TSDHIO -- TSX-Plus DH11 & DHV11 MACRO V05.04 [illegible] 18-Dec-87 07:57
Table of contents

```
                                        .TITLE  TSDHIO -- TSX-Plus DH11 & DHV11 Mux support
                                        .PSECT  TSDHIO
                                TSDHIO:
                                        .ENABL  LC
                                        .ENABL  AMA
                                        .DSABL  GBL
                                ;-------------------------------------------------------------------------------
                                ;  TSDHIO contains the resident TSX-Plus routine to handle I/O to
                                ;  terminals that are connected to DH11 and DHV11 communications lines.
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  TSDHIO
                                        .GLOBL  DHOINT,DHSTRT,DHSTOP
                                        .GLOBL  VHOINT,VHSTRT,VHSTOP
                                        .GLOBL  DHXOFF,VHXOFF,DHXON,VHXON
                                        .GLOBL  DHTIMR
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  MH$BAR,HF$LIN,MH$SCR,MH$BCR,MH$CAR,DHBFSZ
                                        .GLOBL  LOUTIR,PSW,INTPRI,MH$PBR,VF$LIN,VH$CSR,NEDCDO
                                        .GLOBL  VH$BA1,VH$BA2,VH$BCR,VF$TEN,VF$TGO,VH$LCR,LMXLN
                                        .GLOBL  MXLNT,$CTRLS,LSW3,LSW5,$XCHAR,LMXNUM,$HISTP
                                        .GLOBL  VF$ABT,VF$TDV,VH$DBR,$DHBF1,$DHBF2,LSW10,$DHCDO
                                        .GLOBL  LDHB1B,LDHB1P,LDHB2B,LDHB2S,LDHB2R,LCDTYP,CDSTR2
                                        .GLOBL  $DHXOF,$DHXON,VF$XOF,VF$RIE,HF$RIE,HF$TI
                                ;
                                ;-------------------------------------------------------------------------------
                                ;  Macro definitions
                                ;
                                ;  Disable interrupts
                                ;
                                        .MACRO  DISABL          ;Disable interrupts
                                        BIS     #340,@#PSW
                                        .ENDM   DISABL
                                ;
                                ;  Enable interrupts
                                ;
                                        .MACRO  ENABL           ;Enable interrupts
                                        BIC     INTPRI,@#PSW
                                        .ENDM   ENABL
                                ;
                                ;-------------------------------------------------------------------------------
                                ;  Data areas
                                ;
                                ;  Table used to select lines within a DH11 group
                                ;
000000  000001  000002  000004  LINBIT: .WORD   1,2,4,10,20,40,100,200,400
000006  000010  000020  000040
000014  000100  000200  000400
000022  001000  002000  004000          .WORD   1000,2000,4000,10000,20000,40000,100000
000030  010000  020000  040000
000036  100000
                                ;
                                ; Misc data
                                ;
```

```
     54 000040      023                 CTLSBF: .BYTE   23              ;Control-S DMA buffer
     55 000041      021                 CTLQBF: .BYTE   21              ;Control-Q DMA buffer
     56                                         .EVEN
```

```
                                        .SBTTL  DHOINT -- Output interrupt from DH11 line
                                ;-----------------------------------------------------------------------
                                ;  Process an output interrupt from a DH11 multiplexer.
                                ;  The interrupt is vectored directly to an instruction sequence of the
                                ;  form:
                                ;
                                ;       JSR     R5,@#DHOINT
                                ;       .WORD   mux_number
                                ;
                                ;  Thus on entry to DHOINT, R5 has been saved on the stack and currently
                                ;  points to a word containing the mux index number.
                                ;
000042  010046                  DHOINT: MOV     R0,-(SP)
                                ;
                                ;  Set the running interrupt priority level to 7
                                ;  (The hardware priority level is currently 7)
                                ;
000044  013746  0000000                 MOV     INTPRI,-(SP)    ;Save current priority level
000050  005037  0000000                 CLR     INTPRI          ;Say current priority is 7
                                ;
                                ;  Obtain the mux index number
                                ;
000054  011505                          MOV     (R5),R5         ;Get the mux index number
                                ;
                                ;  Call routine which restarts any transmitters that are ready for more output
                                ;
000056  004737  000074'                 CALL    DHOCHK          ;Check for needing more output
                                ;
                                ;  Finished
                                ;
000062  012637  0000000                 MOV     (SP)+,INTPRI    ;Restore interrupt priority level
000066  012600                          MOV     (SP)+,R0
000070  012605                          MOV     (SP)+,R5
000072  000002                          RTI                     ;Return from interrupt
```

```
                                        .SBTTL  DHOCHK -- Check for DH11 lines that need more output
                                ;-------------------------------------------------------------------
                                ;  DHOCHK is called to determine which (if any) DH11 lines are ready for
                                ;  more output and restart those lines.
                                ;
                                ;  Inputs:
                                ;   R5 = DH11 mux index number.
                                ;
000074  010146                 DHOCHK: MOV     R1,-(SP)
000076  010246                         MOV     R2,-(SP)
000100  010346                         MOV     R3,-(SP)
                                ;
                                ;  Get contents of DH11 Buffer Active Register and determine which
                                ;  line(s) are ready for more output.
                                ;
000102                          3$:     DISABL                          ;;;** Disable interrupts **
000110  042775  0000000 0000000         BIC     #HF$TI,@MH$SCR(R5);;;Clear transmitter interrupt flag
000116  016500  0000000                 MOV     MH$PBR(R5),R0   ;;;Get previous contents of BAR
000122  017502  0000000                 MOV     @MH$BAR(R5),R2  ;;;Get current buffer active flags
000126  010265  0000000                 MOV     R2,MH$PBR(R5)   ;;;Save new "previous BAR value"
000132                                  ENABL                   ;** Enable interrupts **
000140  005102                          COM     R2              ;Set flags for all inactive lines
000142  005100                          COM     R0              ;Clear flags for lines that were active
000144  040002                          BIC     R0,R2           ;Get flags for lines that finished output
000146  001422                          BEQ     9$              ;Br if no lines need service
                                ;
                                ;  At this point, R2 contains a flag bit for each of the 16 lines.
                                ;  The corresponding flag is set (1) if the line completed output since
                                ;  the last time this routine was called.
                                ;
                                ;  Begin loop to start new output for each line that finished transmission.
                                ;
000150  005003                          CLR     R3              ;Init index # of line within DH11 mux group
000152  000241                  1$:     CLC                     ;Clear carry flag
000154  006002                          ROR     R2              ;Does this line need service?
000156  103012                          BCC     2$              ;Br if not
                                ;
                                ;  We have found a line that has completed its output.
                                ;  Get TSX-Plus line index number.
                                ;
000160  010301                          MOV     R3,R1           ;Get # of line within mux group
000162  066501  0000000                 ADD     MXLNT(R5),R1    ;Point into correct mux line # table
000166  111101                          MOVB    (R1),R1         ;Get the physical line #
000170  001405                          BEQ     2$              ;Br if not genned into system
000172  042761  0000000 0000000         BIC     #$XCHAR,LSW3(R1);Say transmitter not busy for line
                                ;
                                ;  Try to start more output for this line
                                ;
000200  004737  000224'                 CALL    DHSTRT          ;Try to start output for that line
                                ;
                                ;  Check the next line
                                ;
000204  005203                  2$:     INC     R3              ;Advance line #
000206  005702                          TST     R2              ;Any more lines to check?
000210  001360                          BNE     1$              ;Loop if yes
000212  000733                          BR      3$              ;See if more lines need service
                                ;
```

```
                                        ;  Finished
                                        ;
000214  012603                  9$:       MOV     (SP)+,R3
000216  012602                            MOV     (SP)+,R2
000220  012601                            MOV     (SP)+,R1
000222  000207                            RETURN
```

```
                                        .SBTTL  DHSTRT -- Start output to a DH11 line
                                ;-------------------------------------------------------------
                                ;  DHSTRT is called to initiate transmission to a DH11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number
                                ;
000224  010146                  DHSTRT: MOV     R1,-(SP)
000226  010246                          MOV     R2,-(SP)
000230  010346                          MOV     R3,-(SP)
000232  010546                          MOV     R5,-(SP)
                                ;
                                ;  Return if transmission to the line is in progress
                                ;
000234                                  DISABL                  ;;;** Disable interrupts **
000242  032761  000000G 000000G         BIT     #$XCHAR,LSW3(R1);;;Is a transmission in progress now?
000250  001040                          BNE     9$              ;;;Br if yes
                                ;
                                ;  Line is currently idle.
                                ;  See if there is data ready to be transmitted.
                                ;
000252  052761  000000G 000000G         BIS     #$XCHAR,LSW3(R1);;;Say line is busy
000260  004737  001032'                  CALL    DHSWBF          ;;;Try to get data to transmit
000264  103427                          BCS     1$              ;;;Br if nothing to transmit
                                ;
                                ;  There is data to be transmitted.
                                ;   R0 = Address of DMA buffer.
                                ;   R2 = Number of characters in buffer.
                                ;
000266  016105  000000G                 MOV     LMXNUM(R1),R5   ;;;Get the mux index number
000272  116103  000000G                 MOVB    LMXLN(R1),R3    ;;;Get # of line within mux group (0-15)
000276  052703  000000G                 BIS     #HF$RIE,R3      ;;;Set receiver interrupt enable flag
000302  110375  000000G                 MOVB    R3,@MH$SCR(R5)  ;;;Select line of interest
000306  005402                          NEG     R2              ;;;Get negative # bytes to transmit
000310  010275  000000G                 MOV     R2,@MH$BCR(R5)  ;;;Set negative byte count
000314  010075  000000G                 MOV     R0,@MH$CAR(R5)  ;;;Set address of DMA buffer
000320  042703  000000G                 BIC     #HF$RIE,R3      ;;;Clear receiver interrupt enable flag
000324  006303                          ASL     R3              ;;;Convert line # to word table index
000326  056365  000000' 000000G         BIS     LINBIT(R3),MH$PBR(R5);;;Set flag in shadow cell
000334  056375  000000' 000000G         BIS     LINBIT(R3),@MH$BAR(R5);;;Start transmitter for the line
000342  000403                          BR      9$              ;;;
                                ;
                                ;  There are no more characters to transmit
                                ;
000344  042761  000000G 000000G 1$:     BIC     #$XCHAR,LSW3(R1);;;Say transmitter is not busy
                                ;
                                ;  Finished
                                ;
000352                          9$:     ENABL                   ;** Enable interrupts **
000360  012605                          MOV     (SP)+,R5
000362  012603                          MOV     (SP)+,R3
000364  012602                          MOV     (SP)+,R2
000366  012601                          MOV     (SP)+,R1
000370  000207                          RETURN
```

```
                        .SBTTL  DHSTOP -- Stop transmission to a DH11 line
                ;-----------------------------------------------------------------------
                ;  DHSTOP is called to stop transmission to a DH11 line.
                ;  This is done when a ctrl-S is received from the terminal and we
                ;  want to suspend output until a ctrl-Q is received.
                ;
                ;  Inputs:
                ;   R1 = Physical line index number.
                ;
000372  010546         DHSTOP: MOV     R5,-(SP)
000374  016105  000000G         MOV     LMXNUM(R1),R5   ;Get mux index number
000400  016100  000000G         MOV     LMXLN(R1),R0    ;Get # of line within mux group (0-15)
000404  006300                  ASL     R0              ;Convert line # to word table index
                ;
                ;  Abort the DMA transfer
                ;
000406                          DISABL                  ;;;** Disable interrupts **
000414  046075  000000' 000000G BIC     LINBIT(R0),@MH$BAR(R5);;;Stop transmitter for line
000422  046065  000000' 000000G BIC     LINBIT(R0),MH$PBR(R5);;;Say we know line is stopped
000430  042761  000000G 000000G BIC     #$XCHAR,LSW3(R1);;;Say transmitter idle
000436  052761  000000G 000000G BIS     #$DHCDO,LSW10(R1);;;Request clock-driven processing for line
000444  005237  000000G         INC     NEDCDO          ;;;Say output processing needed
                ;
                ;  See if there is any data remaining to be transmitted.
                ;
000450  006200                  ASR     R0              ;;;Get mux line number
000452  052700  000000G         BIS     #HF$RIE,R0      ;;;Set receiver interrupt enable flag
000456  110075  000000G         MOVB    R0,@MH$SCR(R5)  ;;;Select line of interest
000462  017500  000000G         MOV     @MH$BCR(R5),R0  ;;;Get remaining byte count
000466  001406                  BEQ     4$              ;;;Br if no remaining bytes to transmit
000470  005400                  NEG     R0              ;;;Get positive byte count
000472  010061  000000G         MOV     R0,LDHB2R(R1)   ;;;Save remaining byte count
000476  017561  000000G 000000G MOV     @MH$CAR(R5),LDHB2S(R1);;;Save DMA buffer address
000504                  4$:     ENABL                   ;** Enable interrupts **
                ;
                ;  Finished
                ;
000512  012605                  MOV     (SP)+,R5
000514  000207                  RETURN
```

```
                                        .SBTTL  DHXOFF -- Force transmission of XOFF to DH11 line
                                ;-----------------------------------------------------------------------
                                ;  DHXOFF is called to stuff an XOFF character into the output stream
                                ;  for a DH11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
000516                          DHXOFF:
                                ;
                                ;  Set flag saying we have stopped the sender
                                ;
000516  052761  000000G 000000G         BIS     #$HISTP,LSW10(R1);Set flag saying we have send XOFF to sender
                                ;
                                ;  Set flag to cause DHSWBF to send an XOFF character
                                ;
000524  052761  000000G 000000G         BIS     #$DHXOF,LSW10(R1);Tell DHSWBF to send XOFF
                                ;
                                ;  Stop the current transmission
                                ;
000532  004737  000372'                 CALL    DHSTOP          ;Stop the current transmission
                                ;
                                ;  Now restart the transmitter so it will send the XOFF
                                ;
000536  004737  000224'                 CALL    DHSTRT          ;Start the transmitter
                                ;
                                ;  Finished
                                ;
000542  000207                          RETURN
```

```
                                        .SBTTL  DHXON  -- Force transmission of XON to a DH11 line
                                ;-----------------------------------------------------------------------
                                ;  DHXON is called to stuff an XON (ctrl-Q) character into the output
                                ;  stream for a DH11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
000544                          DHXON:
                                ;
                                ;  Clear flag that says sender has been stopped
                                ;
000544  042761  000000G 000000G        BIC     #$HISTP,LSW10(R1);Clear flag that says XOFF sent to sender
                                ;
                                ;  Set flag to cause DHSWBF to transmit an XON character
                                ;
000552  052761  000000G 000000G        BIS     #$DHXON,LSW10(R1);Set flag to force XON transmission
                                ;
                                ;  Abort any current transmission
                                ;
000560  004737  000372'                CALL    DHSTOP          ;Abort any current transmission
                                ;
                                ;  Start the transmitter to transmit the XON
                                ;
000564  004737  000224'                CALL    DHSTRT          ;Start the transmitter
                                ;
                                ;  Finished
                                ;
000570  000207                         RETURN
```

```
                                        .SBTTL  DHTIMR -- Clock driven routine for DH/DHV output
                                ;-----------------------------------------------------------------------
                                ;  DHTIMR is called on a clock interrupt (50/60 Hz) basis to set up
                                ;  DMA transfers for DH11 and DHV11 multiplexers.
                                ;  This routine moves characters from the TTY output buffer into DMA
                                ;  buffers and starts the DMA transfer when a DMA buffer becomes full
                                ;  or there are no more characters in the TTY output buffer.
                                ;  There are two DMA buffers for each line.  Buffer 1 is the one filled
                                ;  by this routine as characters become available.  Buffer 2 is emptied
                                ;  by DMA transfers. The buffers are switched as DMA transfers are completed.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
000572  010246                  DHTIMR: MOV     R2,-(SP)
000574  010346                          MOV     R3,-(SP)
000576  010446                          MOV     R4,-(SP)
000600  010546                          MOV     R5,-(SP)
                                ;
                                ;  Start output if we need to send either ctrl-S or ctrl-Q
                                ;
000602  032761  000000C 000000G         BIT     #$DHXOF!$DHXON,LSW10(R1);Need to send ctrl-S or ctrl-Q?
000610  001074                          BNE     5$              ;Br if yes
                                ;
                                ;  Start output if transmission was suspended
                                ;
000612  005761  0000000G              TST     LDHB2R(R1)      ;Was output suspended?
000616  001071                          BNE     5$              ;Br if yes
                                ;
                                ;  If buffer 1 is not full, try to add more characters to it
                                ;
000620                                  DISABL                  ;;;** Disable interrupts **
000626  016105  0000000G        6$:     MOV     LDHB1B(R1),R5   ;;;Get pointer to base of buffer 1
000632  062705  0000000G                ADD     #DHBFSZ,R5      ;;;Get pointer past end of buffer 1
000636  166105  0000000G                SUB     LDHB1P(R1),R5   ;;;Compute free space in buffer 1
000642  001454                          BEQ     4$              ;;;Br if buffer 1 is full
000644  042761  0000000G 0000000G       BIC     #$DHBF1,LSW10(R1);;;Say buffer 1 is not ready to go
000652                                  ENABL                   ;** Enable interrupts **
                                ;
                                ;  Buffer 1 is not full.
                                ;  Move as many characters as possible into buffer 1.
                                ;
000660  016102  0000000G                MOV     LDHB1P(R1),R2   ;Get pointer into buffer 1
000664  016103  0000000G                MOV     LOUTIR(R1),R3   ;Get address of routine to supply characters
000670  010104                          MOV     R1,R4           ;Need line # in R4 for NEDCHR
000672                          1$:     DISABL                  ;;;** Disable interrupts **
000700  004713                          CALL    (R3)            ;;;Try to get another character for this line
000702  103405                          BCS     2$              ;;;Br if no more characters are available
000704                                  ENABL                   ;** Enable interrupts **
000712  110022                          MOVB    R0,(R2)+        ;Store char into buffer 1
000714  077512                          SOB     R5,1$           ;Loop if more space in buffer 1
                                ;
                                ;  Buffer 1 is either full or there are no more characters available.
                                ;  If there are any characters in buffer 1, set flag saying it is
                                ;  ready to go.
                                ;
000716                          2$:     ENABL                   ;** Enable interrupts **
```

```
     58 000724  010261  0000000                  MOV     R2,LDHB1P(R1)   ;Save new pointer into buffer 1
     59 000730  166102  0000000                  SUB     LDHB1B(R1),R2   ;Compute # chars in buffer 1
     60 000734  001431                           BEQ     9$              ;Br if no characters in buffer 1
     61                                 ;
     62                                 ;  Buffer 1 has some characters and is ready to go.
     63                                 ;
     64 000736  052761  0000000 0000000 3$:      BIS     #$DHBF1,LSW10(R1);Say buffer 1 is ready to go
     65 000744  016100  0000000                  MOV     LCDTYP(R1),R0   ;Get device type index
     66 000750  004770  0000000                  CALL    @CDSTR2(R0)     ;Call secondary line start routine
     67                                 ;
     68                                 ;  If buffer 1 was used by the output server, go back and try to
     69                                 ;  refill it.
     70                                 ;
     71 000754                                   DISABL                  ;;;** Disable interrupts **
     72 000762  032761  0000000 0000000          BIT     #$DHBF1,LSW10(R1);;;Is buffer 1 now free
     73 000770  001716                           BEQ     6$              ;;;Br if yes
     74 000772  000407                           BR      8$              ;;;Buffer 1 is waiting to be sent
     75                                 ;
     76                                 ;  Try to start output to DH line
     77                                 ;
     78 000774                          4$:      ENABL                   ;** Enable interrupts **
     79 001002  016100  0000000         5$:      MOV     LCDTYP(R1),R0   ;Get device type index
     80 001006  004770  0000000                  CALL    @CDSTR2(R0)     ;Call secondary line start routine
     81                                 ;
     82                                 ;  Finished
     83                                 ;
     84 001012                          8$:      ENABL                   ;** Enable interrupts **
     85 001020  012605                  9$:      MOV     (SP)+,R5
     86 001022  012604                           MOV     (SP)+,R4
     87 001024  012603                           MOV     (SP)+,R3
     88 001026  012602                           MOV     (SP)+,R2
     89 001030  000207                           RETURN
```

```
                                        .SBTTL  DHSWBF -- Switch DH11 output buffers
                                ;-----------------------------------------------------------------------
                                ;  DHSWBF is called when we finish transmitting DMA characters to a DH11 or
                                ;  DHV11 to try to get a new buffer of characters.
                                ;  If buffer 1 is ready to go ($DHBF1 flag set) the buffer pointers are
                                ;  swapped and buffer 1 becomes ready to be sent.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> A buffer switch took place.  Data ready for transmission
                                ;   C-flag set     ==> There is not data ready for transmission.
                                ;   R0 = Address of buffer with data (if C-flag cleared).
                                ;   R2 = Number of characters to transmit.
                                ;
001032                          DHSWBF:
                                ;
                                ;  See if we need to send a control-S.
                                ;
001032  032761  0000000G 0000000G       BIT     #$DHXOF,LSW1O(R1);Do we need to send control-S?
001040  001410                          BEQ     2$              ;Br if not
001042  042761  0000000G 0000000G       BIC     #$DHXOF,LSW1O(R1);Say control-S being sent
001050  012700  000040'                 MOV     #CTLSBF,R0      ;Point to control-S buffer
001054  012702  000001                  MOV     #1,R2           ;Say to send 1 character
001060  000461                          BR      3$              ;Go send ctrl-S
                                ;
                                ;  See if we need to send a control-Q.
                                ;
001062  032761  0000000G 0000000G 2$:   BIT     #$DHXON,LSW1O(R1);Do we need to send control-Q?
001070  001410                          BEQ     4$              ;Br if not
001072  042761  0000000G 0000000G       BIC     #$DHXON,LSW1O(R1);Say control-Q being sent
001100  012700  000041'                 MOV     #CTLQBF,R0      ;Point to buffer
001104  012702  000001                  MOV     #1,R2           ;Say to send 1 character
001110  000445                          BR      3$              ;Go send ctrl-Q
                                ;
                                ;  See if output is suspended because we received a control-S
                                ;
001112  032761  0000000G 0000000G 4$:   BIT     #$CTRLS,LSW3(R1);Is output suspended due to ctrl-S?
001120  001053                          BNE     7$              ;Br if yes
                                ;
                                ;  See if transmission was suspended and we need to restart now.
                                ;
001122  016102  0000000G                MOV     LDHB2R(R1),R2   ;Was transmission of buffer 2 suspended?
001126  001405                          BEQ     5$              ;Br if not
001130  005061  0000000G                CLR     LDHB2R(R1)      ;Say not suspended now
001134  016100  0000000G                MOV     LDHB2S(R1),R0   ;Get pointer to restart point in buffer 2
001140  000431                          BR      3$              ;Go restart transmission
                                ;
                                ;  See if buffer 1 is ready to go
                                ;
001142  032761  0000000G 0000000G 5$:   BIT     #$DHBF1,LSW1O(R1);Is buffer 1 ready to go?
001150  001434                          BEQ     1$              ;Br if not
                                ;
                                ;  Buffer 1 is ready to go
                                ;  Switch buffer pointers
                                ;
```

```
     58 001152  016102  0000000G                    MOV     LDHB1P(R1),R2   ;Get pointer into buffer 1
     59 001156  166102  0000000G                    SUB     LDHB1B(R1),R2   ;Get # characters in buffer 1
     60 001162  016100  0000000G                    MOV     LDHB2B(R1),R0   ;Base of buffer 2
     61 001166  016161  0000000G 0000000G           MOV     LDHB1B(R1),LDHB2B(R1);Switch buffer base pointers
     62 001174  010061  0000000G                    MOV     R0,LDHB1B(R1)
     63 001200  010061  0000000G                    MOV     R0,LDHB1P(R1)   ;Reset pointer into buffer 1
     64 001204  016100  0000000G                    MOV     LDHB2B(R1),R0   ;Return address of buffer 2 in R0
     65                                     ;
     66                                     ;  Now set status indicating that buffer 2 is busy and buffer 1 is free
     67                                     ;
     68 001210  052761  0000000G 0000000G           BIS     #$DHBF2,LSW10(R1);Buffer 2 is busy
     69 001216  042761  0000000G 0000000G           BIC     #$DHBF1,LSW10(R1);Buffer 1 is free
     70                                     ;
     71                                     ;  There is data ready for transmission start.
     72                                     ;
     73 001224  052761  0000000G 0000000G 3$:       BIS     #$DHCDO,LSW10(R1);Request clock-driven processing for line
     74 001232  005237  0000000G                    INC     NEDCDO          ;Say output character processing needed
     75 001236  000241                              CLC                     ;Signal that buffer 2 is ready to go
     76 001240  000404                              BR      9$
     77                                     ;
     78                                     ;  Buffer 1 is not ready to go.
     79                                     ;  Mark buffer 2 as free.
     80                                     ;
     81 001242  042761  0000000G 0000000G 1$:       BIC     #$DHBF2,LSW10(R1);Say buffer 2 is free
     82 001250  000261                     7$:      SEC                     ;Say that buffer 1 not ready to go
     83                                     ;
     84                                     ;  Finished
     85                                     ;
     86 001252  000207                     9$:      RETURN
```

```
                                        .SBTTL  VHSTRT -- Start output to a DHV11 line
                                ;-----------------------------------------------------------------------
                                ;  VHSTRT is called to initiate transmission to a DHV11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number.
                                ;
001254  010146                  VHSTRT: MOV     R1,-(SP)
001256  010246                          MOV     R2,-(SP)
001260  010346                          MOV     R3,-(SP)
001262  010546                          MOV     R5,-(SP)
                                ;
                                ;  Return if transmission to the line is in progress
                                ;
001264                                  DISABL                  ;;;** Disable interrupts **
001272  032761  000000G 000000G         BIT     #$XCHAR,LSW3(R1);;;Is a transmission in progress now?
001300  001036                          BNE     9$              ;;;Br if yes
                                ;
                                ;  Line is currently idle.
                                ;  See if there is data ready to be transmitted.
                                ;
001302  052761  000000G 000000G         BIS     #$XCHAR,LSW3(R1);;;Say line is busy
001310  004737  001032'                 CALL    DHSWBF          ;;;Try to get data to transmit
001314  103420                          BCS     1$              ;;;Br if nothing to transmit
                                ;
                                ;  There is data to be transmitted.
                                ;  R0 = Address of DMA buffer.
                                ;  R2 = Number of characters in buffer.
                                ;
001316  016105  000000G                 MOV     LMXNUM(R1),R5   ;;;Get the mux index number
001322  116103  000000G                 MOVB    LMXLN(R1),R3    ;;;Get line # within mux group
001326  052703  000000G                 BIS     #VF$RIE,R3      ;;;Set receiver interrupt enable flag
001332  110375  000000G                 MOVB    R3,@VH$CSR(R5)  ;;;Select our line
001336  010275  000000G                 MOV     R2,@VH$BCR(R5)  ;;;Set DMA transfer length
001342  010075  000000G                 MOV     R0,@VH$BA1(R5)  ;;;Set address of DMA buffer
001346  012775  000000C 000000G         MOV     #VF$TEN!VF$TGO,@VH$BA2(R5);;;Start the DMA transmission
001354  000410                          BR      9$              ;;;
                                ;
                                ;  There are no more characters to transmit
                                ;
001356  042761  000000G 000000G 1$:     BIC     #$XCHAR,LSW3(R1);;;Say transmitter is not busy
001364  052761  000000G 000000G         BIS     #$DHCDO,LSW10(R1);Request clock-driven processing for line
001372  005237  000000G                 INC     NEDCDO          ;;;Say output processing needed
                                ;
                                ;  Finished
                                ;
001376                          9$:     ENABL                   ;** Enable interrupts **
001404  012605                          MOV     (SP)+,R5
001406  012603                          MOV     (SP)+,R3
001410  012602                          MOV     (SP)+,R2
001412  012601                          MOV     (SP)+,R1
001414  000207                          RETURN
```

```
                                        .SBTTL  VHOINT -- Output interrupt from DHV11 line
                                ;-----------------------------------------------------------------------
                                ;  VHOINT is entered when an output interrupt occurs on a DHV11 line.
                                ;  The interrupt is vectored directly to an instruction sequence of the
                                ;  form:
                                ;
                                ;       JSR     R5,@#VHOINT
                                ;       .WORD   mux_number
                                ;
                                ;  Thus on entry to VHOINT, R5 has been saved on the stack and currently
                                ;  points to a word containing the mux index number.
                                ;
001416  010046                  VHOINT: MOV     R0,-(SP)
001420  010146                          MOV     R1,-(SP)
001422  013746  000000G                 MOV     INTPRI,-(SP)    ;Save current interrupt priority value
001426  005037  000000G                 CLR     INTPRI          ;Say running interrupt priority = 7
                                ;
                                ;  Obtain the interrupting physical line number
                                ;
001432  011505                          MOV     (R5),R5         ;Get mux index number
001434  017501  000000G          2$:     MOV     @VH$CSR(R5),R1  ;Get contents of DHV11 CSR register
001440  002045                          BGE     9$              ;Br if no transmitter needs service
001442  000301                          SWAB    R1              ;Right justify interrupting line #
001444  042701  000000C                 BIC     #^C<VF$LIN>,R1  ;Clear all but line #
001450  066501  000000G                 ADD     MXLNT(R5),R1    ;Point into mux line # table
001454  111101                          MOVB    (R1),R1         ;Get TSX-plus line #
001456  001766                          BEQ     2$              ;Br if line not genned into system
001460  042761  000000G 000000G         BIC     #$XCHAR,LSW3(R1);Say transmitter is not busy for this line
001466  052761  000000G 000000G         BIS     #$DHCDO,LSW10(R1);Request clock-driven processing for line
001474  005237  000000G                 INC     NEDCDO          ;Say output processing needed
                                ;
                                ;  See if a transmission was aborted
                                ;
001500  116100  000000G                 MOVB    LMXLN(R1),R0    ;Get line # within mux group
001504  052700  000000G                 BIS     #VF$RIE,R0      ;Set receiver interrupt enable flag
001510  110075  000000G                 MOVB    R0,@VH$CSR(R5)  ;Select our line in mux
001514  032775  000000G 000000G         BIT     #VF$ABT,@VH$LCR(R5);Was the DMA transmission aborted?
001522  001411                          BEQ     1$              ;Br if not
001524  017561  000000G 000000G         MOV     @VH$BCR(R5),LDHB2R(R1);Save remaining byte count
001532  017561  000000G 000000G         MOV     @VH$BA1(R5),LDHB2S(R1);Save suspended output buffer pointer
001540  042775  000000G 000000G         BIC     #VF$ABT,@VH$LCR(R5);Clear the DMA abort flag
                                ;
                                ;  Try to start more output for this line
                                ;
001546  004737  001254'          1$:     CALL    VHSTRT          ;Try to start more output for the line
001552  000730                          BR      2$              ;See if another line needs service
                                ;
                                ;  Finished -- Return from interrupt
                                ;
001554  012637  000000G          9$:     MOV     (SP)+,INTPRI    ;Restore interrupt priority
001560  012601                          MOV     (SP)+,R1
001562  012600                          MOV     (SP)+,R0
001564  012605                          MOV     (SP)+,R5        ;Pushed by JSR R5
001566  000002                          RTI                     ;Return from interrupt
```

```
                                        .SBTTL  VHSTOP -- Stop DMA transmission to a DHV11 line
                                ;-----------------------------------------------------------------
                                ;  VHSTOP is called to abort a DMA transfer to a DHV11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
001570  010546                  VHSTOP: MOV     R5,-(SP)
                                ;
                                ;  See if a transmission is in progress
                                ;
001572                                  DISABL                          ;;;** Disable interrupts **
001600  032761  0000000 0000000         BIT     #$XCHAR,LSW3(R1);;;Is a transmission in progress now?
001606  001413                          BEQ     9$                      ;;;Br if not
                                ;
                                ;  A transmission is in progress.
                                ;  Set the DMA abort flag.
                                ;
001610  016105  0000000                 MOV     LMXNUM(R1),R5   ;;;Get mux index number
001614  116100  0000000                 MOVB    LMXLN(R1),R0    ;;;Get # of line within mux group
001620  052700  0000000                 BIS     #VF$RIE,R0      ;;;Set receiver interrupt enable flag
001624  110075  0000000                 MOVB    R0,@VH$CSR(R5)  ;;;Select our line
001630  052775  0000000 0000000         BIS     #VF$ABT,@VH$LCR(R5);;;Set DMA abort flag
                                ;
                                ;  Finished
                                ;
001636                          9$:     ENABL                           ;** Enable interrupts
001644  012605                          MOV     (SP)+,R5
001646  000207                          RETURN
```

```
                                        .SBTTL  VHXOFF -- Force transmission of XOFF to DHV11 line
                                ;-----------------------------------------------------------------------
                                ;  VHXOFF is called to stuff an XOFF (ctrl-S) character into the output
                                ;  stream for a DHV11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
001650  010546                  VHXOFF: MOV     R5,-(SP)
                                ;
                                ;  Set flag that says XOFF has been sent to sender
                                ;
001652  052761  0000000 0000000         BIS     #$HISTP,LSW10(R1);Set flag saying XOFF has been sent
                                ;
                                ;  Tell DHV11 to transmit an XOFF
                                ;
001660  016105  0000000                 MOV     LMXNUM(R1),R5   ;Get mux index number
001664  116100  0000000                 MOVB    LMXLN(R1),R0    ;Get # of mux line within mux
001670  052700  0000000                 BIS     #VF$RIE,R0      ;Set receiver interrupt enable flag
001674                                  DISABL                  ;;;** Disable interrupts **
001702  110075  0000000                 MOVB    R0,@VH$CSR(R5)  ;;;Select our mux line
001706  052775  0000000 0000000         BIS     #VF$XOF,@VH$LCR(R5);;;Tell DHV11 to send an XOFF
001714                                  ENABL                   ;** Enable interrupts **
                                ;
                                ;  Finished
                                ;
001722  012605                          MOV     (SP)+,R5
001724  000207                          RETURN

                                        .SBTTL  VHXON  -- Force transmission of XON to a DHV11 line
                                ;-----------------------------------------------------------------------
                                ;  VHXON is called to stuff an XON (ctrl-Q) character into the output
                                ;  stream for a DHV11 line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
001726  010546                  VHXON:  MOV     R5,-(SP)
                                ;
                                ;  Clear flag that says XOFF has been sent to sender
                                ;
001730  042761  0000000 0000000         BIC     #$HISTP,LSW10(R1);Clear flag that says XOFF has been sent
                                ;
                                ;  Tell DHV11 to transmit an XON
                                ;
001736  016105  0000000                 MOV     LMXNUM(R1),R5   ;Get mux index number
001742  116100  0000000                 MOVB    LMXLN(R1),R0    ;Get # of line within mux group
001746  052700  0000000                 BIS     #VF$RIE,R0      ;Set receiver interrupt enable flag
001752                                  DISABL                  ;;;** Disable interrupts **
001760  110075  0000000                 MOVB    R0,@VH$CSR(R5)  ;;;Select our mux line
001764  042775  0000000 0000000         BIC     #VF$XOF,@VH$LCR(R5);;;Tell DHV11 to send an XON
001772                                  ENABL                   ;** Enable interrupts **
                                ;
                                ;  Finished
                                ;
002000  012605                          MOV     (SP)+,R5
002002  000207                          RETURN
```

```
     58         000001                              .END
Errors detected:   0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  117 Words  ( 1 Pages)
Size of core pool:  17920 Words  ( 70 Pages)
Operating  system:  RT-11

Elapsed time:  00:00:20.02
DK:TSDHIO,LP:TSDHIO=DK:TSDHIO.MAC/C/N:SYM
```

Cross reference table (CREF V05.04)

```
$CTRLS   1-24    9-39
$DHBF1   1-25    8-37    8-64    8-72    9-52    9-69
$DHBF2   1-25    9-68    9-81
$DHCDO   1-25    5-21    9-73    10-42   11-29
$DHXOF   1-27    6-17    8-22    9-21    9-23
$DHXON   1-27    7-17    8-22    9-30    9-32
$HISTP   1-24    6-13    7-13    13-13   13-42
$XCHAR   1-24    3-45    4-16    4-22    4-45    5-20    10-16   10-22   10-41   11-28   12-13
CDSTR2   1-26    8-66    8-80
CTLQBF   1-55#   9-33
CTLSBF   1-54#   9-24
DHBFSZ   1-21    8-34
DHOCHK   2-27    3-9#
DHOINT   1-14    2-13#
DHSTOP   1-14    5-10#   6-21    7-21
DHSTRT   1-14    3-49    4-8#    6-25    7-25
DHSWBF   4-23    9-17#   10-23
DHTIMR   1-17    8-15#
DHXOFF   1-16    6-9#
DHXON    1-16    7-9#
HF$LIN   1-21
HF$RIE   1-27    4-32    4-37    5-27
HF$TI    1-27    3-17
INTPRI   1-22    2-18    2-19*   2-31*   3-21    4-49    5-34    8-38    8-49    8-57    8-78    8-84
        10-47   11-15   11-16*  11-50*  12-27   13-23   13-52
LCDTYP   1-26    8-65    8-79
LDHB1B   1-26    8-33    8-59    9-59    9-61    9-62*
LDHB1P   1-26    8-35    8-43    8-58*   9-58    9-63*
LDHB2B   1-26    9-60    9-61*   9-64
LDHB2R   1-26    5-32*   8-27    9-44    9-46*   11-39*
LDHB2S   1-26    5-33*   9-47    11-40*
LINBIT   1-49#   4-39    4-40    5-18    5-19
LMXLN    1-23    4-31    5-12    10-31   11-34   12-20   13-18   13-47
LMXNUM   1-24    4-30    5-11    10-30   12-19   13-17   13-46
LOUTIR   1-22    8-44
LSW10    1-25    5-21*   6-13*   6-17*   7-13*   7-17*   8-22    8-37*   8-64*   8-72    9-21    9-23*
         9-30    9-32*   9-52    9-68*   9-69*   9-73*   9-81*   10-42*  11-29*  13-13*  13-42*
LSW3     1-24    3-45*   4-16    4-22*   4-45*   5-20*   9-39    10-16   10-22*  10-41*  11-28*  12-13
LSW5     1-24
MH$BAR   1-21    3-19    4-40*   5-18*
MH$BCR   1-21    4-35*   5-29
MH$CAR   1-21    4-36*   5-33
MH$PBR   1-22    3-18    3-20*   4-39*   5-19*
MH$SCR   1-21    3-17*   4-33*   5-28*
MXLNT    1-24    3-42    11-25
NEDCDO   1-22    5-22*   9-74*   10-43*  11-30*
PSW      1-22    3-16*   3-21*   4-15*   4-49*   5-17*   5-34*   8-32*   8-38*   8-46*   8-49*   8-57*
         8-71*   8-78*   8-84*   10-15*  10-47*  12-12*  12-27*  13-20*  13-23*  13-49*  13-52*
TSDHIO   1-3#    1-13
VF$ABT   1-25    11-37   11-41   12-23
VF$LIN   1-22    11-24
VF$RIE   1-27    10-32   11-35   12-21   13-19   13-48
VF$TDV   1-25
VF$TEN   1-23    10-36
VF$TGO   1-23    10-36
VF$XOF   1-27    13-22   13-51
```

Cross reference table (CREF V05.04)

Cross reference table (CREF V05.04)