SYMCOD
1171 Notre-Dame O. # 100
Victoriaville, Qc
Canada
G6P 7L1

Telephone: (819) 751-0095
Fax: (819) 751-1292

VT-52 EMULATION

Documentation Ver. 1.3b (April.2, 2003)

Use this software with the product : LBC-02 VT-52

Our Internet site: http:// www.symcod.com/

Programmer Analyst: Steve Bilodeau
Email: sbill@symcod.com

LBC-02 VT-52 Emulation

These control functions are initiated by a specific additional character, the escape code. Escape (or shortened to ESC) has the ASCII code 27. Another letter follows the ESC code, which determines the function as well as additional parameters, if required. The following summary contains a list of all control codes supported by the SYMCOD terminal.

ESC A (Cursor Up)

This enables cursor movement upwards by one line. If the cursor is already located in the uppermost line, nothing happens.

ESC B (Cursor Down)

This ESC sequence moves the cursor down by one line. If the cursor is already located in the bottommost line, nothing happens.

ESC C (Cursor Right)

This sequence moves the cursor one column to the right.

ESC D (Cursor Left)

This sequence moves the cursor one position to the left. This function is identically to the control code 'Backspace' (BS = ASCII code 8). If the cursor is already located in the first column, nothing happens.

ESC E (Clear Home)

This control sequence clears the entire screen and positions the cursor into the upper left screen corner (Home Position).

ESC H (Cursor Home)

This sequence moves the cursor into the upper left screen corner; however, without deleting screen content.

ESC I (Cursor Up)

This sequence moves the cursor up by a line. Unlike ESC A however, a blank line is inserted if the cursor was already in the uppermost line; the rest of the screen scrolls down one line accordingly. The column position of the cursor remains unchanged.

ESC J (Clear Remaining Screen)

This function is used to clear the remaining screen starting at and including the current cursor position. The cursor stays at the same position.

ESC K (Delete Rest of Line)

This ESC sequence deletes the rest of the line in which the cursor is located. Deletion takes place starting at and including the current cursor position. The cursor position itself is not changed.

ESC Y # # (Positioning Cursor)

This is one of the most important functions. It allows for positioning the cursor on the screen any way desired. For this purpose, the function still needs the cursor line and columns as parameters, which are expected in this sequence with an offset of 32. For example, if the user wants to place the cursor in line 5, column 20, the following command has to be issued:

```
CHR$(27)+"Y"+CHR$(32+5)+CHR$(32+20)
```

Lines and columns are counted starting at zero. For a 40 X 2 screen, the lines are numbered from 0 to 1 and the columns from 0 to 39.

ESC d (Clear Screen up to Cursor Position)

This sequence clears the screen starting at and including the current cursor position. The position of the cursor remains unchanged.

ESC e (Cursor On)

This escape sequence makes the cursor become visible.

ESC f (Cursor Off)

The cursor is deactivated again.

ESC j (Save Cursor Position)

This sequence is used to save the current cursor position.

ESC k (Restore Cursor to Saved Position)

This is the counterpart to the above function. The cursor is returned to the position previously stored with ESC j.

ESC l (Delete Line)

The content of the line currently containing the cursor is deleted. All remaining lines are unaffected. After the deletion, the cursor is located in the first column of the deleted line.

ESC o (Delete Beginning of Line)

Deletes the beginning of the cursor line up to and including the cursor position. The position of the cursor remains unchanged.

SPECIAL FUNCTIONS FOR TERMINALS

ESC a # (Mode Block ON/OFF)

Set the terminal "Mode Block" to 0-OFF or 1-ON (Default = OFF)
When in Block Mode, the terminal store into his internal memory all punched keys until the ENTER key is press. After the ENTER key is press, the keyboard is locked and the LED turn on until the host answered. The keys ENTER, ESC, M1 to M5 are always active.
Example Mode Block ON: Chr(27)+"a1"

ESC g # (Terminal TTL IN 1 Enable / Desable / Request Status)

Sets the terminal TTL IN 1 to 0-DESABLE, 1-ENABLE 2-Request Status (Default = Desable)

Example TTL IN 1 ENABLE: Chr(27)+"g1"

Example Request Status of TTL IN 1: Chr(27)+"g2"

ESC h # (Terminal TTL IN 2 Enable / Desable / Request Status)

Sets the terminal TTL IN 2 to 0-DESABLE, 1-ENABLE 2-Request Status (Default = Desable)

Example TTL IN 2 ENABLE: Chr(27)+"h1"

Example Request Status of TTL IN 2: Chr(27)+"h2"

ESC y (Terminal Full Duplex emulation)

Sets the terminal to FULL DUPLEX emulation. In this mode, the "local echo" of the terminal is set to OFF.

ESC z (Terminal Alf Duplex emulation)

Sets the terminal to HALF DUPLEX emulation. In this mode, the "local echo" of the terminal is set to ON (Default).

ESC x # (WatchDog Configuration)

Each terminal has 2 WatchDog protection that you can activate.

In mode:

0 - All WatchDogs are set to OFF (Default).

1- Sets TX WatchDog to ON; if you set this option, the terminal will send the character CHR(251) each 15 seconds.

2- Sets RX WatchDog to ON; if you set this option, after each 13 seconds of inactivity, you must send with your software the character CHR(254); otherwise, the terminal resets the connection TC/IP.

For example: to put all WatchDog OFF on a terminal: CHR(27)+”x0”

Additional Control Characters Not Initiated By ESC:

CHR$(7) (BEL, Audio Output)

Emits a sound for about one second.

CHR$(8) (BS, Backspace)

The cursor moves one position to the left and deletes the character below the cursor.

CHR$(10) (LF, Line Feed)

A line feed is triggered.

CHR$(12) (CLEAR, Clear Home)

This control sequence clears the entire screen and positions the cursor into the upper left screen corner (Home Position).

CHR$(13) (CR, Carriage Return)

The cursor is moved to the left margin of the line.

CHR$(179) (LED, Light led)

Lights LED for about one second.

CHR$(240) (TIME, Show time)

Shows and automatically refreshes time at the current cursor position.

CHR$(241) (DATE, Show date)

Shows and automatically refreshes date at the current cursor position.

CHR$(242) (DISABLE REFRESH, Set refresh to OFF)

Sets the refreshed date and time to OFF.

CHR(177)+"HHMMSS"+CHR(3) (TIME, Set the current time)

Example for 22:10:04 : CHR(177)+"221004"+CHR(3)

CHR(189)+"MMDDYY"+CHR(3) (DATE, Set the current date)

Example for December 28th, 2002: CHR(189)+"122802"+CHR(3)

COMMANDS FOR TTL OUTPUTS

- **Output signal 1-> ON xx SECONDS (220)**
  xx = +- 1 second
  Syntax: CHR(220)+"99"+CHR(3)
  Sets logical output 1 to ON for xx seconds.
  Example for 4 seconds: CHR(220)+"04"+CHR(3)

- **Output signal 2-> ON xx SECONDS (221) (xx = +- 1 second)**
  Syntax: CHR(221)+"99"+CHR(3)

- **Output signal 3-> ON xx SECONDS (222) Available in version 5.8b and higher**
  xx = +- 1 second
  Syntax: CHR(222)+"99"+CHR(3)

- **Output signal 4-> ON xx SECONDS (223) Available in version 5.8b and higher**
  xx = +- 1 second
  Syntax: CHR(223)+"99"+CHR(3)

- **OUTPUT SIGNAL 1-> ON (193)**
  Syntax: CHR(193)+CHR(3)
  Sets logical output 1 to logical level 1.

- **OUTPUT SIGNAL 1-> OFF(196)**
  Syntax: CHR(196)+CHR(3)
  Sets logical output 1 to logical level 0.

- **OUTPUT SIGNAL 2-> ON (192)**
  Syntax: CHR(192)+CHR(3)
  Sets logical output 2 to logical level 1.

- **OUTPUT SIGNAL 2-> OFF(195)**
  Syntax: CHR(195)+CHR(3)

Sets logical output 2 to logical level 0.

- **OUTPUT SIGNAL 3-> ON (226)**
  Syntax: CHR(226)+CHR(3)
  Sets logical output 3 to logical level 1.

- **OUTPUT SIGNAL 3-> OFF(227)**
  Syntax: CHR(227)+CHR(3)
  Sets logical output 3 to logical level 0.

- **OUTPUT SIGNAL 4-> ON (228)**
  Syntax: CHR(228)+CHR(3)
  Sets logical output 4 to logical level 1.

- **OUTPUT SIGNAL 4-> OFF(229)**
  Syntax: CHR(229)+CHR(3)
  Sets logical output 4 to logical level 0.

CHARACTERS RECEIVED FROM THE TERMINAL

ASCII 253 (Terminal BOOT mode)

If the terminal send to your software the character CHR(253), it is to indicate that this terminal is in BOOT mode; then, to go to the TERMINAL mode, you must send:
Chr(1) + Chr(255) + Chr(255) + "A" + Chr(250)

KEYBOARD SPECIAL KEYS

ASCII 8 (Terminal Keyboard key BACKSPACE)

ASCII 13 (Terminal Keyboard key ENTER)

ASCII 27 (Terminal Keyboard key ESC)

ASCII 134 (Terminal Keyboard key M1)

ASCII 135 (Terminal Keyboard key M2)

ASCII 136 (Terminal Keyboard key M3)

ASCII 137 (Terminal Keyboard key M4)

ASCII 138 (Terminal Keyboard key M5)

UDP BROADCAST “FROM TERMINALS”

If the TCPIP connection isn’t establish with a terminal, it will transmit after each three seconds this UDP broadcast:

NET MASK	255.255.255.255
PORT	2024
STRING EXAMPLE	úû10.1.201.6,255.255.0.0,0.0.0.0,57600,0,0,SYMCOD_CONNECT

STRING DESCRIPTION:

CHR(250)+CHR(251)+IP_TERM+”,”+MASK_TERM+”,”+GATE_TERM+”,”+BAUDS+”,”+VAL_PARITY+”,”+VAL_DATABITS+”,SYMCOD_CONNECT”