

```

C850:      79 * versions of these values for its own use.
C850:      80 * COPY USER'S CURSOR IF IT DIFFERS FROM
C850:      81 * WHAT WE LAST PUT THERE:
C850:      82 *
C850:A5 25 83 CSETUP LDA CV ;set up OURCV
C852:8D FB 05 84 STA OURCV
C855:A4 24 85 LDY CH ;GET IT
C857:CC 7B 04 86 CPY OLDCH ;IS IT THE SAME?
C85A:F0 03 C85F 87 BEQ CS2 ;=>YES, USE OUR OWN
C85C:8C 7B 05 88 STY OURCH ;update our cursor
C85F:A5 21 89 CS2 LDA WNDWDTH ;cursor horizontal must not
C861:18 90 CLC ;be greater than window width
C862:ED 7B 05 91 SBC OURCH ;if it is, then put cursor
C865:B0 05 C86C 92 BCS CS3 ;at left edge of window
C867:A0 00 93 LDY #0
C869:8C 7B 05 94 STY OURCH
C86C:AC 7B 05 95 CS3 LDY OURCH ;exit with Y = CH
C86F:60 96 RTS
C870: 97 *
C870: 98 * BIN and BOUT are used when characters are
C870: 99 * input and output by the $F8 ROM while 80VID
C870: 100 * is on. They cannot use the $C3 entry points
C870: 101 * because that switches in the $C8 space, causing
C870: 102 * possible conflict with other $C8 users.
C870: 103 * These routines are only called by the $C100-$C2FF space.
C870: 104 *
C870: 105 * These entry points will only work if the card was
C870: 106 * first initialized using a PR#3. 80 columns will not
C870: 107 * work simply by turning on the 80VID flag.
C870: 108 *
C870:A4 35 109 BOUT LDY SAVY1 ;load Y stuffed by $F8 ROM
C872:18 110 CLC ;signal an output
C873:B0 FE C873 111 BCS * ;skip SEC
C874: 112 ORG *-1
C874:38 C874 113 BIN SEC ;signal an input
C875:8D 7B 06 114 STA CHAR ;save the char
C878:98 115 TYA ;save Y
C879:48 116 PHA
C87A:8A 117 TXA ;save X
C87B:48 118 PHA
C87C: 119 C8BASIC EQU * ;BASIC IN/OUT
C87C:B0 5E C8DC 120 BCS BINPUT ;=>input a character
0000: 0000 1 TEST EQU 0 ;REAL VERSION
C87E: 23 LST ON,A,V
C87E: 24 INCLUDE BPRINT
C87E: 1 *
C87E: 2 * This is the place where characters printed using the
C87E: 3 * CSW hook are actually printed (or executed if they are
C87E: 4 * control characters).
C87E: 5 *
C87E: 6 BPRINT JSR CSETUP ;setup user cursor
C881:AD 7B 06 7 LDA CHAR ;GET CHARACTER
C884:C9 8D 8 CMP #8D ;IS IT C/R?
C886:D0 18 C8A0 9 BNE NOWAIT ;=>don't wait, OURCH ok

C888:AE 00 C0 10 LDX KBD ;IS KEY PRESSED?
C88B:10 13 C8A0 11 BPL NOWAIT ;NO
C88D:E0 93 12 CPX #93 ;IS IT CTL-S?
C88F:D0 0F C8A0 13 BNE NOWAIT ;NO, IGNORE IT
C891:2C 10 C0 14 BIT KBDSTRB ;CLEAR STROBE
C894:AE 00 C0 15 KBDWAIT LDX KBD ;WAIT FOR NEXT KEYPRESS
C897:10 FB C894 16 BPL KBDWAIT
C899:E0 83 17 CPX #83 ;IF CTL-C, LEAVE IT
C89B:F0 03 C8A0 18 BEQ NOWAIT ; IN THE KBD BUFFER
C89D:2C 10 C0 19 BIT KBDSTRB ;CLEAR OTHER CHARACTER
C8A0:29 7F 20 NOWAIT AND #$7F ;drop possible hi bit
C8A2:C9 20 21 CMP #$20 ;IS IT CONTROL CHAR?
C8A4:B0 06 C8AC 22 BCS BPNCTL ;=>NOPE
C8A6:20 D2 CA 23 JSR CTLCHARO ;execute CTL if M.CTL ok
C8A9:4C BD C8 24 JMP CTLOM ;=>enable ctrl chrs
C8AC: 25 *
C8AC: 26 * NOT A CTL CHAR. PRINT IT.
C8AC: 27 *
C8AC: 28 BPNCTL EQU *
C8AC:AD 7B 06 29 LDA CHAR ;get char (all 8 bits)
C8AF:20 38 CE 30 JSR STORCHAR ;and display it
C8B2: 31 *
C8B2: 32 * BUMP THE CURSOR HORIZONTAL:
C8B2: 33 *
C8B2:C8 34 INY ;bump it
C8B3:8C 7B 05 35 STY OURCH ;are we past the
C8B6:C4 21 36 CPY WNDWDTH ; end of the line?
C8B8:90 03 C8BD 37 BCC CTLOM ;>NO, NO PROBLEM
C8BA:20 51 CB 38 JSR X.CR ;YES, DO C/R
C8BD: 39 *
C8BD: 40 * M.CTL is set by RDCHAR and cleared here, after each
C8BD: 41 * character is displayed.
C8BD: 42 *
C8BD:AD FB 04 43 CTLOM LDA MODE ;enable printing of control char
C8C0:29 F7 44 AND #255-M.CTL
C8C2:8D FB 04 45 STA MODE
C8C5:AD 7B 05 46 BIORET LDA OURCH ;get newest cursor position
C8C8:2C 1F C0 47 BIT RDBOVID ;IN 80-MODE?
C8CB:10 02 C8CF 48 BPL SETALL ;>no, set other cursors
C8CD:A9 00 49 LDA #0 ;pin CH to 0 for 80 columns
C8CF:85 24 50 SETALL STA CH
C8D1:8D 7B 04 51 STA OLDCH ;REMEMBER THE SETTING
C8D4:68 52 GETREGS PLA ;RESTORE
C8D5:AA 53 TAX
C8D6:68 54 PLA ;X AND Y
C8D7:A8 55 TAY
C8D8:AD 7B 06 56 LDA CHAR
C8DB:60 57 RTS ;RETURN TO BASIC
C8DC: 25 INCLUDE BINPUT
C8DC: 1 *
C8DC: 2 * BASIC input entry point called by entry point in the
C8DC: 3 * $C3 space. This is the way things normally happen.
C8DC: 4 *
C8DC:A4 24 5 BINPUT LDY CH

```