DH11

OVERLAY FOR INTERPRO TEST

MD-11-DZDHL-B

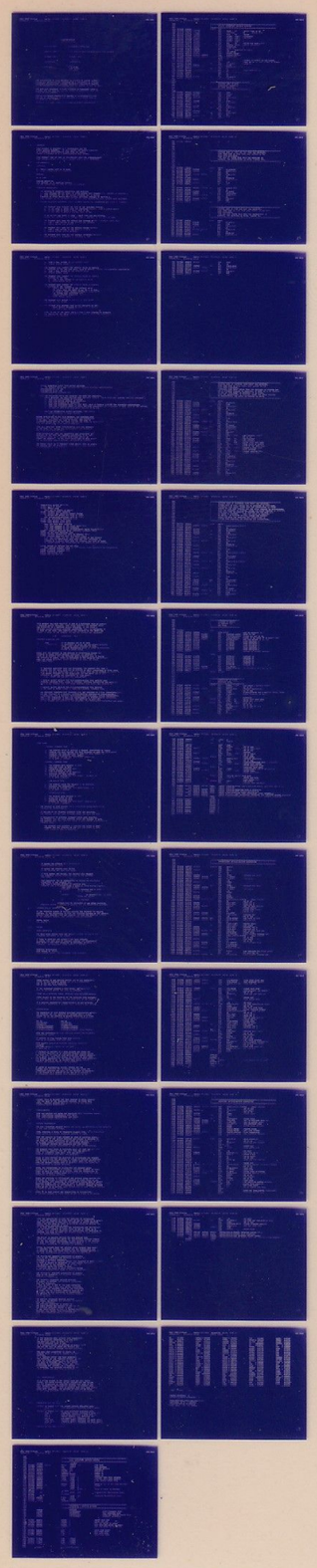
EP DZDHL B DL

JAN 1978

digital

FICHE 1 OF 1

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDHL-B-D
PRODUCT NAME:	DH11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM
PROGRAM DATE:	OCTOBER 1976
MAINTAINER:	DIAGNOSTICS
AUTHORS:	R A JONES JOHN FOOLF FAY BASHAW

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 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. REFFRED 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 61 IF THIS SETUP
WAS FOR DN11 OR DN11BB.

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

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

C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
    1. THE PROGRAM HALTS FOR ISR(INTERFACE SERVICE ROUTINE) SPECIFICATION
    SWR14=SETUP DM-11B ISR
    SWR13=SETUP DN-11 ISR
    SWR=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
    S=14 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 CONTEST/QUICK BROWN FOX/NUMERICS)
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
SW07=1 DO NOT TEST RECEIVED DATA
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
    * IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
    TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
    RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
    OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.

SW04=1 RETURN TO MONITOR FOR END PASS
    WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.
SW03=1 INTERNAL LOOPBACK MODE
SW02=1 EXTERNAL LOOPBACK MODE
SW01=1 ONE-WAY-IN MODE
SW00=1 ONE-WAY-OUT MODE

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE
CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G
(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 (CAP.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 <@@1>)
2. VERFIES THE DATA AGAINST THE DATA SELECTED BY SW@9 AND SW10(SW7=@)
3. TRANSMIT THE DATA SELECTED BY SW@9 AND SW10 (SW8=@) OR
   TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR
   GO TO STEP 1. (SW4=@)

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=@)
7. RETURNS TO MONITOR FOR "END PASS". (SW@4=1) OR
   GO TO STEP 1(SW@4=@)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SW@7=@)
3. RETURNS TO MONITOR FOR "END PASS"(SW@4=1) OR
   GO TO STEP 1 (SW@4=@)

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". (SW@4=1) OR
   GO TO STEP 1 (SW@4=@)

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/RECFIVE 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<001> IS NOT DETECTED
WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

7.0 RESTRICTIONS

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

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

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

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

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.
EXAMPLE:
SWITCHES 14,13,7,4 SHOULD BE THE SAME
ON BOTH CPU S

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

IF USING AN 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 DH11 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 PPOGRAM
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 $ILB: THE RECFIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR
RECEIVER TO FINISH, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC
IF NO ACTION. WHEN RECEIVER IS DONE PROGRAM CHECKS DATA IF SWITCH
SETTINGS PERMIT, AND END PASS IS TYPED IF SWITCH SETTINGS PERMIT.
THEN THE TRANSMITTER IS INITIALIZED, A "WAITING" MESSAGE IS TYPED
EVERY 60 SEC IF NO ACTION. WHEN TPANSMITTER IS DONE PROGRAM RETURNS
TO START OF ROUTINE. ($ILB)

9.34 $XLB: IF IN HALF DUPLEX THE TRANSMITTER IS INITIALIZED,
A "WAITING MESSAGE IS TYPFD 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 CHAPACTER WAS NOT RECOGNIZED

K1

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.

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 INITALIZED BY ITEP,THE PROGRAM WILL HANG IN THE
DH11 TRANSMITTER INITALIZATION 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.

575

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)

```
                                ;*****************************************************************
                                ;       DH11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000          .=11000
011000  044104  000040  DH11:   .ASCIZ  /DH /                   ;ISR NAME
011004  160020          BA:     160020                          ;BUS ADDRESS
011006  000300          RIV:    300                             ;VECTOR ADDRESS
011010  000240          PRIOR:  240                             ;PRIORITY
011012  000000          PARAM1: 0                               ;PARAM #1
011014  006307          PARAM2: 006307                          ;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
                                .WORD   0                       ;
011026  000000          B2016:  .WORD   0                       ;ADDR OF BIN TO OCT TYPE ROUTINE
011030  000000          TIME:   .WORD   0                       ;TIMER
011032  000000                  .WORD   0                       ;
011034  000000                  .WORD   START                   ;ADDR OF START OF PROGRAM
011036  011102
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
```

M1

```
                                ;*****************************************************************
                                ;       DH11-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  014060                  MOV     #PISR,  (R2)+   ;INTERRUPT
011124  013722  011010                  MOV     PRIOR,  (R2)+   ;VECTORS
011130  012722  013510                  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  000000                  MOV     R2,     XCSR(R4);IN CSR'S
011164  012714  004000                  MOV     #MC,@RCSR
011170  053714  011012                  BIS     PARAM1,@RCSR
011174  053764  011014  000004          BIS     PARAM2,LPR(R4)
011202  123727  011012  000017          CMPB    PARAM1,#17
011210  101402                          BLOS    1$
011212  000000                          HALT                    ;NUMBER IN PARAM1 IS TOO LARGE!
011214  000776                          BR      .-2             ;MUST BE 17 OR LESS (LINE# IN OCTAL)
011216  010046                  1$:     MOV     R0,-(SP)        ;SAVE R0
011220  012700  000001                  MOV     #1,R0
011224  013701  011012                  MOV     PARAM1,R1
011230  005701                  2$:     TST     R1
011232  001403                          BEQ     3$              ;CALCULATE BAP BIT
011234  006300                          ASL     R0
011236  005301                          DEC     R1
011240  000773                          BR      2$
011242  010037  013734          3$:     MOV     R0,BARTMP
011246  012600                          MOV     (SP)+,R0
                                ;*****************************************
                                ;       ROUTINE USED TO GOTO
                                ;       SUBROUTINE DEPENDENT
                                ;       ON MODE SELECTED.
                                ;*****************************************
011250  005037  011032          GO:     CLR     TIME
011254  005037  013140                  CLR     DELAY
011260  005037  013144                  CLR     STOP
011264  032700  000001                  BIT     #OWO,MODE
011270  001402                          BEQ     1$
011272  000137  011446                  JMP     $OWO
011276  032700  000002          1$:     BIT     #OWI,MODE
011302  001402                          BEQ     2$
011304  000137  011340                  JMP     $OWI
011310  032700  000010          2$:     BIT     #ILB,MODE
011314  001402                          BEQ     3$
011316  000137  011544                  JMP     $ILB
011322  032700  000004          3$:     BIT     #XLB,MODE
011326  001402                          BEQ     4$
011330  000137  011774                  JMP     $XLB
011334  000000                  4$:     HALT
```

A2

```
 705  011336  000776                          BR      .-2
 706
 707
 708
 709
 710                                  ;*****************************************************************
 711                                  ;       ROUTINE USED IF "ONE WAY IN" MODE WAS SELECTED.
 712                                  ;       NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE
 713                                  ;       ONLY MODE AVAILABLE.
 714                                  ;       "ONE WAY IN" MEANS THAT ONLY THE RECEIVER IS
 715                                  ;       ENABLED. THE TRANSMITTER IS NEVER "TURNED ON".
 716                                  ;*****************************************************************
 717
 718
 719
 720  011340  104416                  $OWI:   KBDIN
 721  011342  004737  013746                  JSR     PC,STARTR
 722  011346  032700  040000          1$:     BIT     #RFLG,STAT
 723  011352  001013                          BNE     2$
 724  011354  023727  011032  000100          CMP     TIME,#100
 725  011362  103771                          BLO     1$
 726  011364  011402                          MOV     @RCSR,R2
 727  011366  016403  000000                  MOV     XCSR(R4),R3
 728  011372  104001                          HLT     1
 729  011374  005037  011032                  CLR     TIME
 730  011400  000762                          BR      1$
 731
 732  011402  032777  000200  177434  2$:     BIT     #NODAT,@SWR
 733  011410  001002                          BNE     3$
 734  011412  004737  012364                  JSR     PC,TESTD
 735  011416  042700  040000          3$:     BIC     #RFLG,STAT
 736  011422  032777  000020  177414          BIT     #LOOP,@SWR
 737  011430  001405                          BEQ     4$
 738  011432  012737  011444  013142          MOV     #4$,BACK
 739  011440  000137  012224                  JMP     EOP
 740  011444  000735                  4$:     BR      $OWI
 741
 742
 743                                  ;*****************************************************************
 744                                  ;       POUTINE USED IF "ONE WAY OUT" WAS SELECTED.
 745                                  ;       NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE ONLY
 746                                  ;       MODE AVAILABLE.
 747                                  ;       "ONE WAY OUT" MEANS THAT ONLY THE TRANSMITTER IS
 748                                  ;       ENABLED. THE RECEIVER IS NEVER "TURNED ON.
 749                                  ;*****************************************************************
 750
 751  011446  104416                  $OWO:   KBDIN
 752  011450  004737  013146                  JSR     PC,STARTX
 753  011454  005037  011032                  CLR     TIME
 754  011460  032700  100000          1$:     BIT     #XFLG,STAT
 755  011464  001013                          BNE     2$
 756  011466  023727  011032  000100          CMP     TIME,#100
 757  011474  103771                          BLO     1$
 758  011476  011402                          MOV     @RCSR,R2
 759  011500  016403  000000                  MOV     XCSR(R4),R3
 760  011504  104001                          HLT     1
```

```
   761  011506  005037  011032                      CLR     TIME
   762  011512  000762                              BR      18
   763  011514  042700  100000              28:     BIC     #XFLG,STAT
   764  011520  032777  000020  177316              BIT     #LOOP,@#SWR
   765  011526  001405                              BEQ     38
   766  011530  012737  011542  013142              MOV     #38,BACK
   767  011536  000137  012224                      JMP     EOP
   768  011542  000741                      38:     BR      #OWO
   769
   770
   771
```

C2

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

011544  104416                  $ILB:   KRDIN
011546  004737  013746                  JSR     PC,STARTR
011552  005037  011032                  CLR     TIME
011556  032700  040000          1$:     BIT     #RFLG,STAT
011562  001013                          BNE     2$
011564  023727  011032  000100          CMP     TIME,#100
011572  103771                          BLO     1$
011574  011402                          MOV     #RCSR,R2
011576  016403  000000                  MOV     XCSR(R4),R3
011602  104001                          HLT     1
011604  005037  011032                  CLR     TIME
011610  000762                          BR      1$
011612  032777  000200  177224  2$:     BIT     #NODAT,#SWR
011620  001002                          BNE     3$
011622  004737  012364                  JSR     PC,TESTD
011626  042700  040000          3$:     BIC     #RFLG,STAT
011632  032777  000020  177204          BIT     #LOOP,#SWR
011640  001405                          BEQ     4$
011642  012737  011654  013142          MOV     #4$,BACK
011650  000137  012224                  JMP     EOP
011654  032777  000400  177162  4$:     BIT     #400,   #SWR    ;USE EXTERNAL DATA?
011662  001416                          BEQ     7$              ;BR IF NO
011664  013702  011020                  MOV     IRDA,   R2      ;SET POINTER
011670  013703  011022                  MOV     IXDA,   R3      ;SET POINTER
011674  010337  011070                  MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
011700  112223                          MOVB    (P2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
011702  001376                          BNE     .-2             ;LOOP IF NOT ZERO CHAR
011704  112743  000177                  MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
011710  005203                          INC     R3              ;BUMP ADDRESS
011712  112723  000177                  MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
011716  105023                          CLRB    (R3)+           ;INSERT ZERO CHAR
011720  005037  011032          7$:     CLR     TIME
011724  004737  013146                  JSR     PC,STARTX
011730  032700  100000          5$:     BIT     #XFLG,STAT
011734  001013                          BNE     6$
011736  023727  011032  000100          CMP     TIME,#100
011744  103771                          BLO     5$
011746  011402                          MOV     #RCSR,R2
011750  016403  000000                  MOV     XCSR(R4),R3
011754  104001                          HLT     1
011756  005037  011032                  CLR     TIME
011762  000762                          BR      5$
011764  042700  100000          6$:     BIC     #XFLG,STAT
011770  000137  011544                  JMP     $ILB
```

D2

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

011774  104416          $XLB:   KBDIN
011776  032737  040000  011014          BIT     #HALF,DUPLEX,PARA+2
012004  001002                        BNE     1$
012006  004737  013746                JSR     PC,STARTR
012012  004737  013146          1$:   JSR     PC,STARTX
012016  005037  011032                CLR     TIME
012022  032737  100000          2$:   BIT     #XFLG,STAT
012026  001016                        BNE     3$
012030  032737  040000          7$:   BIT     #RFLG,STAT
012034  001024                        BNE     4$
012036  023727  011032  000100        CMP     TIME,#100
012044  103766                        BLO     2$
012046  011402                        MOV     @RCSR,R2
012050  016403  000000                MOV     XCSR(R4),R3
012054  104001                        HLT     1
012056  005037  011032                CLR     TIME
012062  000757                        BR      2$
012064  032737  040000  011014  3$:   BIT     #HALF,DUPLEX,PARA+2
012072  001756                        BEQ     7$
012074  042737  100000                BIC     #XFLG,STAT
012100  004737  013746                JSR     PC,STARTR
012104  000746                        BR      2$
012106  032737  040000  011014  4$:   BIT     #HALF,DUPLEX,PARA+2
012114  001020                        BNE     8$
012116  032737  100000                BIT     #XFLG,STAT
012122  001013                        BNE     6$
012124  023727  011032  000100        CMP     TIME,#100
012132  103765                        BLO     4$
012134  011402                        MOV     @RCSR,R2
012136  016403  000000                MOV     XCSR(R4),R3
012142  104001                        HLT     1
012144  005037  011032                CLR     TIME
012150  000756                        BR      4$
012152  042737  100000          6$:   BIC     #XFLG,STAT
012156  042737  040000          8$:   BIC     #RFLG,STAT
012162  005037  011032                CLR     TIME
012166  032777  002000  176650        BIT     #NODAT,@SWR
012174  001002                        BNE     5$
012176  004737  012364                JSR     PC,TESTD
012202  032777  000020  176634  5$:   BIT     #LOOP,@SWR
012210  001671                        BEQ     $XLB
012212  012737  011774  013142        MOV     #$XLB,BACK
012220  000137  012224                JMP     EOP
```

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

012224                          EOP:
012224  104414  000340                  STPS,PPTY7              ;SET PS PRIORITY TO 7
012230  016437  000000  012362          MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012236  042737  157777  012362          BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012244  042764  020000  000000          BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012252  012766  012312  000002          MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012260  010037  013124                  MOV     R0,SAVR0        ;SAVE REGISTER 0
012264  010137  013126                  MOV     R1,SAVR1        ;SAVE REGISTER 1
012270  010237  013130                  MOV     R2,SAVR2        ;SAVE REGISTER 2
012274  010337  013132                  MOV     R3,SAVR3        ;SAVE REGISTER 3
012300  010437  013134                  MOV     R4,SAVR4        ;SAVE REGISTER 4
012304  010537  013136                  MOV     R5,SAVR5        ;SAVE REGISTER 5
012310  000207                          RTS     PC              ;RETURN TO CONTROL PROGRAM

012312                          ENTER:
012312  013700  013124                  MOV     SAVR0,R0        ;RESTORE R0
012316  013701  013126                  MOV     SAVR1,R1        ;RESTORE R1
012322  013702  013130                  MOV     SAVR2,R2        ;RESTORE R2
012326  013703  013132                  MOV     SAVR3,R3        ;RESTORE R3
012332  013704  013134                  MOV     SAVR4,R4        ;RESTORE R4
012336  013705  013136                  MOV     SAVR5,R5        ;RESTORE R5
012342  012737  177777  013140          MOV     #-1,DELAY
012350  053764  012362  000000          BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012356  000177  000560                  JMP     @BACK
012362  000000                  QTPIE:  000000

                                ;*************************************
                                ;       SUBROUTINE TO CHECK
                                ;       RECEIVER DATA.
                                ;*************************************
012364  013746  011056          TESTD:  MOV     ERDBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012370  001413                          BEQ     TSTDAT          ;BR IF NO
012372  032777  020000  176444          BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012400  001007                          BNE     TSTDAT          ;BR IF YES
012402  104400  012564                  TYPE    ,^MSG0          ;<15><12>THERE WAS A RECEIVE ERROR, RBUF=
012406  004077  176416                  JSR     R0,@B2016       ;PRINT CONTENTS OF RBUF
012412  005746                          TST     -(SP)
012414  104400  012645                  TYPF    ,^MSG1          ;<15><12>
012420  013701  011022          TSTDAT: MOV     IXDA,   R1      ;SETUP XMIT DATA ADDR
012424  013702  011020                  MOV     IRDA,   R2      ;SETUP RCV DATA ADDR
012430  122122                  SCAN4:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012432  001776                          BEQ     SCAN4           ;BR IF OK
012434  123741  011040                  CMPB    TX,TERM,-(R1)   ;IS IT END OF DATA
012440  001447                          BEQ     TESTDX          ;BR IF YES
012442  122742  000002                  CMPB    #002,-(R2)
012446  001005                          BNE     2$
012450  010237  012456                  MOV     R2,1$
012454  104400                          TYPE
```

F2

```
939  012456  000000                  18:     .WORD   0
940  012460  000437                          BR      TESTDX
941  012462                          28:
942  012462  105712                          TSTB    (R2)            ;
943  012464  001435                          BEQ     TESTDX          ;BR IF YES
944  012466  122721  000177                  CMPB    #177,   (R1)+   ;IS IT FILL CHAR?
945  012472  001756                          BEQ     SCAN4           ;BR IF YES
946  012474  005301                          DEC     R1              ;BACKUP
947  012476  122722  000177                  CMPB    #177,   (R2)+   ;IS IT FILL?
948  012502  001752                          BEQ     SCAN4           ;BR IF YES
949  012504  000240                  SCAN5:  NOP                     ;DATA ERROR
950  012506  032777  020000  176330          BIT     #BIT13,@#SWR    ;INHIBIT PRINTOUTS
951  012514  001016                          BNE     DERR            ;BR IF YES
952  012516  104400  012650                  TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
953  012522  013737  011020  012532          MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
954  012530  104400                          TYPE                    ;PRINT RECEIVED DATA
955  012532  000000                  RDAX:   0                       ;RECEIVED DATA ADDR.
956  012534  104400  012675                  TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
957  012540  013737  011022  012550          MOV     IXDA,   .+10    ;SETUP ADDR.
958  012546  104400                          TYPE                    ;PRINT GOOD DATA
959  012550  011022                          IXDA
960  012552  111103                  DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
961  012554  114202                          MOVB    -(R2),R2        ;SETUP RCV DATA
962  012556  104007                          HLT+7                   ;DATA ERROR HALT
963  012560  005726                  TESTDX: TST     (SP)+           ;POP STACK
964  012562  000207                          RTS     PC              ;RETURN FROM SUB/ROUT
965
966  012564  005015  044124  051105  MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR, REGISTER (SEL 2) =/
(1)  012645     015  000012          MSG1:   .ASCIZ  <15><12>
(1)  012650  005015  042522  042503  MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
(1)  012675     015  042012  052101  MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
(1)  012720  005015  046120  040505  MSG4:   .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER),/
(1)  012767     015  053412  042510          .ASCIZ <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH,/<15><12>
(1)  013052  005015  046120  040505  MSG5:   .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER),/<15><12>
(1)                                  .EVEN
(1)  013124  000000          SAVR0:  0
967  013126  000000          SAVR1:  0
968  013130  000000          SAVR2:  0
969  013132  000000          SAVR3:  0
970  013134  000000          SAVR4:  0
971  013136  000000          SAVR5:  0
972  013140  000000          DELAY:  0
973  013142  000000          BACK:   0
974  013144  000000          STOP:   0
975
```

```
                        ;*****************************************************************
                        ;       TRANSMITTER INITIALIZATION SUBROUTINE
                        ;*****************************************************************

013146  005737  011024        STARTX: TST     SETTLE
013152  001404                        BNE     68
013154  005737  013140                TST     DELAY
013160  001015                        BNE     58
013162  000434                        BR      18
013164  005037  013730        68:     CLR     TEMP1           ;PREPARE FOR DELAY
013170  012737  000007  013732        MOV     #7,TEMP2
013176  062737  000001  013730        ADD     #1,TEMP1        ;INC DELAY
013204  001374                        BNE     .-6
013206  005337  013732                DEC     TEMP2
013212  001371                        BNE     .-14
013214  005037  013730        58:     CLR     TEMP1           ;PREPARE FOR DELAY
013220  012737  000007  013732        MOV     #7,TEMP2
013226  062737  000001  013730        ADD     #1,TEMP1        ;INC DELAY
013234  001374                        BNE     .-6
013236  005337  013732                DEC     TEMP2
013242  001371                        BNE     .-14
013244  005037  013140                CLR     DELAY
013250  005037  011024                CLR     SETTLE
013254  032737  040000  011014  18:     BIT     #HALF.DUPLEX,PARAM2     ;HALF DUPLEX?
013262  001440                        BEQ     48              ;BR IF NO
013264  013746  000004                MOV     @#4,-(SP)       ;SAVE LOC 4
013270  013746  000006                MOV     @#6,-(SP)       ;SAVE LOC 6
013274  012737  013352  000004        MOV     #38,@#4         ;SET UP TRAP CATCHER
013302  005037  000006                CLR     @#6             ;CLEAR VECT+2
013306  005737  013144                TST     STOP            ;FIRST TIME HERE?
013312  001407                        BEQ     88              ;BR IF YES
013314  012737  177777  013144        MOV     #-1,STOP
013322  032777  000100  000406        BIT     #100,@DMBB      ;CARRIER UP?
013330  001374                        BNE     .-6             ;BR IF YES
013332  052777  000004  000376  88:     BIS     #BIT2,@DMBB     ;SET RQTS IN DMBB
013340  032777  000040  000370  28:     BIT     #BIT5,@DMBB     ;SPIN ON CTS
013346  001774                        BEQ     28
013350  024646                        CMP     -(SP),-(SP)     ;ADJUST STACK
013352  022626                38:     CMP     (SP)+,(SP)+     ;POP STACK
013354  012637  000006                MOV     (SP)+,@#6       ;RESTORE LOC 6
013360  012637  000004                MOV     (SP)+,@#4       ;RESTORE LOC 4
013364  013737  011022  011070  48:     MOV     IXDA,XDA        ;SET UP XMIT DATA ADD
013372  042700  100000                BIC     #XFLG,STAT      ;CLEAR XFLG
013376  013764  011070  000006        MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
013404  032737  040000  011014        BIT     #HALF.DUPLEX,PARAM2     ;HALF DUPLEX?
013412  001022                        BNE     78              ;BR IF YES
013414  032700  000004                BIT     #XLB,MODE       ;XLB MODE?
013420  001417                        BEQ     78              ;BR IF NO
013422  012737  177777  013742        MOV     #-1,TRNFLG      ;SET SOFTWARE FLAG
013430  012764  177777  000010        MOV     #-1,BCP(R4)
013436  052714  020000                BIS     #TIE,@RCSR
013442  013764  013734  000012        MOV     BARTMP,BAR(R4)
013450  000001                        WAIT
013452  005737  013740                TST     SNCFLG          ;HAS RECEIVER GOT FIRST CHAR?
013456  001375                        BNE     .-4             ;NO WAIT FOR IT
013460  013764  011070  000006  78:     MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
```

#2

```
1032  013466  012764  177777  000010          MOV     #-1,BCR(R4)     ;LOAD BYTE COUNT REG
1033  013474  052714  020000                  BIS     #TIE,@RCSR      ;SET INTERUPT ENABLE
1034  013500  013764  013734  000012          MOV     BARTMP,BAR(R4)  ;LOAD BAR REG
1035  013506  000207                          RTS     PC
1036
1037  013510  042714  100000          XISP:   BIC     #TI,@RCSR       ;CLEAR XMIT DONE
1038  013514  032714  002000                  BIT     #NEM,@RCSR      ;NON-EXISTENT MEM ERROR?
1039  013520  001407                          BEQ     1$              ;BR IF NO
1040  013522  011402                          MOV     @RCSR,R2        ;SAVE CSR FOR TYPE OUT
1041  013524  005003                          CLR     R3
1042  013526  104010                          HLT     10              ;ERROR HLT
1043  013530  104400  014337                  TYPE    ,NONEX          ;TYPE ERROR MESS
1044  013534  000000                          HALT
1045  013536  000776                          BR      .-2             ;BR HALT
1046  013540  127737  175324  011040  1$:     CMPB    @XDA,TX,TERM    ;IS CHAR TERMINATION CHAR?
1047  013546  001033                          BNE     XISP1           ;BR IF NO
1048  013550  052737  100000                  BIS     #XFLG,STAT      ;SET XMIT DONE FLAG
1049  013554  042714  020000                  BIC     #TIE,@RCSR      ;CLEAR INTERUPT ENABLE
1050  013560  032737  040000  011014          BIT     #HALF,DUPLEX,PARAM2     ;HALF DUPLEX?
1051  013566  001422                          BEQ     3$              ;BR IF NO
1052  013570  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
1053  013574  013746  000006                  MOV     @#6,-(SP)       ;SAVE LOC 6
1054  013600  012737  013622  000004          MOV     #2$,@#4         ;SET UP TRAP CATCHER
1055  013606  005037  000006                  CLR     @#6             ;CLEAR VECT+2
1056  013612  042777  000004  000116          BIC     #BIT2,@DMRB     ;CLEAR RQTS
1057  013620  024646                          CMP     -(SP),-(SP)     ;ADJUST STACK
1058  013622  022626                  2$:     CMP     (SP)+,(SP)+     ;POP STACK
1059  013624  012637  000006                  MOV     (SP)+,@#6       ;RESTORE LOC 6
1060  013630  012637  000004                  MOV     (SP)+,@#4       ;RESTORE LOC 4
1061  013634  000430                  3$:     BR      XISP2
1062  013636  032777  000100  175200  XISP1:  BIT     #100,@SWP       ;MONITOR XMIT DATA?
1063  013644  001406                          BEQ     NOXMON          ;BR IF NO
1064  013646  105777  175224                  TSTB    @TPS            ;TTY READY?
1065  013652  100003                          BPL     NOXMON          ;BR IF NO
1066  013654  117777  175210  175216          MOVB    @XDA,@TPB       ;TYPE CHAR
1067  013662  005237  011070          NOXMON: INC     XDA             ;INC TXBUF POINTER
1068  013666  013764  011070  000006          MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
1069  013674  005737  013742                  TST     TRNFLG          ;IS THIS FIRST TIME?
1070  013700  001006                          BNE     XISR2           ;BR IF YES
1071  013702  012764  177777  000010          MOV     #-1,BCR(R4)     ;LOAD BYTE COUNT REG
1072  013710  013764  013734  000017          MOV     BARTMP,BAR(R4)  ;SET BAR BIT
1073  013716  005037  011032          XISP2:  CLR     TIME
1074  013722  005037  013742                  CLR     TRNFLG
1075  013726  000002                          RTI
1076  013730  000000                  TEMP1:  0
1077  013732  000000                  TEMP2:  0
1078  013734  000000                  BARTMP: 0
1079  013736  170502                  DMBB:   170502                  ;LINE STATUS REG IN DMBB
1080  013740  000000                  SNCFLG: 0
1081  013742  000000                  TRNFLG: 0
1082  013744     177                  FILL:   .BYTE   177
1083        013746                    .EVEN
```

```
1084                                    ;*****************************************************************
1085                                    ;       RECEIVER INITIALIZATION SUBROUTINE
1086                                    ;*****************************************************************
1087
1088  013746  032737  040000  011014  STARTR: BIT     #HALF,DUPLEX,PARAM2     ;HALF DUPLEX?
1089  013754  001010                          BNE     2$                      ;BR IF YES
1090  013756  032700  000004                  BIT     #XLB,MODE               ;XLB MODE?
1091  013762  001405                          BEQ     2$                      ;BR IF NO
1092  013764  005037  013730                  CLR     TEMP1                   ;START DELAY
1093  013770  005237  013730          1$:     INC     TEMP1
1094  013774  001375                          BNE     1$
1095  013776  042700  040000          2$:     BIC     #RFLG,STAT              ;CLEAR RFLG
1096  014002  012737  011020  011066          MOV     IRDA,RDA                ;SET UP RECEIVER DATA ADD
1097  014010  012737  001000  011064          MOV     #1000,RCC               ;SET UP BUFFER LIMIT
1098  014016  012737  177777  013740          MOV     #-1,SNCFLG              ;SET SOFTWARE FLAG
1099  014024  005037  011054                  CLR     ERCSR                   ;CLEAR ERROR RECORDS
1100  014030  005037  011056                  CLR     ERDBR
1101  014034  052714  004000                  BIS     #BIT11,@RCSR            ;MASTER CLEAR
1102  014040  053714  011012                  BIS     PARAM1,@RCSR            ;SET LINE NUMBER
1103  014044  053764  011014  000004          BIS     PARAM2,LPR(R4)          ;LINE PARAMETERS
1104  014052  052714  010100                  BIS     #RIE+SIE,@RCSR          ;SET INTERUPT ENABLES
1105  014056  000207                          RTS     PC
1106
1107  014060  032714  040000          RISR:   BIT     #SI,@RCSR               ;SILO OVERFLOW?
1108  014064  001407                          BEQ     1$                      ;BR IF NO
1109  014066  011402                          MOV     @RCSR,R2                ;SAVE CSR FOR TYPEOUT
1110  014070  005003                          CLR     R3
1111  014072  104010                          HLT     10                      ;ERROR HLT
1112  014074  104403  014310                  TYPE    ,SILO                   ;TYPE ERROR MESS
1113  014100  000003                          HALT
1114  014102  000776                          BR      .-2                     ;BR HALT
1115  014104  016401  000002          1$:     MOV     NRCR(R4),R1             ;PUT CHAR IN R1
1116  014110  042701  000200                  BIC     #200,R1                 ;STRIP A BIT
1117  014114  005701                          TST     R1                      ;VALID DATA?
1118  014116  100403                          BMI     4$                      ;BR IF YES
1119  014120  011402                          MOV     @PCSR,R2                ;SAVE CSR FOR TYPEOUT
1120  014122  005003                          CLR     R3
1121  014124  104010                          HLT     10                      ;ERROR HLT
1122  014126  032701  070000          4$:     BIT     #DO+FE+PE,R1            ;OVERRUN,FRAMING OR PARITY ERROR?
1123  014132  001404                          BEQ     3$                      ;BR IF NO
1124  014134  011437  011054                  MOV     @RCSR,ERCSR             ;SAVE CSR
1125  014140  010137  011056                  MOV     R1,ERDBR                ;SAVE CHAR
1126  014144  110177  174716          3$:     MOVB    R1,@RDA                 ;STORE CHAR IN BUFFER
1127  014150  032777  000040  174666          BIT     #BIT5,@SWR              ;MONITOR RECEIVE DATA?
1128  014156  001405                          BEQ     NORMON                  ;BR IF NO
1129  014160  105777  174712                  TSTB    @TPS                    ;TTY READY?
1130  014164  100002                          BPL     NORMON                  ;BR IF NO
1131  014166  110177  174706                  MOVB    R1,@TPB                 ;TYPE CHAR
1132  014172  005237  011066          NORMON: INC     RDA                     ;INC RECEIVER BUFFER POINTER
1133  014176  105077  174664                  CLRB    @RDA                    ;CLEAR NEXT LOCATION
1134  014202  005337  011064                  DEC     RCC                     ;DEC CHAR COUNT
1135  014206  001005                          BNE     1$                      ;BR IF BUFFER NOT FULL
1136  014210  000005                          RESET
1137  014212  104000                          HLT     0                       ;STOP THE SHOW,BUFFER OVERFLOWED!
1138  014214  104006                          HLT+6                           ;RECEIVER BUFFER FULL
1139  014216  000000                          HALT
```

```
  1140  014220  000776                          BR      .-2             ;BR HALT
  1141  014222  123701  011041          18:     CMPB    RX,TERM,R1      ;IS CHAR RCV TERMINATION CHAR
  1142  014226  001004                          BNE     RISR1           ;BR IF NOT
  1143  014230  042714  010100                  BIC     #RIE+SIE,@RCSR  ;CLEAR INTERUPT ENABLES
  1144  014234  052700  040000                  BIS     #RFLG,STAT      ;SET RCV DONE FLAG
  1145  014240  005037  011032          RISR1:  CLR     TIME
  1146  014244  005037  013740                  CLR     SNCFLG          ;CLEAR FLAG
  1147  014250  000002                          RTI
  1148  014252  005015  051105  047522  MFULL:  .ASCIZ<15><12>/ERROR: RECEIVER BUFFER FULL/
        014310  005015  051105  047522  SILO:   .ASCIZ<15><12>/ERROR: SILO OVERFLOW/
        014337     015  047012  047117  NONEX:  .ASCIZ<15><12>/NON EXISTENT MEMORY ERROR/
        014373     015  050012  042514  LINES:  .ASCIZ<15><12>/PLEASE SELECT ONLY ONE LINE AT A TIME(PARAM3)/
                014454                  .EVEN
                000001                  .END
```

K2

```
BA      011004
BACK    013142
BAR   = 000012
BARTMP  013734
BCR   = 000010
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
B2016   011030
CAR   = 000006
DELAY   013140
DERR    012552
DH11    011000
DISPLA  011046
DMBB    013736
DO    = 040000
DSFLG = 020000
DSSTAT  011060
ENTER   012312
EOP     012224
ERCSR   011054
ERDBR   011056
FE    = 020000
FILL    013744
FLAG    011042
FULL.D= 000001
GO      011250
HALF.D= 040000
ILB   = 000010
IPDA    011020
IXDA    011022
KRDIN = 104416
LINES   014373
LOOP  = 000020
LPR   = 000004
MC    = 004000
MFULL   014252
MODE  =%000000
MSG0    012564
MSG1    012645
MSG2    012650
MSG3    012675
MSG4    012720
MSG5    013052
NEM   = 002000
NODAT = 000200
NONEX   014337
NORMON  014172
NOXMON  013662
NRCR  = 000002
OWI   = 000002
OWO   = 000001
PARAM1  011012
PARAM2  011014
PARAM3  011016
PE    = 010000
PRIOR   011010
PRTY0 = 000000
PRTY1 = 000040
PRTY2 = 000100
PRTY3 = 000140
PRTY4 = 000200
PRTY5 = 000240
PRTY6 = 000300
PRTY7 = 000340
QTPIE   012362
RCC     011064
RCSR  =%000004
RDA     011066
RDAX    012532
RFLG  = 040000
RIE   = 000100
RISR    014060
RISP1   014240
RIV     011006
RWAIT = 104410
RX.TER  011041
R6    =%000006
R7    =%000007
SAVR0   013124
SAVR1   013126
SAVR2   013130
SAVR3   013132
SAVR4   013134
SAVR5   013136
SCAN4   012430
SCAN5   012504
SETTLE  011024
SI    = 040000
SIE   = 010000
SILO    014310
SNCFLG  013740
SRCSR   011052
SSP   = 000016
START   011102
STARTP  013746
STARTX  013146
STAT  =%000000
STOP    013144
STPS  = 104414
SWR     011044
SW12  = 010000
SXCSR   011050
TEMP1   013730
TEMP2   013732
TESTD   012364
TESTDX  012560
TI    = 100000
TIE   = 020000
TIME    011032
TKB     011074
TKS     011072
TPB     011100
TPS     011076
TRNFLG  013742
TSTDAT  012420
TX.TER  011040
TYPE  = 104400
XCC     011062
XCSR  = 000000
XDA     011070
XFLG  = 100000
XISR    013510
XISR1   013636
XISR2   013716
XLB   = 000004
XWAIT = 104412
$ILB    011544
$OWI    011340
$OWO    011446
$XLB    011774
.     = 014454
```

```
. ABS.  014454     000
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

DHLB,DHLB/SOL_ITEP1,MAC,DZDHLB,P11
RUN-TIME:  10 13  .3 SECONDS
RUN-TIME RATIO:  66/25=2.5
CORE USED:  16K  (31 PAGES)
```

L2