

DH11

OVERLAY FOR ITEP
CZDHLC0

AH-8489C-MC

COPYRIGHT © 75-78

FICHE 1 OF 1

JUN 1978

digital

MADE IN USA

The microfiche strip contains approximately 15 frames. Each frame displays a table with multiple columns and rows of data. The text is small and difficult to read, but the layout appears to be a structured data set, possibly a schedule or a list of items. The frames are arranged vertically, with each frame containing a similar table structure.

EOF1CVKUAASEQ
PDP80040001
CZDHLCO.P11

00010000 780519
CZDHLCO DH11 ITEP OVERLAY
23-MAR-78 12:56

PDP10 411
MACY11 30A(1052) 23-MAR-78 13:03 PAGE 2

00010000

780519

SEQ 0001

IDENTIFICATION

PRODUCT CODE: AC-8488C-MC
PRODUCT NAME: CZDHLCO DH11 OVRLY FOR ITEP
PROGRAM DATE: MARCH 1978
MAINTAINER: DIAGNOSTICS
AUTHORS: R A JONES
JOHN EGOLF
FAY BASHAW

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR IT'S AFFILIATED COMPANIES.

COOPYRIGHT (C)1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUSS

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 CZDHLCO DH11 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 DM11BB.

- 2. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
 - A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
 - B. 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 DN-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, DN11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWITCHES THEN HIT CONTINUE.)
 - A. HALT FOR BUS ADDRESS OF INTERFACE
 - B. HALT FOR VECTOR ADDRESS OF INTERFACE
 - C. HALT FOR PRIORITY OF INTERFACE
 - D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - E. HALT FOR INTERFACE PARAM #2 (DN11 AND 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 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

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

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

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

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

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

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY 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 MANUALLY EXAMINE THE TRANSMIT OR RECEIVE BUFFERS, DO THE FOLLOWING: TO FIND THE STARTING ADDRESS OF THE RECEIVE BU. FER, 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<OO1> IS NOT DETECTED
WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

7.0 RESTRICTIONS

THE OPERATION OF THIS PROGRAM REQUIRES COORDINATION BETWEEN
THE OPERATOR AND THE OPERATOR OF ANOTHER PDP-11 SYSTEM
UNLESS ONE OF THE SYSTEMS IS ALWAYS OPERATING IN A FIXED
MODE. THE FOLLOWING TABLE LISTS THE VALID COMBINATIONS:

CPU #1	CPU #2
ONE-WAY-OUT	ONE-WAY-IN
ONE-WAY-IN	ONE-WAY-OUT
EXTERNAL-LOOPBACK	INTERNAL-LOOPBACK
INTERNAL-LOOPBACK	EXTERNAL-LOOPBACK
EXTERNAL-LOOPBACK	EXTERNAL-LOOPBACK (FULL DUPLEX)

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

IF RUNNING IN FULL DUPLEX MODE BOTH SYSTEMS
MUST BE IN EXTERNAL LOOP BACK MODE.

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.

EXAMPLE:
SWITCHES 14, 13, 7, 4 SHOULD BE THE SAME
ON BOTH CPU'S

IF PROGRAM IS WAITING IN A SCAN ROUTINE AND TYPES OUT
A "WAITING MESSAGE", IF AN INCOMING MESSAGE STARTS DURING
THE TYPE OUT IT WILL BE LOST BECAUSE THE TYPEOUT PRIORITY
IS AT LEVEL 7. THIS WILL RESULT IN OVERRUN OR SILO OVER-
RUN ERRORS, DEPENDING ON THE DEVICE. TO AVOID THIS SITUATION
RUN WITH SWITCH 13 UP. IF OVERRUN DOES OCCURE DURING A
TYPEOUT THE PROGRAM SHOULD BE RESTARTED.

IF USING AN ASYNCHRONOUS DEVICE, MODEMS AND THE
MAYNARD TEST STATION AND INITIALIZE DOES NOT CLEAR THE
CONNECTION (EXAMPLE THE DJ11) IF THE PROGRAM IS RESTARTED
IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING CR
AN IMMEDIATE ERROR MESSAGE FROM MAYNARD WILL BE RE-

CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING FOR THE REST OF THE INTERRUPTED MESSAGE. TO AVOID THIS ERROR, RESTART PROGRAM ONLY AT THE END OF THE MESSAGE CURRENTLY BEING TRANSMITTED.

8.0 MISCELLANEOUS

ITEP WAS CHECKED OUT USING THE FOLLOWING BELL TELEPHONE MODEMS.
201A (HALF-DUPLEX SYNCHRONOUS 2000 BAUD)
202C (HALF-DUPLEX ASYNCHRONOUS 1200 BAUD)
103A (FULL-DUPLEX ASYNCHRONOUS 110 BAUD)

9.0 PROGRAM DESCRIPTION

9.1 THE CZDHLCO DH11 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 TYPING "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 DCR 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.

DH11 RESTRICTIONS

IF A DM11BB EXISTS IN THE SYSTEM WITH THE DH11 BEING
TESTED, BUT MODEM CONTROL IS NOT DESIRED AND THE DM11BB
WAS NOT INITIALIZED BY ITEP THE PROGRAM WILL HANG IN THE
DH11 TRANSMITTER INITIALIZATION ROUTINE. TO CORRECT THIS
LOAD LOCATION "DMBB" WITH AN ADDRESS THAT WILL TIME OUT (NO
SLAVE SYNC RESPONSE). THE ADDRESS OF DMBB CAN BE FOUND
IN THE CROSS REFERENCE TABLE IN THE BACK OF THIS LISTING.

576

10.0 PARAMETERS FOR THE DH11

PARAM#1 IS LOADED INTO THE SYSTEM CONTROL REGISTER. (SCR)
BITS 0-3 LINE SELECTION, DEFAULT= LINE 0 (0000)

PARAM#2 IS LOADED INTO THE LINE PARAMETER REGISTER. (LPR)
BITS 0,1 CHARACTER LENGTH, DEFAULT= 8 BITS (11)
BIT 2 STOP BITS, DEFAULT= 2 STOP BITS (1)
BIT 4 PARITY ENABLED (1), DEFAULT= (0)
BIT 5 ODD PARITY (1), DEFAULT= (0)
BITS 6-9 RECEIVER SPEED, DEFAULT= 110 BAUD (0011)
BITS 10-13 TRANSMIT SPEED, DEFAULT= 110 BAUD (0011)
BIT 14 HALF DUPLEX (1), DEFAULT= FULL DUPLEX (0)

PARAM#3 IS NOT USED (177777)

```

596
597
598
599
600
601 011000 011000 000040
602 011004 044104
603 011006 160020
604 011010 000300
605 011012 000240
606 011014 000000
607 011016 006307
608 011020 177777
609 011022 000000
610 011024 000000
611 011026 000000
612 011030 000000
613 011032 000000
614 011034 000000
615 011036 011102
616 011040
617 011040 000
618 011041
619 011041 001
620 011042 000000
621 011044 177570
622 011046 177570
623
624
625
626
627
628 000000
629 100000
630 040000
631 020000
632 020000
633 011050 000000
634 011052 000000
635 011054 000000
636 011056 000000
637 011060 000000
638
639 011062 000000
640 011064 000000
641 011066 000000
642 011070 000000
643
644 011072 177560
645 011074 177562
646 011076 177564
647 011100 177566
648
649 000001

```

```

*****
: DH11 INTERFACE SERVICE PARAMS
*****

```

```

DH11: .=11000
BA: .ASC=2 2Ch
RIY: 300
PRIOX: 240
PARAM1: 0
PARAM2: 006307
PARAM3: 177777
IRDA: .WORD 0
IXDA: .WORD 0
SETTLE: .WORD 0
B2016: .WORD 0
TIME: .WORD 0
TX. TERM: .WORD START
RX. TERM: .BYTE 000
FLAG: .BYTE 001
SWR: .WORD 0
DISPLAY: 177570

```

```

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

```

```

*****
: CONSTANTS + WORKING STORAGE
*****

```

```

STAT=RO
XFLG=100000
RFLG=40000
DSFLG=20000
BIT13=20000
SXCSR: 0
SRCR: 0
ERCSR: 0
ERDBR: 0
DSSTAT: 0
XCC: 0
RCC: 0
RD4: 0
XDA: 0
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566

```

```

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

```

```

FULL DUPLEX=000001

```

650
651
652
653 011102 000240
654 011104 017700 177734
655 011110 042700 177400
656 011114 013702 011006
657 011120 012722 014040
658 011124 013722 011010
659 011130 012722 013470
660 011134 013722 011010
661 011140 013704 011004
662 011144 012714 004000
663 011150 053714 011012
664 011154 053764 011014 000004
665 011162 123727 011012 000017
666 011170 101402
667 011172 000000
668 011174 000776
669 011176 010046
670 011200 012700 000001
671 011204 013701 011012
672 011210 005701
673 011212 001403
674 011214 006300
675 011216 005301
676 011220 000773
677 011222 010037 013714
678 011226 012600
679
680
681
682
683
684
685
686 011230 005037 011032
687 011234 005037 013120
688 011240 005037 013124
689 011244 032700 000001
690 011250 001402
691 011252 000137 011426
692 011256 032700 000002
693 011262 001402
694 011264 000137 011320
695 011270 032700 000010
696 011274 001402
697 011276 000137 011524
698 011302 032700 000004
699 011306 001402
700 011310 000137 011754
701 011314 000000
702 011316 000776
703
704
705

```
*****  
: DH11-X INTERFACE SERVICE ROUTINE  
:*****  
START: NOP  
MOV JSWR, 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 #MC, JRCSR  
BIS PARAM1, JRCSR  
BIS PARAM2, LPR(R4)  
CMPB PARAM1, #17  
BLOS 1$  
HALT ;NUMBER IN PARAM1 IS TOO LARGE!  
BR -2 ;MUST BE 17 OR LESS (LINE# IN OCTAL)  
1$: MOV R0, -(SP) ;SAVE R0  
MOV #1, R0  
MOV PARAM1, R1  
2$: TST R1  
BEQ 3$ ;CALCULATE BAR BIT  
ASL R0  
DEC R1  
BR 2$  
3$: MOV R0, BARTMP  
MOV (SP)+, R0
```

```
*****  
: ROUTINE USED TO GOT0  
: SUBROUTINE DEPENDENT  
: ON MODE SELECTED.  
:*****
```

```
G0: CLR TIME  
CLR DELAY  
CLR STOP  
BIT #OWO, MODE  
BEQ 1$  
1$: JMP $OWO  
BIT #OWI, MODE  
BEQ 2$  
2$: JMP $OWI  
BIT #ILB, MODE  
BEQ 3$  
3$: JMP $ILB  
BIT #XLB, MODE  
BEQ 4$  
4$: JMP $XLB  
HALT  
BR -2
```

706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761

011320 104416
011322 004737 013726
011326 032700 040000
011332 001013
011334 023727 011032 000100
011342 103771
011344 011402
011346 016403 000000
011352 104001
011354 005037 011032
011360 000762

011362 032777 000200 177454
011370 001002
011372 004737 012344
011376 042700 040000
011402 032777 000020 177434
011410 001405
011412 012737 011424 013122
011420 000137 012204
011424 000735

SOWI: KBDIN
JSR PC_STARTR
1\$: BIT #RFLG, STAT
BNE 2\$
CMP TIME, #100
BLO 1\$
MOV #RCSR, 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 SOWI

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".

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".

011426 104416
011430 004737 013126
011434 005037 011032
011440 032700 100000
011444 001013
011446 023727 011032 000100
011454 103771
011456 011402
011460 016403 000000
011464 104001
011466 005037 011032
011472 000762
011474 042700 100000
011500 032777 000020 177336

SOWO: KBDIN
JSR PC_STARTX
CLR TIME
1\$: BIT #XFLG, STAT
BNE 2\$
CMP TIME, #100
BLO 1\$
MOV #RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 1\$
2\$: BIC #XFLG, STAT
BIT #LOOP, #SWR

762	011506	001405				BEO	3\$
763	011510	012737	011522	013122		MOV	#3\$, BACK
764	011516	000137	012204			JMP	EOP
765	011522	000741			3\$:	BR	\$OWO
766							
767							
768							

769
770
771
772
773
774
775
776
777
778
779
780 011524 104416
781 011526 004737 013726
782 011532 005037 011032
783 011536 032700 040000
784 011542 001013
785 011544 023727 011032 000100
786 011552 103771
787 011554 011402
788 011556 016403 000000
789 011562 104001
790 011564 005037 011032
791 011570 000762
792 011572 032777 000200 177244 2S:
793 011600 001002
794 011602 004737 012344
795 011606 042700 040000 3S:
796 011612 032777 000020 177224
797 011620 001405
798 011622 012737 011634 013122
799 011630 000137 012204
800 011634 032777 000400 177202 4S:
801 011642 001416
802 011644 013702 011020
803 011650 013703 011022
804 011654 010337 011070
805 011660 112223
806 011662 001376
807 011664 112743 000177
808 011670 005203
809 011672 112723 000177
810 011676 105023
811 011700 005037 011032 7S:
812 011704 004737 013126
813 011710 032700 100000 5S:
814 011714 001013
815 011716 023727 011032 000100
816 011724 103771
817 011726 011402
818 011730 016403 000000
819 011734 104001
820 011736 005037 011032
821 011742 000762
822 011744 042700 100000 6S:
823 011750 000137 011524

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

SILB: KBDIN
JSR PC, STARTR
CLR TIME
1S: BIT #RFLG, STAT
BNE 2S
CMP TIME, #100
BLO 1S
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 1S
2S: BIT #NODAT, @SWR
BNE 3S
JSR PC, TESTD
3S: BIC #RFLG, STAT
BIT #LOOP, @SWR
BEQ 4S
MOV #4S, BACK
JMP EOP
4S: BIT #400, @SWR ; USE EXTERNAL DATA?
BEQ 7S ; 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
7S: CLR TIME
JSR PC, STARTX
5S: BIT #XFLG, STAT
BNE 6S
CMP TIME, #100
BLO 5S
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
C R TIME
BR 5S
6S: BIC #XFLG, STAT
JMP SILB

```

824
825
826
827
828
829
830
831
832
833
834
835
836
837 011754 104416
838 011756 032737 040000 011014
839 011764 001002
840 011766 004737 013726
841 011772 004737 013126
842 011776 005037 011032
843 012002 032700 100000
844 012006 001016
845 012010 032700 040000
846 012014 001024
847 012016 023727 011032 000100
848 012024 103766
849 012026 011402
850 012030 016403 000000
851 012034 104001
852 012036 005037 011032
853 012042 000757
854 012044 032737 040000 011014
855 012052 001756
856 012054 042700 100000
857 012060 004737 013726
858 012064 000746
859 012066 032737 040000 011014
860 012074 001020
861 012076 032700 100000
862 012102 001013
863 012104 023727 011032 000100
864 012112 103765
865 012114 011402
866 012116 016403 000000
867 012122 104001
868 012124 005037 011032
869 012130 000756
870 012132 042700 100000
871 012136 042700 040000
872 012142 005037 011032
873 012146 032777 000200 176670
874 012154 001002
875 012156 004737 012344
876 012162 032777 000020 176654
877 012170 001671
878 012172 012737 011754 013122
879 012200 000137 012204

```

```

*****
ROUTINE USED IF "EXTERNAL LOOP BACK" WAS SELECTED.
EITHER HALF OR FULL DUPLEX MAY BE SELECTED IN THIS MODE.
"EXTERNAL LOOP BACK" MEANS THAT THE TRANSMITTER IS FIRST
TURNED ON (IF HALF DUPLEX) AND THE WHOLE MESSAGE IS TRANSMITTED;
THEN THE RECEIVER IS ENABLED. AFTER THE WHOLE MESSAGE IS RECEIVED;
DATA WILL THEN BE CHECKED IF DESIRED AND END PASS WILL
BE GIVEN IF DESIRED. THEN THE CYCLE IS REPEATED
AS ABOVE. IF RUNNING IN FULL DUPLEX THE PROGRAM
WAITS FOR BOTH THE RECEIVER AND TRANSMITTER TO
FINISH THEN RESTARTS THE RECEIVER AND TRANSMITTER.
*****

```

```

$XLB: KBDIN
      BIT      #HALF.DUPLEX,PARAM2
      BNE     1$
      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$:   BIT     #HALF.DUPLEX,PARAM2
      BEQ     7$
      BIC     #XFLG,STAT
      JSR     PC,STARTR
      BR     2$
4$:   BIT     #HALF.DUPLEX,PARAM2
      BNE     8$
      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
      JMP     EOP

```

G02

```
880  
881  
882  
883  
884  
885  
886 012204  
887 012204 104414 000340  
888 012210 016437 000000 012342  
889 012216 042737 157777 012342  
890 012224 042764 020000 000000  
891 012232 012766 012272 000002  
892 012240 010037 013104  
893 012244 010137 013106  
894 012250 010237 013110  
895 012254 010337 013112  
896 012260 010437 013114  
897 012264 010537 013116  
898 012270 000207  
899  
900 012272  
901 012272 013700 013104  
902 012276 013701 013106  
903 012302 013702 013110  
904 012306 013703 013112  
905 012312 013704 013114  
906 012316 013705 013116  
907 012322 012737 177777 013120  
908 012330 053764 012342 000000  
909 012336 000177 000560  
910 012342 000000  
911  
912  
913  
914  
915  
916  
917  
918 012344 013746 011056  
919 012350 001413  
920 012352 032777 020000 176464  
921 012360 001007  
922 012362 104400 012544  
923 012366 004077 176436  
924 012372 005746  
925 012374 104400 012625  
926 012400 013701 011022  
927 012404 013702 011020  
928 012410 122122  
929 012412 001776  
930 012414 123741 011040  
931 012420 001447  
932 012422 122742 000002  
933 012426 001005  
934 012430 010237 012436  
935 012434 104400
```

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

```
EOP: STPS, PRTY7 ; SET PS PRIORITY TO 7  
MOV XCSR(R4), QTPIE ; SAVE TX CSR  
BIC #1<TIE>, QTPIE ; CLEAR ALL BUT TX IE.  
BIC #TIE, XCSR(R4) ; CLEAR TX IE (EVEN IF IT WASN'T SET)  
MOV #ENTER, 2(SP) ; SET FOR RETURN IF SW 14=1  
MOV R0, SAVR0 ; SAVE REGISTER 0  
MOV R1, SAVR1 ; SAVE REGISTER 1  
MOV R2, SAVR2 ; SAVE REGISTER 2  
MOV R3, SAVR3 ; SAVE REGISTER 3  
MOV R4, SAVR4 ; SAVE REGISTER 4  
MOV R5, SAVR5 ; SAVE REGISTER 5  
RTS PC ; RETURN TO CONTROL PROGRAM  
  
ENTER: MOV SAVR0, R0 ; RESTORE R0  
MOV SAVR1, R1 ; RESTORE R1  
MOV SAVR2, R2 ; RESTORE R2  
MOV SAVR3, R3 ; RESTORE R3  
MOV SAVR4, R4 ; RESTORE R4  
MOV SAVR5, R5 ; RESTORE R5  
MOV #-1, DELAY  
BIS QTPIE, XCSR(R4) ; IF ORGINALLY SET; SET TX IE  
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, JSWR ; 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 RCY 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 Z$  
MOV R2, 1$  
TYPE
```



```

936 012436 000000      1$:      .WORD      0
937 012440 000437      BR          TESTDX
938 012442
939 012442 105712
940 012444 001435
941 012446 122721 000177      2$:      TSTB      (R2)
942 012452 001756      BEQ      TESTDX
943 012454 005301      CMPB     #177, (R1)+ ; BR IF YES
944 012456 122722 000177      BEQ      SCAN4 ; IS IT FILL CHAR?
945 012462 001752      DEC     R1 ; BR IF YES
946 012464 000240      CMPB     #177, (R2)+ ; BACKUP
947 012466 032777 020000 176350 SCANS:    BEQ      SCAN4 ; IS IT FILL?
948 012474 001016      NOP      ; BR IF YES
949 012476 104400 012630      BIT     #BIT13,DSWR ; DATA ERROR
950 012502 013737 011020 012512      BNE     DERR ; INHIBIT PRINTOUTS
951 012510 104400      TYPE    MSG2 ; BR IF YES
952 012512 000000      MOV     IRDA, RDAX ; <15><12>RECEIVED DATA = <15><12>
953 012514 104400 012655      TYPE    MSG3 ; SETUP DATA ADRESS
954 012520 013737 011022 012530      MOV     IXDA, .+10 ; PRINT RECEIVED DATA
955 012526 104400      TYPE    MSG3 ; RECEIVED DATA ADDR.
956 012530 011022      MOV     IXDA ; <15><12>DATA SHOULD BE<15><12>
957 012532 111103      HLT+7   ; SETUP ADDR.
958 012534 114202      MOV     (R1),R3 ; SETUP XMIT DATA
959 012536 104007      MOV     -(R2),R2 ; SETUP RCV DATA
960 012540 005726      HLT+7   ; DATA ERROR HALT
961 012542 000207      TESTDX: TST     (SP)+ ; POP STACK
962
963 012544 005015 044124 051105 MSG0:    .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1) 012625 015 000012 MSG1:    .ASCIZ  <15><12>
(1) 012630 005015 042522 042503 MSG2:    .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
(1) 012655 015 042012 052101 MSG3:    .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
(1) 012700 005015 046120 040505 MSG4:    .ASCII  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1) 012747 015 053412 042510 .ASCIZ  <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
(1) 013032 005015 046120 040505 MSG5:    .ASCIZ  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)
(1) 013104 000000 .EVEN
964 013106 000000 SAVR0:  0
965 013110 000000 SAVR1:  0
966 013112 000000 SAVR2:  0
967 013114 000000 SAVR3:  0
968 013116 000000 SAVR4:  0
969 013120 000000 SAVR5:  0
970 013122 000000 DELAY:  0
971 013124 000000 BACK:   0
972
STOP:    0

```

```

973 ;*****
974 ; TRANSMITTER INITIALIZATION SUBROUTINE
975 ;*****
976
977 013126 005737 011024 STARTX: TST SETTLE
978 013132 001004 BNE 6$
979 013134 005737 013120 TST DELAY
980 013140 001015 BNE 5$
981 013142 000434 BR 1$
982 013144 005037 013710 6$: CLR TEMP1 ;PREPARE FOR DELAY
983 013150 012737 000007 013712 MOV #7,TEMP2
984 013156 062737 000001 013710 ADD #1,TEMP1 ;INC DELAY
985 013164 001374 BNE -6
986 013166 005337 013712 DEC TEMP2
987 013172 001371 BNE -14
988 013174 005037 013710 5$: CLR TEMP1 ;PREPARE FOR DELAY
989 013200 012737 000007 013712 MOV #7,TEMP2
990 013206 062737 000001 013710 ADD #1,TEMP1 ;INC DELAY
991 013214 001374 BNE -6
992 013216 005337 013712 DEC TEMP2
993 013222 001371 BNE -14
994 013224 005037 013120 CLR DELAY
995 013230 005037 011024 CLR SETTLE
996 013234 032737 040000 011014 1$: BIT #HALF.DUPLEX,PARAM2 ;HALF DUPLEX?
997 013242 001440 BEQ 4$ ;BR IF NO
998 013244 013746 000004 MOV #4,-(SP) ;SAVE LOC 4
999 013250 013746 000006 MOV #6,-(SP) ;SAVE LOC 6
1000 013254 012737 013332 000004 MOV #35,#4 ;SET UP TRAP CATCHER
1001 013262 005037 000006 CLR #6 ;CLEAR VECT+2
1002 013266 005737 013124 TST STOP ;FIRST TIME HERE?
1003 013272 001407 BEQ 8$ ;BR IF YES
1004 013274 012737 177777 013124 MOV #-1,STOP
1005 013302 032777 000100 000406 BIT #100,#DMBB ;CARRIER UP?
1006 013310 001374 BNE -6 ;BR IF YES
1007 013312 052777 000004 000376 8$: BIS #BIT2,#DMBB ;SET RQTS IN DMBB
1008 013320 032777 000040 000370 2$: BIT #BIT5,#DMBB ;SPIN ON CTS
1009 013326 001774 BEQ 2$
1010 013330 024646 CMP -(SP),-(SP) ;ADJUST STACK
1011 013332 022626 3$: CMP (SP)+,(SP)+ ;POP STACK
1012 013334 012637 000006 MOV (SP)+,#6 ;RESTORE LOC 6
1013 013340 012637 000004 MOV (SP)+,#4 ;RESTORE LOC 4
1014 013344 013737 011022 011070 4$: MOV IXDA,XDA ;SET UP XMIT DATA ADD
1015 013352 042700 100000 BIC #XFLG,STAT ;CLEAR XFLG
1016 013356 013764 011070 000006 MOV XDA,CAR(R4) ;LOAD CURRENT ADDRESS REG
1017 013364 032737 040000 011014 BIT #HALF.DUPLEX,PARAM2 ;HALF DUPLEX?
1018 013372 001022 BNE 7$ ;BR IF YES
1019 013374 032700 000004 BIT #XLB,MODE ;XLB MODE?
1020 013400 001417 BEQ 7$ ;BR IF NO
1021 013402 012737 177777 013722 MOV #1,TRNFLG ;SET SOFTWARE FLAG
1022 013410 012764 177777 000010 MOV #-1,BCR(R4)
1023 013416 052714 020000 BIS #TIE,#ACSR
1024 013422 013764 013714 000012 MOV BARTMP,BAR(R4)
1025 013430 000001 WAIT
1026 013432 005737 013720 TST SNCF LG ;HAS RECEIVER GOT FIRST CHAR?
1027 013436 001375 BNE -4 ;NO WAIT FOR IT
1028 013440 013764 011070 000006 7$: MOV XDA,CAR(R4) ;LOAD CURRENT ADDRESS REG

```

1029	013446	012764	177777	000010		MOV	#-1,BCR(R4)	:LOAD BYTE COUNT REG
1030	013454	052714	020000			BIS	#TIE,QRCSR	:SET INTERRUPT ENABLE
1031	013460	013764	013714	000012		MOV	BARTMP,BAR(R4)	:LOAD BAR REG
1032	013466	000207				RTS	PC	
1033								
1034	013470	042714	100000		XISR:	BIC	#TI,QRCSR	:CLEAR XMIT DONE
1035	013474	032714	002000			BIT	#NEM,QRCSR	:NON-EXISTENT MEM ERROR?
1036	013500	001407				BEQ	1\$:BR IF NO
1037	013502	011402				MOV	QRCSR,R2	:SAVE CSR FOR TYPE OUT
1038	013504	005003				CLR	R3	
1039	013506	104010				HLT	10	:ERROR HLT
1040	013510	104400	014317			TYPE	,NONEX	:TYPE ERROR MESS
1041	013514	000000				HALT		
1042	013516	000776				BR	-2	:BR HALT
1043	013520	127737	175344	011040	1\$:	CMPB	QXDA,TX.TERM	:IS CHAR TERMINATION CHAR?
1044	013526	001033				BNE	XISR1	:BR IF NO
1045	013530	052700	100000			BIS	#XFLG,STAT	:SET XMIT DONE FLAG
1046	013534	042714	020000			BIC	#TIE,QRCSR	:CLEAR INTERRUPT ENABLE
1047	013540	032737	040000	011014		BIT	#HALF.DUPLEX,PARAM2	:HALF DUPLEX?
1048	013546	001422				BEQ	3\$:BR IF NO
1049	013550	013746	000004			MOV	Q#4,-(SP)	:SAVE LOC 4
1050	013554	013746	000006			MOV	Q#6,-(SP)	:SAVE LOC 6
1051	013560	012737	013602	000004		MOV	Q#5,Q#4	:SET UP TRAP CATCHER
1052	013566	005037	000006			CLR	Q#6	:CLEAR VECT+2
1053	013572	042777	000004	000116		BIC	#BIT2,QDMBB	:CLEAR RQTS
1054	013600	024646				CMP	-(SP),-(SP)	:ADJUST STACK
1055	013602	022626			2\$:	CMP	(SP)+,(SP)+	:POP STACK
1056	013604	012637	000006			MOV	(SP)+,Q#6	:RESTORE LOC 6
1057	013610	012637	000004			MOV	(SP)+,Q#4	:RESTORE LOC 4
1058	013614	000430			3\$:	BR	XISR2	
1059	013616	032777	000100	175220	XISR1:	BIT	#100,QSWR	:MONITOR XMIT DATA?
1060	013624	001406				BEQ	NOXMON	:BR IF NO
1061	013626	105777	175244			TSTB	QTPS	:TTY READY?
1062	013632	100003				BPL	NOXMON	:BR IF NO
1063	013634	117777	175230	175236		MOVB	QXDA,QTPB	:TYPE CHAR
1064	013642	005237	011070		NOXMON:	INC	XDA	:INC TXBUF POINTER
1065	013646	013764	011070	000006		MOV	XDA,CAR(R4)	:LOAD CURRENT ADDRESS REG
1066	013654	005737	013722			TST	TRNFLG	:IS THIS FIRST TIME?
1067	013660	001006				BNE	XISR2	:BR IF YES
1068	013662	012764	177777	000010		MOV	#-1,BCR(R4)	:LOAD BYTE COUNT REG
1069	013670	013764	013714	000012		MOV	BARTMP,BAR(R4)	:SET BAR BIT
1070	013676	005037	011032		XISR2:	CLR	TIME	
1071	013702	005037	013722			CLR	TRNFLG	
1072	013706	000002				RTI		
1073	013710	000000			TEMP1:0			
1074	013712	000000			TEMP2:0			
1075	013714	000000			BARTMP:0			
1076	013716	170502			DMBB:170502			:LINE STATUS REG IN DMBB
1077	013720	000000			SNCFLG:0			
1078	013722	000000			TRNFLG:0			
1079	013724	177			FILL:	.BYTE 177		
1080		013726			.EVEN			

```

1081
1082
1083
1084
1085 013726 032737 040000 011014 STARTR: BIT #HALF.DUPLEX,PARAM2 ;HALF DUPLEX?
1086 013734 001010 BNE 2$ ;BR IF YES
1087 013736 032700 000004 BIT #XLB,MODE ;XLB MODE?
1088 013742 001405 BEQ 2$ ;BR IF NO
1089 013744 005037 013710 CLR TEMP1 ;START DELAY
1090 013750 005237 013710 1$: INC TEMP1
1091 013754 001375 BNE 1$
1092 013756 042700 040000 2$: BIC #RFLG,STAT ;CLEAR RFLG
1093 013762 013737 011020 011066 MOV IRDA,RDA ;SET UP RECEIVER DATA ADD
1094 013770 012737 001000 011064 MOV #1000,RCC ;SET UP BUFFER LIMIT
1095 013776 012737 177777 013720 MOV #-1,SNCFLG ;SET SOFTWARE FLAG
1096 014004 005037 011054 CLR ERCSR ;CLEAR ERROR RECORDS
1097 014010 005037 011056 CLR ERDBR
1098 014014 052714 004000 BIS #BIT11,RCRCSR ;MASIER CLEAR
1099 014020 053714 011012 BIS PARAM1,RCRCSR ;SET LINE NUMBER
1100 014024 053764 011014 000004 BIS PARAM2,LPR(R4) ;LINE PARAMETERS
1101 014032 052714 01010C BIS #RIE+SIE,RCRCSR ;SET INTERRUPT ENABLES
1102 014036 000207 RTS PC
1103
1104 014040 032714 040000 RISR: BIT #SI,RCRCSR ;SILO OVERFLOW?
1105 014044 001407 BNE 1$ ;BR IF NO
1106 014046 011402 MOV RCRCSR,R2 ;SAVE CSR FOR TYPEOUT
1107 014050 005003 CLR R3
1108 014052 104010 HLT 10 ;ERROR HLT
1109 014054 104400 014270 TYPE ,SILO ;TYPE ERROR MESS
1110 014060 000000 HALT
1111 014062 000776 BR -2 ;BR HALT
1112 014064 016401 000002 1$: MOV NCR(R4),R1 ;PUT CHAR IN R1
1113 014070 042701 000200 BIC #200,R1 ;STRIP A BIT
1114 014074 005701 TST R1 ;VALID DATA?
1115 014076 100403 BMI 4$ ;BR IF YES
1116 014100 011402 MOV RCRCSR,R2 ;SAVE CSR FOR TYPEOUT
1117 014102 005003 CLR R3
1118 014104 104010 HLT 10 ;ERROR HLT
1119 014106 032701 070000 4$: BIT #OO+FE+PE,R1 ;OVERRUN,FRAMING OR PARITY ERROR?
1120 014112 001404 BNE 3$ ;BR IF NO
1121 014114 011437 011054 MOV RCRCSR,ERCSR ;SAVE CSR
1122 014120 010137 011056 MOV R1,ERDBR ;SAVE CHAR
1123 014124 110177 174736 3$: MOV R1,IRDA ;STORE CHAR IN BUFFER
1124 014130 032777 000040 174706 BIT #BITS,SWR ;MONITOR RECEIVE DATA?
1125 014136 001405 BEQ NORMON ;BR IF NO
1126 014140 105777 174732 TSTB JTPS ;TTY READY?
1127 014144 100002 BPL NORMON ;BR IF NO
1128 014146 110177 174726 MOV R1,JTPB ;TYPE CHAR
1129 014152 005237 011066 NORMON: INC RDA ;INC RECEIVER BUFFER POINTER
1130 014156 105077 174704 CLRB JRDA ;CLEAR NEXT LOCATION
1131 014162 005337 011064 DEC RCC ;DEC CHAR COUNT
1132 014166 001005 BNE 1$ ;BR IF BUFFER NOT FULL
1133 014170 000005 RESET
1134 014172 104000 HLT 0 ;STOP THE SHOW,BUFFER OVERFLOWED!
1135 014174 104006 HLT+B ;RECEIVER BUFFER FULL
1136 014176 000000 HALT

```

1137	014200	000776	
1138	014202	123701	011041
1139	014206	001004	
1140	014210	042714	010100
1141	014214	052700	040000
1142	014220	005037	011032
1143	014224	005037	013720
1144	014230	000002	
1145	014232	005015	051105
	014270	005015	051105
	014317	015	047012
	014353	015	050012
		014434	
		000001	

```

1S: BR .-2 ;BR HALT
CMPB RX.TERM,R1 ;IS CHAR RCV TERMINATION CHAR
BNE RISR1 ;BR IF NOT
BIC #R1E+SIE,DRCSR ;CLEAR INTERRUPT ENABLES
BIS #RFLG,STAT ;SET RCV DONE FLAG
RISR1: CLR TIME ;CLEAR FLAG
CLR SMCFLG
RTI
MFULL: .ASCIZ<15><12>/ERROR! RECEIVER BUFFER FULL/
SILO: .ASCIZ<15><12>/ERROR! SILO OVERFLOW/
NONEX: .ASCIZ<15><12>/NON EXISTENT MEMORY ERROR/
LINES: .ASCIZ<15><12>/PLEASE SELECT ONLY ONE LINE AT A TIME(PARAM3)/
.EVEN
.END

```


TEMP1	013710	982*	984*	988*	990*	1073*	1089*	1090*								
TEMP2	013712	983*	986*	989*	992*	1074*										
TEST0	012344	731	794	875	918*											
TESTDX	012540	931	937	940	960*											
TI =	100000	596*	1034													
TIE =	020000	596*	989	890	1023	1030	1046									
TIME	011032	613*	686*	721	726*	750*	753	758*	782*	785	790*	811*	815	820*		
		842*	847	852*	863	868*	872*	1070*	1142*							
TKB	011074	645*														
TKS	011072	644*														
TPB	011100	647*	1063*	1128*												
TPS	011076	646*	1061	1126												
TRNFLG	013722	1021*	1066	1071*	1078*											
ISTDAT	012400	919	921	926*												
TX.TER	011040	616*	930	1043												
TYPE =	104400	596*	922	925	935	949	951	953	955	1040	1109					
XCC	011062	639*														
XCSR =	000000	596*	724	756	788	818	850	866	888	890*	908*					
XDA	011070	642*	804*	1014*	1016	1028	1043	1063	1064*	1065	1080*					
XFLG =	100000	628*	751	760	813	822	843	856	861	870	1015	1045				
XISR	013470	659	1034*													
XISR1	013616	1044	1059*													
XISR2	013676	1058	1067	1070*												
XLB =	000004	596*	698	1019	1087											
XWAIT =	104412	596*														
SILB	011524	697	780*	823												
SOWI	011320	694	717*	737												
SOWO	011426	691	748*	765												
SXLB	011754	700	837*	877	878											
.	= 014434	600*	668	702	806	954*	985	987	991	993	1006	1027	1042	1080*		
		1111	1137	1145*												

ABS. 014434 000

ERRORS DETECTED: 0

DSKZ: CZDHLC, DSKZ: CZDHLC, SEQ=DSKZ: ITP1.MAC, DSKZ: CZDHLC.P11

RUN-TIME: 3.4.2 SECONDS

RUN-TIME RATIO: 15/8=1.7

CORE USED: 16K (31 PAGES)

DOCUMENT PAGES: 27

