H 6

```
TTTTTTTTTTTTTTT  TTTTTTTTTTTTTTT  DDDDDDDDDDDD      RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
TTTTTTTTTTTTTTT  TTTTTTTTTTTTTTT  DDDDDDDDDDDD      RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
TTTTTTTTTTTTTTT  TTTTTTTTTTTTTTT  DDDDDDDDDDDD      RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD   RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD   RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD   RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD   RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD   RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD   RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD   RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD   RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD   RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD   RRR   RRR        VVV         VVV  RRR   RRR
      TTT              TTT        DDD         DDD   RRR   RRR        VVV         VVV  RRR   RRR
      TTT              TTT        DDD         DDD   RRR   RRR        VVV         VVV  RRR   RRR
      TTT              TTT        DDD         DDD   RRR      RRR        VVV   VVV     RRR      RRR
      TTT              TTT        DDD         DDD   RRR      RRR        VVV   VVV     RRR      RRR
      TTT              TTT        DDD         DDD   RRR      RRR        VVV   VVV     RRR      RRR
      TTT              TTT        DDDDDDDDDDDD      RRR         RRR        VVV        RRR         RRR
      TTT              TTT        DDDDDDDDDDDD      RRR         RRR        VVV        RRR         RRR
      TTT              TTT        DDDDDDDDDDDD      RRR         RRR        VVV        RRR         RRR
```

_S

Va
--
00
00
00
00
00
00
00
48
80
80
80
80
80
80
80
80
80
80
80
80
80
80
80
80
80
80

FILEID**TTYSUB

M 8

```
TTTTTTTTTT  TTTTTTTTTT  YY      YY    SSSSSSSS  UU      UU  BBBBBBBB
TTTTTTTTTT  TTTTTTTTTT  YY      YY    SSSSSSSS  UU      UU  BBBBBBBB
    TT          TT      YY      YY  SS          UU      UU  BB      BB
    TT          TT      YY      YY  SS          UU      UU  BB      BB
    TT          TT        YY  YY    SS          UU      UU  BB      BB
    TT          TT        YY  YY    SS          UU      UU  BB      BB
    TT          TT          YY        SSSSSS    UU      UU  BBBBBBBB
    TT          TT          YY        SSSSSS    UU      UU  BBBBBBBB
    TT          TT          YY              SS  UU      UU  BB      BB
    TT          TT          YY              SS  UU      UU  BB      BB
    TT          TT          YY              SS  UU      UU  BB      BB
    TT          TT          YY              SS  UU      UU  BB      BB        ....
    TT          TT          YY      SSSSSSSS    UUUUUUUUUU  BBBBBBBB          ....
    TT          TT          YY      SSSSSSSS    UUUUUUUUUU  BBBBBBBB          ....
                                                                              ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000        .TITLE  TTYSUB - Terminal driver miscellaneous subroutines
0000        .IDENT  'V04-001'
0000
0000  ;
0000  ;****************************************************************************
0000  ;*                                                                          *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000  ;*  ALL RIGHTS RESERVED.                                                    *
0000  ;*                                                                          *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000  ;*  TRANSFERRED.                                                            *
0000  ;*                                                                          *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000  ;*  CORPORATION.                                                            *
0000  ;*                                                                          *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000  ;*                                                                          *
0000  ;*                                                                          *
0000  ;****************************************************************************
0000  ;
0000  ;++
0000  ;
0000  ; FACILITY:
0000  ;
0000  ;       VAX/VMS TERMINAL DRIVER
0000  ;
0000  ; ABSTRACT:
0000  ;
0000  ;       THIS MODULE CONTAINS ROUTINES COMMON TO ALL TERMINAL DRIVER FUNCTIONS.
0000  ;
0000  ; AUTHOR:
0000  ;
0000  ;       R.HEINEN  11-AUG-1976
0000  ;
0000  ; Revision history:
0000  ;
0000  ;
0000  ;       V04-001 MIR1100         Michael I. Rosenblum    7-Sep-1984
0000  ;               Add code to catch fired hangup ast's.
0000  ;
0000  ;       V03-057 JLV0394         Jake VanNoy             13-AUG-1984
0000  ;               Change exit path in read error such that condition code is
0000  ;               is correctly on return to port driver.
0000  ;
0000  ;       V03-056 MIR0460         Michael I. Rosenblum            07-Aug-1984
0000  ;               Fix bug described in QAR 915 that points out a condition
0000  ;               Where the DMF hardware XON/XOFF handling get's out of synch
0000  ;               With the terminal driver's reconing.
0000  ;
0000  ;       V03-055 MIR0455         MICHAEL I. ROSENBLUM            20-JUN-1984
```

```
0000    ;               Add check to catch late rtimout ast's.
0000    ;               make detach cause a hangup if the line is set as such.
0000    ;
0000    ;       V03-054 RKS0054         RICK SPITZ                      10-APR-1984
0000    ;               Show Physical terminal UCB redirected when associated
0000    ;               with logical TT UCB. Remove characteristics logic on rebind
0000    ;               as it is done in the connect startio I/O action code.
0000    ;
0000    ;       V03-053 MIR0390         Michael I. Rosenblum            04-Apr-1984
0000    ;               Return the offset when a readverify read terminates
0000    ;               Due to timeout.  Terminate a Read verify read if
0000    ;               it is restarted by another io also return the correct
0000    ;               offset.
0000    ;
0000    ;       V03-052 MIR0370         Michael I. Rosenblum            20-Mar-1984
0000    ;               Remove local definition of TTY$V_ST_CTSLOW.
0000    ;
0000    ;       V03-051 RKS0051         RICK SPITZ                      15-MAR-1984
0000    ;
0000    ;               Cleanup modem disconnect routines and
0000    ;               add login timeout logic to modem processing.
0000    ;               Cleanup routine headers.
0000    ;
0000    ;       V03-050 RKS0050         RICK SPITZ                      05-MAR-1984
0000    ;               Enhance virtual terminal timeout code to delete
0000    ;               a virtual terminal when specified time period expires.
0000    ;
0000    ;       V03-048 MIR0310         Michael Rosenblum               09-Feb-1984
0000    ;               Fix bugs.
0000    ;               Clear breakthru classes on last deassign.
0000    ;               Maintnance functions shold not always return abort.
0000    ;               Add entry point to Transition to not check modem in
0000    ;               Devdepend so that the modem can be shutdown when
0000    ;               modem is turned off.
0000    ;
0000    ;       V03-047 MHB0100         Mark H. Bramhall                1-Feb-1984
0000    ;               Add the setting/resetting of DEV$M_DET in DEVCHAR2 during
0000    ;               LUCB unlinks/(re-)links.
0000    ;
0000    ;       V03-046 MIR0200         Michael I. Rosenblum            15-Oct-1983
0000    ;               make calls to EXE$FORK call TTY$SYNCH to allow the terminal
0000    ;               driver to provide different synchronization methods.
0000    ;               Change input IPL to the RTIMOU routine to allow the
0000    ;               class driver to provide it's own synchronization.
0000    ;
0000    ;       V03-045 MIR0082         Michael I. Rosenblum            19-Aug-1983
0000    ;               Add argument in R0 to the PORT DISCONNECT routine.
0000    ;
0000    ;       V03-044 MIR1080         Michael I. Rosenblum            11-Aug-1983
0000    ;               remove restriction that modem state table routines must
0000    ;               be forward references.
0000    ;
0000    ;       V03-043 MIR0080         Michael I. Rosenblum            28-Jul-1983
0000    ;               Reposition the routines in the module.
0000    ;
0000    ;       V03-042 MIR0070         Michael I. Rosenblum            13-Jul-1983
0000    ;               Re-write the code that handles modem timers to allow
```

```
0000  ;              any number of lines of modem control per controler.
0000  ;              Make the UNLINK routine turn around reads with timers
0000  ;              rather than requeueing them.
0000  ;              change setup ucb to  not setup the ucb if there are
0000  ;              channels assigned.
0000  ;              Cause transition when called from the ports to turn
0000  ;              a call to INIT with channels assigned to a SHUTDOWN.
0000  ;              This will blow the processes away when a power fail
0000  ;              happens.
0000  ;
0000  ;      V03-041 MIR0051          Michael I. Rosenblum            23-Jun-1983
0000  ;              Add TTY$NOTIFY service.  Cause break on NON-autobaud terminals
0000  ;              to login.  Remove unnecessary jsb's and replace with bsbw's.
0000  ;
0000  ;      V03-040 RKS0040          RICK SPITZ                      8-JUN-1983
0000  ;              ADD SUPPORT FOR CLONING DETACHED TERMINAL UCB AND LOGIC
0000  ;              TO LINK AND UNLINK LOGICAL AND PHYSICAL UCBs.
0000  ;              RESTRUCTURE CANCEL I/O CODE TO BE MORE MODULAR.
0000  ;              MOVE UCB$V_TT_HANGUP INTO LUCB (EXCEPT FOR CANCEL_RESET SPECIAL CASE
0000  ;
0000  ;              REMOVE STOP BROADCASTS CODE SINCE THEY ARE NOW BREAKTHRU WRITES
0000  ;
0000  ;      V03-039 MIR0050          Michael I. Rosenblum            11-May-1983
0000  ;              Remove code that specail cases broadcasts.
0000  ;              Remove TTY$STOP2
0000  ;
0000  ;      V03-038 MIR0041          Michael I. Rosenblum            29-Apr-1983
0000  ;              Add code to allow autobaud to work with parity, autoparity
0000  ;              for even parity checking.  Change autobaud code to
0000  ;              require a readable <CR> before invoking LOGINOUT.
0000  ;
0000  ;      V03-037 MIR0032          Michael I. Rosenblum            05-Apr-1983
0000  ;              Change control character echoing representation.
0000  ;
0000  ;      V03-036 MIR0031          Michael I. Rosenblum            01-Apr-1983
0000  ;              add code to disable the transmiting of characters with
0000  ;              parity errors to the user.
0000  ;
0000  ;      V03-035 RKS0035          RICK SPITZ                      14-MAR-1983
0000  ;              ADD SUPPORT FOR LOGICAL UCB
0000  ;
0000  ;      V03-034 MIR8026          Michael I. Rosenblum            14-Mar-1983
0000  ;              Fix problem with initial string offsets and origional
0000  ;              cursor position
0000  ;
0000  ;      V03-033 MIR0026          Michael I. Rosenblum            01-Mar-1983
0000  ;              Clear out the size of the recall buffer when creating
0000  ;              the typeahead buffer.
0000  ;
0000  ;      V03-032 MIR0024          Michael I. Rosenblum            28-Jan-1983
0000  ;              Change code to reflect the new read buffer.
0000  ;
0000  ;      V03-031 MIR0022          Michael I. Rosenblum            18-Jan-1983
0000  ;              Move common code from port drivers into class jacket routines.
0000  ;              Change  references to UCB$B_ERTCNT to UCB$W_TT_UNITBIT to
0000  ;              allow more maintainability.
0000  ;
```

```
0000   ;       V03-030 MIR0017         Michael I. Rosenblum            05-Jan-1983
0000   ;               Move common powerfail code into this module, and change
0000   ;               the ATTENTION routine to use this common code.  Add
0000   ;               TTY$POWERACTION routine as a class survice to the port
0000   ;               drivers that will setup everything for postive action
0000   ;               when a powerfail occurs.
0000   ;
0000   ;       V03-029 MIR0015         Michael I. Rosenblum            20-Dec-1982
0000   ;               Change fork dispatching code to use state dispatch
0000   ;               Class entry vectors:
0000   ;                       CLASS_FORK - Schedules a fork interupt for the
0000   ;                                    port driver
0000   ;                       CLASS_DISCONNECT - Performs cleanup then delivers
0000   ;                                    DCL's hangup ast.
0000   ;               Add class survice:
0000   ;                       CLASS_MODEM_DIS - Forces modem shutdown transision
0000   ;               Make CLASS_DISCONNECT fork to FIPL
0000   ;               Add class jacket routines for all port functions.
0000   ;
0000   ;       V03-028 MIR0014         Michael I. Rosenblum            17-Dec-1982
0000   ;               Add CLASS_XON and CLASS_XOFF to remove some of the
0000   ;               code duplicated in all of the port drivers.
0000   ;
0000   ;       V03-027 MIR0013         Michael I. Rosenblum            16-Dec-1982
0000   ;               Fix up refferences to new ucb structure
0000   ;
0000   ;       V03-026 MIR0011         Michael I. Rosenblum            18-Nov-1982
0000   ;               Remove HOLDSCREEN code.
0000   ;               Change RESTARTIO to use new EDITREAD state and the
0000   ;               new MULTIECHO functionality.
0000   ;
0000   ;       V03-025 MIR0010         Michael I Rosenblum     9-Nov-1982
0000   ;               Moved the count of characters in the typeahead buffer
0000   ;               from the UCB into the typeahead structure.
0000   ;
0000   ;       V03-024 RKS0024         RICK SPITZ              8-NOV-1982
0000   ;               PREVENT PURGE TYPEAHEAD ROUTINE FROM DESTROYING CONTENTS
0000   ;               OF R4 (ADDRESS OF TYPEAHAED STRUCTURE)
0000   ;
0000   ;       V03-023 RKS0023         RICK SPITZ              04-OCT-1982
0000   ;               CORRECT ATTENTION ROUTINE TO CALL START OUTPUT
0000   ;               AFTER CALLING RESUME FOR BROADCAST. THIS FIXES A PROBLEM
0000   ;               WITH DMF CONTROLLERS HANGING SHUTDOWN IN MWAIT
0000   ;
0000   ;       V03-022 RKS0022         RICK SPITZ              23-SEP-1982
0000   ;               ABORT PORT OUTPUT IN MODEM HANGUP SERVICE ROUTINE.
0000   ;               THIS PREVENTS A TERMINAL IN CONTROL S STATE FROM
0000   ;               KEEPING A PROCESS RUNNING LOGINOUT AROUND UNTIL
0000   ;               NEXT CONTROL Q IS ISSUED.
0000   ;
0000   ;               MOVE LOGOUT MODEM ACTION ROUTINE CALL FROM SHUTDOWN
0000   ;               STATE TO SHUT1. THIS GUARANTEES THAT A TIMER INTERRUPT
0000   ;               OCCURS BETWEEN MODEM TRANSITION AND DELIVERY OF ANY
0000   ;               ATTENTION AST. IT IS NEEDED TO KEEP FROM
0000   ;               CALLING COM$DELATTNAST
0000   ;               AT DEVICE IPL (IN THE CASE OF CONTROLLERS WHICH
0000   ;               INTERRUPT ON MODEM TRANSITIONS).
```

```
0000 ;
0000 ;       V03-021 KDM0002           Kathleen D. Morse       28-Jun-1982
0000 ;               Added $PRDEF.
0000 ;
0000 ;
0000 ;--
```

```
    0000        .SBTTL  Declarations
    0000
    0000
    0000  ;
    0000  ; EXTERNAL SYMBOLS
    0000  ;
    0000
    0000        $ACBDEF                         ;
    0000        $CANDEF                         ; DEFINE CANCEL DEFINITIONS
    0000        $CRBDEF                         ; DEFINE CRB
    0000        $IDBDEF                         ; DEFINE IDB OFFSETS
    0000        $DCDEF                          ; DEFINE DEVICE CODES
    0000        $DEVDEF                         ; DEFINE DEV STATUS AND DEVCHAR BITS
    0000        $DPTDEF                         ; DEFINE DPT OFFSETS
    0000        $DYNDEF                         ; DEFINE DYNAMIC REGION TYPES
    0000        $IODEF                          ; DEFINE I/O FUNCTION CODES
    0000        $IPLDEF                         ; DEFINE INTERRUPT PRIORITY LEVELS
    0000        $IRPDEF                         ; DEFINE IRP
    0000        $MSGDEF                         ; DEFINE MESSAGE TYPES
    0000        $PCBDEF                         ; DEFINE PCB
    0000        $PRDEF                          ; DEFINE PROCESSOR REGISTERS
    0000        $SSDEF                          ; DEFINE SYSTEM STATUS CODES
    0000        $UCBDEF                         ; DEFINE UCB
    0000        $TTDEF                          ; DEFINE TERMINAL CHARACTERISTICS
    0000        $TT2DEF                         ; DEFINE TERMINAL CHARACTERISTICS
    0000        $TTYDEF                         ; DEFINE TERMINAL DRIVER SYMBOLS
    0000        $VECDEF                         ; DEFINE CRB VECTOR
    0000        $TTYMACS                        ; DEFINE TERMINAL MACROS
    0000        $TTYDEFS                        ; DEFINE TERMINAL DEFINITIONS
    0000        $TTYMODEM                       ; DEFINE TERMINAL MODEM SYMBOLS
    0000        $TQEDEF                         ; TIMER QUEUE DEFINITIONS
    0000
00000000        .PSECT  $$$115_DRIVER,QUAD      ; DEFINE NON-PAGED PSECT
```

```
                          0000          .SBTTL  TTY$ABORT_IO - ABORT I/O
                          0000  ;++
                          0000  ;
                          0000  ; TTY$ABORT_IO - Cancel I/O as a result of ABORT out-of-band.
                          0000  ;
                          0000  ; INPUTS:
                          0000  ;         R2 - STATE VECTOR
                          0000  ;         R5 - UCB
                          0000  ;
                          0000  ;--
                          0000
                          0000  TTY$ABORT_IO::
                          0000
            0368    30    0000          BSBW    TTY$PURGE_AHEAD             ; PURGE THE TYPEAHEAD
                          0003
   51    00CC C5    9E    0003          MOVAB   UCB$L_TT_WFLINK(R5),R1      ; Get start of write queue.
            53 51   D0    0008          MOVL    R1,R3                       ; Copy it to check for end.
                          000B  30$:
            53 63   D0    000B          MOVL    TTY$L_WB_FLINK(R3),R3       ; Get next entry.
            51 53   D1    000E          CMPL    R3,R1                       ; At end of queue?
               13   13    0011          BEQL    50$                         ; Yes. Branch forward.
      54    24 A3   D0    0013          MOVL    TTY$L_WB_IRP(R3),R4         ; Get associated IRP address.
               F2   13    0017          BEQL    30$                         ; If no IRP, it's a broadcast,
                          0019                                              ; leave it in the queue.
      53    04 A3   D0    0019          MOVL    TTY$L_WB_BLINK(R3),R3       ; Get entry's backward link.
      54    00 B3   0F    001D          REMQUE  @TTY$L_WB_FLINK(R3),R4      ; Remove the link.
            FFDC'   30    0021          BSBW    TTY$WRITEPOST               ; Queue it for completion
               E5   11    0024          BRB     30$                         ; Try for next link.
                          0026
                          0026  50$:                                        ; Stop the current I/Os.
                          0026          IF_NOT_STATE -                      ; Branch forward if no read is
                          0026                  READ,60$                    ; currently in progress.
                          002A          CLR_STATE -                         ; Clear escape bits.
                          002A                  <ESC,BADESC>
      53    78 A5   D0    0030          MOVL    UCB$L_SVAPTE(R5),R3         ; GET THE PACKET ADDRESS
      54    58 A5   D0    0034          MOVL    UCB$L_IRP(R5),R4            ; Get address of IRP.
                          0038          IF_STATE EOL,55$                    ; IF END OF LINE THEN ALREADY SUBTRACTED
      51    3A A4   9B    003C          MOVZBW  IRP$L_MEDIA+2(R4),R1        ; GET THE LENGTH OF THE TERMINATOR
   3C A3    51      A2    0040          SUBW    R1,TTY$W_RB_TXTOFF(R3)      ; AND FIXUP THE COUNT
                          0044  55$:
         38 A4      D4    0044          CLRL    IRP$L_MEDIA(R4)             ; Clear terminators.
            FFB6'   30    0047          BSBW    TTY$READONE                 ; Complete the read.
                          004A
                          004A  60$:                                        ; Check for a write to complete.
         00DC C5    B5    004A          TSTW    UCB$W_TT_MULTILEN(R5)       ; Test for non-zero
               04   13    004E          BEQL    65$                         ; Skip if zero
                          0050          SET_STATE MULTI                     ; Set multiecho here because
                          0054                                              ;  READONE will clear it if called
                          0054  65$:    IF_NOT_STATE -                      ; Branch forward if no write is
                          0054                  WRITE,70$                   ; in progress.
            09CB    30    0058          BSBW    TTY$ABORT                   ; Abort any current output activity
            0A02    30    005B          BSBW    TTY$RESUME                  ; Restart output
   53    00D4 C5    D0    005E          MOVL    UCB$L_TT_WRTBUF(R5),R3      ; Get address of write packet.
         7C A5      B0    0063          MOVW    UCB$W_BOFF(R5),-            ; Put status code in the
         28 A3            0066                  TTY$W_WB_STATUS(R3)         ; packet.
            FF95'   30    0068          BSBW    TTY$WRITEDONE               ; Complete the I/O.
                          006B
```

```
    006B   327 70$:                                 ; And echo the control char.
05  006B   328          RSB
```

```
                         006C          .SBTTL  TTY$AUTOBAUD -  SENSE AND SET BAUD RATE
                         006C
                         006C  ;++
                         006C  ; TTY$AUTOBAUD - Sense and Set baud rate
                         006C  ;
                         006C  ; Functional description:
                         006C  ;
                         006C  ; This routine is called from the TTYCHARI module when a job controller
                         006C  ; character has been detected. It assumes the character typed was a carriage
                         006C  ; return. The default sampling speed is 9600. If the terminal is set at baud
                         006C  ; rates 1200 - 9600, the algorithm will detect the proper baud rate on one
                         006C  ; character and set to it. If the baud rate is other than that, READERROR will
                         006C  ; detect a framing error and set the sampling rate to 600. A second carriage
                         006C  ; return then sampled at 600 baud can map the 50-600 baud range. Using the 9600
                         006C  ; or 600 baud table the algorithm matches the character read to a specific baud
                         006C  ; rate. No split speed detection is allowed.
                         006C  ;
                         006C  ; Inputs:
                         006C  ;
                         006C  ;       IPL = Device IPL
                         006C  ;       R5 = UCB address
                         006C  ;
                         006C  ; Outputs:
                         006C  ;
                         006C  ;       Baud rate can change.
                         006C  ;       R3 will be changed to <CR>, or a space
                         006C  ;       R4 is destroyed.
                         006C  ;       R5 is preserved.
                         006C  ;--
                         006C
                         006C  TTY$AUTOBAUD::
                         006C
        00B0 C5     D1   006C          CMPL    UCB$L_TT_RDUE(R5),-
    00000000'GF          0070                  G^EXE$GL_ABSTIM          ; compare to current
             4E     13   0075          BEQL    50$                      ; branch if too soon
                         0077
     00 53   07     E5   0077          BBCC    #7,R3,10$                ; Clear high order bit
                         007B
                         007B  ; check baud rate
                         007B
54   00000000'EF    9E   007B  10$:    MOVAB   TTY$AB_9600,R4           ; set 9600 baud table
   00F4 C5   0F     91   0082          CMPB    #TT$C_BAUD_9600,UCB$W_TT_SPEED(R5) ; speed at 9600 ?
             16     13   0087          BEQL    20$                      ; branch if yes
                         0089
54   00000000'EF    9E   0089          MOVAB   TTY$AB_600,R4            ; set 600 baud table
   00F4 C5   07     91   0090          CMPB    #TT$C_BAUD_600,UCB$W_TT_SPEED(R5) ; speed at 600 ?
             08     13   0095          BEQL    20$                      ; branch if no
                         0097
        54   0F     9A   0097          MOVZBL  #TT$C_BAUD_9600,R4       ; Wrong baud baud rate for sampling
           002C     30   009A          BSBW    SET_SPEED                ; Set to 9600
             26     11   009D          BRB     50$                      ; Exit
                         009F
                         009F  ; now test character
                         009F
        84   53     91   009F  20$:    CMPB    R3,(R4)+                 ; input same as table ?
             08     13   00A2          BEQL    30$                      ; branch to set speed if yes
             54     D6   00A4          INCL    R4                       ; increment past baud rate
```

```
               64  95  00A6          TSTB    (R4)                      ; end of list
               F5  18  00A8          BGEQ    20$                       ; loop if >= 0
               19  11  00AA          BRB     50$                       ; no match
                       00AC
                       00AC  ; Set speed, zero speed+1 to indicate not split speed
                       00AC
          54   64  9B  00AC  30$:    MOVZBW  (R4),R4                   ; set baud rate
               18  10  00AF          BSBB    SET_SPEED                 ; Set line speed
                       00B1
       64 A5   01  AA  00B1          BICW    #UCB$M_TIM,UCB$W_STS(R5)  ; Clear timeout expected
                       00B5          CLR_STATE <AUTOP>                 ; Clear AUTOBAUD pending
          53   0D  9A  00B9          MOVZBL  #TTY$C_CR,R3              ; Set carriage return as input
  00000000'GF      D0  00BC          MOVL    G^EXE$GL_ABSTIM,-
     00B0 C5           00C2                  UCB$L_TT_RDUE(R5)         ; Set time this path taken
                       00C5
                       00C5  40$:
                       00C5  ;
                       00C5  ; No appropriate mapping was found, make sure that character isn't a terminator
                       00C5  ;
          53   20  9A  00C5  50$:    MOVZBL  #TTY$C_BLANK,R3           ; Invalidate character
                   05  00C8          RSB
                       00C9
                       00C9  SET_SPEED:
     00F4 C5   54  9B  00C9          MOVZBW  R4,UCB$W_TT_SPEED(R5)     ; set speed, zero out speed+1
             09A8  30  00CE          BSBW    TTY$SET_LINE              ; Set line speed
                   05  00D1          RSB                               ; Return
```

```
                      00D2          .SBTTL  TTY$CANCELIO - CANCEL I/O ON TERMINAL
                      00D2
                      00D2     ;++
                      00D2     ; TTY$CANCELIO - CANCEL I/O, DEASSIGN, AND DEALLOCATE
                      00D2     ;
                      00D2     ; FUNCTIONAL DESCRIPTION:
                      00D2     ;
                      00D2     ; THIS ROUTINE IS ENTERED TO STOP CURRENT I/O  ON A TERMINAL UNIT.
                      00D2     ;
                      00D2     ; THE ASSOCIATED CHANNEL'S READ OR WRITE I/O IS CANCELED ALONG WITH
                      00D2     ; A FLUSH OF CONTROL-C AST REQUESTS.
                      00D2     ; IF THIS OPERATION IS DONE ON A UNIT WITH A ZERO REFERENCE COUNT, THE CONTROL
                      00D2     ; Y ASTS ARE FLUSHED, THE TYPEAHEAD BUFFER IS DEALLOCATED AND THE UNIT
                      00D2     ; IS INITIALIZED FOR THE NEXT USER.
                      00D2     ;
                      00D2     ; This routine always clears the control-O state bit, and also clears
                      00D2     ; the control-S state bit if the UCB reference count is zero.
                      00D2     ;
                      00D2     ; INPUTS:
                      00D2     ;
                      00D2     ;       R2 = NEGATIVE OF THE CHANNEL NUMBER
                      00D2     ;       R3 = CURRENT I/O PACKET ADDRESS
                      00D2     ;       R4 = PCB OF CANCELING PROCESS
                      00D2     ;       R5 = UCB ADDRESS
                      00D2     ;
                      00D2     ; OUTPUTS:
                      00D2     ;
                      00D2     ;       R4,R5 ARE PRESERVED
                      00D2     ;--
                      00D2
                      00D2     TTY$CANCELIO::                                  ; CANCELIO TTY USAGE
        53    52  D0  00D2             MOVL    R2,R3                           ; SAVE THE CHANNEL NUMBER
           FF28'  30  00D5             BSBW    TTY$LOCK                        ; SETUP IPL AND REGISTERS
                      00D8
       02D0 8F    BB  00D8             PUSHR   #^M<R4,R6,R7,R9>
        56    53  D0  00DC             MOVL    R3,R6                           ; COPY CHANNEL NUMBER
  53    58 A5     D0  00DF             MOVL    UCB$L_IRP(R5),R3                ; GET THE CURRENT IRP
        59    55  D0  00E3             MOVL    R5,R9                           ; SAVE LOGICAL UCB ADDRESS
  55  00A0 C5     D0  00E6             MOVL    UCB$L_TL_PHYUCB(R5),R5          ; GET THE PHYSICAL UCB
            03    12  00EB             BNEQ    5$                              ; CHECK AGAIN AFTER LOCKING THE DAT BASE
          0115    31  00ED             BRW     CANCEL_DETACH
   00000227'EF    9F  00F0     5$:     PUSHAB  CANCEL_DONE                     ; EXIT TO HERE WHEN FINISHED
                      00F6     ;
                      00F6     ; Empty the write queue.
                      00F6     ;
                      00F6
  51  00CC C5     9E  00F6             MOVAB   UCB$L_TT_WFLINK(R5),R1          ; Get first buffer in queue.
        57    51  D0  00FB             MOVL    R1,R7                           ; Copy buffer address.
                      00FE
                      00FE     10$:                                            ; Empty write queue.
        57    67  D0  00FE             MOVL    TTY$L_WB_FLINK(R7),R7           ; Get next buffer.
        51    57  D1  0101             CMPL    R7,R1                           ; Is this the end of queue?
            39    13  0104             BEQL    40$                             ; Yes. Branch past loop.
  50    24 A7     D0  0106             MOVL    TTY$L_WB_IRP(R7),R0             ; Get address of associated IRP.
            F2    13  010A             BEQL    10$                             ; Branch if broadcast buffer
                      010C                                                     ; (leave these in queue).
            04    E0  010C             BBS     #IRP$V_VIRTUAL,-                ; Don't dequeue virtual I/O
```

```
        ED 2A A0      010E                     IRP$W_STS(R0),10$         ; packet.
           0C A0  D1  0111         CMPL        IRP$L_PID(R0),-           ; Compare packet's PID with
           60 A4      0114                     PCB$L_PID(R4)             ; current process' PID.
              E6  12  0116         BNEQ        10$                       ; No match. Go to next packet.
     28 A0    56  B1  0118         CMPW        R6,IRP$W_CHAN(R0)         ; Does the channel match?
              E0  12  011C         BNEQ        10$                       ; No. Go to next packet.
              00  E1  011E         BBC         #IRP$V_BUFIO,-            ; Branch on direct I/O.
        04 2A A0      0120                     IRP$W_STS(R0),20$
              02  AA  0123         BICW        #IRP$M_FUNC,-             ; Otherwise, clear function of
           2A A0      0125                     IRP$W_STS(R0)             ; buffered I/O.
                      0127
                      0127  20$:                                         ; Set up I/O completion.
      0830 8F     3C  0127         MOVZWL      #SS$_CANCEL,-             ; Return CANCEL error status
           7C A5      012B                     UCB$W_BOFF(R5)            ; code.
              54  DD  012D         PUSHL       R4                        ; Save R4
     57    04 A7  D0  012F         MOVL        TTY$L_WB_BLINK(R7),R7     ; Get backward link.
     54    00 B7  0F  0133         REMQUE      @TTY$L_WB_FLINK(R7),R4    ; Remove buffer from queue.
           FEC6'  30  0137         BSBW        TTY$WRITEPOST             ; QUEUE IT FOR COMPLETION
           54 8ED0    013A         POPL        R4                        ; Restore R4
              BF  11  013D         BRB         10$                       ; Get next buffer in queue.
                      013F
                      013F  ;
                      013F  ; See if a read is in progress. Cancel it if it was initiated by the
                      013F  ; canceling process.
                      013F  ;
                      013F
                      013F  40$:
                      013F
                      013F                                               ; Check for read in progress.
              08  E1  013F         BBC         #UCB$V_BSY,-              ; Branch if no read in progress.
        29 64 A9      0141                     UCB$W_STS(R9),50$
OC A3    60 A4    D1  0144         CMPL        PCB$L_PID(R4),IRP$L_PID(R3); PID MATCH?
              22  12  0149         BNEQ        50$                       ; Branch if no match.
     28 A3    56  B1  014B         CMPW        R6,IRP$W_CHAN(R3)         ; CHANNEL MATCH?
              1C  12  014F         BNEQ        50$                       ; Branch if no match.
                      0151         IF_NOT_STATE -                        ; Proceed if this is not a read.
                      0151                     READ,50$
                      0155         CLR_STATE -                           ; Clear escape bits.
                      0155                     <ESC,BADESC>
              10  BB  015B         PUSHR       #^M<R4>                   ; Save the PCB address.
     54    58 A5  D0  015D         MOVL        UCB$L_IRP(R5),R4          ; Get address of IRP.
           38 A4  D4  0161         CLRL        IRP$L_MEDIA(R4)           ; Clear terminators.
              2C  3C  0164         MOVZWL      #SS$_ABORT,-              ; Set up abort status code.
           7C A5      0166                     UCB$W_BOFF(R5)
           FE95'  30  0168         BSBW        TTY$READONE               ; Complete the read.
                      016B
                      016B  45$:                                         ; Remember to restore register.
              10  BA  016B         POPR        #^M<R4>                   ; Restore PCB address.
                      016D
                      016D  50$:                                         ; Check for a write in progress.
                      016D         IF_NOT_STATE -                        ; If no write request is in
                      016D                     WRITE,60$                 ; progress, branch.
     57  00D4 C5  D0  0171         MOVL        UCB$L_TT_WRTBUF(R5),R7    ; Get address of write buffer.
     50    24 A7  D0  0176         MOVL        TTY$L_WB_IRP(R7),R0       ; Get associated IRP.
              25  13  017A         BEQL        60$                       ; no irp then don't cancel
           0C A0  D1  017C         CMPL        IRP$L_PID(R0),-           ; See if packet's PID matches
           60 A4      017F                     PCB$L_PID(R4)             ; process' PID.
```

```
            1E   12  0181            BNEQ    60$                        ; Branch if no match.
      28 A0 56   B1  0183            CMPW    R6,IRP$W_CHAN(R0)          ; See if channels match.
            18   12  0187            BNEQ    60$                        ; No. Don't stop this write.
          089A   30  0189            BSBW    TTY$ABORT                  ; Abort any current output activity
            2C   3C  018C            MOVZWL  #SS$_ABORT,-               ; Set abort status in the
         7C A5       018E                    UCB$W_BOFF(R5)             ; UCB status word.
            10   BB  0190            PUSHR   #^M<R4>                    ; Save the PCB address.
  53   00D4 C5   D0  0192            MOVL    UCB$L_TT_WRTBUF(R5),R3     ; Get address of write packet.
         7C A5   B0  0197            MOVW    UCB$W_BOFF(R5),-           ; Put status code in the
         28 A3       019A                    TTY$W_WB_STATUS(R3)        ; packet.
         FE61'   30  019C            BSBW    TTY$WRITEDONE              ; Complete the I/O.
            10   BA  019F            POPR    #^M<R4>                    ; Restore PCB address.
                     01A1
                     01A1   ;
                     01A1   ;        PROCESS CONTROL AND OUT OF BAND ASTS
                     01A1   ;
                     01A1
          002D   30  01A1   60$:     BSBW    CANCEL_AST                 ; CANCEL OUTSTANDING ASTS
                     01A4
                     01A4   ;
                     01A4   ; SEE IF THIS IS THE LAST DEASSIGN
                     01A4   ;
         5C A9   B5  01A4            TSTW    UCB$W_REFC(R9)             ; REF COUNT 0?
            10   12  01A7            BNEQ    65$
                     01A9
                     01A9
   64 A9   01    8A  01A9            BICB    #UCB$M_TIM,UCB$W_STS(R9); RESET HANGUP TIMER IN LUCB
                     01AD
            3C   E4  01AD            BBSC    #TTY$V_SX_RECONNECT,-
    12 00B8 C5       01AF                    UCB$Q_TT_STATE(R5),70$
                     01B3
          0079   30  01B3            BSBW    CANCEL_MODEM               ; SIGNAL HANGUP IF NEEDED
          009E   30  01B6            BSBW    CANCEL_RESET               ; RESET PHYSICAL UCB
                     01B9   65$:
  52   00B8 C5   DE  01B9            MOVAL   UCB$Q_TT_STATE(R5),R2      ; RESTORE STATE ADDRESS
                     01BE            CLR_STATE       <CTRLO>            ; Clear control-O state always.
         FE3B'   31  01C2            BRW     TTY$STARTOUTPUT            ; RESTART ANY STOPPED OUTPUT
                     01C5
                     01C5
                     01C5   70$:
            54   DD  01C5            PUSHL   R4                         ; PRESERVE R4
       54   05   D0  01C7            MOVL    #TTY$V_FD_LINK,R4          ; SCHEDULE LINK FORK FUNCTION
          045E   30  01CA            BSBW    TTY$CRE_FORK
         54 8ED0     01CD            POPL    R4
            05       01D0            RSB
```

```
                          01D1
                          01D1       ;
                          01D1       ;       CANCEL CONTROL AND OUT OF BAND ASTS
                          01D1       ;
                          01D1       ;       INPUTS: R9 - LUCB
                          01D1       ;
                          01D1
                          01D1  CANCEL_AST:
                          01D1       ;
                          01D1       ; RELEASE THE CONTROL C AST BLOCKS
                          01D1       ;
  57   0094 C9     9E     01D1          MOVAB   UCB$L_TL_CTRLC(R9),R7    ; ADDRESS THE CONTROL C ENABLE LIST
               52  D4     01D6          CLRL    R2                       ; SPECIFY NULL MASK
     00000000'GF   16     01D8          JSB     G^COM$FLUSHATTNS         ; FLUSH THE LIST
                          01DE       ;
                          01DE       ; FLUSH OUT OF BAND ASTS
                          01DE       ;
  57   009C C9     9E     01DE          MOVAB   UCB$L_TL_BANDQUE(R9),R7  ; ADDRESS THE OUT OF BAND LIST
  52   0098 C9     DE     01E3          MOVAL   UCB$L_TL_OUTBAND(R9),R2  ; GET CURRENT SUMMARY MASK
     00000000'GF   16     01E8          JSB     G^COM$FLUSHCTRLS         ; DO IT
                          01EE       ;
                          01EE       ; FLUSH CONTROL Y AST QUEUE
                          01EE       ;
       01     58   B1     01EE          CMPW    R8,#CAN$C_DASSGN         ; IS THIS A DEASSIGN?
              11   12     01F1          BNEQ    10$                      ; NO, SKIP
  57   0090 C9     9E     01F3          MOVAB   UCB$L_TL_CTRLY(R9),R7    ; ADDRESS LIST HEAD FOR CONTROL Y ENABLE
              52   D4     01F8          CLRL    R2                       ; SPECIFY NULL MASK
     00000000'GF   16     01FA          JSB     G^COM$FLUSHATTNS         ; FLUSH LIST
       00A4 C5     D4     0200          CLRL    UCB$L_TL_CTLPID(R5)      ; RESET CONTROLLING PID
                          0204  10$:
                   05     0204          RSB
                          0205
                          0205
```

```
                          0205
                          0205   ;
                          0205   ;       PROCESS CANCEL FOR DETACHED LOGICAL UCB.
                          0205   ;       THIS ROUTINE CANCELS ANY INTERNAL STRUCTURES FOR
                          0205   ;       A DETACHED LUCB.
                          0205   ;
                          0205   ;       INPUTS: R9 - LUCB
                          0205   ;
                          0205
                          0205   CANCEL_DETACH:
          55   59    D0   0205           MOVL    R9,R5                          ; RESTORE LOGICAL UCB ADDRESS
             FFC6    30   0208           BSBW    CANCEL_AST                     ; PROCESS OUTSTANDING CONTROL ASTS
            5C A5    B5   020B           TSTW    UCB$W_REFC(R5)                 ; CHECK FOR LAST DEASSIGN
               12    12   020E           BNEQ    10$
       64 A9   01    8A   0210           BICB    #UCB$M_TIM,UCB$W_STS(R9)       ; RESET HANGUP TIMER IN LUCB
            6C A9    D4   0214           CLRL    UCB$L_DUETIM(R9)               ; RESET HANGUP TIMEOUT VALUE IN LUCB
     50    0084 C5   D0   0217           MOVL    UCB$L_PDT(R5),R0               ; CHECK FOR PENDING REBIND FUNCTION
               04    13   021C           BEQL    10$                            ; NONE
          0084 C0    D4   021E           CLRL    UCB$L_PDT(R0)                  ; SIGNAL THAT THIS UCB IS DELETED
          02D0 8F    BA   0222   10$:    POPR    #^M<R4,R6,R7,R9>               ; Restore registers.
                     05   0226           RSB                                    ; RETURN
                          0227
                          0227
```

```
                 0227
                 0227
                 0227  CANCEL_DONE:
   55   59  D0   0227          MOVL    R9,R5                   ; RESTORE LOGICAL UCB ADDRESS
 02D0   8F  BA   022A          POPR    #^M<R4,R6,R7,R9>        ; Restore registers.
            05   022E          RSB                             ; RETURN
                 022F
                 022F
                 022F
```

```
                        022F   ;
                        022F   ;       FINAL MODEM PROCESSING FOR LAST DEASSIGN
                        022F   ;
                        022F   ;
                        022F   ;       INPUTS: R5 - PUCB
                        022F   ;
                        022F   CANCEL_MODEM:
                        022F
                        022F   ;
                        022F   ; CHECK FOR HANGUP
                        022F   ;
            02    E1    022F           BBC     #TT2$V_HANGUP,-
      03 48 A5          0231                   UCB$L_DEVDEPND2(R5),15$ ; BRANCH IF LOGOUT/NOHANGUP
          0C92    30    0234           BSBW    CLASS_MODEM_DIS         ; RESET SPEED AND UCB
                        0237   15$:
                        0237
                        0237   ;
                        0237   ; CHECK FOR NEED TO REACTIVATE MODEM PROCESSING
                        0237   ;
                        0237
   0E 44 A5    15    E0    0237           BBS     #TT$V_MODEM,UCB$L_DEVDEPEND(R5),20$     ; CURRENT STATE IS MODEM
 14 00C4 C5    15    E1    023C           BBC     #TT$V_MODEM,UCB$L_TT_DECHAR(R5),25$     ; NO CHANGE
         51    00    9A    0242           MOVZBL  #MODEM$C_INIT,R1                        ; NEED TO RESTART MODEM
       08F4    30          0245           BSBW    TRANSITION
         0C    11          0248           BRB     25$
                           024A
                           024A   20$:
 06 00C4 C5    15    E0    024A           BBS     #TT$V_MODEM,UCB$L_TT_DECHAR(R5),25$     ; NO CHANGE
         51    01    9A    0250           MOVZBL  #MODEM$C_SHUTDWN,R1                     ; SHUT DOWN MODEM
       08E6    30          0253           BSBW    TRANSITION
                           0256   25$:
                     05    0256           RSB
                           0257
```

```
                         0257  ;
                         0257  ;       RESET PHYSICAL UCB AND MAKE IT AVAILABLE FOR FUTURE USE
                         0257  ;
                         0257  ;
                         0257  ;       INPUTS: R5 - PUCB
                         0257  ;
                         0257  CANCEL_RESET:
                         0257
      68 A5    08    AA  0257          BICW    #UCB$M_TT_HANGUP,UCB$W_DEVSTS(R5); RESET HANGUP FLAG
                         025B                                                    ; NOTE THIS IS CORRECTLY REFERENCED
                         025B                                                    ; BUT IS NORMALLY MANIPULATED IN THE
                         025B  ;
                         025B  ;       DELETE TYPEAHEAD BUFFER
                         025B  ;
      50   00E4 C5   D0  025B          MOVL    UCB$L_TT_TYPAHD(R5),R0   ; GET ADDRESS OF TYPE AHEAD
              0A     13  0260          BEQL    10$                      ; NONE
           00E4 C5   D4  0262          CLRL    UCB$L_TT_TYPAHD(R5)
      00000000'GF    16  0266          JSB     G^COM$DRVDEALMEM         ; DEALLOCATE TYPE AHEAD BUFFER
            00FC     30  026C  10$:    BSBW    TTY$PURGE_AHEAD          ; PURGE TYPE AHEAD BUFFER
                         026F
                         026F  ;
                         026F  ; SET UP DISCONNECT SIGNAL TYPE FOR PORT
                         026F  ;
          7E   01    D0  026F          MOVL    #1,-(SP)                 ; INDICATE NOHANGUP
               02    E1  0272          BBC     #TT2$V_HANGUP,-
         02 48 A5        0274                  UCB$L_DEVDEPND2(R5),15$  ; BRANCH IF LOGOUT/NOHANGUP
               6E    D4  0277          CLRL    (SP)                     ; ASSUME HANGUP
                         0279  15$:
                         0279                                           ; OF LAST DEASSIGN
                         0279  ;
                         0279  ; RESTORE PERM CHARACTERISTICS
                         0279  ;
                         0279
           00A8 C5   7C  0279          CLRQ    UCB$Q_TL_BRKTHRU(R5)     ; RESET ALL THE BREAKTHRU CLASSES
              02E8   30  027D          BSBW    TTY$SETUP_UCB            ; SETUP UCB FIELDS, AND RESET PUCB
                         0280
      00000000'GF    D0  0280          MOVL    G^EXE$GL_ABSTIM,-
         00B0 C5         0286                  UCB$L_TT_RDUE(R5)        ; SET TIME TO ''TURN OFF'' AUTOBAUD
            07ED     30  0289          BSBW    TTY$SET_LINE             ; RESET SPEED AND UCB
                         028C
   00 68 A5    00    E5  028C          BBCC    #UCB$V_JOB,UCB$W_DEVSTS(R5),30$; RESET JOB CONTROLLER OWNERSHIP BIT
                         0291  30$:
       7E  0122 C5   B0  0291          MOVW    UCB$W_TT_PRTCTL(R5),-(SP); SAVE THE PORT CONTROL WORD
  0122 C5  0040 8F   AA  0296          BICW    #TTY$M_PC_XOFENA,UCB$W_TT_PRTCTL(R5); MAKE SURE THAT THE CONTROL
                         029D                                           ; AND THE DRIVER AGREE ON XOFF SETTING
            07C0     30  029D          BSBW    TTY$RESUME               ; CONTINUE ANY PORT OUTPUT
         0122 C5 8E  B0  02A0          MOVW    (SP)+,UCB$W_TT_PRTCTL(R5)
                         02A5
      52   00B8 C5   DE  02A5          MOVAL   UCB$Q_TT_STATE(R5),R2    ; RESTORE STATE ADDRESS
                         02AA          CLR_STATE       <CTRLO>          ; Clear control-O state always.
                         02AE
              50 8ED0    02AE          POPL    R0                       ; GET HANGUP INFO AS ARGUMENT TO
            0785     30  02B1          BSBW    TTY$DISCONNECT           ; INFORM THE PORT
              05         02B4          RSB
                         02B5
                         02B5
```

```
                              02B5      .SBTTL  TTY$SETUP_READ - Set up UCB and states for a read
                              02B5
                              02B5 ;++
                              02B5 ; TTY$SETUP_READ - get ready to initiate a read operation
                              02B5 ;
                              02B5 ; Functional description:
                              02B5 ;
                              02B5 ;      This routine copies IRP fields into the UCB, disables mailbox,
                              02B5 ;      sets up a timed-out read, cancels control-O, purges the type
                              02B5 ;      ahead buffer, etc. as instructed by the IRP.
                              02B5 ;
                              02B5 ;      This routine is called by both TTY$STARTIO to start a read, and
                              02B5 ;      TTY$RESTARTIO to restart a read after a powerfailure or after
                              02B5 ;      a write completes.
                              02B5 ;
                              02B5 ; Inputs:
                              02B5 ;
                              02B5 ;      R2      - address of the UCB state bits
                              02B5 ;      R3      - address of the IRP
                              02B5 ;      R5      - address of the UCB
                              02B5 ;
                              02B5 ; Outputs:
                              02B5 ;
                              02B5 ;      R0, R2, R3, and R5 must be preserved.
                              02B5 ;
                              02B5 ;      R1,R4 destroyed.
                              02B5 ;
                              02B5 ;--
                              02B5
                              02B5 TTY$SETUP_READ::
           54    2C A3    D0  02B5      MOVL    IRP$L_SVAPTE(R3),R4     ; Get buffered block address.
              00FC C5     B0  02B9      MOVW    UCB$W_TT_CURSOR(R5),-   ; Save the current cursor
                 3A A4        02BD              TTY$W_RB_CPZORG(R4)     ; position.
                              02BF
                              02BF ;
                              02BF ; Check to see if mailbox should be disabled.
                              02BF ;
                              02BF
        0D       4A A4    91  02BF      CMPB    TTY$A_RB_PRM(R4),#TTY$C_CR; IS THE FIRST CHARACTER A <CR>
                    03    12  02C3      BNEQ    5$                      ; NO THEN LEAVE CPZORG HERE
                 3A A4    B4  02C5      CLRW    TTY$W_RB_CPZORG(R4)     ; ELSE ZERO THE CURSOR POSITION
                    0A    E1  02C8 5$:  BBC     #IO$V_DSABLMBX,-        ; Branch forward if disable
              15 20 A3        02CA              IRP$W_FUNC(R3),10$      ; mailbox not requested.
                              02CD
          51   00C0 C5    D0  02CD      MOVL    UCB$L_TT_LOGUCB(R5),R1  ; GET LUCB ADDRESS
 44 A5   00010000 8F      C8  02D2      BISL    #TT$M_MBXDSABL,UCB$L_DEVDEPEND(R5)
 44 A1   00010000 8F      C8  02DA      BISL    #TT$M_MBXDSABL,UCB$L_DEVDEPEND(R1)
                              02E2 ;
                              02E2 ; Set up read with time out if necessary.
                              02E2 ;
                              02E2 10$:
                    07    E1  02E2      BBC     #IO$V_TIMED,-           ; Branch forward if read with
              1D 20 A3        02E4              IRP$W_FUNC(R3),30$      ; timeout not requested.
         00000000'EF      D0  02E7      MOVL    TTY$A_MAXTIME,-         ; Assume zero second timeout
              00B0 C5         02ED              UCB$L_TT_RDUE(R5)       ; (meaning just empty typeahead
                              02F0                                      ; buffer into read buffer).
           54    36 A4    3C  02F0      MOVZWL  TTY$W_RB_TIMOS(R4),R4   ; Get number of seconds to wait
```

```
                      02F4                                          ; between each character.
                0A 13 02F4          BEQL    20$                      ; Branch if none.
   00000000'GF  54 C1 02F6          ADDL3   R4,G^EXE$GL_ABSTIM,-     ; Calculate and store the due
           00B0 C5    02FD                  UCB$L_TT_RDUE(R5)        ; time.
                      0300
                      0300  20$:
                02 A8 0300          BISW    #UCB$M_TT_TIMO,-         ; Set the ''this is a read with
             68 A5    0302                  UCB$W_DEVSTS(R5)         ; timeout'' bit.
                      0304
                      0304  ;
                      0304  ; Always turn off control-O state.
                      0304  ;
                      0304
                      0304  30$:
                      0304          CLR_STATE -                      ; Always clear control-O state.
                      0304                  <CTRLO>
                      0308
                      0308  ;
                      0308  ; Handle a readsync terminal.
                      0308  ;
                      0308
                12 E1 0308          BBC     #TT$V_READSYNC,-         ; Branch forward if terminal is
          0F 44 A5    030A                  UCB$L_DEVDEPEND(R5),40$  ; not in readsync mode.
                      030D          IF_STATE -                       ; If in EOL state, don't issue
                      030D                  <TYPFUL,EOL>,40$         ; an XON code.
           07BA 30    0319          BSBW    TTY$XON                  ; Send XON
                      031C
                      031C  40$:
                   05 031C          RSB                              ; Return to caller.
```

```
                          031D         .SBTTL  TTY$NOTIFY - NOTIFY JOB CONTROLER OF LOGIN IF NECESSARY
                          031D  ;++
                          031D  ; TTY$NOTIFY - NOTIFY THE JOB CONTROLER OF LOGIN PENDING IF NECESSARY
                          031D  ;
                          031D  ; FUCNTIONAL DESCRIPTION:
                          031D  ;
                          031D  ; THIS ROUTINE WILL VALIDATE A REQUEST TO NOTIFY THE JOB CONTROLLER
                          031D  ; OF A LOGIN REQUEST. IF THERE IS A TYPEAHEAD BUFFER, THE JOB CONTROLER
                          031D  ; HAS NOT ALREADY BEEN NOTIFIED AND TERMINALS ARE ENABLED A FORK IS
                          031D  ; QUEUED TO PERFORM THE FUNCTION.
                          031D  ;
                          031D  ; INPUTS:
                          031D  ;
                          031D  ;         R5 = PHYSICAL UCB ADDRESS
                          031D  ;
                          031D  ; OUTPUTS:
                          031D  ;
                          031D  ;         ALL REGISTERS SAVED
                          031D  ;--
                          031D  TTY$NOTIFY::
              50     DD   031D          PUSHL   R0                      ; SAVE A SCRATCH REGISTER
    43 44 A5  02     E0   031F          BBS     #TT$V_NOTYPEAHD,UCB$L_DEVDEPEND(R5),100$; IF SLAVED TERMINAL, IGNORE
     50   00C0 C5    D0   0324          MOVL    UCB$L_TT_LOGUCB(R5),R0  ; GET LOGICAL UCB ADDRESS
           5C A0     B5   0329          TSTW    UCB$W_REFC(R0)          ; UNIT REF COUNT 0?
              11     13   032C          BEQL    10$                     ; IF EQL THEN JOB CONTROLLER POSSIBILITY
           60 A0     D5   032E          TSTL    UCB$L_AMB(R0)           ; USER ASSOCIATED MAILBOX?
              34     13   0331          BEQL    100$                    ; IF EQL THEN NO
    2F 44 A5  10     E0   0333          BBS     #TT$V_MBXDSABL,UCB$L_DEVDEPEND(R5),100$; BR IF NOT ENABLED
    2A 68 A5  02     E0   0338          BBS     #UCB$V_TT_NOTIF,UCB$W_DEVSTS(R5),100$; BR IF ALREADY NOTIFIED
              1E     11   033D          BRB     15$                     ; CONTINUE IN COMMON
    23 68 A5  00     E0   033F  10$:    BBS     #UCB$V_JOB,UCB$W_DEVSTS(R5),100$; IF ALREADY NOTIFIED THEN GO
    1E 38 A5  06     E0   0344          BBS     #DEV$V_SPL,UCB$L_DEVCHAR(R5),100$;IGNORE IF DEVICE IS SPOOLED
 50   00000000'GF    9E   0349          MOVAB   G^OPA$UCB0,R0           ; ADDRESS CONSOLE UCB
           55 50     D1   0350          CMPL    R0,R5                   ; IS THIS THE CONSOLE
              08     13   0353          BEQLU   15$                     ; IF EQL THEN SKIP TERMINAL ENABLED TEST
      00000068'GF    B5   0355          TSTW    G^SYS$GL_JOBCTLMB+UCB$W_DEVSTS; TERMINALS ENABLED FOR JOBCTLR?
              0A     15   035B          BLEQ    100$                    ; IF LEQ THEN DISMISS
              18     BB   035D  15$:    PUSHR   #^M<R3,R4>              ; SAVE THESE REGISTERS OVER THE FORK
           54 00     D0   035F          MOVL    #TTY$V_FD_UNSOL,R4      ; ASK FOR UNSOLICITED DATA FORK
            02C6     30   0362          BSBW    TTY$CRE_FORK            ; FORK TO SEND MESSAGE
              18     BA   0365  20$:    POPR    #^M<R3,R4>
                          0367
           50 8ED0        0367  100$:   POPL    R0
                     05   036A          RSB
```

```
                        036B         .SBTTL TTY$PURGE_AHEAD - PURGE TYPEAHEAD BUFFER
                        036B
                        036B     ;++
                        036B     ; TTY$PURGE_AHEAD - PURGE THE CONTENTS OF THE TYPEAHEAD BUFFER
                        036B     ;
                        036B     ; FUNCTIONAL DESCRIPTION:
                        036B     ;
                        036B     ; THIS ROUTINE RESETS THE TYPEAHEAD BUFFER POINTERS TO REFLECT NO
                        036B     ; DATA PRESENT.
                        036B     ;
                        036B     ; INPUTS:
                        036B     ;
                        036B     ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                        036B     ;       R5 = UCB ADDRESS
                        036B     ;
                        036B     ; OUTPUTS:
                        036B     ;
                        036B     ;       R4 = ADDRESS OF TYPEAHEAD BUFFER STRUCTURE
                        036B     ;       R2,R3,R5 ARE PRESERVED.
                        036B     ;--
                        036B
                        036B     TTY$PURGE_AHEAD::                               ; PURGE TYPEAHEAD
      54   00E4 C5  D0  036B             MOVL    UCB$L_TT_TYPAHD(R5),R4          ; GET TYPE AHEAD BUFFER ADDRESS
              1A  13    0370             BEQL    55$                             ; IF EQL THEN NONE
           0C A4  B4    0372             CLRW    TTY$W_TA_INAHD(R4)              ; SET NO DATA IN TYPEAHEAD BUFFER
      64   0118 C4  9E  0375             MOVAB   TTY$L_TA_DATA(R4),TTY$L_TA_PUT(R4); RESET PUT POINTER
   04 A4   0118 C4  9E  037A             MOVAB   TTY$L_TA_DATA(R4),TTY$L_TA_GET(R4); RESET GET POINTER
 07 04 A2      0C   E5  0380     50$:    BBCC    #TTY$V_ST_TYPFUL,4(R2),55$; TYPE AHEAD FULL AND INPUT STOPPED?
                        0385             CLR_STATE <OVRFLO>                      ; RESET OVERFLOW CONDITION
            074A  30    0389             BSBW    TTY$XON                         ; SEND XON
                  05    038C     55$:    RSB
```

```
                          038D          .SBTTL  TTY$READERROR - Line error occured.
                          038D  ;++
                          038D  ; TTY$READERROR - READ ERROR OCCURED ON LINE
                          038D  ;
                          038D  ; Functional description:
                          038D  ;
                          038D  ; This routine is called from a terminal port driver to report
                          038D  ; a read error on a line. It completes the read with error in the
                          038D  ; event a read is active, or just returns if no read is active.
                          038D  ;
                          038D  ; Inputs:
                          038D  ;
                          038D  ;       IPL = Device IPL
                          038D  ;       R5 = UCB address
                          038D  ;
                          038D  ; Outputs:
                          038D  ;
                          038D  ;       R0,R1,R2,R3 are destroyed.
                          038D  ;       R4,R5 are preserved.
                          038D  ;--
                          038D          .ENABLE LSB
                          038D
                          038D  TTY$READERROR::
                  11  BB  038D          PUSHR   #^M<R0,R4>                      ; SAVE REGISTERS
      54   01F4 8F    3C  038F          MOVZWL  #SS$_PARITY,R4                  ; ASSUME PARITY
      52   00B8 C5    9E  0394          MOVAB   UCB$Q_TT_STATE(R5),R2           ; ADDRESS STATE DATA
        09 53   0C    E0  0399          BBS     #12,R3,10$                      ; BRANCH IF PARITY ERROR
        7E 53   0D    E0  039D          BBS     #13,R3,TTY$FRAMERROR            ; BRANCH IF FRAME ERROR
      54   0838 8F    3C  03A1          MOVZWL  #SS$_DATAOVERUN,R4              ; OVERUN ERROR
      51   00C0 C5    D0  03A6  10$:    MOVL    UCB$L_TT_LOGUCB(R5),R1          ; Get logical UCB  address
              5C A1   B5  03AB          TSTW    UCB$W_REFC(R1)                  ; Terminal in use ?
                  27  12  03AE          BNEQ    12$                             ; Branch if yes
     22 48 A5     01  E1  03B0          BBC     #TT2$V_AUTOBAUD,UCB$L_DEVDEPND2(R5),12$; Check AUTOBAUD
           53     0D  91  03B5          CMPB    #TTY$C_CR,R3                    ; IS THIS CHARACTER A <CR>
                  06  13  03B8          BEQL    11$                             ; YES THEN DO AUTOPARITY
                FCAF  30  03BA          BSBW    TTY$AUTOBAUD                    ; YES THEN JUMP THERE
                0042  31  03BD          BRW     15$                             ; RETURN TO THE USER
                          03C0
   0B 00F8 C5     06  E5  03C0  11$:    BBCC    #UCB$V_TT_PARTY,UCB$B_TT_PARITY(R5),70$; CLEAR PARITY
   05 00F8 C5     02  E0  03C6          BBS     #UCB$V_TT_USERFRAME,UCB$B_TT_PARITY(R5),70$; DON'T CHANGE THE FRAME
                          03CC                                                  ; IF THE USER SPECIFYED IT.
      00F8 C5     18  88  03CC          BISB    #UCB$M_TT_LEN,UCB$B_TT_PARITY(R5); SET EIGHT BIT FRAME SIZE
                06A5  30  03D1  70$:    BSBW    TTY$SET_LINE                    ; THEN TELL THE PORT
                002B  31  03D4          BRW     15$                             ; NOW EXIT
                          03D7
                          03D7  12$:    IF_NOT_STATE    READ,90$                ; SKIP IF NO READ ACTIVE
   09 00F8 C5     01  E1  03DB          BBC     #UCB$V_TT_DISPARERR,UCB$B_TT_PARITY(R5),13$
 000001F4 8F      54  D1  03E1          CMPL    R4,#SS$_PARITY                  ; IS THIS A PARITY ERROR
                  18  13  03E8          BEQL    15$                             ; YES THEN IGNORE IT
         7C A5    54  B0  03EA  13$:    MOVW    R4,UCB$W_BOFF(R5)               ; SET ERROR CODE TO RETURN
               FC0F'  30  03EE          BSBW    TTY$READONE                     ; COMPLETE THE REQUEST
                  0F  11  03F1          BRB     15$
                          03F3
 00000838 8F      54  D1  03F3  90$:    CMPL    R4,#SS$_DATAOVERUN              ; IS THIS A DATA OVERRUN
                  06  12  03FA          BNEQ    15$                             ; NO THEN EXIT
                          03FC          SET_STATE <TYPFUL,OVRFLO>               ; YES THEN MAKE SURE IT
                          0402                                                  ; IS SIGNALED
```

```
                      0402   954 15$:
           11   BA    0402   955          POPR    #^M<R0,R4>                      ; RESTORE REGISTERS
      010B C5   94    0404   956          CLRB    UCB$B_TT_OUTYPE(R5)             ; ASSUME NO CHARACTER TO OUTPUT
           53   95    0408   957          TSTB    R3                              ; TEST FOR DATA
           05   13    040A   958          BEQL    16$                             ; BRANCH IF ZERO
 010B C5   01   90    040C   959          MOVB    #1,UCB$B_TT_OUTYPE(R5)          ; SET CHARACTER VALID
           05         0411   960 16$:     RSB                                     ; RETURN TO PORT
                      0412   961
                      0412   962 18$:
 54  01F4  8F   3C    0412   963          MOVZWL  #SS$_PARITY,R4                  ; SET ERROR TYPE
           06   E1    0417   964          BBC     #TT$V_PARITY,-
  E5 00F8  C5         0419   965                  UCB$B_TT_PARITY(R5),15$         ; IF NOT ENABLED, THEN DROP
           87   11    041D   966          BRB     10$                             ; DROP CHARACTER, NO ERROR
```

```
                          041F          .SBTTL  TTY$FRAMERROR - FRAME ERROR ON A READ
                          041F  ;++
                          041F  ; TTY$FRAMERROR - FRAME READ ERROR OCCURED ON LINE
                          041F  ;
                          041F  ; Functional description:
                          041F  ;
                          041F  ; This routine is branched to from a TTY$READERROR to report a frame
                          041F  ; read error on a line. It compares time against the
                          041F  ; last autobaud attempt, and if a "sufficient" (avg 0.5 seconds) has
                          041F  ; passed, the sampling baud rate is toggled between 9600 and 600.
                          041F  ; The toggling to 600 baud will time out in less than 3 seconds.
                          041F  ;
                          041F  ; Inputs:
                          041F  ;
                          041F  ;       R0,R4 saved on stack
                          041F  ;       IPL = Device IPL
                          041F  ;       R5 = UCB address
                          041F  ;
                          041F  ; Outputs:
                          041F  ;
                          041F  ;       Baud rate may change.
                          041F  ;       R4,R5 are preserved.
                          041F  ;--
                          041F  TTY$FRAMERROR:
                          041F
                          041F  ; Check if we should attempt autobaud on this line
                          041F
 54    00C0 C5      D0    041F          MOVL    UCB$L_TT_LOGUCB(R5),R4  ; Get logical UCB  address
            5C A4   B5    0424          TSTW    UCB$W_REFC(R4)          ; Terminal in use ?
               0F   13    0427          BEQL    40$                     ; Branch if yes
                          0429
E4 48 A5       10   E1    0429          BBC     #TT2$V_SECURE,UCB$L_DEVDEPND2(R5),18$   ; NOT SECURE THEN
                          042E                                          ; GIVE ERROR
               53   95    042E          TSTB    R3                      ; NOT NULL DATA THEN NOT A BREAK
               E0   12    0430          BNEQ    18$                     ; SO GIVE AN ERROR
             0A9F   30    0432          BSBW    TTY$CLASS_DISCONNECT    ; ELSE SIGNAL HANGUP STATUS
             FFCA   31    0435          BRW     15$                     ; THEN RETURN NOTHING
                          0438
0A 48 A5       01   E0    0438  40$:    BBS     #TT2$V_AUTOBAUD,UCB$L_DEVDEPND2(R5),43$ ; Check AUTOBAUD
               53   95    043D          TSTB    R3                      ; BREAK?
               D1   12    043F          BNEQ    18$                     ; NO THEN GIVE ERROR
             FED9   30    0441          BSBW    TTY$NOTIFY              ; LET THE JOB CONTROLER KNOW ABOUT
             FFBB   31    0444          BRW     15$                     ; THE BREAK
                          0447
                          0447  ; check time to see that it's been "awhile".
                          0447  ; note that you want to throw away multiple-character, single-keystroke
                          0447  ; characters after the first. This check gives between 0 and one second.
                          0447  ; It doesn't seem necessary to make it be at least one second.
                          0447  43$:
          00B0 C5   D1    0447          CMPL    UCB$L_TT_RDUE(R5),-
        00000000'GF       044B                  G^EXE$GL_ABSTIM         ; check against current
               C0   13    0450          BEQL    18$                     ; exit if equal
                          0452
                          0452  ; make sure there is no valid data in R3, if so, leave it for TTY$AUTOBAUD and
                          0452  ; don't toggle baud rate
                          0452
010B C5        01   90    0452          MOVB    #1,UCB$B_TT_OUTYPE(R5)  ; ASSUME GOOD DATA
```

```
          53   95  0457  1025           TSTB    R3                      ; test
          34   12  0459  1026           BNEQ    30$                     ; Branch if data there
   010B   C5   94  045B  1027           CLRB    UCB$B_TT_OUTYPE(R5)     ; BUT WE FOUND NO GOOD DATA
                   045F  1028
                   045F  1029 ; time is OK - continue by checking baud rate
                   045F  1030
     54   0F   9A  045F  1031           MOVZBL  #TT$C_BAUD_9600,R4      ; Assume set to 9600 baud
          0F   91  0462  1032           CMPB    #TT$C_BAUD_9600,-
   00F4   C5       0464  1033                   UCB$W_TT_SPEED(R5)      ; At 9600 ?
          1A   12  0467  1034           BNEQ    20$                     ; no, set 9600
                   0469  1035 ;
                   0469  1036 ; Toggle to 600 baud for 2 to 3 seconds
                   0469  1037 ;
     54   07   9A  0469  1038           MOVZBL  #TT$C_BAUD_600,R4       ; otherwise, set 600 baud
                   046C  1039 80$:      SET_STATE -
                   046C  1040                   <AUTOP>                 ; Set state autobaud pending
                   0470  1041           TIMSET  2
                   0483  1042
        FC43   30  0483  1043 20$:      BSBW    SET_SPEED               ; Set line speed
 00000000'GF   D0  0486  1044           MOVL    G^EXE$GL_ABSTIM,-
     00B0 C5       048C  1045                   UCB$L_TT_RDUE(R5)       ; set time
        FF70   31  048F  1046 30$:      BRW     15$                     ; and exit via main module
                   0492  1047
                   0492  1048
                   0492  1049           .DISABLE LSB
```

```
                          0492         .SBTTL  TTY$RESTARTIO - RESTART CURRENT I/O
                          0492
                          0492  ;
                          0492  ;++
                          0492  ; TTY$RESTARTIO - RESTART CURRENT I/O
                          0492  ;
                          0492  ; FUNCTIONAL DESCRIPTION:
                          0492  ;
                          0492  ;       This routine restarts an interrupted read or write request. Such
                          0492  ;       requests may have been interrupted due to a powerfail or
                          0492  ;       to coincident read and write requests.
                          0492  ;
                          0492  ; INPUTS:
                          0492  ;
                          0492  ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                          0492  ;       R5 = UNIT UCB ADDRESS
                          0492  ;
                          0492  ; OUTPUTS:
                          0492  ;
                          0492  ;       R2,R5 ARE PRESERVED.
                          0492  ;--
                          0492
                          0492  TTY$RESTARTIO::                                 ; RESTART THE I/O
                          0492         IF_NOT_STATE WRITE,10$                   ; IF NOT WRITE THEN CONTINUE
                          0496         IF_STATE WRTALL,30$                      ; If passall mode, no special
                          049A                                                  ; restart is required.
       54    00D4 C5  D0  049A         MOVL    UCB$L_TT_WRTBUF(R5),R4           ; Get address of write buffer.
                0584  30  049F         BSBW    TTY$ABORT                        ; Abort any output activity
               2A A4  B4  04A2         CLRW    TTY$W_WB_BCNT(R4)                ; No. Clear output byte count.
               30 A4  9E  04A5         MOVAB   TTY$L_WB_DATA(R4),-              ; And reset the start of data
               1C A4      04A8                 TTY$L_WB_NEXT(R4)                ; pointer.
                      05  04AA         RSB                                      ; Return.
                          04AB
                          04AB  10$:
                          04AB         IF_NOT_STATE READ,30$                    ; BR IF READ NOT IN PROGRESS
       53      58 A5  D0  04AF         MOVL    UCB$L_IRP(R5),R3                 ; Get address of IRP.
                          04B3         IF_STATE RDVERIFY,40$                    ; NOTHING SPECIAL FOR READ VERIFY
                FDFB  30  04B7         BSBW    TTY$SETUP_READ                   ; Setup for the read operation.
                          04BA         IF_STATE -                               ; Continue without special
                          04BA                 PASALL,20$                       ; handling if PASSALL
                          04BE         IF_NOT_STATE -                           ; or NOECHO ...
                          04BE                 NOECHO,15$                       ;
    1D 48 A5      00  E1  04C2         BBC     #TT2$V_LOCALECHO, -              ; without local echo.
                          04C7                 UCB$L_DEVDEPND2(R5),20$          ;
                          04C7  15$:   IF_NOT_STATE -                           ; Also no special handling if
                          04C7                 REFRSH,30$                       ; REFRESH state is not set.
             0001'CF  9E  04CB         MOVAB   W^TTY$A_CTRLU+1,-                ; GET A <CR> TO MULTIECHO
             00D8 C5      04CF                 UCB$L_TT_MULTI(R5)
     00DC C5      01  B0  04D2         MOVW    #1,UCB$W_TT_MULTILEN(R5)                 ; AND ONLY 1 CHARACTER HERE.
                          04D7         CLR_STATE -
                          04D7                 <CTRLR,SKIPCRLF>
                          04DB         SET_STATE -                              ; Set the control-R and
                          04DB                 <EDITREAD,MULTI,SKIPLF>; send a linefeed and then control-r the read
                          04E4  20$:
                  0D  E0  04E4         BBS     #IO$V_REFRESH,-                  ; If the read specified a
            04 20 A3      04E6                 IRP$W_FUNC(R3),30$               ; refresh, don't cancel it.
                          04E9         CLR_STATE -                              ; Otherwise, cancel refresh
```

```
                     04E9                 REFRSH                          ; state.
                     04ED
                     04ED  30$:
                 05  04ED                 RSB                             ; CONTINUE
 7C A5   02D4 8F  B0  04EE  40$:   MOVW    #SS$_OPINCOMPL,UCB$W_BOFF(R5) ; Set I/O completion status
    54     78 A5  D0  04F4         MOVL    UCB$L_SVAPTE(R5),R4     ; GET THE READ BUFFER ADDRESS
           38 A3  D4  04F8         CLRL    IRP$L_MEDIA(R3)         ; ZERO ALL OTHER READ VERIFY FIELDS
           30 A4  90  04FB         MOVB    TTY$W_RB_LINOFF(R4),-   ; PUT THE INDEX INTO THE IOSB
           3B A3      04FE                 IRP$L_MEDIA+3(R3)
           FAFD'  31  0500         BRW     TTY$READONE             ; Go complete the read
                      0503
```

```
                         0503        .SBTTL  TTY$RTIMOU - READ TIMED OUT ENTRY
                         0503
                         0503  ;++
                         0503  ; TTY$RTIMOU - Read timed out entry (from time dependent schedular)
                         0503  ;
                         0503  ; Functional description:
                         0503  ;
                         0503  ; This routine is called from the time dependent schedular via the UCB
                         0503  ; offset UCB$L_TT_RTIMOU whenever a timed read times out.  It completes
                         0503  ; the read with time SS$_TIMEOUT error and, then, continues terminal
                         0503  ; processing.
                         0503  ;
                         0503  ; Inputs:
                         0503  ;
                         0503  ;       IPL = IPL$_POWER
                         0503  ;       R5 = UCB address
                         0503  ;
                         0503  ; Outputs:
                         0503  ;
                         0503  ;       R0,R1,R2,R3,R4 are destroyed.
                         0503  ;       R5 is preserved.
                         0503  ;--
                         0503
                         0503  TTY$RTIMOU:                                   ; Read timed out entry
                         0503          SETIPL  #IPL$_TIMER                   ; REDUCE IPL TO BEFORE LOCKING THE DATA BASE
            FAF7'   30   0506          BSBW    TTY$LOCK                      ; MAKE SURE THE UCB IS LOCKED
   52   00B8 C5     9E   0509          MOVAB   UCB$Q_TT_STATE(R5),R2         ; Address state vector
                         050E          IF_NOT_STATE READ,20$                 ; IF THE AST IS LATE (I.E. THE READ IS
                         0512                                                ; NO LONGER ACTIVE) THEN EXIT.
7C A5   022C 8F     B0   0512          MOVW    #SS$_TIMEOUT,UCB$W_BOFF(R5) ; Set I/O completion status
                         0518          IF_NOT_STATE RDVERIFY,5$              ; NOTHING SPECIAL FOR OTHER THAN READ VERIFY
     53    58 A5    D0   051C          MOVL    UCB$L_IRP(R5),R3              ; GET THE IRP ADDRESS IN R3
     54    78 A5    D0   0520          MOVL    UCB$L_SVAPTE(R5),R4           ; GET THE READ BUFFER ADDRESS
           38 A3    D4   0524          CLRL    IRP$L_MEDIA(R3)               ; ZERO ALL OTHER READ VERIFY FIELDS
           30 A4    90   0527          MOVB    TTY$W_RB_LINOFF(R4),-         ; PUT THE INDEX INTO THE IOSB
           3B A3         052A                  IRP$L_MEDIA+3(R3)
            FAD1'   30   052C  5$:     BSBW    TTY$READONE                   ; Go complete the read
            FACE'   31   052F  10$:    BRW     TTY$STARTOUTPUT               ; then startup any waiting activit)
                    05   0532  20$:    RSB
                         0533
```

```
                        0533            .SBTTL TTY$VT_TIMEOUT - DETACHED TERMINAL TIMEOUT
                        0533
                        0533 ;++
                        0533 ; TTY$VT_TIMEOUT - DETACHED TERMINAL TIMEOUT ROUTINE
                        0533 ;
                        0533 ; FUNCTIONAL DESCRIPTION:
                        0533 ;
                        0533 ; THIS ROUTINE IS CALLED BY THE SYSTEM TIMER SERVICE WHEN
                        0533 ; A LUCB HAS BEEN DETACHED FOR A SPECIFIED PERIOD OF TIME
                        0533 ; WITHOUT BEING REATTACHED. A HANGUP AST IS FIRED FOR THAT
                        0533 ; LUCB TO GET THE PROCESS KILLED. ALL PENDING IO ON THE INPUT QUEUE
                        0533 ; IS COMPLETED WITH SS$_HANGUP STATUS. IN THIS STATE THERE SHOULD BE
                        0533 ; NO CURRENT READ OR WRITE. LAST DEASSIGN CODE IN CANCEL
                        0533 ; WILL CLEAN UP THE LUCB, AND DEASSIGN WILL DELETE IT BACK TO POOL.
                        0533 ;
                        0533 ; INPUTS:
                        0533 ;          INTERLOCKED AT DIPL
                        0533 ;          R5 = LUCB ADDRESS
                        0533 ;
                        0533 ;--
                 0002'  0533            .WORD   TTY$VT_TIMEOUT-.
                        0535 TTY$VT_TIMEOUT:
            56    DD    0535            PUSHL   R6                      ; SAVE SCRATCH REGISTER
        56  55    D0    0537            MOVL    R5,R6                   ; LUCB REQUIRED IN R6
53    0084 C6    D0    053A            MOVL    UCB$L_PDT(R6),R3; GET ADDRESS OF ANY REBINDING PUCB
            04    13    053F            BEQL    5$                      ; NONE
       0084 C3    D4    0541            CLRL    UCB$L_PDT(R3)           ; CANCEL ANY PENDING REBIND OPERATION
            10    AA    0545 5$:        BICW    #UCB$M_ONLINE,-         ; SET LUCB OFFLINE TO PREVENT ANY MORE IO
         64 A6          0547                    UCB$W_STS(R6)
          019B    30    0549            BSBW    DISC_FIREAST            ; SIGNAL HANGUP AST TO OWNER
53      4C B6    0F    054C 10$:       REMQUE  @UCB$L_IOQFL(R6),R3     ; DRAIN IO QUEUE
            12    1D    0550            BVS     20$                     ; NO MORE PACKETS QUEUED
       02CC 8F    3C    0552            MOVZWL  #SS$_HANGUP,-           ; FILL IN COMPLETION STATUS
         38 A3          0556                    IRP$L_MEDIA(R3)
   2A A3  02    AA    0558            BICW    #IRP$M_FUNC,IRP$W_STS(R3) ; RESET ANY BUFFERED READS
  00000000'GF    16    055C            JSB     G^COM$POST              ; AND COMPLETE THE IO
            E8    11    0562            BRB     10$
                        0564
         56 8ED0        0564 20$:       POPL    R6
            05          0567            RSB
                        0568
                        0568
```

```
                                  0568          .SBTTL  TTY$SETUP_UCB - SETUP TERMINAL UCB
                                  0568
                                  0568  ;++
                                  0568  ; TTY$SETUP_UCB - SETUP TERMINAL UCB
                                  0568  ;
                                  0568  ; FUNCTIONAL DESCRIPTION:
                                  0568  ;
                                  0568  ; THIS ROUTINE IS ENTERED DURING SYSTEM STARTUP TO SET UP TERMINAL UCBS.
                                  0568  ; THE UCB IS ZEROED EXCEPT FOR THE SPEED AND FILL COUNTS. THE CURSOR IS
                                  0568  ; SET TO 1 TO FORCE A CR-LF. THE HOLDING TANK IS INVALIDATED. THE FORK BLOCK
                                  0568  ; IS INITIALIZED.
                                  0568  ;
                                  0568  ; If the write queue is not yet initialized (indicated by zeroes in the
                                  0568  ; listhead), initialize the queue.
                                  0568  ;
                                  0568  ; Initialize the read timed out dispatch (UCB$L_TT_RTIMOU).
                                  0568  ;
                                  0568  ; INPUTS:
                                  0568  ;
                                  0568  ;       R5 = UCB ADDRESS
                                  0568  ;
                                  0568  ; OUTPUTS:
                                  0568  ;
                                  0568  ;       R5 = UCB ADDRESS
                                  0568  ;
                                  0568  ;       R0-R4 ARE PRESERVED.
                                  0568  ;--
                                  0568
                                  0568  TTY$SETUP_UCB::                                 ; SET UP THE UCB (PORT INTERFACE)
                       50   DD    0568          PUSHL   R0
           50   00C0 C5     D0    056A          MOVL    UCB$L_TT_LOGUCB(R5),R0  ; GET THE LOGICAL UCB ADDRESS
                       09   13    056F          BEQL    3$                      ; NONE THEN INIT THE UCB
                    5C A0   B5    0571          TSTW    UCB$W_REFC(R0)          ; IS THIS UCB ACTIVE?
                       04   13    0574          BEQL    3$                      ; NO THEN INIT IT
                       50 8ED0    0576          POPL    R0                      ; IF ACTIVE THEN RETURN NORMALY
                            05    0579          RSB
                                  057A  3$:
              00C0 C5  55   D0    057A          MOVL    R5,UCB$L_TT_LOGUCB(R5)  ; INIT LOGICAL UCB ADDRESS
              00A0 C5  55   D0    057F          MOVL    R5,UCB$L_TL_PHYUCB(R5)  ; INIT PHYSICAL UCB
                                  0584
           00040000 8F      C8    0584          BISL    #DEV$M_AVL,-            ; SHOW PUCB AVALABLE FOR USE
                    38 A5         058A                  UCB$L_DEVCHAR(R5)
   3C A5   00000100 8F      CA    058C          BICL    #DEV$M_RED,UCB$L_DEVCHAR2(R5) ; SHO PUCB NOT REDIRECTED
                                  0594
                  00A4 C5   D4    0594          CLRL    UCB$L_TL_CTLPID(R5)     ; CLEAN OUT A LEFT OVER PID
              00FC C5  01   B0    0598          MOVW    #1,UCB$W_TT_CURSOR(R5)  ; SET BAD CURSOR TO FORCE CRLF
                  0108 C5   B4    059D          CLRW    UCB$W_TT_HOLD(R5)       ; INIT HOLDING TANK
         0C A5   064D'CF    9E    05A1          MOVAB   W^TTY$FORK_ADDR,UCB$L_FPC(R5); SET UP FORK ADDRESS
       00F4 C5   00E8 C5    D0    05A7          MOVL    UCB$W_TT_DESPEE(R5),UCB$W_TT_SPEED(R5); SET UP TEMP CHARS
       00F8 C5   00EC C5    90    05AE          MOVB    UCB$B_TT_DEPARI(R5),UCB$B_TT_PARITY(R5);SET UP DEFAULT PARITY
         41 A5   00F0 C5    D0    05B5          MOVL    UCB$B_TT_DETYPE(R5),UCB$B_DEVTYPE(R5);
                                  05BB
    50   44 A5   FFFFDFFF 8F   CB 05BB          BICL3   #^C<TT$M_REMOTE>,UCB$L_DEVDEPEND(R5),R0; SAVE CURRENT REMOTE STATE
 44 A5   00C4 C5  00002000 8F  CB 05C4          BICL3   #TT$M_REMOTE,UCB$L_TT_DECHAR(R5),UCB$L_DEVDEPEND(R5);
                   44 A5  50   C8 05CF          BISL    R0,UCB$L_DEVDEPEND(R5)  ; KEEP PREVIOUS SETTING
    50   44 A5      FC 8F      8B 05D3          BICB3   #^C<TT$M_PASSALL!TT$M_NOECHO>,UCB$L_DEVDEPEND(R5),R0; GET PASSALL AN
                                  05D9
```

```
                                05D9  ;           Set up default DMA characteristics
                                05D9
      48 A5    00C8 C5    D0    05D9              MOVL    UCB$L_TT_DECHA1(R5),UCB$L_DEVDEPND2(R5) ; SET UP DEFAULT CHAR WORD 2
         1E 48 A5    06   E1    05DF              BBC     #TT2$V_DMA,UCB$L_DEVDEPND2(R5),8$        ; SKIP IF NOT SET
      13 0122 C5     02   E0    05E4              BBS     #TTY$V_PC_DMAAVL,UCB$W_TT_PRTCTL(R5),5$  ; DMA FEATURE AVAILABLE
  00C8 C5  00000040 8F    CA    05EA              BICL    #TT2$M_DMA,UCB$L_TT_DECHAT(R5)           ; SHOW DMA NOT AVAILABLE
     48 A5  00000040 8F   CA    05F3              BICL    #TT2$M_DMA,UCB$L_DEVDEPND2(R5)           ; IF DMA IS NOT AVAILABLE
                     05   11    05FB              BRB     8$
                                05FD  5$:
          0122 C5    02   A8    05FD              BISW    #TTY$M_PC_DMAENA,UCB$W_TT_PRTCTL(R5)     ; ENABLE DMA USAGE
                                0602  8$:
                                0602
  00BC C5    02    02    50   F0    0602          INSV    R0,#TTY$V_ST_PASALL,#2,UCB$Q_TT_STATE+4(R5); RESET CURRENT STATE
                         50 8ED0    0609          POPL    R0
                                060C
                                060C  ;
                                060C  ; Initialize the write queue.
                                060C  ;
                                060C
             00CC C5      D5    060C              TSTL    UCB$L_TT_WFLINK(R5)        ; Is the queue initialized?
                  0E      12    0610              BNEQ    10$                        ; Yes. Don't redo it.
             00CC C5      9E    0612              MOVAB   UCB$L_TT_WFLINK(R5),-      ; Initialize queue to point to
             00CC C5            0616                      UCB$L_TT_WFLINK(R5)        ; itself in both directions.
             00CC C5      9E    0619              MOVAB   UCB$L_TT_WFLINK(R5),-      ; Now the backward link.
             00D0 C5            061D                      UCB$L_TT_WBLINK(R5)
                                0620
                                0620  10$:
                                0620
                                0620  ;
                                0620  ; Initialize read timed out dispatch.
                                0620  ;
                                0620
     00B4 C5    FEDF CF   9E    0620              MOVAB   TTY$RTIMOU,UCB$L_TT_RTIMOU(R5) ; Set read timed out dispatch
                                0627
                          05    0627              RSB                                ; Return.
```

```
                0628          .SBTTL  TTY$CLASS_FORK - PORT DRIVER FORK ROUTINE
                0628  ;++
                0628  ; TTY$CLASS_FORK
                0628  ;
                0628  ; DESCRIPTION
                0628  ;
                0628  ;       WILL CAUSE A FORK TO BE CREATED FOR A PORT DRIVER.
                0628  ; WHEN THE FORK IS CREATED THE DRIVER IS CALLED BACK AT IT'S PORT_FORKRET
                0628  ; DISPATCH VECTOR.
                0628  ;
                0628  ; INPUTS:
                0628  ;
                0628  ;       R2 = UNIT STATE VECTOR
                0628  ;       R5 = UCB ADDRESS
                0628  ;
                0628  ; OUTPUTS:
                0628  ;       R4,R3 DESTROYED
                0628  ;       R2,R5 ARE PRESERVED
                0628  ;--
                0628  TTY$CLASS_FORK::                        ; CREATE A FORK FOR THE TERMINAL PORT
54    03   D0   0628          MOVL    #TTY$V_FD_PORTFORK,R4   ; SET THE PORT FORK BIT.
                062B                                          ; AND FALL INTO CREATING THE FORK
```

```
                          062B          .SBTTL  TTY$CRE_FORK - General purpose driver fork routine
                          062B      ;++
                          062B      ; TTY$CRE_FORK - CREATE FORK ROUTINE
                          062B      ;
                          062B      ; FUNCTIONAL DESCRIPTION:
                          062B      ;
                          062B      ; THIS ROUTINE FORKS IN THE UCB FOR EITHER OF TWO REASONS:
                          062B      ;
                          062B      ;       1. UNSOLICITED DATA - SEND MESSAGE TO OWNER
                          062B      ;       2. NO TYPEAHEAD BUFFER - ALLOCATE ONE
                          062B      ;
                          062B      ; INPUTS:
                          062B      ;
                          062B      ;       R2 = UNIT STATE VECTOR ADDRESS
                          062B      ;       R3 = CHARACTER TO BUFFER FOR TYPEAHEAD NEEDED
                          062B      ;       R4 = FORK REASON STATE BIT
                          062B      ;       R5 = UCB ADDRESS
                          062B      ;
                          062B      ; OUTPUTS:
                          062B      ;
                          062B      ;       R3,R4 ARE DESTROYED
                          062B      ;       R2,R5 ARE PRESERVED
                          062B      ;--
                          062B      TTY$CRE_FORK::                                         ; CREATE FORK
15 14 A5     06     E2    062B              BBSS    #TTY$V_FD_BUSY,UCB$L_FR4(R5),20$; ARE WE BUSY
00 14 A5     54     E2    0630              BBSS    R4,UCB$L_FR4(R5),3$             ; SET THE APROPRIATE BIT
   54     14 A5     D0    0635      3$:     MOVL    UCB$L_FR4(R5),R4                ; AND GET THE BIT INTO R4
          2C A5     DD    0639      5$:     PUSHL   UCB$L_PID(R5)                   ; SAVE THE PID ON THE STACK
          4D'AF     9F    063C              PUSHAB  B^TTY$FORK_ADDR                 ; SET FORK ROUTINE ADDRESS
   00000000'EF      17    063F              JMP     TTY$SYNCH                       ; CREATE FORK PROCESS
                          0645      ;
                          0645      ; FORK ALREADY ACTIVE
                          0645      ;
00 14 A5     54     E2    0645      20$:    BBSS    R4,UCB$L_FR4(R5),25$            ; SET THE APROPRIATE BIT
                    05    064A      25$:    RSB                                     ; AND RETURN TO THE CALLER
```

```
                          064B
                          064B               .SBTTL  FORK DISPATCHER
                   035B'  064B               .WORD   ATTENTION-.                 ; SET UP ATTENTION INTERRUPT
                          064D  TTY$FORK_ADDR::
                          064D  ;
                          064D  ; ATTENTION FORK PROCESS
                          064D  ;
        52   00B8 C5   9E 064D  10$:        MOVAB   UCB$Q_TT_STATE(R5),R2       ; ADDRESS UNIT STATE
                          0652
                          0652               DSBINT  UCB$B_DIPL(R5)              ; DISABLE INTERUPTS FOR THE CHECK
   54   14 A5   07   00   EA 0659            FFS     #0,#TTY$V_FD_BUSY+1,UCB$L_FR4(R5),R4; FIND THE FIRST BIT SET
     00 14 A5   54   E4   065F               BBSC    R4,UCB$L_FR4(R5),15$        ; CLEAR THAT BIT
                          0664  15$:        ENBINT                              ; ENABLE INTERUPTS
                          0667
             E3 AF   DF   0667               PUSHAL  10$                         ; SAVE A RETURN ADDRESS
                          066A               CASE    R4,TYPE=B,<-                ; AND DISPATCH TO THE NECESSARY ROUTINE
                          066A               UNSOL -
                          066A               GETAHD -
                          066A               DISCONNECT -
                          066A               PORTFORK -
                          066A               UNLINKFORK -
                          066A               LINKFORK -
                          066A               FORKEXIT>
                          067C
                          067C  FORKEXIT:
              54 8ED0     067C               POPL    R4                          ; POP OFF  THE ADDRESS OF THE DISPATCHER
                   05     067F               RSB                                 ; THEN RETURN
                          0680
                          0680  PORTFORK:
        51   0118 C5   D0 0680               MOVL    UCB$L_TT_PORT(R5),R1        ; GET THE PORT ADDRESS
             34 B1     17 0685               JMP     @PORT_FORKRET(R1)           ; AND CALL THE PORTS FORK ADDRESS
                          0688
                          0688
```

```
                         0688          .SBTTL  FORK TO DISCONNECT THE TERMINAL AND DELIVER THE HANGUP AST
                         0688  ;++
                         0688  ;DISCONNECT
                         0688  ;
                         0688  ; Descrpition:
                         0688  ;         Delivers all AST's, aborts all IO and then sends a terminal hangup
                         0688  ; message to the job controler.
                         0688  ;
                         0688  ; Inputs:
                         0688  ;         R2 = Unit state vector
                         0688  ;         R5 = PUcb address
                         0688  ;
                         0688  ; Outputs:
                         0688  ;
                         0688  ;         R5 = PRESERVED
                         0688  ;--
                         0688  DISCONNECT:
        0040 8F    BB    0688          PUSHR   #^M<R6>
   56   00C0 C5    D0    068C          MOVL    UCB$L_TT_LOGUCB(R5),R6  ; GET LOGICAL UCB ADDRESS
                         0691
                         0691          DSBINT  UCB$B_DIPL(R5)
        038B       30    0698          BSBW    TTY$ABORT               ; ABORT ANY OUTPUT,
        03C2       30    069B          BSBW    TTY$RESUME              ; AND RESET CONTROL S
                         069E          ENBINT
                         06A1
 00 44 A5   0D     E5    06A1          BBCC    #TT$V_REMOTE,UCB$L_DEVDEPEND(R5),10$     ;RESET REMOTE TERMINAL
 44 A6   44 A5     D0    06A6  10$:    MOVL    UCB$L_DEVDEPEND(R5),UCB$L_DEVDEPEND(R6) ;UPDATE MODIFIED FIELD
                         06AB
 08 48 A5   11     E1    06AB          BBC     #TT2$V_DISCONNECT,UCB$L_DEVDEPND2(R5),15$;SKIP IF DISCONNECT NOT AVA
      56    55     D1    06B0          CMPL    R5,R6                                    ; MAKE SURE ITS REASONABLE
            03     13    06B3          BEQL    15$                                      ; NO, LUCB/PUCB THE SAME
        0028       31    06B5          BRW     50$
                         06B8
                         06B8  15$:
        002C       30    06B8          BSBW    DISC_FIREAST                             ; FIRE HANGUP AST
 7C A5  02CC 8F    3C    06BB          MOVZWL  #SS$_HANGUP,UCB$W_BOFF(R5)               ; SET THE RETURN STATUS
        0062       30    06C1          BSBW    DISC_STOPIO                              ; STOP CURRENT I/O
                         06C4
                         06C4
      54    06     D0    06C4          MOVL    #MSG$_TRMHANGUP,R4      ; SET MESSAGE TYPE
      55    56     D0    06C7          MOVL    R6,R5                   ; SWITCH TO LUCB CONTEXT
   53    60 A5     D0    06CA          MOVL    UCB$L_AMB(R5),R3        ; GET ASSOC MAILBOX
            06     13    06CE          BEQL    20$                     ; IF EQL THEN NONE
   00000000'GF     16    06D0          JSB     G^EXE$SNDEVMSG          ; INSERT MAILBOX MESSAGE
                         06D6
   55   00A0 C5    D0    06D6  20$:    MOVL    UCB$L_TL_PHYUCB(R5),R5  ; SWITCH BACK TO PUCB CONTEXT
        0040 8F    BA    06DB          POPR    #^M<R6>
                   05    06DF          RSB
                         06E0
                         06E0  50$:
        0040 8F    BA    06E0          POPR    #^M<R6>                 ; RESTORE SAVED REGISTERS
        016D       31    06E4          BRW     UNLINKFORK              ; AND DO UNLINK OPERATION
```

```
                          06E7   ;
                          06E7   ;       FIRE HANGUP AST
                          06E7   ;
                          06E7   ;       THIS ROUTINE DELIVERS ALL OUTSTANDING ^Y ASTS WITH
                          06E7   ;       HANGUP STATUS TO REQUEST THE PROCESS BE RUN DOWN. IN THE
                          06E7   ;       EVENT NONE ARE CURRENTLY QUEUED, THEN A FLAG IS SET TO
                          06E7   ;       SIGNAL THIS CONDITION WHEN THE NEXT ^Y AST IS QUEUED.
                          06E7   ;       *NOTE* THIS ROUTINE MAY NOT UTILIZE ANY FIELDS IN THE PUCB
                          06E7   ;       AS IT IS ALSO USED BY DISCONNECTED LOGICAL UCB'S.
                          06E7   ;
                          06E7   ;       INPUTS:
                          06E7   ;               R6 - LUCB
                          06E7   ;       OUTPUTS:
                          06E7   ;               R5,R6 PRESERVED
                          06E7
                          06E7   DISC_FIREAST:
                          06E7           DSBINT  UCB$B_DIPL(R6)          ; INTERLOCK TO DEVICE IPL
      54   0090 C6    9E  06EE           MOVAB   UCB$L_TL_CTRLY(R6),R4   ; ADDRESS CONTROL Y AST LIST
           50   54    D0  06F3           MOVL    R4,R0                   ; COPY LIST ADDRESS
   00A8 C6     10 A0  D0  06F6           MOVL    ACB$L_AST(R0),UCB$Q_TL_BRKTHRU(R6)
                60    D5  06FC           TSTL    (R0)                    ;LIST HEAD EMPTY?
                09    12  06FE           BNEQ    5$                      ;NO,PROCESS HANGUP ASTS
           5C A6      B5  0700           TSTW    UCB$W_REFC(R6)          ;IS THIS UNIT OWNED OR ACTIVE?
                          0703                                           ;(OR IS IT LOGGED OUT)
                04    13  0703           BEQL    5$                      ;NOT ACTIVE,SO SKIP FLAG SETTING
                08    A8  0705           BISW    #UCB$M_TT_HANGUP,-
           68 A6          0707                   UCB$W_DEVSTS(R6)        ;SET HANGUP FLAG TO NOTIFY NEXT CTRL-Y
                          0709                                           ;ATTENTION AST WHEN IT IS QUEUED.
           50   60    D0  0709   5$:     MOVL    (R0),R0                 ; GET NEXT ENTRY
                0E    13  070C           BEQL    10$                     ; IF EQL THEN DONE
   00AC C6     10 A0  D0  070E           MOVL    ACB$L_AST(R0),UCB$Q_TL_BRKTHRU+4(R6)
   1C A0     02CC 8F  3C  0714           MOVZWL  #SS$_HANGUP,ACB$L_KAST+4(R0); INSERT NEW PARAM FOR AST
                ED    11  071A           BRB     5$                      ; CONTINUE UNTIL DONE
       00000000'GF    16  071C   10$:    JSB     G^COM$DELATTNAST        ; DELIVER THE AST'S
                          0722           ENBINT
                      05  0725           RSB
                          0726
                          0726
```

```
                     0726
                     0726      ;       ABORT CURRENT AND QUEUED READS AND WRITES
                     0726      ;
                     0726      ;       THIS ROUTINE IS USED TO ABORT INTERNALLY QUEUED WRITES AND
                     0726      ;       ANY ACTIVE READ. IT RUNS IN THE CONTEXT OF THE PUCB.
                     0726      ;
                     0726      ;       INPUT:  R5 - PUCB
                     0726      ;       OUPUT:  R5 - PRESERVED
                     0726
                     0726      DISC_STOPIO:
                     0726              DSBINT  UCB$B_DIPL(R5)             ; INTERLOCK TO DEVICE IPL
52   00B8 C5   9E    072D              MOVAB   UCB$Q_TT_STATE(R5),R2      ; SET UP LUCB STATE POINTER
51   00CC C5   9E    0732              MOVAB   UCB$L_TT_WFLINK(R5),R1     ; Get start of write queue.
     53   51   D0    0737              MOVL    R1,R3                      ; Copy it to check for end.
                     073A
                     073A      10$:
     53   63   D0    073A              MOVL    TTY$L_WB_FLINK(R3),R3      ; Get next entry.
     51   53   D1    073D              CMPL    R3,R1                      ; At end of queue?
          0D   13    0740              BEQL    15$                        ; Yes. Branch forward.
                     0742                                                 ; leave it in the queue.
53     04 A3   D0    0742              MOVL    TTY$L_WB_BLINK(R3),R3      ; Get entry's backward link.
54     00 B3   0F    0746              REMQUE  @TTY$L_WB_FLINK(R3),R4     ; Remove the link.
       F8B3'   30    074A              BSBW    TTY$WRITEPOST              ; Queue it for completion
          EB   11    074D              BRB     10$                        ; Try for next link.
                     074F
                     074F      15$:                                       ; Stop the current I/Os.
                     074F              IF_NOT_STATE -                     ; Branch forward if no read is
                     074F                      READ,20$                   ; currently in progress.
                     0753              CLR_STATE -                        ; Clear escape bits.
                     0753                      <ESC,BADESC>
54     58 A5   D0    0759              MOVL    UCB$L_IRP(R5),R4           ; Get address of IRP.
       38 A4   D4    075D              CLRL    IRP$L_MEDIA(R4)            ; Clear terminators.
       F89D'   30    0760              BSBW    TTY$READONE                ; Complete the read.
                     0763      20$:                                       ; Check for a write to complete.
                     0763              IF_NOT_STATE -                     ; Branch forward if no write is
                     0763                      WRITE,30$                  ; in progress.
53   00D4 C5   D0    0767              MOVL    UCB$L_TT_WRTBUF(R5),R3     ; Get address of write packet.
       7C A5   B0    076C              MOVW    UCB$W_BOFF(R5),-           ; Put status code in the
       28 A3         076F                      TTY$W_WB_STATUS(R3)        ; packet.
       F88C'   30    0771              BSBW    TTY$WRITEDONE              ; Complete the I/O.
                     0774      30$:
                     0774              ENBINT
          05         0777              RSB
                     0778
```

```
                              0778          .SBTTL  FORK TO CHECK FOR TYPEAHEAD BUFFER FETCH
                              0778  ;++
                              0778  ; GETAHD
                              0778  ;
                              0778  ; Description:
                              0778  ;       Will create a typeahead buffer for the user, then insert
                              0778  ; the character that caused the buffer to be allocated and cause
                              0778  ; that character to be processed.
                              0778  ;
                              0778  ; Inputs:
                              0778  ;
                              0778  ;       R2 = UNIT STATE VECTOR
                              0778  ;       R5 = UCB ADDRESS
                              0778  ;       UCB$L_FR3 = CHARACTER THAT CAUSED TYPEAHEAD BUFFER TO BE CREATED
                              0778  ;
                              0778  ; Outputs:
                              0778  ;
                              0778  ;       R1,R4 are Destroyed
                              0778  ;       R2, R5 are Preserved
                              0778  ;--
                              0778  GETAHD:
          00E4 C5        D5   0778          TSTL    UCB$L_TT_TYPAHD(R5)           ; ALREADY ALLOCATED?
                64       12   077C          BNEQ    100$                          ; IF NEQ THEN YES
                07       E1   077E          BBC     #TT2$V_ALTYPEAHD,-            ; SKIP IF NORMAL SIZE
           09 48 A5           0780                  UCB$L_DEVDEPND2(R5),72$
51    00000000'GF        3C   0783          MOVZWL  G^TTY$GW_ALTYPAHD,R1          ; ALLOCATE THE TYPEAHEAD BUFFER
                07       11   078A          BRB     73$
51    00000000'GF        3C   078C  72$:    MOVZWL  G^TTY$GW_TYPAHDSZ,R1          ; ALLOCATE THE TYPEAHEAD BUFFER
51    00000118 8F        C0   0793  73$:    ADDL    #TTY$L_TA_DATA,R1             ; PLUS SPACE FOR HEADER
      00000000'GF        16   079A          JSB     G^EXE$ALONONPAGED             ; ALLOCATE THE TYPEAHEAD BUFFER
             54 52       D0   07A0          MOVL    R2,R4                         ; COPY BLOCK ADDRESS
             3C 50       E9   07A3          BLBC    R0,100$                       ; IF FAILURE THEN TOUGH
          08 A4 51       B0   07A6          MOVW    R1,TTY$W_TA_SIZE(R4)          ; SET UP SIZE
      10 A4   6441       9E   07AA          MOVAB   (R4)[R1],TTY$L_TA_END(R4);    SET UP END POINTER
          0A A4 14       90   07AF          MOVB    #DYN$C_TYPAHD,TTY$B_TA_TYPE(R4);
             14 A4       B4   07B3          CLRW    TTY$W_TA_RCLSIZ(R4)           ; CLEAR OUT THE RECALL SIZE
                              07B6          DSBINT  UCB$B_DIPL(R5)                ; LOCK TERMINAL DATA BASE
       00E4 C5 54        D0   07BD          MOVL    R4,UCB$L_TT_TYPAHD(R5)        ; STORE ADDRESS
      52   00B8 C5       9E   07C2          MOVAB   UCB$Q_TT_STATE(R5),R2         ; RESTORE THE UNIT STATE VECTOR ADDRESS
              FBA1       30   07C7          BSBW    TTY$PURGE_AHEAD               ; SET UP THE BLOCK
  0118 C4    10 A5       90   07CA          MOVB    UCB$L_FR3(R5),TTY$L_TA_DATA(R4); INSERT DATA
                08       13   07D0          BEQL    80$                           ; SKIP INSERT IF NULL
                64       D6   07D2          INCL    TTY$L_TA_PUT(R4)              ; POINT TO NEXT PLACE
             0C A4       B6   07D4          INCW    TTY$W_TA_INAHD(R4)            ; AND BUMP THE COUNT
             F826'       30   07D7          BSBW    TTY$STARTOUTPUT               ; BEGIN NEXT STATE
   00 14 A5   01         E4   07DA  80$:    BBSC    #TTY$V_FD_GETAHD,UCB$L_FR4(R5),90$; CLEAN THE BIT OUT AGAIN
                              07DF  90$:    ENBINT                                ; LOWER IPL
                         05   07E2  100$:   RSB                                   ; THEN RETURN
```

```
                                07E3          .SBTTL LINKFORK - RECONNECT LUCB/PUCB
                                07E3
                                07E3  ;++
                                07E3  ; LINKFORK - LINK LOGICAL AND PHYSICAL UCB
                                07E3  ;
                                07E3  ; FUNCTIONAL DESCRIPTION:
                                07E3  ;
                                07E3  ;       THIS FORK ROUTINE IS QUEUED TO LINK A PUCB AND LUCB.
                                07E3  ;       THE LUCB ADDRESS TO BE LINKED TO IS CONTAINED IN PUCB
                                07E3  ;       OFFSET UCB$L_PDT. THIS ROUTINE RESETS BSY IN UCB$W_STS
                                07E3  ;       AND RESTARTS PENDING IRP'S IN THE I/O QUEUE.
                                07E3  ;
                                07E3  ; INPUTS:
                                07E3  ;
                                07E3  ;       R5 = PUCB ADDRESS
                                07E3  ;
                                07E3  ;--
                                07E3  LINKFORK:
        54    0084 C5   D0      07E3          MOVL    UCB$L_PDT(R5),R4                        ; GET SPECIFIED LUCB
     0B A4      0B A5   90      07E8          MOVB    UCB$B_FIPL(R5),UCB$B_FIPL(R4)           ; FORK IPL
     5E A4      5E A5   90      07ED          MOVB    UCB$B_DIPL(R5),UCB$B_DIPL(R4)           ; DEVICE IPL
        64 A4      01   8A      07F2          BICB    #UCB$M_TIM,UCB$W_STS(R4)                ; AND TURN OFF TIMER BIT IN
                                07F6
  48 A5   00020000 8F   C8      07F6          BISL    #TT2$M_DISCONNECT,UCB$L_DEVDEPND2(R5)   ; SET DISCONNECT BIT
                                07FE                                                          ; BECAUSE WE ARE DISCONNECTA
     44 A4      44 A5   D0      07FE          MOVL    UCB$L_DEVDEPEND(R5),UCB$L_DEVDEPEND(R4) ; MAKE SURE LUCB IS
     48 A4      48 A5   D0      0803          MOVL    UCB$L_DEVDEPND2(R5),UCB$L_DEVDEPND2(R4) ; IMAGE COPY OF PUCB
                                0808
     00A0 C4       55   D0      0808          MOVL    R5,UCB$L_TL_PHYUCB(R4)                  ; LINK LUCB TO PUCB
     00C0 C5       54   D0      080D          MOVL    R4,UCB$L_TT_LOGUCB(R5)                  ; LINK PUCB TO LUCB
     00A0 C5       55   D0      0812          MOVL    R5,UCB$L_TL_PHYUCB(R5)                  ; SHOW PUCB=PUCB
             0084 C4    D4      0817          CLRL    UCB$L_PDT(R4)                           ; RESET REBIND FIELDS
             0084 C5    D4      081B          CLRL    UCB$L_PDT(R5)                           ; RESET REBIND FIELDS
                                081F
        3C A4      02   CA      081F          BICL    #DEV$M_DET,UCB$L_DEVCHAR2(R4)           ; LUCB NOW NOT DETACHED
         00040000 8F    CA      0823          BICL    #DEV$M_AVL,-                            ; SHOW PUCB NOT AVALABLE FOR
                38 A5           0829                  UCB$L_DEVCHAR(R5)
  3C A5   00000100 8F   C8      082B          BISL    #DEV$M_RED,UCB$L_DEVCHAR2(R5)           ; SHO PUCB REDIRECTED
                                0833
           55      54   D0      0833          MOVL    R4,R5                                   ; SWITCH TO LUCB CONTEXT
        53      4C B5   0F      0836          REMQUE  @UCB$L_IOQFL(R5),R3                     ; GET A PACKET OFF QUEUE
                   0C   1D      083A          BVS     10$                                     ; QUEUE EMPTY
          0000084E'EF   DF      083C          PUSHAL  20$                                     ; SAVE RETURN ADDRESS
          00000000'GF   17      0842          JMP     G^IOC$INITIATE                          ; START THE PACKET
     64 A5    0100 8F   AA      0848  10$:    BICW    #UCB$M_BSY,UCB$W_STS(R5)                ; RESET BSY BIT IN LUCB
                                084E                                                          ; TO ALLOW NEW OUTPUT
        55    00A0 C5   D0      084E  20$:    MOVL    UCB$L_TL_PHYUCB(R5),R5                  ; SWITCH BACK TO PUCB
                        05      0853          RSB                                             ; DONE
                                0854
```

```
                          0854          .SBTTL UNLINKFORK - DISCONNECT LUCB/PUCB
                          0854
                          0854  ;++
                          0854  ; UNLINKFORK - UNLINK LOGICAL AND PHYSICAL UCB
                          0854  ;
                          0854  ; FUNCTIONAL DESCRIPTION:
                          0854  ;
                          0854  ;       THIS FORK ROUTINE IS QUEUED TO ALLOW ANY READS AND WRITE FORKS
                          0854  ;       TO FINISH PRIOR TO UNLINKING LOGICAL AND PHYSICAL UCBS.
                          0854  ;       CURRENTLY QUEUED WRITES ARE COMPLETED WITH SS$_NORMAL. ANY
                          0854  ;       CURRENT READ, IS INSERTED AT THE HEAD OF THE I/O QUEUE. THE
                          0854  ;       UCB "BUSY" BIT IS SET TO PREVENT ANY NEW I/O FROM STARTING.
                          0854  ;
                          0854  ; INPUTS:
                          0854  ;
                          0854  ;       R5 = PUCB ADDRESS
                          0854  ;
                          0854  ;--
                          0854  UNLINKFORK:
       54  00C0 C5    D0  0854          MOVL    UCB$L_TT_LOGUCB(R5),R4          ; GET LOGICAL UCB ADDRESS
       52  00B8 C5    9E  0859          MOVAB   UCB$Q_TT_STATE(R5),R2           ; GET STATE ADDRESS
                          085E
                          085E  ;       CHECK FOR ANY ACTIVE I/O
                          085E          IF_STATE READ,40$
                          0862          IF_STATE WRITE,30$
                          0866
                          0866  ;       THE LUCB IS IDLE, SO WE FORCE BSY TO PREVENT ANY NEW I/O FROM
                          0866  ;       BEING STARTED.  THIS WILL ALLOW THE LUCB TO BE DETACHED, AND THE
                          0866  ;       PUCB TO BE FREE FOR FUTURE CONNECTS
                          0866
    64 A4  0100 8F    A8  0866          BISW    #UCB$M_BSY,UCB$W_STS(R4)        ; FORCE BSY
              58 A4   D4  086C          CLRL    UCB$L_IRP(R4)                   ; SHOW NO I/O ACTIVE
            00A0 C4   D4  086F          CLRL    UCB$L_TL_PHYUCB(R4)             ; AND BREAK LINK TO PUCB
    00C0 C5      55   D0  0873          MOVL    R5,UCB$L_TT_LOGUCB(R5)          ; AND MAKE PUCB=LUCB
      3C A4      02   C8  0878          BISL    #DEV$M_DET,UCB$L_DEVCHAR2(R4)   ; LUCB IS NOW DETACHED
                          087C
                          087C  ;       INIT TIMER ON LUCB TO INSURE IT DISCONNECTS
                          087C  ;       IF NOONE RECLAIMS IT WITHIN 30 MINUTES
                          087C
    0C A4   FCB5 CF   9E  087C          MOVAB   TTY$VT_TIMEOUT,UCB$L_FPC(R4)    ; LOAD VECTOR FOR TIMEOUT ENTRY
00000000'GF  00000000'GF  C1  0882      ADDL3   G^TTY$GL_TIMEOUT,G^EXE$GL_ABSTIM,-; AND SET TIMER
               6C A4      088D                  UCB$L_DUETIM(R4)
      64 A4      01   88  088F          BISB    #UCB$M_TIM,UCB$W_STS(R4)        ; AND TURN ON TIMER BIT IN LUCB
                          0893
                          0893
                          0893  ;       IF REBINDING STATE, THEN GOTO LINKFORK
                 3C   E4  0893          BBSC    #TTY$V_SX_RECONNECT,-
        08 00B8 C5        0895                  UCB$Q_TT_STATE(R5),20$
                          0899
              F993    30  0899          BSBW    CANCEL_MODEM                    ; CANCEL THE MODEM (HANGUP IF
                          089C                                                  ; NECESSARY)
            F9B7 CF   16  089C          JSB     CANCEL_RESET                    ; GO RESET PUCB. THIS WILL ALSO
                          08A0                                                  ; MAKE IT AVAILABLE
                      05  08A0          RSB
                          08A1
              FF3F    31  08A1  20$:    BRW     LINKFORK                        ; GO LINK PUCB TO REQUESTED LUCB
                          08A4
```

```
                    08A4
                    08A4    ;       STOP IO ON PUCB PRIOR TO DISCONNECT
                    08A4    30$:
  7C A5    01   3C  08A4            MOVZWL  #SS$_NORMAL,UCB$W_BOFF(R5)  ; SET THE RETURN STATUS
         FE7B   30  08A8            BSBW    DISC_STOPIO                 ; STOP CURRENT IO
           06   E5  08AB            BBCC    #TTY$V_FD_BUSY,-            ; FORCE NEW FORK TO QUEUE BEHIND ALL OTHERS
     00 14 A5       08AD                    UCB$L_FR4(R5),35$
     54    04   D0  08B0    35$:    MOVL    #TTY$V_FD_UNLINK,R4         ; AND QUEUE THE FORK
         FD75   30  08B3            BSBW    TTY$CRE_FORK
       54 8ED0      08B6            POPL    R4                          ; POP OFF TERMINAL FORK CONTROL RETURN PC
           05       08B9            RSB                                 ; AND EXIT
                    08BA
                    08BA    ;       REQUEUE ANY ACTIVE READ TO THE HEAD OF THE QUEUE
                    08BA    40$:
                    08BA            DSBINT  UCB$B_DIPL(R5)              ; INTERLOCK WITH DEVICE INTERRUPTS
  53    58 A5   D0  08C1            MOVL    UCB$L_IRP(R5),R3            ; GET IRP OF ACTIVE READ
                    08C5
                    08C5    ;       INIT READ STATUS FIELDS
                    08C5
16 68 A5   01   E4  08C5            BBSC    #UCB$V_TT_TIMO,UCB$W_DEVSTS(R5),50$; RESET ANY TIMED READ STATE
                    08CA
  4C A4    63   0E  08CA            INSQUE  (R3),UCB$L_IOQFL(R4)        ; AND PUT AT HEAD OF THE QUEUE
                    08CE
                    08CE            CLR_STATE -
                    08CE                    <READ,DEL,EOL,-             ; RESET READ STATE BITS, BUT LEAVE ''BSY'' SET
                    08CE                    PROMPT,CTRLR,NOFLTR,-
                    08CE                    ESC,ESC_O,BADESC,PRE,-
                    08CE                    REFRSH,EDITREAD,SKIPCRLF,RDVERIFY,-
                    08CE                    MULTI,RECALL,OVERSTRIKE,EDITING,-
                    08CE                    QUOTING,BACKSPACE,TERMNORM>
                    08DB
                    08DB    45$:    ENBINT
           C4   11  08DE            BRB     30$                         ; AND FULSH COMPLETE ANY WRITES
                    08E0
7C A5  02CC 8F  3C  08E0    50$:    MOVZWL  #SS$_HANGUP,UCB$W_BOFF(R5)              ; SET THE RETURN STATUS
        F717'   30  08E6            BSBW    TTY$READONE                 ; AND COMPLETE THE READ
           F0   11  08E9            BRB     45$
                    08EB
                    08EB
```

```
                            08EB                .SBTTL  FORK TO SEND UNSOLICITED DATA MESSAGE
                            08EB         ;++
                            08EB         ; UNSOL
                            08EB         ;
                            08EB         ; Description:
                            08EB         ;       Will send a message to the job controler indicating that
                            08EB         ; unsoliced input has been recieved
                            08EB         ;
                            08EB         ; Note: This routine runs in context of the logical UCB.
                            08EB         ;
                            08EB         ; INPUTS:
                            08EB         ;       R2 = UNIT STATE VECTOR
                            08EB         ;       R5 = UCB ADDRESS
                            08EB         ;
                            08EB         ; OUTPUTS:
                            08EB         ;       R3 IS DESTROYED
                            08EB         ;       R2,R5 ARE PRESERVED
                            08EB         ;
                            08EB         ; REGISTER USAGE:
                            08EB         ;       R5 = LOGICAL UCB
                            08EB         ;       R6 = PHYSICAL UCB
                            08EB         ;--
                            08EB         UNSOL:
            0060 8F     BB  08EB                PUSHR   #^M<R5,R6>                  ; SAVE REGISTERS
      55    00C0 C5     D0  08EF                MOVL    UCB$L_TT_LOGUCB(R5),R5      ; LOAD LUCB ADDRESS
      56    00A0 C5     D0  08F4                MOVL    UCB$L_TL_PHYUCB(R5),R6      ; LOAD PUCB ADDRESS
                            08F9
         54     01      9A  08F9                MOVZBL  #MSG$_TRMUNSOLIC,R4         ; SET MESSAGE TYPE
             5C A5      B5  08FC                TSTW    UCB$W_REFC(R5)              ; IF REF COUNT 0 THEN
                0D      13  08FF                BEQL    50$                         ; JOB CONTROLLER
         53   60 A5     D0  0901                MOVL    UCB$L_AMB(R5),R3            ; GET USER'S MAILBOX
                33      13  0905                BEQL    65$                         ; IF EQL THEN NONE
      2E 68 A6  02      E0  0907                BBS     #UCB$V_TT_NOTIF,UCB$W_DEVSTS(R6),65$; BR IF ALREADY NOTIFIED
                14      11  090C                BRB     55$                         ; CONTINUE
                            090E         50$:
   53   00000000'GF     D0  090E                MOVL    G^TTY$GL_JOBCTLMB,R3        ; ADDRESS JOB CONTROLLER MAILBOX
      20 68 A6  00      E0  0915                BBS     #UCB$V_JOB,UCB$W_DEVSTS(R6),65$; BR IF NOTIFIED ALREADY
                            091A
                11      E1  091A                BBC     #TT2$V_DISCONNECT,-         ; SKIP IF UNIT NOT ABLE TO DISCONNECT
         03 48 A6           091C                        UCB$L_DEVDEPND2(R6),55$
                            091F
              001D      30  091F                BSBW    CLONE_UCB                   ; GO CREATE LOGICAL UCB
                            0922
      00000000'GF       16  0922         55$:   JSB     G^EXE$SNDEVMSG              ; SEND THE MESSAGE
             0F 50      E9  0928                BLBC    R0,65$                      ; IF FAILED THEN LEAVE DATA BASE UNCHANGED
             5C A5      B5  092B                TSTW    UCB$W_REFC(R5)              ; REF COUNT 0?
                06      12  092E                BNEQ    60$                         ; IF NEQ THEN USER
         68 A6  01      A8  0930                BISW    #UCB$M_JOB,UCB$W_DEVSTS(R6); SET NOTIFIED
                04      11  0934                BRB     65$                         ; LOOK TO UNSOLICITED DATA
         68 A6  04      A8  0936         60$:   BISW    #UCB$M_TT_NOTIF,UCB$W_DEVSTS(R6); SET NOTIFIED
            0060 8F     BA  093A         65$:   POPR    #^M<R5,R6>                  ; RESTORE REGISTERS AND SWITCH BACK TO PUCB
                        05  093E                RSB                                 ; RETURN TO THE DISPATCHER
                            093F
```

```
                              093F          .SBTTL  CLONE NEW LOGICAL UCB
                              093F     ;++
                              093F     ; CLONE_UCB
                              093F     ;
                              093F     ; Description:
                              093F     ;        This routine will clone a logical ucb and cross link it to the
                              093F     ;        specified physical ucb. The logical ucb fields are initialized
                              093F     ;        as appropriate.
                              093F     ;
                              093F     ; Inputs:
                              093F     ;
                              093F     ;        R6 = PHYSICAL UCB ADDRESS
                              093F     ;
                              093F     ; Outputs:
                              093F     ;
                              093F     ;        R5 = LOGICAL UCB ADDRESS
                              093F     ;        R6 = PHYSICAL UCB ADDRESS
                              093F     ;
                              093F     ;        R0-R4 DESTROYED
                              093F     ;--
                              093F     CLONE_UCB:
                   18     BB  093F          PUSHR   #^M<R3,R4>
      55    00000000'EF   D0  0941          MOVL    VT$UCB,R5                    ; GET ADDRESS OF TEMPLATE UCB ADDRESS
                   56     13  0948          BEQL    50$                          ; NONE ESTABLISHED
            00000000'GF   16  094A          JSB     G^IOC$CLONE_UCB              ; GO CREATE A DETACHED UCB
                4D 50     E9  0950          BLBC    R0,50$                       ; ERROR, SO SKIP ITS USE
             55    52     D0  0953          MOVL    R2,R5                        ; GET NEW LUCB ADDRESS
                5C A5     B4  0956          CLRW    UCB$W_REFC(R5)               ; INIT REFERENCE COUNT
                              0959
            00010000 8F   C8  0959          BISL    #UCB$M_DELETEUCB,-           ; MARK THIS LUCB TO BE DELETED ON LAST DEASS
                   64 A5      095F                  UCB$L_STS(R5)
            00040000 8F   CA  0961          BICL    #DEV$M_AVL,-                 ; SHOW PUCB NOT AVALABLE FOR USE
                   38 A6      0967                  UCB$L_DEVCHAR(R6)
   3C A6    00000100 8F   C8  0969          BISL    #DEV$M_RED,UCB$L_DEVCHAR2(R6) ; SHO PUCB REDIRECTED
                              0971
                              0971
                              0971     ;        INIT LUCB FIELDS
                              0971
     41 A5      41 A6     90  0971          MOVB    UCB$B_DEVTYPE(R6),UCB$B_DEVTYPE(R5)      ; TERMINAL TYPE
     0B A5      0B A6     90  0976          MOVB    UCB$B_FIPL(R6),UCB$B_FIPL(R5)            ; FORK IPL
     5E A5      5E A6     90  097B          MOVB    UCB$B_DIPL(R6),UCB$B_DIPL(R5)            ; DEVICE IPL
     42 A5      42 A6     B0  0980          MOVW    UCB$W_DEVBUFSIZ(R6),UCB$W_DEVBUFSIZ(R5)  ; BUFFER SIZE
     44 A5      44 A6     D0  0985          MOVL    UCB$L_DEVDEPEND(R6),UCB$L_DEVDEPEND(R5)  ; DEVICE CHARACTERISTICS
     48 A5      48 A6     D0  098A          MOVL    UCB$L_DEVDEPND2(R6),UCB$L_DEVDEPND2(R5)  ;
     5F A5      5F A6     90  098F          MOVB    UCB$B_AMOD(R6),UCB$B_AMOD(R5)            ; DEVICE PROTECTION
      00A0 C5      56     D0  0994          MOVL    R6,UCB$L_TL_PHYUCB(R5)                   ; LINK LUCB TO PUCB
      00C0 C6      55     D0  0999          MOVL    R5,UCB$L_TT_LOGUCB(R6)                   ; LINK PUCB TO LUCB
                   03     11  099E          BRB     55$
                              09A0
             55    56     D0  09A0     50$: MOVL    R6,R5                        ; ERROR, NO LOGICAL UCB CREATED
                   18     BA  09A3     55$: POPR    #^M<R3,R4>
                   05         09A5          RSB
                              09A6
                              09A6
```

```
                          09A6          .SBTTL  ATTENTION - UNIT TIMEOUT, POWERFAIL ATTENTION ROUTINE
                          09A6
                          09A6    ;++
                          09A6    ; ATTENTION - UNIT TIMEOUT, POWERFAIL ATTENTION ROUTINE
                          09A6    ;
                          09A6    ; THIS ROUTINE IS ENTERED VIA THE UCB$L_FPC ON POWERFAIL OR TIMEOUT.
                          09A6    ; THE ACTION IS TO RESTART THE I/O IF THE POWER FAILED AND TO CANCEL IT
                          09A6    ; IF A TIMEOUT OCCURRED.
                          09A6    ;
                          09A6    ; INPUTS:
                          09A6    ;
                          09A6    ;       R5 = UCB ADDRESS
                          09A6    ;
                          09A6    ; OUTPUTS:
                          09A6    ;
                          09A6    ;       R5 = UCB ADDRESS
                          09A6    ;--
                          09A6
                          09A6    ATTENTION:                                     ; ATTENTION INTERRUPT
               F657'  30  09A6          BSBW    TTY$LOCK                        ; SET IPL AND REGISTERS
    06 64 A5     05   E1  09A9          BBC     #UCB$V_POWER,UCB$W_STS(R5),20$; CANCEL IF NOT POWERFAIL
                          09AE
                          09AE    ;
                          09AE    ; POWER FAILED WHILE THIS UNIT WAS WAITING FOR AN INTERRUPT.
                          09AE    ;
                          09AE
               004A   30  09AE          BSBW    RESTART                         ; Restart in progress I/O.
               F64C'  31  09B1          BRW     TTY$STARTOUTPUT                 ; BEGIN THE I/O AGAIN
                          09B4    20$:
                          09B4    ;
                          09B4    ; CHECK FOR AUTOBAUD AT 600 BAUD TIMEOUT
                          09B4    ;
                          09B4          IF_NOT_STATE    AUTOP,50$               ; SKIP IF NOT
                 0F   9B  09B8          MOVZBW  #TT$C_BAUD_9600,-
            00F4 C5       09BA                  UCB$W_TT_SPEED(R5)              ; SET 9600 BAUD
    00F8 C5 00EC C5   90  09BD          MOVB    UCB$B_TT_DEPARI(R5),UCB$B_TT_PARITY(R5); RESET THE PARITY
               00B2   30  09C4          BSBW    TTY$SET_LINE                    ; RESET SPEED ON LINE,
                          09C7                                                  ;  SINCE TIMER EXPIRED
                      05  09C7          RSB
                          09C8
                          09C8    ;
                          09C8    ; ACTUAL HARDWARE TIMEOUT
                          09C8    ;
                          09C8
                          09C8    50$:                                          ; Handle timeout.
            0082 C5   B6  09C8          INCW    UCB$W_ERRCNT(R5)                ; BUMP ERROR COUNTER
            022C 8F   3C  09CC          MOVZWL  #SS$_TIMEOUT,-                  ; Load status code into the
              7C A5       09D0                  UCB$Q_BOFF(R5)                  ; UCB status word.
                          09D2
                          09D2    60$:
                          09D2          IF_NOT_STATE -                          ; Branch forward if not in a
                          09D2                  WRITE,70$                       ; write state.
               004D   30  09D6          BSBW    TTY$ABORT                       ; Abort any output activity
       53   00D4 C5   D0  09D9          MOVL    UCB$L_TT_WRTBUF(R5),R3          ; Get address of write buffer.
              7C A5   B0  09DE          MOVW    UCB$W_BOFF(R5),-                ; Save status in the write
              28 A3       09E1                  TTY$W_WB_STATUS(R3)             ; packet.
               F61A'  30  09E3          BSBW    TTY$WRITEDONE                   ; Complete this write.
```

```
                    09E6
                    09E6  70$:
                    09E6              IF_NOT_STATE -                  ; Check for a read.
                    09E6                      READ,80$                ; Branch forward if not reading.
                    09EA              CLR_STATE -                     ; Clear escape states.
                    09EA                      <ESC,BADESC>
   54   58 A5   D0  09F0              MOVL    UCB$L_IRP(R5),R4        ; GET CURRENT PACKET
        38 A4   D4  09F4              CLRL    IRP$L_MEDIA(R4)         ; SET NO TERMINATORS
        F606'   30  09F7              BSBW    TTY$READONE             ; COMPLETE READ
                    09FA  80$:
                05  09FA              RSB
```

```
                             09FB        .SBTTL  RESTART - RESTART EVERYTHING ON POWERFAIL
                             09FB  ;++
                             09FB  ; RESTART
                             09FB  ;
                             09FB  ; Description:
                             09FB  ;
                             09FB  ;         Power fail recovery code.  Called by the state dispatcher when
                             09FB  ; a powerfail is detected by the port unit-init routines.
                             09FB  ;
                             09FB  ; Inputs:
                             09FB  ;
                             09FB  ;         R5 = UCB address
                             09FB  ;
                             09FB  ; Outputs:
                             09FB  ;
                             09FB  ;         R2 = UNIT state vector
                             09FB  ;         R5 = UCB address
                             09FB  ;
                             09FB  RESTART::
      64 A5    20      AA    09FB          BICW    #UCB$M_POWER,UCB$W_STS(R5); RESET POWERFAIL
   52  00B8 C5         9E    09FF          MOVAB   UCB$Q_TT_STATE(R5),R2   ; ADDRESS STATE VECTOR
          001F         30    0A04          BSBW    TTY$ABORT               ; ABORT PORT ACTIVITY
          0056         30    0A07          BSBW    TTY$RESUME              ; AND THEN RESUME PORT ACTIVITY
                             0A0A          CLR_STATE POWER                 ; RESET POWER FAIL STATE
                             0A0D          SET_STATE REFRSH                ; FORCE READS TO BE REFRESHED
       FA7D CF         17    0A11          JMP     TTY$RESTARTIO           ; SET UP THE CURRENT I/O
                             0A15
```

```
                        0A15          .SBTTL  TTY$POWERACTION - PORT ENTRY FOR POWERFAIL RECOVERY ACTION
                        0A15     ;++
                        0A15     ; TTY$POWERACTION
                        0A15     ;
                        0A15     ; Description:
                        0A15     ;
                        0A15     ;        Called by the port driver's unit-init routine when a powerfail is
                        0A15     ; detected.  This routine will clear the interupt expected bit and setup
                        0A15     ; a timer to call us back during the first pass of the timer survice routine.
                        0A15     ; This routine also sets the POWER bit in the unit state vector to perform
                        0A15     ; powerfail recovery if the timer is reset by a burst or character output.
                        0A15     ;
                        0A15     ; Inputs:
                        0A15     ;
                        0A15     ;        R5 = UCB address
                        0A15     ;
                        0A15     ; Outputs:
                        0A15     ;
                        0A15     ;        All registers are preserved
                        0A15     ;
                        0A15     ;--
                        0A15    TTY$POWERACTION::
      64 A5    02   AA  0A15          BICW    #UCB$M_INT,UCB$W_STS(R5)
      64 A5    01   A8  0A19          BISW    #UCB$M_TIM,UCB$W_STS(R5)
         6C    A5   D4  0A1D          CLRL    UCB$L_DUETIM(R5)
    00B8 C5    01   C8  0A20          BISL    #TTY$M_ST_POWER,UCB$Q_TT_STATE(R5)
               05       0A25          RSB
                        0A26
```

```
                        0A26        .SBTTL  CLASS DRIVER JACKET INTERFACE TO PORT DRIVERS
                        0A26  ;++
                        0A26  ;
                        0A26  ; FUNCTIONAL DESCRIPTION:
                        0A26  ;
                        0A26  ;       These routines are the only place in the driver that the terminal
                        0A26  ; port driver is actually called.  These routines allow the terminal class
                        0A26  ; driver to add code that would have been duplicated in every port driver.
                        0A26  ;
                        0A26  ; Inputs to all routines:
                        0A26  ;
                        0A26  ;       R5 - ucb address
                        0A26  ;
                        0A26  ;--
                        0A26  ;
                        0A26  ; TTY$ABORT - Abort current output activity
                        0A26  ;
                        0A26
                        0A26  TTY$ABORT::
                01  BB  0A26        PUSHR   #^M<R0>                         ; SAVE A REGISTER
   00B8 C5      01  E5  0A28        BBCC    #TTY$V_ST_CTRLS,UCB$Q_TT_STATE(R5),-    ; ALLOW CLASS OUTPUT
                00      0A2D        5$
                        0A2E  5$:
50    0118 C5       D0  0A2E        MOVL    UCB$L_TT_PORT(R5),R0            ; GET THE ADDRESS OF THE PORT
            20 B0   16  0A33        JSB     @PORT_ABORT(R0)                 ; GOTO THE ROUTINE
                01  BA  0A36        POPR    #^M<R0>
                    05  0A38        RSB                                     ; and return
                        0A39
```

```
                        0A39  ;
                        0A39  ; TTY$DISCONNECT - Indicates last deassign.
                        0A39  ;
                        0A39  ; INPUTS:
                        0A39  ;
                        0A39  ;       R5 - UCB address
                        0A39  ;
                        0A39  TTY$DISCONNECT::
              02   BB   0A39          PUSHR   #^M<R1>                 ; SAVE A REGISTER
   51    0118 C5   D0   0A3B          MOVL    UCB$L_TT_PORT(R5),R1    ; GET THE ADDRESS OF THE PORT
           04 B1   16   0A40          JSB     @PORT_DISCONNECT(R1)    ; GOTO THE ROUTINE
              02   BA   0A43          POPR    #^M<R1>
                   05   0A45          RSB                             ; and return
                        0A46
```

```
                          0A46  2003
                          0A46  2004 ; TTY$DS_SET - SET OUTPUT MODEM SIGNALS
                          0A46  2005 ;
                          0A46  2006 ; INPUTS:
                          0A46  2007 ;
                          0A46  2008 ;       R2 - LOW BYTE - SIGNALS TO ACTIVATE
                          0A46  2009 ;            HIGH BYTE - SIGNALS TO DEACTIVATE
                          0A46  2010 ;       R5 - UCB ADDRESS
                          0A46  2011 ;
                          0A46  2012 ; OUTPUTS:
                          0A46  2013 ;       R1 - R3 ARE USED
                          0A46  2014 ;--
                          0A46  2015 TTY$DS_SET::
              01     BB   0A46  2016         PUSHR   #^M<R0>                 ; SAVE A REGISTER
  50    0118 C5      D0   0A48  2017         MOVL    UCB$L_TT_PORT(R5),R0    ; GET THE ADDRESS OF THE PORT
           0C B0     16   0A4D  2018         JSB     @PORT_DS_SET(R0)        ; GOTO THE ROUTINE
              01     BA   0A50  2019         POPR    #^M<R0>
                     05   0A52  2020         RSB                             ; and return
```

```
                    0A53  2022 ;
                    0A53  2023 ; TTY$MAINT - MAINTNENCE ROUTINES
                    0A53  2024 ;
                    0A53  2025 TTY$MAINT::
            02  BB  0A53  2026         PUSHR   #^M<R1>                 ; SAVE A REGISTER
  51  0118 C5  D0  0A55  2027         MOVL    UCB$L_TT_PORT(R5),R1    ; GET THE ADDRESS OF THE PORT
        30 B1  16  0A5A  2028         JSB     @PORT_MAINT(R1)         ; GOTO THE ROUTINE
            02  BA  0A5D  2029         POPR    #^M<R1>
                05  0A5F  2030         RSB                             ; and return
```

```
                        0A60  ;
                        0A60  ; TTY$RESUME - CONTINUE OUTPUT ON A LINE
                        0A60  ;
                        0A60  TTY$RESUME::
                01   BB 0A60          PUSHR   #^M<R0>                       ; SAVE A REGISTER
   0E 00BC C5   1D   E0 0A62          BBS     #TTY$V_ST_CTSLOW,UCB$Q_TT_STATE+4(R5),10$;IF CLEAR TO SEND IS LOW
                        0A68                                                ; THEN DON'T RESUME YET
      00B8 C5   01   E5 0A68          BBCC    #TTY$V_ST_CTRLS,UCB$Q_TT_STATE(R5),-     ; ALLOW CLASS OUTPUT
                00      0A6D                  5$
                        0A6E  5$:
     50    0118 C5   D0 0A6E          MOVL    UCB$L_TT_PORT(R5),R0          ; GET THE ADDRESS OF THE PORT
             24 B0   16 0A73          JSB     @PORT_RESUME(R0)              ; GOTO THE ROUTINE
                01   BA 0A76  10$:    POPR    #^M<R0>
                     05 0A78          RSB                                   ; and return
```

```
                    0A79  2047 ; TTY$SET_LINE - SET SPEED AND PARITY
                    0A79  2048 ;
                    0A79  2049 ; IMPLICIT INPUTS:
                    0A79  2050 ;         UCB$B_TT_PARITY - CONTAINS PARITY, STOP BIT AND FRAME SIZE INFO
                    0A79  2051 ;         UCB$W_TT_SPEED - LOW BYTE XMIT SPEED
                    0A79  2052 ;                          HIGH BYTE ZERO OR RECIEVE SPEED
                    0A79  2053 ;         UCB$B_TT_PRTCTL - DMA AND AUTOXON ENABLE FLAGS
                    0A79  2054 ;
                    0A79  2055 TTY$SET_LINE::
           01   BB  0A79  2056         PUSHR   #^M<R0>                 ; SAVE A REGISTER
 50   0118 C5   D0  0A7B  2057         MOVL    UCB$L_TT_PORT(R5),R0    ; GET THE ADDRESS OF THE PORT
        08 B0   16  0A80  2058         JSB     @PORT_SET_LINE(R0)      ; GOTO THE ROUTINE
           01   BA  0A83  2059         POPR    #^M<R0>
                05  0A85  2060         RSB                             ; and return
```

```
                          0A86  ; TTY$SET_MODEM - START MODEM POLLING
                          0A86  ;
                          0A86  ; R0 - R4 USED
                          0A86  ;
                          0A86  TTY$SET_MODEM:
                01   BB   0A86          PUSHR   #^M<R0>                 ; SAVE A REGISTER
     50   0118 C5    D0   0A88          MOVL    UCB$L_TT_PORT(R5),R0    ; GET THE ADDRESS OF THE PORT
             28 B0   16   0A8D          JSB     @PORT_SET_MODEM(R0)     ; GOTO THE ROUTINE
                01   BA   0A90          POPR    #^M<R0>
                     05   0A92          RSB                             ; and return
```

```
                 0A93  2073 ; TTY$STOP - STOP OUTPUT TO THE TERMINAL LINE
                 0A93  2074 ;
                 0A93  2075 TTY$STOP::
         01  BB  0A93  2076         PUSHR   #^M<R0>                    ; SAVE A REGISTER
         01  E2  0A95  2077         BBSS    #TTY$V_ST_CTRLS,-
   00 00B8 C5    0A97  2078                 UCB$Q_TT_STATE(R5),5$           ; BLOCK NEW OUTPUT
                 0A9B  2079 5$:
50    0118 C5  D0  0A9B  2080         MOVL    UCB$L_TT_PORT(R5),R0      ; GET THE ADDRESS OF THE PORT
         18 B0  16  0AA0  2081         JSB     @PORT_STOP(R0)           ; GOTO THE ROUTINE
         01  BA  0AA3  2082         POPR    #^M<R0>
             05  0AA5  2083         RSB                               ; and return
```

```
                        0AA6
                        0AA6   ; TTY$XOFF - CALCULATE CHARACTER TO SEND FOR XOFF AND SEND IT TO THE PORT
                        0AA6   ; TTY$XON  - SAME AS XOFF FOR XON FLOW CONTROL.
                        0AA6   .ENABLE LSB
                        0AA6   TTY$XOFF::
            0A    BB    0AA6            PUSHR   #^M<R1,R3>                      ; SAVE REGISTERS
       53   13    9A    0AA8            MOVZBL  #TTY$C_XOFF,R3                  ; USE XOFF
            2C    E1    0AAB            BBC     #TTY$V_SX_TYPFUL,-
     19 00B8 C5         0AAD                    UCB$Q_TT_STATE(R5),30$          ; SKIP IF NOT TYPEAHEAD FULL
   09 44 A5  04   E0    0AB1            BBS     #TT$V_HOSTSYNC,UCB$L_DEVDEPEND(R5),10$ ; XOFF CORRECT
            22    E0    0AB6            BBS     #TTY$V_SX_PASALL,-
     0B 00B8 C5         0AB8                    UCB$Q_TT_STATE(R5),20$          ; SKIP IF PASSALL MODE
       53   07    9A    0ABC            MOVZBL  #TTY$C_BELL,R3                  ; USE BELL CHARACTER
   51     0118 C5 D0    0ABF   10$:     MOVL    UCB$L_TT_PORT(R5),R1            ; GET THE ADDRESS OF THE PORT VECTOR
         14 B1    16    0AC4            JSB     @PORT_XOFF(R1)                  ;
            0A    BA    0AC7   20$:     POPR    #^M<R1,R3>
                  05    0AC9            RSB
                        0ACA
  00040010 8F     D3    0ACA   30$:     BITL    #TT$M_HOSTSYNC!TT$M_READSYNC,-
         44 A5          0AD0                    UCB$L_DEVDEPEND(R5)             ; HOST OR READ SYNC TERMINAL?
            EB    12    0AD2            BNEQ    10$                             ; YES THEN OUTPUT THE CHARACTER
            F1    11    0AD4            BRB     20$                             ; NO, SKIP OUTPUT
                        0AD6
                        0AD6   TTY$XON::
            0A    BB    0AD6            PUSHR   #^M<R1,R3>
       53   11    9A    0AD8            MOVZBL  #TTY$C_XON,R3                   ; SEND XON
  00040010 8F     D3    0ADB            BITL    #TT$M_HOSTSYNC!TT$M_READSYNC,-
         44 A5          0AE1                    UCB$L_DEVDEPEND(R5)             ; HOST OR READ SYNC TERMINAL?
            E2    13    0AE3            BEQL    20$                             ; NO, SKIP OUTPUT
   51     0118 C5 D0    0AE5            MOVL    UCB$L_TT_PORT(R5),R1            ; GET THE ADDRESS OF THE PORT VECTOR
         10 B1    16    0AEA            JSB     @PORT_XON(R1)                   ;
            D8    11    0AED            BRB     20$                             ; GO BACK AND RETURN
                        0AEF   .DISABLE LSB
```

```
                0AEF         .SBTTL  MODEM ROUTINES
                0AEF
                0AEF  ;      MODEM CONTROL DATA AREA
                0AEF
                0AEF
                0AEF
                0AEF  TT$L_DIALUP::                                     ;ROOT OF MODEM TIMER CRB LIST
      00000000  0AEF         .LONG   0
                0AF3  TT$W_REFCNT:
          0000  0AF3         .WORD   0                                  ;COUNT OF TIMERS IN USE
                0AF5  TT$W_TIMCTRL:
          0000  0AF5         .WORD   0                                  ;FLAG WORD
                0AF7
                0AF7
                0AF7         .ALIGN  QUAD
                0AF8  TT$TIMQUENT:                                      ;TIMER QUE ENTRY FOR MODEM TIMER
      00000B28  0AF8         .BLKB   TQE$C_LENGTH                       ;ALLOCATE BLOCK HERE
                0B28         STO_TQE TQE$W_SIZE,WORD,TQE$C_LENGTH,TT$TIMQUENT
                0B28         STO_TQE TQE$B_TYPE,BYTE,DYN$C_TQE,TT$TIMQUENT
                0B28         STO_TQE TQE$B_RQTYPE,BYTE,TQE$C_SSREPT,TT$TIMQUENT
                0B28
                0B28
```

```
                          0B28  ;++
                          0B28  ; TRANSITION_NOCHECK -  process transition element without the devdepend bit cc heck
                          0B28  ; TRANSITION     -      PROCESS TRANSITION ELEMENTS ON MODEM/TIMER STATUS EVENT
                          0B28  ; PORT_TRANSITION -     PROCESSES PORT CALLS THEN FALLS INTO TRANSITION
                          0B28  ;
                          0B28  ; FUNCTIONAL DESCRIPTION:
                          0B28  ;
                          0B28  ; THIS ROUTINE IS CALLED ON ALL MODEM TRANSITIONS SUCH AS MODEM SIGNAL CHANGES,
                          0B28  ; TIMER EXPIRATIONS AND CONTROL. THE TRANSITION ELEMENTS CONTAINED IN THE
                          0B28  ; STATE TABLE ASSOCIATED WITH THE CURRENT STATE ARE SCANNED. IF A MATCH IS
                          0B28  ; FOUND, THEN A NEW STATE IS DECLARED. UPON DECLARING A NEW STATE,
                          0B28  ; OUTPUT MODEM SIGNALS MAY BE SET OR RESET, TIMERS MAY BE STARTED OR
                          0B28  ; CANCELLED AND ACTION ROUTINES MAY BE INVOKED.
                          0B28  ;
                          0B28  ; PORT_TRANSITION
                          0B28  ;       THIS ROUTINE PRE-PROCESSES THE CALLS FROM THE PORT DRIVERS
                          0B28  ; TO SCREEN OUT CALLS TO MODEM INIT WITH REF-COUNT NON-ZERO AND CHANGE
                          0B28  ; THOSE CALLS INTO SHUTDOWN CALLS.  ALL INPUTS AND OUTPUTS ARE THE SAME
                          0B28  ; AS TRANSITION.
                          0B28  ;
                          0B28  ; INPUTS:
                          0B28  ;
                          0B28  ;       R1 =    TRANSITION TYPE
                          0B28  ;       R2 =    TYPE SPECIFIC ARGUMENT
                          0B28  ;                       DATASET - NEW RCV MODEM MASK
                          0B28  ;       R5 =    UCB ADDRESS
                          0B28  ;
                          0B28  ; OUTPUTS:
                          0B28  ;       R0-R4   DESTROYED
                          0B28  ;--
                          0B28  PORT_TRANSITION::
        51    00     D1   0B28          CMPL    #MODEM$C_INIT,R1                ; IS THIS AN INIT?
              0F     12   0B2B          BNEQ    TRANSITION                      ; NO THEN DON'T INTERPRET
   53     00C0 C5    D0   0B2D          MOVL    UCB$L_TT_LOGUCB(R5),R3          ; GET THE LOGICAL UCB ADDRESS
              08     13   0B32          BEQL    TRANSITION                      ; NO LOGICAL THEN EXIT
        5C    A3     B5   0B34          TSTW    UCB$W_REFC(R3)                  ; ANY REFCOUNT IN LOGICAL
              03     13   0B37          BEQL    TRANSITION                      ; NO THEN INIT THE LINE
        51    01     9A   0B39          MOVZBL  #MODEM$C_SHUTDWN,R1             ; ELSE SHUT IT DOWN
                          0B3C  TRANSITION::
            0126 C5  B5   0B3C          TSTW    UCB$W_TT_DS_ST(R5)              ; IS MODEM STATE IDLE?
              05     12   0B40          BNEQ    TRANSITION_NOCHECK              ; NO, SO ALLOW CLEANUP TO FINISH
     47 44 A5    15  E1   0B42          BBC     #TT$V_MODEM,UCB$L_DEVDEPEND(R5),END;DON'T BOTHER IF NOT MODEM
                          0B47  TRANSITION_NOCHECK::
   53   00000D7B'EF  DE   0B47          MOVAL   STATE_INIT+MODEM$C_ST_LENGTH,R3 ;ROOT OF STATE TABLE + OFFSET
     54     0126 C5  3C   0B4E          MOVZWL  UCB$W_TT_DS_ST(R5),R4
        54    53     C0   0B53          ADDL    R3,R4
                          0B56          CASE    R1,TYPE=B,<INIT,SHUTDWN,TRAN_LOOP,TRAN_LOOP,TRAN_LOOP>
                          0B64  INIT:
            01EA     30   0B64          BSBW    MODEM$LINK_CRB                  ;LINK INTO CRB CHAIN
   52   00000007'8F  D0   0B67          MOVL    #IDLE-STATE_INIT,R2
            008C     31   0B6E          BRW     DECLARE_STATE                   ;INIT MODEM CONTROL
                          0B71  SHUTDWN:
   52   000000BE'8F  D0   0B71          MOVL    #SHUTDOWN-STATE_INIT,R2         ;DECLARE SHUTDOWN STATE
            0082     31   0B78          BRW     DECLARE_STATE                   ;TRANSITION TO NEW STATE
                          0B7B  TRAN_LOOP:
        53    64     9A   0B7B          MOVZBL  (R4),R3                         ;GET TRANSITION ELEMENT TYPE
                          0B7E          CASE    R3,TYPE=B,<DATASET,TIME,END,DIALTYPE,DZ11,NOMODEM>
```

```
                    0B8E                                                 ;DISPATCH TO TRANSITION SERVICE
                    0B8E  ;
                    0B8E  END:                ;LAST TRANSITION FOR STATE
              05    0B8E          RSB
                    0B8F
                    0B8F  ;
                    0B8F  TIME:               ;TIMEOUT TRANSITION ELEMENT TYPE
     51   04  91    0B8F          CMPB    #MODEM$C_TIMER,R1              ;TIMEOUT TRANSITION CALL
          5F  12    0B92          BNEQ    NEXT_TRAN                      ;NO PROCESS NEXT ELEMENT
        0062  31    0B94          BRW     NEW_STATE                      ;DECLARE NEW STATE
                    0B97
                    0B97  ;
                    0B97  DATASET:            ;DATASET TRANSITION ELEMENT TYPE
     51   03  91    0B97          CMPB    #MODEM$C_DATASET,R1            ;MODEM INTERRUPT?
          05  12    0B9A          BNEQ    10$                            ;NO , DONT UPDATE STATUS
 0124 C5  52  90    0B9C          MOVB    R2,UCB$B_TT_DS_RCV(R5)         ;UPDATE STATUS
                    0BA1  10$:
 52   0124 C5  9A   0BA1          MOVZBL  UCB$B_TT_DS_RCV(R5),R2         ;GET CURRENT STATUS
       05 A4  95    0BA6          TSTB    MODEM$B_TRAN_ONMASK(R4)        ;MASK ACTIVE
          0D  13    0BA9          BEQL    15$                            ;NO
 53    05 A4  92    0BAB          MCOMB   MODEM$B_TRAN_ONMASK(R4),R3     ;GET COMPLEMENT OF MASK
      52  53  8A    0BAF          BICB    R3,R2                          ;ISOLATE BITS OF INTEREST
 52    05 A4  8C    0BB2          XORB    MODEM$B_TRAN_ONMASK(R4),R2     ;TEST IF BOTH ON
          13  13    0BB6          BEQL    20$                            ;YES
                    0BB8  15$:
 52   0124 C5  9A   0BB8          MOVZBL  UCB$B_TT_DS_RCV(R5),R2         ;GET CURRENT STATUS
       04 A4  95    0BBD          TSTB    MODEM$B_TRAN_OFFMASK(R4)       ;MASK ACTIVE
          31  13    0BC0          BEQL    NEXT_TRAN                      ;NO
 52    04 A4  93    0BC2          BITB    MODEM$B_TRAN_OFFMASK(R4),R2    ;TEST SIGNALS OFF CASE
          03  13    0BC6          BEQL    20$                            ;YES OFF, DECLARE NEW STATE
        0028  31    0BC8          BRW     NEXT_TRAN
                    0BCB  20$:
        002B  31    0BCB          BRW     NEW_STATE                      ;ENTER NEW STATE
                    0BCE
                    0BCE  ;
                    0BCE  DIALTYPE:
       05 A4  93    0BCE          BITB    MODEM$B_TRAN_ONMASK(R4),-      ;CHECK FOR MATCH
   00000000'GF      0BD1                  G^TTY$GB_DIALTYP
          1B  13    0BD6          BEQL    NEXT_TRAN                      ;NO MATCH
        001E  31    0BD8          BRW     NEW_STATE                      ;ENTER NEW STATE
                    0BDB  DZ11:
     51   24 A5 D0  0BDB          MOVL    UCB$L_CRB(R5),R1               ;GET CRB ADDRESS
 42 8F    0B A1 91  0BDF          CMPB    CRB$B_TT_TYPE(R1),#DT$_DZ11    ;IS CONTROLLER A DZ11?
          0D  12    0BE4          BNEQ    NEXT_TRAN
        0010  31    0BE6          BRW     NEW_STATE
                    0BE9  NOMODEM:
  44 A5   15  E0    0BE9          BBS     #TT$V_MODEM,UCB$L_DEVDEPEND(R5),-
          03        0BED                  10$                            ;IF MODEM LINE, CONTINUE
        0008  31    0BEE          BRW     NEW_STATE                      ;LINE NO LONGER MODEM
                    0BF1  10$:
          00  11    0BF1          BRB     NEXT_TRAN
                    0BF3
                    0BF3  ;
                    0BF3  NEXT_TRAN:          ;LOOK AT NEXT TRANSITION ELEMENT
     54   06  C0    0BF3          ADDL    #MODEM$C_TRAN_LENGTH,R4        ;POINT TO NEXT ELEMENT
        FF82  31    0BF6          BRW     TRAN_LOOP
                    0BF9
```

```
                 0BF9  2256
```

```
                          0BF9  ;++
                          0BF9  ; DECLARE_STATE -        DECLARE NEW STATE
                          0BF9  ;
                          0BF9  ; FUNCTIONAL DESCRIPTION:
                          0BF9  ;
                          0BF9  ; THIS ROUTINE IS USED TO DECLARE A NEW MODEM STATE. IT HANDLES STARTING
                          0BF9  ; OR CANCELLING TIMERS, SETTING NEW MODEM OUTPUT SIGNALS (VIA THE PORT)
                          0BF9  ; AND INVOKING ACTION ROUTINES.
                          0BF9  ;
                          0BF9  ; INPUTS:
                          0BF9  ;
                          0BF9  ;         R2 =    NEW STATE BEING DECLARED
                          0BF9  ;         R5 =    UCB ADDRESS
                          0BF9  ;
                          0BF9  ; OUTPUTS:
                          0BF9  ;         R0-R4   DESTROYED
                          0BF9  ;--
                          0BF9  NEW_STATE:                                              ;ALTERNATE ENTRY FOR CHANGING
                          0BF9                                                          ;STATE FOR TRANSITION ELEMENT
        52    02 A4   3C  0BF9          MOVZWL  MODEM$W_TRAN_NSTATE(R4),R2              ;GET OFFSET TO NEXT STATE
                          0BFD
                          0BFD  DECLARE_STATE:
     0126 C5      52  B0  0BFD          MOVW    R2,UCB$W_TT_DS_ST(R5)                   ;SET NEW STATE
  54   00000D75'EF    DE  0C02          MOVAL   STATE_INIT,R4                           ;ROOT OF STATE TABLE
          54      52  C0  0C09          ADDL    R2,R4                                   ;POINT TO STATE ENTRY
        52    02 A4   3C  0C0C          MOVZWL  MODEM$W_ST_TIMER(R4),R2                 ;GET TIMER VALUE
            0023      30  0C10          BSBW    MODEM_TIMER                             ;SET NEW TIMER OR
                          0C13                                                          ;CANCEL PREVIOUS IF ZERO
          52      64  3C  0C13          MOVZWL  MODEM$B_ST_ONMASK(R4),R2                ;GET SET SIGNAL MASK
                          0C16                                                          ;TURN LOW BYTE ON,HIGH BYTE OFF
              03      13  0C16          BEQL    5$                                      ;NULL FIELDS
            FE2B      30  0C18          BSBW    TTY$DS_SET                              ;SET NEW MODEM SIGNAL(S)
                          0C1B  5$:
        52    04 A4   32  0C1B          CVTWL   MODEM$W_ST_ROUTINE(R4),R2               ;ACTION ROUTINE PRESENT?
              0F      13  0C1F          BEQL    10$                                     ;NO
  50   00000D75'EF    DE  0C21          MOVAL   STATE_INIT,R0
          52      50  C0  0C28          ADDL    R0,R2                                   ;COMPUTE ADDRESS OF ACTION ROUTINE
              62      16  0C2B          JSB     (R2)                                    ;INVOKE ACTION ROUTINE
          03      50  E9  0C2D          BLBC    R0,15$                                  ;SPECIAL TRANSITION TYPE IF ERROR
                          0C30  10$:
          51      02  D0  0C30          MOVL    #MODEM$C_NULL,R1
            FF11      31  0C33  15$:    BRW     TRANSITION_NOCHECK                      ;DECLARE NULL TRANSITION IN NEW STAT
                          0C36                                                          ;(TO PROCESS ANY PRESET
                          0C36                                                          ; MODEM SIGNALS)
                          0C36
                          0C36
```

```
                              0C36  ;++
                              0C36  ; MODEM_TIMER   -       SET/RESET MASTER MODEM TIMER ROUTINE
                              0C36  ;
                              0C36  ; FUNCTIONAL DESCRIPTION:
                              0C36  ;
                              0C36  ; THIS ROUTINE STARTS OR CANCELS A MODEM PROTECOL TIMER FOR A LINE.
                              0C36  ; ONE TQE IS USED TO TIME ALL ACTIVE LINES. THE TQE ONLY REMAINS ACTIVE
                              0C36  ; SO LONG AS AT LEAST ONE TIMER IS NEEDED. SUBSEQUENT TIMERS USE THE
                              0C36  ; SAME TQE. WHEN ALL CURRENT TIMERS EXPIRE THE TQE IS REMOVED FROM
                              0C36  ; THE SYSTEM TIMER QUEUE UNTIL NEEDED AGAIN.
                              0C36  ; SINCE TIMERS ARE ONLY NEEDED DURING TRANSITIONS, THIS IS A LOW OVERHEAD
                              0C36  ; ACTIVITY.
                              0C36  ;
                              0C36  ; INPUTS:
                              0C36  ;
                              0C36  ;         R2 =     NEW TIMEOUT VALUE OR 0 TO CANCEL ANY OUTSTANDING
                              0C36  ;         R5 =     UCB ADDRESS
                              0C36  ;
                              0C36  ; OUTPUTS:
                              0C36  ;         R0-R3   DESTROYED
                              0C36  ;--
                              0C36  MODEM_TIMER:
                  54     DD   0C36          PUSHL   R4
         54    24 A5     D0   0C38          MOVL    UCB$L_CRB(R5),R4                ;CRB ADDRESS
                  52     D5   0C3C          TSTL    R2                              ;CANCEL?
                  75     13   0C3E          BEQL    TIMER_CANCEL                    ;YES
                              0C40
                              0C40          ;NEED TO INIT TIMER FOR LINE
             0128 C5     B5   0C40          TSTW    UCB$W_TT_DS_TIM(R5)             ; ANY CURRENT TIMER ACTIVE
                  07     12   0C44          BNEQ    5$                              ; YES THEN JUST CHANGE THE TIME
               1F A4     96   0C46          INCB    CRB$B_TT_TIMREFC(R4)            ; BUMP THE REFCOUNT IN THE CRB
             FEA6 CF     B6   0C49          INCW    TT$W_REFCNT                     ;BUMP IN USE COUNTER
        0128 C5   52     B0   0C4D  5$:     MOVW    R2,UCB$W_TT_DS_TIM(R5)  ;SET TIME TO WAIT
                              0C52
        FE9E CF   01     AA   0C52          BICW    #TIMCTRL$M_CANCEL,TT$W_TIMCTRL  ;RESET ANY CANCEL REQUEST
     02 FE99 CF   01     E1   0C57          BBC     #TIMCTRL$V_ACTIVE,TT$W_TIMCTRL,10$ ;TIMER NOT CURRENTLY ACTIVE
                  72     11   0C5D          BRB     TIMER_END
                              0C5F  10$:                                            ;NEED TO RESTART TIMER
                  55     DD   0C5F          PUSHL   R5                              ;SAVE UCB ADDRESS
       55    FE93 CF     DE   0C61          MOVAL   TT$TIMQUENT,R5                  ;
           0B A5  06     90   0C66          MOVB    #IPL$_QUEUEAST,TQE$B_RQTYPE(R5) ;SET UP FORK IPL
        00000C72'EF      9F   0C6A          PUSHAB  20$                             ;RETURN ADDRESS
                  0A     11   0C70          BRB     30$                             ;QUEUE FORK
                              0C72
               55 8ED0        0C72  20$:    POPL    R5                              ;RESTORE UCB ADDRESS
        FE7B CF   02     A8   0C75          BISW    #TIMCTRL$M_ACTIVE,TT$W_TIMCTRL  ;SHOW TIMER ACTIVE
                  55     11   0C7A          BRB     TIMER_END
                              0C7C  30$:
        00000000'GF      16   0C7C          JSB     G^EXE$FORK
                              0C82          DSBINT  #IPL$_SYNCH                     ;INTERLOCK TO ENTER TIMER ENTRY
   0C A5    0CD5'CF      9E   0C88          MOVAB   W^TT$TIMER,TQE$L_FPC(R5)        ;SET TIMEOUT ROUTINE ADDRESS
 20 A5   004C4B40 8F     D0   0C8E          MOVL    #5000000,TQE$Q_DELTA(R5)        ;SET REPEAT AMOUNT (500MS)
           0B A5  05     90   0C96          MOVB    #TQE$C_SSREPT,TQE$B_RQTYPE(R5)  ;SET REPEATING TYPE
    50   00000000'GF     7D   0C9A          MOVQ    G^EXE$GQ_SYSTIME,R0             ;CURRENT SYSTEM TIME
    50   004C4B40 8F     C0   0CA1          ADDL    #5000000,R0                     ;TIMEOUT DUE IN 500MS
              51  00     D8   0CA8          ADWC    #0,R1                           ;
        00000000'GF      16   0CAB          JSB     G^EXE$INSTIMQ                   ;INSERT IN TIMER QUEUE
```

```
                    0CB1                  ENBINT                                    ;RESTORE IPL
               05   0CB4                  RSB
                    0CB5
                    0CB5   TIMER_CANCEL:
      0128 C5  B5   0CB5                  TSTW     UCB$W_TT_DS_TIM(R5)              ; IS THE TIMER
           0B  13   0CB9                  BEQL     20$                              ; ACTIVE NO THEN EXIT
      0128 C5  B4   0CBB                  CLRW     UCB$W_TT_DS_TIM(R5)              ;RESET TO INDICATE IDLE
        1F A4  97   0CBF                  DECB     CRB$B_TT_TIMREFC(R4)             ; DECREMENT THE REFCOUNTS
      FE2D CF  B7   0CC2                  DECW     TT$W_REFCNT
                    0CC6   20$:
      FE29 CF  B5   0CC6                  TSTW     TT$W_REFCNT                      ;ANY ACTIVE TIMERS?
           05  12   0CCA                  BNEQ     TIMER_END                        ;YES
                    0CCC
                    0CCC                  ;NO ACTIVE TIMERS  SO,
                    0CCC                  ;TIMER NEEDS TO BE CANCELLED
                    0CCC
  FE24 CF  01  A8   0CCC                  BISW     #TIMCTRL$M_CANCEL,TT$W_TIMCTRL   ;REQUEST TIMER CANCEL
                    0CD1
                    0CD1   TIMER_END:
          54 8ED0   0CD1                  POPL     R4
               05   0CD4                  RSB
                    0CD5
```

```
                        0CD5  ;++
                        0CD5  ; TT$TIMER        -       TIMER INTERRUPT ROUTINE
                        0CD5  ;
                        0CD5  ; FUNCTIONAL DESCRIPTION:
                        0CD5  ;
                        0CD5  ; THIS ROUTINE IS INVOKED UPON THE EXPIRATION OF THE MODEM PROTECOL TQE.
                        0CD5  ; IT COUNTS DOWN ALL ACTIVE TIMERS AND INVOKES THE TRANSITION
                        0CD5  ; ROUTINE FOR ANY LINES THAT HAVE TIMED OUT. THE ROUTINE REQUEUES THE
                        0CD5  ; TQE INTO THE SYSTEM TIMER QUEUE UNLESS ALL ACTIVE TIMERS HAVE
                        0CD5  ; EXPIRED.
                        0CD5  ;
                        0CD5  ; INPUTS:
                        0CD5  ;
                        0CD5  ;         R5 =    TQE ADDRESS
                        0CD5  ;
                        0CD5  ; OUTPUTS:
                        0CD5  ;         R0-R4   DESTROYED
                        0CD5  ;--
                        0CD5  TT$TIMER:
            20     BB   0CD5          PUSHR   #^M<R5>
54     FE14 CF     DE   0CD7          MOVAL   TT$L_DIALUP,R4                          ;GET ROOT OF CRB LIST
                        0CDC  10$:
54       64        D0   0CDC          MOVL    (R4),R4                                 ;GET NEXT CRB ADDRESS
         52        13   0CDF          BEQL    60$                                     ;PROCESS LINES FOR CRB
      0B A4        95   0CE1          TSTB    CRB$B_TT_TIMREFC-CRB$L_TT_MODEM(R4); ANY TIMERS ON THIS LINE?
         F6        13   0CE4          BEQL    10$                                     ; NO  THEN EXIT
                        0CE6  ;
                        0CE6  ; TIMERS ARE ACTIVE ON THIS CONTROLER
                        0CE6  ;
         54        DD   0CE6          PUSHL   R4                                      ;SAVE TIMER THREAD
54       14        C2   0CE8          SUBL    #CRB$L_TT_MODEM,R4                      ;GET ACTUAL CRB ADDRESS
                        0CEB
                        0CEB          ;PROCESS LINES WITH ACTIVE TIMERS
53    2C A4        D0   0CEB          MOVL    CRB$L_INTD+VEC$L_IDB(R4),R3             ;GET ADRESS OF IDB
52    0C A3        3C   0CEF          MOVZWL  IDB$W_UNITS(R3),R2                      ; GET THE NUMBER OF UNITS
         52        D7   0CF3          DECL    R2                                      ; ON THIS CONTROLER (ZERO BASED)
                        0CF5  20$:
55  18 A342        D0   0CF5          MOVL    IDB$L_UCBLST(R3)[R2],R5                 ;GET UCB FOR THAT LINE
         10        13   0CFA          BEQL    35$                                     ; NO UNIT THEN GO ON TO THE NEXT
                        0CFC          DSBINT  UCB$B_DIPL(R5)                          ;INTERLOCK WITH DEVICE INTERRUPTS
     0128 C5       B5   0D03          TSTW    UCB$W_TT_DS_TIM(R5)                     ; DOES THIS UNIT HAVE THE TIMER
         0C        12   0D07          BNEQ    50$                                     ; YES THEN HANDLE IT
                        0D09  30$:    ENBINT                                          ; NO MORE THEN ENABLE THE INTERRUPTS
      E6 52        F4   0D0C  35$:    SOBGEQ  R2,20$                                  ; NO THEN GO ON
         54      8ED0   0D0F  40$:    POPL    R4                                      ; RESTORE TIMER THREAD
       FFC7        31   0D12          BRW     10$
                        0D15  ;
                        0D15  ; DECREMENT/FIRE TIMER REQUESTS
                        0D15  ;
51       04        90   0D15  50$:    MOVB    #MODEM$C_TIMER,R1                       ;SET TRANSITION TYPE
     0128 C5       B7   0D18          DECW    UCB$W_TT_DS_TIM(R5)                     ;COUNT DOWN LINE
         EB        12   0D1C          BNEQ    30$                                     ;NOT TIME YET
     FDD1 CF       B7   0D1E          DECW    TT$W_REFCNT                             ;
         1C        BB   0D22          PUSHR   #^M<R2,R3,R4>                           ; SAVE SOME REGISTERS
       FE15        30   0D24          BSBW    TRANSITION                              ;INVOKE TRANSITION ROUTINE
         1C        BA   0D27          POPR    #^M<R2,R3,R4>
      1F A4        B7   0D29          DECW    CRB$B_TT_TIMREFC(R4)                    ; DECREMENT THE REFCOUNT
```

```
                  DB    12  0D2C  2442          BNEQ    30$                           ; CONTINUE
                            0D2E  2443          ENBINT                                ; RE-ENABLE INTERUPTS IF
                  DC    11  0D31  2444          BRB     40$                           ; NO MORE TIMERS ON THIS
                            0D33  2445                                                ; CONTROLER AND MOVE TO THE NEXT
                            0D33  2446                                                ; CONTROLER
                            0D33  2447  ;
                            0D33  2448  ; CHECK TO SEE IF WE ARE TO CANCEL THE TIMER OR RE-ENABLE IT
                            0D33  2449  ;
                  20    BA  0D33  2450 60$:     POPR    #^M<R5>
                            0D35  2451          DSBINT  #31                           ;INTERLOCK
      0C FDB5 CF  00    E1  0D3B  2452          BBC     #TIMCTRL$V_CANCEL,TT$W_TIMCTRL,70$;DONT CANCEL TIMER
                  03    AA  0D41  2453          BICW    #<TIMCTRL$M_ACTIVE!TIMCTRL$M_CANCEL>-
             FDAF CF        0D43  2454                  ,TT$W_TIMCTRL                 ;RESET CANCEL +TIMER ACTIVE BITS
   55    00000000'GF    DE  0D46  2455          MOVAL   G^EXE$AL_TQENOREPT, R5        ; Point R5 to no-repeat TQE
                            0D4D  2456                                                ; thus ending timer thread.
                            0D4D  2457 70$:
                            0D4D  2458          ENBINT
                        05  0D50  2459          RSB
                            0D51  2460
```

```
                      0D51  ;++
                      0D51  ; MODEM$LINK_CRB
                      0D51  ;
                      0D51  ; FUNCTIONAL DESCRIPTION:
                      0D51  ;
                      0D51  ; THIS ROUTINE IS USED TO LINK THE ASSOCIATED CRB OF A UCB
                      0D51  ; REQUIRING MODEM CONTROL INTO THE MODEM PROTOCOL TIMER QUEUE.
                      0D51  ; NOTE THAT THIS QUEUE IS SEPERATE FROM THE DZ-11 SPECIFIC
                      0D51  ; MODEM TRANSITION POLLING QUEUE.
                      0D51  ;
                      0D51  ; INPUTS:
                      0D51  ;
                      0D51  ;         R5 =    UCB ADDRRESS
                      0D51  ;
                      0D51  ; OUTPUTS:
                      0D51  ;         R0-R4   DESTROYED
                      0D51  ;--
                      0D51  ;
                      0D51  ;         CHECK IF CRB NEEDS INSERTION ON MODEM TIMER QUEUE
                      0D51  ;
                      0D51  MODEM$LINK_CRB:
                      0D51
   54    24 A5   D0   0D51          MOVL    UCB$L_CRB(R5),R4              ; GET CRB ADDRESS
 51    FD96 CF   DE   0D55          MOVAL   TT$L_DIALUP,R1                ; ADDRESS OF LIST HEAD
   53    14 A4   DE   0D5A          MOVAL   CRB$L_TT_MODEM(R4),R3         ; GET CRB THREAD
      52    51   D0   0D5E          MOVL    R1,R2
                      0D61  10$:
      53    62   D1   0D61          CMPL    (R2),R3                       ; CRB ON LIST
            0B   13   0D64          BEQL    20$                           ; YES
      52    62   D0   0D66          MOVL    (R2),R2                       ; POINT TO NEXT
            F6   12   0D69          BNEQ    10$                           ; IF NOT END, LOOK AGAIN
      63    61   D0   0D6B          MOVL    (R1),(R3)                     ; LINK AT LIST HEAD
      61    53   D0   0D6E          MOVL    R3,(R1)
                      0D71  20$:
          FD12   30   0D71          BSBW    TTY$SET_MODEM                 ; INVOKE ANY PORT SPECIFIC SETUP
                 05   0D74          RSB
                      0D75
```

```
0D75            .sbttl  MODEM STATE TABLES
0D75    ;++
0D75    ; MODEM CONTROL STATE TABLES
0D75    ;
0D75    ; FUNCTIONAL DESCRIPTION:
0D75    ;
0D75    ;       THIS TABLE CONTAINS STATE ENTRIES. EACH STATE CORRESPONDS TO
0D75    ; A MODEM PROTECOL STATE. EACH STATE ENTRY CONSISTS OF A MODEM SIGNAL
0D75    ; MASK TO BE SET ON STATE ENTRY, AN INITIAL TIMER VALUE TO BE STARTED ON
0D75    ; STATE ENTRY AND AN OPTIONAL ACTION ROUTINE.
0D75    ;       FOLLOWING THE STATE ENTRY ARE TRANSITION ELEMENTS. ON ANY
0D75    ; TRANSITION : TIMEOUT, MODEM INTERRUPT, OR PROGRAM DECLARED, EACH
0D75    ; TRANSITION ELEMENT IS PROCESSED SEQUENTIALLY. THE LAST TRANSITION ELEMENT
0D75    ; IS ALWAYS AN END TRANSITION ELEMENT WHICH SIGNALS THAT
0D75    ; NO TRANSITION OCCURS IF IT IS PROCESSED. OTHER TRANSITION ELEMENT TYPES
0D75    ; MAKE CONDITIONAL TESTS AND MAY DECLARE A NEW STATE IF THE APPROPRIATE
0D75    ; CONDITIONS HOLD TRUE.
0D75    ;--
0D75
0D75
0D75    ;       MODEM STATE MACRO DEFINITIONS
0D75    ;
0D75
0D75            .macro  state   name,onmask=0,offmask=0,timer,routine
0D75    name:
0D75            .byte   onmask
0D75            .byte   offmask
0D75            .if     nb      timer
0D75            .word   <timer*2>
0D75            .if_false
0D75            .word   0
0D75            .endc
0D75            .if     nb      routine
0D75            .word   routine-state_init
0D75            .if_false
0D75            .word   0
0D75            .endc
0D75            .endm   state
0D75
0D75            .macro  tran    type,nstate,onmask=0,offmask=0
0D75            .if     idn type,end
0D75            .byte   modem$c_tran_end
0D75            .if_false
0D75            .byte   modem$c_tran_'type',0
0D75            .word   nstate-state_init
0D75            .byte   offmask
0D75            .byte   onmask
0D75            .endc
0D75            .endm   tran
0D75
```

```
0D75
0D75    STATE_INIT:
0D75
0D75    ;
0D75    ; LINE NOT UNDER MODEM CONTROL
0D75    ;
0D75            STATE   OFF
0D7B            TRAN    END
0D7C
0D7C    ;
0D7C    ; INITIAL MODEM STATE
0D7C    ;
0D7C            STATE   IDLE,OFFMASK=<TT$M_DS_DTR!TT$M_DS_RTS>,TIMER=2
0D82            TRAN    DIALTYPE,ONMASK=02,NSTATE=RINGWAIT
0D88            TRAN    TIME,NSTATE=WAIT
0D8E            TRAN    END
0D8F
0D8F    ;
0D8F    ; WAIT FOR RING PRIOR TO SETTING DTR
0D8F    ;
0D8F            STATE   RINGWAIT
0D95            TRAN    DATASET,ONMASK=<TT$M_DS_RING>,NSTATE=INIT2
0D9B            TRAN    END
0D9C
0D9C    ;
0D9C    ; WAIT FOR DTR AND RTS
0D9C    ; (IF DZ-11 GO DO SUBSET SUPPORT)
0D9C    ;
0D9C            STATE   WAIT,ONMASK=<TT$M_DS_DTR!TT$M_DS_RTS>
0DA2            TRAN    DZ11,NSTATE=DZWAIT
0DA8            TRAN    DATASET,ONMASK=TT$M_DS_DSR,NSTATE=INIT1
0DAE            TRAN    END
0DAF
0DAF    ;
0DAF    ; DZ-11 SUBSET SUPPORT
0DAF    ; WAIT FOR CARRIER OR RING
0DAF    ;
0DAF            STATE   DZWAIT
0DB5            TRAN    DATASET,ONMASK=<TT$M_DS_CARRIER>,NSTATE=TRANSMIT0
0DBB            TRAN    DATASET,ONMASK=<TT$M_DS_RING>,NSTATE=INIT2
0DC1            TRAN    END
0DC2
0DC2    ;
0DC2    ; DELAY
0DC2    ;
0DC2            STATE   INIT1,TIMER=1
0DC8            TRAN    TIME,NSTATE=INIT2
0DCE            TRAN    END
0DCF
0DCF    ;
0DCF    ; START TIMER AND WAIT FOR CTS AND CARRIER
0DCF    ; IF TIMER EXPIRES, SHUTDOWN LINE
0DCF    ;
0DCF            STATE   INIT2,ONMASK=<TT$M_DS_DTR!TT$M_DS_RTS>,TIMER=30
0DD5            TRAN    TIME,NSTATE=SHUTDOWN
0DDB            TRAN    DATASET,ONMASK=<TT$M_DS_CTS!TT$M_DS_CARRIER!TT$M_DS_DSR>,-
```

```
0DDB                            NSTATE=TRANSMIT0
0DE1                    TRAN    END
0DE2
0DE2    ;
0DE2    ; TRANSMIT0 STATE (SIGNAL CONNECT AND ALLOW TIME TO ASSIGN CHANNELS)
0DE2    ;
0DE2                    STATE   TRANSMIT0,ROUTINE=LOGIN,TIMER=30
0DE8                    TRAN    DATASET,OFFMASK=TT$M_DS_DSR,NSTATE=SHUTDOWN
0DEE                    TRAN    DATASET,OFFMASK=TT$M_DS_CARRIER,NSTATE=TRANSMIT1
0DF4                    TRAN    TIME,NSTATE=TRANSMIT
0DFA                    TRAN    END
0DFB    ;
0DFB    ; NORMAL TRANSMIT STATE
0DFB    ;
0DFB                    STATE   TRANSMIT,ROUTINE=VERIFY ; ROUTINE VERIFY WILL SHUTDOWN IF REFC=0
0E01                    TRAN    DATASET,OFFMASK=TT$M_DS_DSR,NSTATE=SHUTDOWN
0E07                    TRAN    DATASET,OFFMASK=TT$M_DS_CARRIER,NSTATE=TRANSMIT1
0E0D                    TRAN    DATASET,OFFMASK=TT$M_DS_CTS,NSTATE=CTSLOW
0E13                    TRAN    END
0E14
0E14    ;
0E14    ; LOSS OF CARRIER DETECTED
0E14    ;
0E14                    STATE   TRANSMIT1,TIMER=2,ROUTINE=CTSHIGH
0E1A                    TRAN    DIALTYPE,ONMASK=01,NSTATE=SHUTDOWN
0E20                    TRAN    TIME,NSTATE=SHUTDOWN
0E26                    TRAN    DATASET,OFFMASK=TT$M_DS_DSR,NSTATE=SHUTDOWN
0E2C                    TRAN    DATASET,ONMASK=<TT$M_DS_CARRIER!TT$M_DS_CTS>,NSTATE=TRANSMIT
0E32                    TRAN    END
0E33
0E33    ;
0E33    ; DELAY PRIOR TO COMPLETE SHUTDOWN
0E33    ;
0E33                    STATE   SHUTDOWN,TIMER=1,OFFMASK=TT$M_DS_DTR,ROUTINE=LOGOUT
0E39                    TRAN    TIME,NSTATE=SHUT1
0E3F                    TRAN    END
0E40
0E40    ;
0E40    ; COMPLETE SHUTDOWN
0E40    ; AND THEN REINIT
0E40    ;
0E40                    STATE   SHUT1,TIMER=2
0E46                    TRAN    NOMODEM,NSTATE=OFF
0E4C                    TRAN    TIME,NSTATE=IDLE
0E52                    TRAN    DATASET,OFFMASK=TT$M_DS_DSR,NSTATE=IDLE
0E58                    TRAN    END
0E59    ;
0E59    ; CTSLOW
0E59    ; STOP SENDING DATA AND WAIT FOR CTS TO GO HIGH
0E59    ;
0E59                    STATE   CTSLOW,ROUTINE=LOSTCTS
0E5F                    TRAN    DATASET,ONMASK=TT$M_DS_CTS,NSTATE=TRANSMIT
0E65                    TRAN    DATASET,OFFMASK=TT$M_DS_DSR,NSTATE=SHUTDOWN
0E6B                    TRAN    DATASET,OFFMASK=TT$M_DS_CARRIER,NSTATE=TRANSMIT1
0E71                    TRAN    END
0E72
```

```
                                 0E72        .SBTTL  MODEM ACTION ROUTINES
                                 0E72   ;
                                 0E72   ;
                                 0E72   ;       THESE ROUTINES ACT AS ACTION ROUTINES FOR MODEM CONNECTION
                                 0E72   ;       AND DISCONECTION. ON CONNECTION THE REMOTE BIT IS SET.
                                 0E72   ;       ON DISCONNECT A CONTROL Y AST IS FIRED WITH HANGUP STATUS
                                 0E72   ;       TO SIGNAL THE CLI. ALSO ANY OUTSTANDING ATTENTION ASTS ARE
                                 0E72   ;       FIRED.
                                 0E72   ;
                                 0E72   LOGIN:
             51   00C0 C5   D0   0E72           MOVL    UCB$L_TT_LOGUCB(R5),R1          ; GET LUCB ADDRESS
     44 A5   00002000 8F    C8   0E77           BISL    #TT$M_REMOTE,UCB$L_DEVDEPEND(R5); UPDATE CHARACTERISTIC
     44 A1   00002000 8F    C8   0E7F           BISL    #TT$M_REMOTE,UCB$L_DEVDEPEND(R1)
                      1B    10   0E87           BSBB    CTSHIGH                         ; SAY CTS IS HIGH
                      28    11   0E89           BRB     ACTION_EXIT
                                 0E8B
                                 0E8B   LOGOUT:
                    0016    30   0E8B           BSBW    CTSHIGH
               54     02    D0   0E8E           MOVL    #TTY$V_FD_DISCONNECT,R4         ; SIGNAL HANGUP CONDITION
                    F797    30   0E91           BSBW    TTY$CRE_FORK
                      1D    11   0E94           BRB     ACTION_EXIT
                                 0E96
                                 0E96   LOSTCTS:
  00BC C5   20000000 8F     C8   0E96           BISL    #TTY$M_ST_CTSLOW,UCB$Q_TT_STATE+4(R5); SAY CTS IS LOW
                    FBF1    30   0E9F           BSBW    TTY$STOP
                      0F    11   0EA2           BRB     ACTION_EXIT
                                 0EA4   CTSHIGH:
        03 00BC C5    1D    E4   0EA4           BBSC    #TTY$V_ST_CTSLOW,UCB$Q_TT_STATE+4(R5),10$; SAY CTS IS HIGH
                    FBB3    31   0EAA           BRW     TTY$RESUME
                    FBB0    30   0EAD   10$:    BSBW    TTY$RESUME
                   F14D'    30   0EB0           BSBW    TTY$STARTOUTPUT
                                 0EB3
                                 0EB3   ACTION_EXIT:
               50     01    9A   0EB3           MOVZBL  #SS$_NORMAL,R0
                            05   0EB6           RSB
                                 0EB7
                                 0EB7   ;
                                 0EB7   ; VERIFY THAT UNIT HAS CHANNELS ASSIGNED
                                 0EB7   ;
                                 0EB7   VERIFY:
                      EB    10   0EB7           BSBB    CTSHIGH                         ; SAY CTS IS HIGH
             51   00C0 C5   D0   0EB9           MOVL    UCB$L_TT_LOGUCB(R5),R1          ; GET LUCB ADDRESS
                   5C A1    B5   0EBE           TSTW    UCB$W_REFC(R1)                  ; CHANNELS ASSIGNED
                      F0    12   0EC1           BNEQ    ACTION_EXIT                     ; YES, RETURN SUCCESS
                      50    D4   0EC3           CLRL    R0                              ; SIGNAL FAILURE
               51     01    9A   0EC5           MOVZBL  #MODEM$C_SHUTDWN,R1             ; AND RETURN TRANSITION CODE IN R1
                            05   0EC8           RSB
                                 0EC9
```

```
                     0EC9          .SBTTL  CLASS_MODEM_DIS - CLASS SERVICE TO FORCE MODEM SHUTDOWN
                     0EC9          .SBTTL  TTY$CLASS_DISCONNECT - CLASS SERVICE TO SIGNAL HANGUP CONDITION
                     0EC9  ;++
                     0EC9  ; CLASS_MODEM_DIS
                     0EC9  ;
                     0EC9  ; FUNCTION:
                     0EC9  ;       WILL FORCE MODEM SIGNALS TO HANGUP THE LINE AND SIGNAL HANGUP STATUS
                     0EC9  ;
                     0EC9  ; INPUTS:
                     0EC9  ;
                     0EC9  ;       R5 = UCB ADDRESS
                     0EC9  ;
                     0EC9  ; OUTPUTS:
                     0EC9  ;
                     0EC9  ;       R3,R4 DESTROYED
                     0EC9  ;       R0,R1,R2 ARE PRESERVED
                     0EC9  ;--
                     0EC9  CLASS_MODEM_DIS::                              ; INITIATE MODEM DISCONNECT
             07  BB  0EC9          PUSHR   #^M<R0,R1,R2>
      51     01  D0  0ECB          MOVL    #MODEM$C_SHUTDWN,R1            ; SIGNAL RESET
           FC6B  30  0ECE          BSBW    TRANSITION                     ; INVOKE MODEM TRANSITION ROUTINE
             07  BA  0ED1          POPR    #^M<R0,R1,R2>
                 05  0ED3          RSB                                    ; RESTORE THE REGISTERS AND RETURN
                     0ED4
                     0ED4  ; GENERALIZED CLASS ROUTINE TO SIGNAL HANGUP STATUS
                     0ED4
                     0ED4  TTY$CLASS_DISCONNECT::
             07  BB  0ED4          PUSHR   #^M<R0,R1,R2>
 54     00C0 C5  D0  0ED6          MOVL    UCB$L_TT_LOGUCB(R5),R4              ; GET THE LOGICAL UCB TO CHECK
           5C A4 B5  0EDB          TSTW    UCB$W_REFC(R4)                      ; THE REFCOUNT.
             05  12  0EDE          BNEQ    5$                                  ; IF CHANNELS THEN AST'S TO FIRE OR
             50  D4  0EE0          CLRL    R0                                  ; ALWAYS INDICATE HANGUP IF
           FB54  30  0EE2          BSBW    TTY$DISCONNECT                      ; NO CHANNELS THEN NONE OF THE ABOFE
           FFA3  30  0EE5  5$:     BSBW    LOGOUT                              ; AND QUEUE FORK TO SIGNAL HANGUP ST
             07  BA  0EE8          POPR    #^M<R0,R1,R2>
                 05  0EEA          RSB
```

```
                 0EEB  2751          .SBTTL  End of module
                 0EEB  2752
                 0EEB  2753 TT_END::                                     ; End of Terminal Class Driver
                 0EEB  2754          .END
```

```
$$$$$$                         = 00000B28 R       02
ACB$L_AST                      = 00000010
ACB$L_KAST                     = 00000018
ACTION_EXIT                      00000EB3 R       02
ATTENTION                        000009A6 R       02
CAN$C_DASSGN                   = 00000001
CANCEL_AST                       000001D1 R       02
CANCEL_DETACH                    00000205 R       02
CANCEL_DONE                      00000227 R       02
CANCEL_MODEM                     0000022F R       02
CANCEL_RESET                     00000257 R       02
CLASS_MODEM_DIS                  00000EC9 RG      02
CLONE_UCB                        0000093F R       02
CNT                            = 00000001
COM$DELATTNAST                   ********   X     02
COM$DRVDEALMEM                   ********   X     02
COM$FLUSHATTNS                   ********   X     02
COM$FLUSHCTRLS                   ********   X     02
COM$POST                         ********   X     02
CRB$B_TT_TIMREFC               = 0000001F
CRB$B_TT_TYPE                  = 0000000B
CRB$L_INTD                     = 00000024
CRB$L_TT_MODEM                 = 00000014
CTSHIGH                          00000EA4 R       02
CTSLOW                           00000E59 R       02
DATASET                          00000B97 R       02
DECLARE_STATE                    00000BFD R       02
DEV$M_AVL                      = 00040000
DEV$M_DET                      = 00000002
DEV$M_RED                      = 00000100
DEV$V_SPL                      = 00000006
DIALTYPE                         00000BCE R       02
DISCONNECT                       00000688 R       02
DISC_FIREAST                     000006E7 R       02
DISC_STOPIO                      00000726 R       02
DT$_DZ11                       = 00000042
DYN$C_TQE                      = 0000000F
DYN$C_TYPAHD                   = 00000014
DZ11                             00000BDB R       02
DZWAIT                           00000DAF R       02
END                              00000B8E R       02
EXE$ALONONPAGED                  ********   X     02
EXE$AL_TQENOREPT                 ********   X     02
EXE$FORK                         ********   X     02
EXE$GL_ABSTIM                    ********   X     02
EXE$GQ_SYSTIME                   ********   X     02
EXE$INSTIMQ                      ********   X     02
EXE$SNDEVMSG                     ********   X     02
F                              = 00000000
FORKEXIT                         0000067C R       02
GETAHD                           00000778 R       02
IDB$L_UCBLST                   = 00000018
IDB$W_UNITS                    = 0000000C
IDLE                             00000D7C R       02
INIT                             00000B64 R       02
INIT1                            00000DC2 R       02
INIT2                            00000DCF R       02
IO$V_DSABLMBX                  = 0000000A
IO$V_REFRESH                   = 0000000D
IO$V_TIMED                     = 00000007
IOC$CLONE_UCB                    ********   X     02
IOC$INITIATE                     ********   X     02
IPL$_QUEUEAST                  = 00000006
IPL$_SYNCH                     = 00000008
IPL$_TIMER                     = 00000008
IRP$C_MEDIA                    = 00000038
IRP$L_PID                      = 0000000C
IRP$L_SVAPTE                   = 0000002C
IRP$M_FUNC                     = 00000002
IRP$V_BUFIO                    = 00000000
IRP$V_VIRTUAL                  = 00000004
IRP$W_CHAN                     = 00000028
IRP$W_FUNC                     = 00000020
IRP$W_STS                      = 0000002A
LINKFORK                         000007E3 R       02
LOGIN                            00000E72 R       02
LOGOUT                           00000E8B R       02
LOSTCTS                          00000E96 R       02
MODEM$B_ST_ONMASK              = 00000000
MODEM$B_TRAN_OFFMASK           = 00000004
MODEM$B_TRAN_ONMASK            = 00000005
MODEM$C_DATASET                = 00000003
MODEM$C_INIT                   = 00000000
MODEM$C_NULL                   = 00000002
MODEM$C_SHUTDWN                = 00000001
MODEM$C_ST_LENGTH              = 00000006
MODEM$C_TIMER                  = 00000004
MODEM$C_TRAN_DATASET           = 00000000
MODEM$C_TRAN_DIALTYPE          = 00000003
MODEM$C_TRAN_DZ11              = 00000004
MODEM$C_TRAN_END               = 00000002
MODEM$C_TRAN_LENGTH            = 00000006
MODEM$C_TRAN_NOMODEM           = 00000005
MODEM$C_TRAN_TIME              = 00000001
MODEM$LINK_CRB                   00000D51 R       02
MODEM$W_ST_ROUTINE             = 00000004
MODEM$W_ST_TIMER               = 00000002
MODEM$W_TRAN_NSTATE            = 00000002
MODEM_TIMER                      00000C36 R       02
MSG$_TRMHANGUP                 = 00000006
MSG$_TRMUNSOLIC                = 00000001
NEW_STATE                        00000BF9 R       02
NEXT_TRAN                        00000BF3 R       02
NOMODEM                          00000BE9 R       02
OFF                              00000D75 R       02
OPA$UCB0                         ********   X     02
PCB$L_PID                      = 00000060
PORTFORK                         00000680 R       02
PORT_ABORT                     = 00000020
PORT_DISCONNECT                = 00000004
PORT_DS_SET                    = 0000000C
PORT_FORKRET                   = 00000034
PORT_MAINT                     = 00000030
PORT_RESUME                    = 00000024
```

```
PORT_SET_LINE                  = 00000008
PORT_SET_MODEM                 = 00000028
PORT_STOP                      = 00000018
PORT_TRANSITION                  00000B28 RG      02
PORT_XOFF                      = 00000014
PORT_XON                       = 00000010
PR$_IPL                        = 00000012
RESTART                          000009FB RG      02
RINGWAIT                         00000D8F R       02
SET_SPEED                        000000C9 R       02
SHUT1                            00000E40 R       02
SHUTDOWN                         00000E33 R       02
SHUTDWN                          00000B71 R       02
SS$_ABORT                      = 0000002C
SS$_CANCEL                     = 00000830
SS$_DATAOVERUN                 = 00000838
SS$_HANGUP                     = 000002CC
SS$_NORMAL                     = 00000001
SS$_OPINCOMPL                  = 000002D4
SS$_PARITY                     = 000001F4
SS$_TIMEOUT                    = 0000022C
STATE_INIT                       00000D75 R       02
SYS$GL_JOBCTLMB                  ********    X    02
T                              = 0000002A
TIMCTRL$M_ACTIVE               = 00000002
TIMCTRL$M_CANCEL               = 00000001
TIMCTRL$V_ACTIVE               = 00000001
TIMCTRL$V_CANCEL               = 00000000
TIME                             00000B8F R       02
TIMER_CANCEL                     00000CB5 R       02
TIMER_END                        00000CD1 R       02
TQE$B_RQTYPE                   = 0000000B
TQE$B_TYPE                     = 0000000A
TQE$C_LENGTH                   = 00000030
TQE$C_SSREPT                   = 00000005
TQE$L_FPC                      = 0000000C
TQE$Q_DELTA                    = 00000020
TQE$W_SIZE                     = 00000008
TRANSITION                       00000B3C RG      02
TRANSITION_NOCHECK               00000B47 RG      02
TRANSMIT                         00000DFB R       02
TRANSMIT0                        00000DE2 R       02
TRANSMIT1                        00000E14 R       02
TRAN_LOOP                        00000B7B R       02
TT$C_BAUD_600                  = 00000007
TT$C_BAUD_9600                 = 0000000F
TT$L_DIALUP                      00000AEF RG      02
TT$M_DS_CARRIER                = 00000020
TT$M_DS_CTS                    = 00000010
TT$M_DS_DSR                    = 00000080
TT$M_DS_DTR                    = 00000002
TT$M_DS_RING                   = 00000040
TT$M_DS_RTS                    = 00000010
TT$M_HOSTSYNC                  = 00000010
TT$M_MBXDSABL                  = 00010000
TT$M_NOECHO                    = 00000002
TT$M_PASSALL                   = 00000001
TT$M_READSYNC                  = 00040000
TT$M_REMOTE                    = 00002000
TT$TIMER                         00000CD5 R       02
TT$TIMQUENT                      00000AF8 R       02
TT$V_HOSTSYNC                  = 00000004
TT$V_MBXDSABL                  = 00000010
TT$V_MODEM                     = 00000015
TT$V_NOTYPEAHD                 = 00000002
TT$V_PARITY                    = 00000006
TT$V_READSYNC                  = 00000012
TT$V_REMOTE                    = 0000000D
TT$W_REFCNT                      00000AF3 R       02
TT$W_TIMCTRL                     00000AF5 R       02
TT2$M_DISCONNECT               = 00020000
TT2$M_DMA                      = 00000040
TT2$V_ALTYPEAHD                = 00000007
TT2$V_AUTOBAUD                 = 00000001
TT2$V_DISCONNECT               = 00000011
TT2$V_DMA                      = 00000006
TT2$V_HANGUP                   = 00000002
TT2$V_LOCALECHO                = 00000000
TT2$V_SECURE                   = 00000010
TTY$ABORT                        00000A26 RG      02
TTY$ABORT_IO                     00000000 RG      02
TTY$AB_600                       ********    X    02
TTY$AB_9600                      ********    X    02
TTY$AUTOBAUD                     0000006C RG      02
TTY$A_CTRLU                      ********    X    02
TTY$A_MAXTIME                    ********    X    02
TTY$A_RB_PRM                   = 0000004A
TTY$B_TA_TYPE                  = 0000000A
TTY$CANCELIO                     000000D2 RG      02
TTY$CLASS_DISCONNECT             00000ED4 RG      02
TTY$CLASS_FORK                   00000628 RG      02
TTY$CRE_FORK                     0000062B RG      02
TTY$C_BELL                     = 00000007
TTY$C_BLANK                    = 00000020
TTY$C_CR                       = 0000000D
TTY$C_XOFF                     = 00000013
TTY$C_XON                      = 00000011
TTY$DISCONNECT                   00000A39 RG      02
TTY$DS_SET                       00000A46 RG      02
TTY$FORK_ADDR                    0000064D RG      02
TTY$FRAMERROR                    0000041F R       02
TTY$GB_DIALTYP                   ********    X    02
TTY$GL_JOBCTLMB                  ********    X    02
TTY$GL_TIMEOUT                   ********    X    02
TTY$GW_ALTYPAHD                  ********    X    02
TTY$GW_TYPAHDSZ                  ********    X    02
TTY$LOCK                         ********    X    02
TTY$L_TA_DATA                  = 00000118
TTY$L_TA_END                   = 00000010
TTY$L_TA_GET                   = 00000004
TTY$L_TA_PUT                   = 00000000
TTY$L_WB_BLINK                 = 00000004
TTY$L_WB_DATA                  = 00000030
TTY$L_WB_FLINK                 = 00000000
```

```
TTY$L_WB_IRP              = 00000024              TTY$V_PC_NOTIME           = 00000000
TTY$L_WB_NEXT             = 0000001C              TTY$V_ST_CTRLS            = 00000001
TTY$MAINT                   00000A53 RG    02     TTY$V_ST_CTSLOW           = 0000001D
TTY$M_PC_DMAENA           = 00000002              TTY$V_ST_PASALL           = 00000002
TTY$M_PC_XOFENA           = 00000040              TTY$V_ST_TYPFUL           = 0000000C
TTY$M_ST_AUTOP            = 00020000              TTY$V_SX_AUTOP            = 00000031
TTY$M_ST_BACKSPACE        = 00000020              TTY$V_SX_BACKSPACE        = 00000005
TTY$M_ST_BADESC           = 00000100              TTY$V_SX_BADESC           = 00000028
TTY$M_ST_CTRLO            = 00000001              TTY$V_SX_CTRLO            = 00000020
TTY$M_ST_CTRLR            = 00040000              TTY$V_SX_CTRLR            = 00000032
TTY$M_ST_CTSLOW           = 20000000              TTY$V_SX_DEL              = 00000021
TTY$M_ST_DEL              = 00000002              TTY$V_SX_EDITING          = 00000034
TTY$M_ST_EDITING          = 00100000              TTY$V_SX_EDITREAD         = 00000009
TTY$M_ST_EDITREAD         = 00000200              TTY$V_SX_EOL              = 00000008
TTY$M_ST_EOL              = 00000100              TTY$V_SX_ESC              = 00000027
TTY$M_ST_ESC              = 00000080              TTY$V_SX_ESC_O            = 0000002E
TTY$M_ST_ESC_O            = 00004000              TTY$V_SX_MULTI            = 00000006
TTY$M_ST_MULTI            = 00000040              TTY$V_SX_NOECHO           = 00000023
TTY$M_ST_NOFLTR           = 00000040              TTY$V_SX_NOFLTR           = 00000026
TTY$M_ST_OVERSTRIKE       = 00800000              TTY$V_SX_OVERSTRIKE       = 00000037
TTY$M_ST_OVRFLO           = 00010000              TTY$V_SX_OVRFLO           = 00000030
TTY$M_ST_POWER            = 00000001              TTY$V_SX_PASALL           = 00000022
TTY$M_ST_PRE              = 04000000              TTY$V_SX_POWER            = 00000000
TTY$M_ST_PROMPT           = 00000020              TTY$V_SX_PRE              = 0000003A
TTY$M_ST_QUOTING          = 00400000              TTY$V_SX_PROMPT           = 00000025
TTY$M_ST_RDVERIFY         = 00000400              TTY$V_SX_QUOTING          = 00000036
TTY$M_ST_READ             = 00001000              TTY$V_SX_RDVERIFY         = 0000000A
TTY$M_ST_RECALL           = 00000800              TTY$V_SX_READ             = 0000000C
TTY$M_ST_REFRSH           = 00000400              TTY$V_SX_RECALL           = 0000000B
TTY$M_ST_SKIPCRLF         = 00080000              TTY$V_SX_RECONNECT        = 0000003C
TTY$M_ST_SKIPLF           = 00002000              TTY$V_SX_REFRSH           = 0000002A
TTY$M_ST_TERMNORM         = 01000000              TTY$V_SX_SKIPCRLF         = 00000033
TTY$M_ST_TYPFUL           = 00001000              TTY$V_SX_SKIPLF           = 0000002D
TTY$NOTIFY                  0000031D RG    02     TTY$V_SX_TERMNORM         = 00000038
TTY$POWERACTION             00000A15 RG    02     TTY$V_SX_TYPFUL           = 0000002C
TTY$PURGE_AHEAD             0000036B RG    02     TTY$V_SX_WRITE            = 00000007
TTY$READERROR               0000038D RG    02     TTY$V_SX_WRTALL           = 00000024
TTY$READONE                 ********  X    02     TTY$WRITEDONE               ********  X    02
TTY$RESTARTIO               00000492 RG    02     TTY$WRITEPOST               ********  X    02
TTY$RESUME                  00000A60 RG    02     TTY$W_RB_CPZORG           = 0000003A
TTY$RTIMOU                  00000503 R     02     TTY$W_RB_LINOFF           = 00000030
TTY$SETUP_READ              000002B5 RG    02     TTY$W_RB_TIMOS            = 00000036
TTY$SETUP_UCB               00000568 RG    02     TTY$W_RB_TXTOFF           = 0000003C
TTY$SET_LINE                00000A79 RG    02     TTY$W_TA_INAHD            = 0000000C
TTY$SET_MODEM               00000A86 R     02     TTY$W_TA_RCLSIZ           = 00000014
TTY$STARTOUTPUT             ********  X    02     TTY$W_TA_SIZE             = 00000008
TTY$STOP                    00000A93 RG    02     TTY$W_WB_BCNT             = 0000002A
TTY$SYNCH                   ********  X    02     TTY$W_WB_STATUS           = 00000028
TTY$VT_TIMEOUT              00000535 R     02     TTY$XOFF                    00000AA6 RG    02
TTY$V_FD_BUSY             = 00000006              TTY$XON                     00000AD6 RG    02
TTY$V_FD_DISCONNECT       = 00000002              TT_END                      00000EEB RG    02
TTY$V_FD_GETAHD           = 00000001              UCB$B_AMOD                = 0000005F
TTY$V_FD_LINK             = 00000005              UCB$B_DEVTYPE             = 00000041
TTY$V_FD_PORTFORK         = 00000003              UCB$B_DIPL                = 0000005E
TTY$V_FD_UNLINK           = 00000004              UCB$B_FIPL                = 0000000B
TTY$V_FD_UNSOL            = 00000000              UCB$B_TT_DEPARI           = 000000EC
TTY$V_PC_DMAAVL           = 00000002              UCB$B_TT_DETYPE           = 000000F0
```

```
UCB$B_TT_DS_RCV                 = 00000124
UCB$B_TT_OUTYPE                 = 0000010B
UCB$B_TT_PARITY                 = 000000F8
UCB$L_AMB                       = 00000060
UCB$L_CRB                       = 00000024
UCB$L_DEVCHAR                   = 00000038
UCB$L_DEVCHAR2                  = 0000003C
UCB$L_DEVDEPEND                 = 00000044
UCB$L_DEVDEPND2                 = 00000048
UCB$L_DUETIM                    = 0000006C
UCB$L_FPC                       = 0000000C
UCB$L_FR3                       = 00000010
UCB$L_FR4                       = 00000014
UCB$L_IOQFL                     = 0000004C
UCB$L_IRP                       = 00000058
UCB$L_PDT                       = 00000084
UCB$L_PID                       = 0000002C
UCB$L_STS                       = 00000064
UCB$L_SVAPTE                    = 00000078
UCB$L_TL_BANDQUE                = 0000009C
UCB$L_TL_CTLPID                 = 000000A4
UCB$L_TL_CTRLC                  = 00000094
UCB$L_TL_CTRLY                  = 00000090
UCB$L_TL_OUTBAND                = 00000098
UCB$L_TL_PHYUCB                 = 000000A0
UCB$L_TT_DECHA1                 = 000000C8
UCB$L_TT_DECHAR                 = 000000C4
UCB$L_TT_LOGUCB                 = 000000C0
UCB$L_TT_MULTI                  = 000000D8
UCB$L_TT_PORT                   = 00000118
UCB$L_TT_RDUE                   = 000000B0
UCB$L_TT_RTIMOU                 = 000000B4
UCB$L_TT_TYPAHD                 = 000000E4
UCB$L_TT_WBLINK                 = 000000D0
UCB$L_TT_WFLINK                 = 000000CC
UCB$L_TT_WRTBUF                 = 000000D4
UCB$M_BSY                       = 00000100
UCB$M_DELETEUCB                 = 00010000
UCB$M_INT                       = 00000002
UCB$M_JOB                       = 00000001
UCB$M_ONLINE                    = 00000010
UCB$M_POWER                     = 00000020
UCB$M_TIM                       = 00000001
UCB$M_TT_HANGUP                 = 00000008
UCB$M_TT_LEN                    = 00000018
UCB$M_TT_NOTIF                  = 00000004
UCB$M_TT_TIMO                   = 00000002
UCB$Q_TL_BRKTHRU                = 000000A8
UCB$Q_TT_STATE                  = 000000B8
UCB$V_BSY                       = 00000008
UCB$V_JOB                       = 00000000
UCB$V_POWER                     = 00000005
UCB$V_TT_DISPARERR              = 00000001
UCB$V_TT_NOTIF                  = 00000002
UCB$V_TT_PARTY                  = 00000006
UCB$V_TT_TIMO                   = 00000001
UCB$V_TT_USERFRAME              = 00000002

UCB$W_BOFF                      = 0000007C
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_DEVSTS                    = 00000068
UCB$W_ERRCNT                    = 00000082
UCB$W_REFC                      = 0000005C
UCB$W_STS                       = 00000064
UCB$W_TT_CURSOR                 = 000000FC
UCB$W_TT_DESPEE                 = 000000E8
UCB$W_TT_DS_ST                  = 00000126
UCB$W_TT_DS_TIM                 = 00000128
UCB$W_TT_HOLD                   = 00000108
UCB$W_TT_MULTILEN               = 000000DC
UCB$W_TT_PRTCTL                 = 00000122
UCB$W_TT_SPEED                  = 000000F4
UNLINKFORK                        00000854 R     02
UNSOL                             000008EB R     02
VEC$L_IDB                       = 00000008
VERIFY                            00000EB7 R     02
VT$UCB                            ********   X   02
W0                              = 00000000
W1                              = 00000004
WAIT                              00000D9C R     02
X                               = 00000000
X0                              = 00000000
X1                              = 00000001
Z0                              = 00000003
Z1                              = 00000001
```

```
                                   +----------------+
                                   ! Psect synopsis !
                                   +----------------+

PSECT name                   Allocation          PSECT No.  Attributes
----------                   ----------          ---------  ----------
.  ABS  .                    00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                        00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$$$115_DRIVER                00000EEB  ( 3819.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC QUAD

                                +------------------------+
                                ! Performance indicators !
                                +------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                    29    00:00:00.05    00:00:02.47
Command processing               137    00:00:00.43    00:00:03.30
Pass 1                           813    00:00:25.13    00:01:31.40
Symbol table sort                  0    00:00:03.35    00:00:12.59
Pass 2                           412    00:00:06.29    00:00:22.94
Symbol table output                1    00:00:00.25    00:00:01.07
Psect synopsis output              0    00:00:00.02    00:00:00.02
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals            1394    00:00:35.52    00:02:13.79

The working set limit was 2700 pages.
212887 bytes (416 pages) of virtual memory were used to buffer the intermediate code.
There were 170 pages of symbol table space allocated to hold 3037 non-local and 147 local symbols.
2754 source lines were read in Pass 1, producing 26 object records in Pass 2.
65 pages of virtual memory were used to define 62 macros.

                               +--------------------------+
                               ! Macro library statistics !
                               +--------------------------+

Macro library name                     Macros defined
------------------                     --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                     28
_$255$DUA28:[SYSLIB]STARLET.MLB;2                  11
TOTALS (all libraries)                             39

3384 GETS were required to define 39 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:TTYSUB/OBJ=OBJ$:TTYSUB MSRC$:TTYSUB/UPDATE=(ENH$:TTYSUB)+EXECML$/LIB
```

0404 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

TTYSYNCH
LIS

TTYSTRSTP
LIS

TTYSUB
LIS

YCDRIVER
LIS