTERRUPT SERVICE ROUTNE
1068                                    ;*****************************************************************
1069  013662  000240                    RISR:   NOP
1070  013664  105714                            TSTB    @RCSR           ;IS RECEIVER DONE BIT SET
1071  013666  100055                            BPL     RISR2           ;BR IF NO
1072  013670  116401  000002                    MOVB    RBUF(R4),R1     ;STORE CHAR.
1073  013674  142701  000200                    BICB    #200,   R1      ;STRIP A BIT
1074  013700  110177  175162                    MOVB    R1,     @RDA    ;MOVE CHAR TO INBUF
1075  013704  032777  000040  175132            BIT     #40,    @SWR    ;MONITOR INPUT?
1076  013712  001405                            BEQ     NORMON          ;BR IF NO
1077  013714  105777  175156                    TSTB    @TPS            ;IS TTY AVAILABLE?
1078  013720  100002                            BPL     NORMON          ;BR IF NO
1079  013722  110177  175152                    MOVB    R1,     @TPB    ;TYPE THE CHAR
1080  013726                          NORMON:
1081  013726  005237  011066                    INC     RDA             ;BUMP POINTER
1082  013732  105077  175130                    CLRB    @RDA            ;CLEAR NEXT CHAR POSITION
1083  013736  005337  011064                    DEC     RCC             ;DECREMENT CHAR. COUNTER
1084  013742  001010                            BNE     1$              ;BR IF BUFFER NOT FULL
1085  013744  042714  000100                    BIC     #RIE,   @RCSR   ;RESET INTERRUPT ENAB
1086  013750  011402                            MOV     @RCSR,  R2      ;SETUP RCV CSR
1087  013752  016403  000004                    MOV     XCSR(R4),R3     ;SETUP XMIT CSR
1088  013756  104006                            HLT+6                   ;RECEIVER BUFFER FULL
1089  013760  004737  013402                    JSR     PC,STARTR       ;INITIALIZE RECEIVER
1090
1091  013764  123701  011041          1$:       CMPB    RX.TERM,R1      ;IS IT LINE FEED?
1092  013770  001004                            BNE     RISR1           ;BR IF NO
1093  013772  042714  000140                    BIC     #RIE+DIE,@RCSR  ;DISABLE INTERRUPTS
1094  013776  052700  040000                    BIS     #RFLG,  STAT    ;SET RCVR COMPLETE FLAG
1095  014002  005764  000002          RISR1:    TST     RBUF(R4)        ;IS THERE A DATA ERROR
1096  014006  100005                            BPL     RISR2           ;BR IF NO
```

```
 1097  014010  011437  011054                MOV     @RCSR, ERCSR    ;SAVE RCSR
 1098  014014  016437  000002  011056        MOV     RBUF(R4),ERDBR  ;SAVE RDBR
 1099  014022  005714          RISR2:  TST     @RCSR           ;IS THERE A DATA SET STATUS CHANGE
 1100  014024  100004                  BPL     RISR3           ;BR IF NO
 1101  014026  011437  011060          MOV     @RCSR, DSSTAT   ;SAVE STATUS
 1102  014032  052700  020000          BIS     #DSFLG, STAT    ;SET FLAG
 1103  014036  005037  013540  RISR3:  CLR     SNCFLG          ;CLEAR FLAG
 1104  014042  005037  011032          CLR     TIME            ;RESET TIMER
 1105  014046  000002                  RTI                     ;RETURN FROM INTERRUPT
 1106
 1107          000001                  .END
```

```
BA       011004              587#    646
BACK     013056              707*    735*    770*    850*    881     942#
BIT0   = 000001              581#
BIT1   = 000002              581#
BIT10  = 002000              581#
BIT11  = 004000              581#
BIT12  = 010000              581#
BIT13  = 020000              581#    616#    892     919
BIT14  = 040000              581#
BIT15  = 100000              581#
BIT2   = 000004              581#
BIT3   = 000010              581#
BIT4   = 000020              581#
BIT5   = 000040              581#
BIT6   = 000100              581#
BIT7   = 000200              581#
BIT8   = 000400              581#
BIT9   = 001000              581#
B2016    011030              597#    895
CD     = 010000              581#
CTS    = 020000              581#    974
CTSOK    013266              972     986     990#
CTSW     013206              974#    977     984
DELAY    013054              659*    879*    941#    950     958*
DERR     012466              920     929#
DIE    = 000040              581#   1034    1093
DISPLA   011046              607#
DL11     011000              586#
DSC    = 100000              581#
DSFLG  = 020000              581#    615#    1102
DSSTAT   011060              622#   1101*
DTR    = 000002              581#    960     971     1016    1034
ENTER    012226              863     872#
EOP      012140              708     736     771     851     858#
ER     = 100000              581#
ERCSR    011054              620#   1031*   1097*
ERDBR    011056              621#    890    1032*   1098*
FE     = 020000              581#
FLAG     011042              605#
FULL.D=  000001              634#    810     826     831     966     999     1020
GO       011164              658#
ILB    = 000010              581#    667
IRDA     011020              593#    774     899     922    1028
IXDA     011022              594#    775     898     926     928     959
KBDIN  = 104416              581#    689     720     752     809
LOOP   = 000020              581#    705     733     768     848
MSG0     012500              894     935#
MSG1     012561              897     935#
MSG2     012564              921     935#
MSG3     012611              925     935#
MSG4     012634              935#    963
MSG5     012766              935#   1017
NODAT  = 000200              581#    701     764     845
NORMON   013726             1076    1078    1080#
NOXMON   013630             1054    1056    1058#
OR     = 040000              581#
```

```
OWI     = 000002            581#    664
OWO     = 000001            581#    661
PARAM1    011012            590#    647
PARAM2    011014            591#    648    810    826    831    966    999   1020
PARAM3    011016            592#    985
PE      = 010000            581#
PRIOR     011010            589#    643    645
PRTY0   = 000000            581#
PRTY1   = 000040            581#
PRTY2   = 000100            581#
PRTY3   = 000140            581#
PRTY4   = 000200            581#
PRTY5   = 000240            581#
PRTY6   = 000300            581#
PRTY7   = 000340            581#    859
QTPIE     012276            860*   861*    880    882#
RA      = 004000            581#
RBUF    = 000002            581#   1033   1072   1095   1098
RCC       011064            625#   1029*  1083*
RD      = 000200            581#
RDA       011066            626#   1028*  1074*  1081*  1082*
RDAX      012446            922*    924#
RE      = 000001            581#   1034
RFLG    = 040000            614#    691    704    755    767    817    843   1030   1094
RI      = 040000            581#
RIE     = 000100            581#   1034   1085   1093
RISR      013662            642    1069#
RISR1     014002           1092    1095#
RISR2     014022           1071    1096   1099#
RISR3     014036           1100    1103#
RIV       011006            588#    641
RQTS    = 000004            581#    970   1048
RWAIT   = 104410            581#
RX.TER    011041            603#   1091
SAVR0     013040            864*    873    935#
SAVR1     013042            865*    874    936#
SAVR2     013044            866*    875    937#
SAVR3     013046            867*    876    938#
SAVR4     013050            868*    877    939#
SAVR5     013052            869*    878    940#
SCAN4     012344            900#    901    914    917
SCAN5     012420            918#
SETTLE    011024            595#    990    998*  1018*
SNCFLG    013540           1006    1027*  1039#  1103*
SRCSR     011052            619#
SRD     = 002000            581#
START     011102            600     638#
STARTR    013402            690     753    812    829   1014#   1089
STARTX    013062            721     784    813    949#
STD     = 000010            581#
STOP      013060            660*    943#   961    965*   1014   1019*
STPS    = 104414            581#    859
SWR       011044            606#    639    701    705    733    764    768    772    845    848    892    919    980
                           1053    1075
SW12    = 010000            581#    980
SXCSR     011050            618#
```

```
TEMP1   013534            952*   954*   992*   994*  1024*  1025*  1037#
TEMP2   013536            953*   956*   993*   996*  1038#
TESTD   012300            703    766    847    890#
TESTDX  012474            903    909    912    932#
TIE   = 000100            581#   861    862   1004   1008   1049   1062
TIME    011032            598#   658*   693    698*   722*   725    730*   754*   757    762*   783*   787    792*
                          814#   819    824*   835    840*   844*   976    983*  1063*  1104*
TKB     011074            630#
TKS     011072            629#
TPB     011100            632#  1057*  1079*
TPS     011076            631#  1055   1077
TR    = 000200            581#
TRNFLG  013542           1003*  1040#  1060   1064*
TSTDAT  012334            891    893    898#
TX.TER  011040            601#   902    1045
TYPE  = 104400            581#   894    897    907    921    923    925    927    963    1017
XBUF  = 000006            581#  1052*
XCC     011062            624#
XCSR  = 000004            581#   650*   696    728    760    790    822    838    860    862*   880*   979    1004*
                         1008*  1049*  1062*  1087
XDA     011070            627#   776*   959*  1045   1052   1057   1059*
XFLG  = 100000            613#   723    732    785    794    815    828    833    842    1047
XISR    013544            644   1044#
XISR1   013576           1046   1052#
XISR2   013634           1050   1060#
XLB   = 000004            581#   670    1001   1022
XWAIT = 104412            581#
$ILB    011460            669    752#   795
$OWI    011254            666    689#   709
$OWO    011362            663    720#   737
$XLB    011710            672    809#   849    850
.     = 014050            585#   674    778    926*   955    957    969    995    997    1007   1026

. ABS.  014050     000

 ERRORS DETECTED:  0

DSKZ:CZDLOD,DSKZ:CZDLOD.SEQ=DSKZ:ITEP1.MAC,DSKZ:CZDLOD.P11
RUN-TIME:  3  5  .3 SECONDS
RUN-TIME RATIO: 65/9=7.1
CORE USED:  16K  (31 PAGES)

DOCUMENT PAGES:  26
```