

# melabs Serial LCD

Firmware Version 1.2  
8/11/09

The melabs Serial LCD (SLCD) can display serial data from either asynchronous RS232-style or synchronous I<sup>2</sup>C input. A range of baud rates from 2400 to 57,600 is available and the serial data is buffered so that no delay is necessary when transmitting data to the SLCD.

The SLCD is available with a 16-character by 2-line or a 20-character by 4-line liquid crystal display. Other display formats may be available by request. The display includes an LED backlight with variable brightness. The melabs Serial LCD board is also available separately to attach to other displays that utilize a Hitachi HD44780 or equivalent controller IC. Instructions for attaching a different display are toward the end of this document.

## Serial Connections

Serial data, along with power, is supplied to the SLCD through connectors J1 and J2. Connector J1 is for asynchronous RS232-style serial data and connector J2 is for synchronous I<sup>2</sup>C serial data. The SLCD includes an 80 character buffer so that no delays are necessary when sending usual amounts of serial data to it.

The SLCD can also accept data from both the asynchronous and synchronous inputs at the same time. However, it does not differentiate between the inputs so care must be taken when sending commands from one input so that no characters from the other input interfere with their interpretation.

Also note that jumpers JP1 and JP2 affect both serial inputs. If a specific baud rate is necessary, be sure to use the I<sup>2</sup>C address offset that matches that jumper setting.

## RS232-style USART Serial Connection

Asynchronous RS232-style serial data is applied to the pin labeled 'RX' on connector J1. The data format is a start bit followed by 8 data bits and a stop bit (8N1). There is no parity.

J1:

Pin	Name	Function
1	+5	+5 volts
2	G	Ground
3	RX	Serial data

The default baud rate is 9600. Other baud rates may be selected by shorting jumpers JP1 and/or JP2. The table below shows the baud rate selections. Note that these same jumpers are also used to select the I<sup>2</sup>C address offset.

<b>Baud</b>	<b>JP1</b>	<b>JP2</b>
2400	SHORTED	SHORTED
9600 (default)	OPEN	OPEN
19,200	OPEN	SHORTED
57,600	SHORTED	OPEN

Jumper JP3 selects whether the serial signal is true or inverted. The default (open) state is inverted. This allows for direct connection to a PC without the need for an RS232 driver IC. The true state can be selected by shorting JP3.

As mentioned, direct connection to a PC is possible. The SLCD includes a series resistor and diode so that it can handle voltages between +15 and -15 volts on the RX line only, as may be present on a PC RS232 port.

Below is an example of sending data to the display using the PICBASIC™ or PICBASIC PRO™ Compiler. In this example, the data is sent on PORTB.0 at 9600 baud.

```
Serout 0, N9600, ("Hello") ' Display Hello using RS232-style connection
```

### **I<sup>2</sup>C Serial Connection**

I<sup>2</sup>C synchronous serial data is applied to connector J2. ‘SC’ is the serial clock connection and ‘SD’ is the serial data connection. The SLCD board includes a 4.7k pullup resistor on each of these lines.

J2:

<b>Pin</b>	<b>Name</b>	<b>Function</b>
1	+5	+5 volts
2	SD	Serial data
3	G	Ground
4	SC	Serial clock

The default I<sup>2</sup>C address is hexadecimal 50. This default can be changed using a command. See the section on commands later in these instructions for more information. Jumpers JP1 and JP2 are used to select an address offset. The alternate addresses for the default I<sup>2</sup>C address are shown in the table below. Note that these same jumpers are also used to select the baud rate.

<b>I<sup>2</sup>C Address</b>	<b>JP1</b>	<b>JP2</b>
Hex 50 (default)	OPEN	OPEN
Hex 52	SHORTED	OPEN
Hex 54	OPEN	SHORTED
Hex 56	SHORTED	SHORTED

Below is an example of sending data to the display using the PICBASIC PRO™ Compiler. In this example, the I2C serial data is sent on PORTC.4 and the I2C serial clock is sent on PORTC.3 to address hexadecimal 50.

```
I2CWrite PORTC.4,PORTC.3,$50,["Hello"]      ' Display Hello using I2C connection
```

### **Power Supply**

The SLCD is powered by 5 volts to either of the 2 serial input connectors, J1 or J2. The connectors are labeled '+5' for the 5 volt connection and 'G' for the ground connection. During normal operation, very little current is consumed by the SLCD. At 5 volts with the backlight off, the current consumption is typically less than 5ma. However, turning on the LED backlight can add hundreds of milliamps to the power requirement.

### **Contrast Control**

POT1 on the SLCD is used to set the contrast for the LCD. If it is turned too far in one direction, the screen will just show dark rectangles. If it is turned too far the other direction, the screen will appear to be blank. Adjust it for the best character visibility, darkest characters on the lightest background. A test screen can be displayed to provide visible characters for adjusting the potentiometer.

### **Test Screen**

Connecting both 'SC' and 'SD' on connector J2 to ground on power up will display a test screen for a short time. Depending on the size of the display, this screen shows the SLCD firmware version number, baud rate and mode, I2C address, backlight level, and the number of characters and lines the controller is set for. The LED backlight will also be lit to approximately half brightness. The display will clear and enter regular operation after 10 seconds.

### **Characters**

The melabs Serial LCD displays ASCII characters sent to it over the serial connection. These are actually numeric codes that tell the LCD what character to display. Depending on the software

that you use to send data to the display, you might send messages by typing strings literally (“Hello”), or you might type the numbers that represent the characters that you want to see (hexadecimal 58 or decimal 88 is the same as “X”). Refer to a standard ASCII character chart to learn what numbers will display a given character.

There are also 8 additional characters available that are not shown in the ASCII character table for the display. These custom characters are accessed by their numeric value only as there is no keyboard letter to type for them. They are character numbers 0 through 7. Character 0 is a symbol for plus/minus. Character 1 is ½ (one half) represented as a single character. Character 2 is a superscript 2. Character 3 is a superscript 3. Character 4 is the copyright symbol. Character 5 is the Greek character mu (used for micro). Character 6 is the Greek character omega (used for ohms). Character 7 is the degree symbol.

These custom characters are loaded by default when the display is first turned on and anytime the reset command is executed. The first 5 are overwritten when the horizontal bar graph is initialized. All 8 are overwritten when one of the vertical bar graphs are initialized.

Serout 0, N9600, (0, “100”)	‘ Display plus/minus 100
Serout 0, N9600, (“55”, 1)	‘ Display 55 and a half
Serout 0, N9600, (“X”, 2)	‘ Display X squared
Serout 0, N9600, (“Y”, 3)	‘ Display Y cubed
Serout 0, N9600, (4, “melabs”)	‘ Display copyright melabs
Serout 0, N9600, (5)	‘ Display the Greek letter mu (micro symbol)
Serout 0, N9600, (“100”, 6)	‘ Display 100 followed by the ohms symbol
Serout 0, N9600, (“75”, 7)	‘ Display 75 followed by the degree symbol

You can also load your own custom characters to replace these or the bar graph characters. The data sheet for the Hitachi HD44780 on the melabs web site at <http://melabs.com/slcd> has the details.

There are a few other characters with special meaning to the SLCD. The form feed character clears the screen and puts the cursor at the beginning of the first line. Backspace moves the cursor back one position without clearing the character at that position. Carriage return causes the cursor to be placed at the beginning of the current line. Line feed moves the cursor down to the next line, or optionally scrolls the display up if it was on the last line. See the section on commands for more information on different carriage return and line feed modes.

There are also 2 characters that start 2 different format commands. Each is described in the sections on commands below.

## **Character Positioning**

The melabs Serial LCD offers several ways to specify the cursor location (i.e. the next position for characters to appear on the display). Characters normally appear on the display one after the

other as they are written to the next location in memory. When the end of the line is reached, the next character is also written to the next location in memory. This location may not be the first position of the next line on the display. It may not even be in displayable memory. Using one of the methods outlined below allows the cursor to be positioned at any desired location.

The first method for moving the cursor to a new location is detailed in the Command Format 1 section of these instructions and can be used to move the cursor directly to any position on the display. It uses a 2-byte sequence to send an address directly to the display controller:

Serout 0, N9600, (\$fe, \$83) ‘ Move cursor to 4<sup>th</sup> character position of first line

The second method uses escape sequences shown in the section on Command Format 2. Separate commands are available to move to the beginning of each line on the display. For example:

Serout 0, N9600, (\$1b, \$32) ‘ Move cursor to start of second line

The cursor can also be moved to a particular character position on the current line using escape sequences. These consist of an escape followed by the number of the desired position (starting with 1) with hexadecimal 80 (decimal 128) added to it. As examples:

Serout 0, N9600, (\$1b, \$80 + 1) ‘ Move cursor to the first character on current line

Serout 0, N9600, (\$1b, \$80 + 8) ‘ Move cursor to the 8<sup>th</sup> character on current line

The cursor can also be moved around using carriage return and line feed. Carriage return moves the cursor to first position of the current line. Line feed moves the cursor down to the next line on the display, in the same column. The normal sequence at the end of a line would be a carriage return followed by a line feed to move the cursor to the first column of the next line.

There is a command that may be used to alter the way carriage returns and line feeds work. The Set Carriage Return Line Feed Modes command can be used to cause a line feed to automatically be added after a carriage return is received, or a carriage return to be added after a line feed is received. This makes the cursor move to the beginning of the next line if only a carriage return or a line feed is received - receiving both is not required. This command can also be used to specify what happens if the cursor is on the last line of the display and a line feed is received. By default, when a line feed is received with the cursor on the last line of the display, the cursor will move up to the first line. The Set Carriage Return Line Feed Modes command can be used to specify that the lines on the display are scrolled up one position if a line feed is received with the cursor on the last line. The cursor would then remain on the last line.

## **Command Format 1**

The first command format is always 2 bytes long, starting with a hexadecimal fe (decimal 254) followed by a command byte that is sent directly to the Hitachi HD44780 controller on the actual

LCD. The commands that may be sent are outlined in the data sheet for the Hitachi controller available on the melabs web site at <http://melabs.com/slcd>. Some of the more useful commands are shown here.

Commands are available to clear the display and set the cursor position. For example, to clear the display, send the hexadecimal value fe (or decimal 254), followed by the value 1. Some of the other commands are shown in the table below.

Hex Command	Operation
<b>fe, 1</b>	Clear display and move cursor to beginning of first line
<b>fe, 80</b>	Move cursor to beginning of first line
<b>fe, c0</b>	Move cursor to beginning of second line
<b>fe, 94</b>	Move cursor to beginning of third line
<b>fe, d4</b>	Move cursor to beginning of fourth line

Serout 0, N9600, (\$fe, \$1) ‘ Clear screen and move cursor to start of first line

The characters are mapped to addresses on the display, as shown in the charts below. To move the cursor to a specific address, send the hexadecimal value fe (or decimal 254), followed by the address from one of the charts below. For example, to move the cursor to the 4<sup>th</sup> character on line 1, send the hexadecimal value fe, followed by the hexadecimal value 83.

Serout 0, N9600, (\$fe, \$83) ‘ Move cursor to 4<sup>th</sup> character position of first line

Character positions in hexadecimal for 16x2 display:

80	81	82	83	84	85	85	87	88	89	8a	8b	8c	8d	8e	8f
c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf

Character positions in hexadecimal for 20x4 display:

80	81	82	83	84	85	85	87	88	89	8a	8b	8c	8d	8e	8f	90	91	92	93
c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf	d0	d1	d2	d4
94	95	96	97	98	99	9a	9b	9c	9d	9e	9f	a0	a1	a2	a3	a4	a5	a6	a7
d4	d5	d6	d7	d8	d9	da	db	dc	dd	de	df	e0	e1	e2	e3	e4	e5	e6	e7

## Command Format 2

The second of the two command formats uses an escape sequence which starts with an escape character, (hexadecimal 1b or decimal 27) and is followed by a command character and optionally followed by one or more data characters. For example, to clear the screen, send an escape (hexadecimal 1b or decimal 27) followed by a ‘0’ (hexadecimal 30 or decimal 48) :

Serout 0, N9600, (\$1b, \$30) ‘ Clear screen and move cursor to start of first line

The escape sequences are summarized in the following table. Detailed information for each command, including examples written in PICBASIC™, follows it.

Escape Code	Hexadecimal Code	Data	Command
ESC *	1b 2a	<i>n</i>	Set backlight brightness to <i>n</i>
ESC +	1b 2b		Scroll display down one line
ESC -	1b 2d		Scroll display up one line
ESC 0	1b 30		Clear screen and go to beginning of line 1
ESC 1	1b 31		Go to beginning of line 1
ESC 2	1b 32		Go to beginning of line 2
ESC 3	1b 33		Go to beginning of line 3
ESC 4	1b 34		Go to beginning of line 4
ESC <	1b 3c	<i>n</i>	Display horizontal bar graph of <i>n</i> bars from right to left
ESC >	1b 3e	<i>n</i>	Display horizontal bar graph of <i>n</i> bars from left to right
ESC B	1b 42		Blinking cursor on
ESC F	1b 46		Blinking or underline cursor off
ESC H	1b 48	<i>n</i>	Set up horizontal bar graph of <i>n</i> bars maximum length
ESC R	1b 52	<i>n c</i>	Repeat character <i>c</i> , <i>n</i> times
ESC S	1b 53	<i>n</i>	Display string starting at EEPROM address <i>n</i>
ESC U	1b 55		Underline cursor on
ESC V	1b 56	<i>n</i>	Set up vertical bar graph of <i>n</i> bars maximum length going from top down
ESC ^	1b 5e	<i>n</i>	Set up vertical bar graph of <i>n</i> bars maximum length going from bottom up
ESC c	1b 63	<i>n</i>	Set number of characters per line to <i>n</i>
ESC dw	1b 64 77		Save defaults to EEPROM
ESC i	1b 69	<i>n</i>	Set I <sup>2</sup> C address to <i>n</i>
ESC l	1b 6c	<i>n</i>	Set number of lines to <i>n</i>
ESC m	1b 6d	<i>n</i>	Set carriage return line feed modes
ESC w	1b 77	<i>n s</i>	Write string <i>s</i> to EEPROM address <i>n</i>
ESC z	1b 7a		Reset the display
ESC	1b 7c	<i>n</i>	Display vertical bar graph of <i>n</i> bars

ESC 80 + $n$	1b 80 + $n$		Move cursor to position $n$ on current line
--------------	-------------	--	---

**Set Backlight Brightness** ESC \*  $n$  (1b 2a  $n$ )

The brightness of the LED backlight of the display may be set from off to full on in 256 steps using this command. First an escape character is sent (hexadecimal 1b or decimal 27) followed by an asterisk (hexadecimal 2a or decimal 42) and then a single character, represented by  $n$  above, of hexadecimal 0 to ff (decimal 0 to 255) where 0 is off and ff (255) is full on.

Serout 0, N9600, (\$1b, \$2a, \$80) ‘ Set LCD backlight to about half brightness

**Clear Screen** ESC 0 (1b 30)

Erase the display and position the cursor at the first character of the first line.

Serout 0, N9600, (\$1b, \$30) ‘ Clear screen and move cursor to start of first line

**Line 1** ESC 1 (1b 31)

Position the cursor at the first character of the first line.

Serout 0, N9600, (\$1b, \$31) ‘ Move cursor to start of first line

**Line 2** ESC 2 (1b 32)

Position the cursor at the first character of the second line.

Serout 0, N9600, (\$1b, \$32) ‘ Move cursor to start of second line

**Line 3** ESC 3 (1b 33)

Position the cursor at the first character of the third line.

Serout 0, N9600, (\$1b, \$33) ‘ Move cursor to start of third line

**Line 4** ESC 4 (1b 34)

Position the cursor at the first character of the fourth line.

Serout 0, N9600, (\$1b, \$34) ‘ Move cursor to start of fourth line

**Position cursor on line** ESC 80 + n (1b 80 + n)

The cursor can be moved to a particular character position on the current line using this command. This command is a bit different than the other commands and consists of an escape followed by the number of the desired position (starting with 1) with hexadecimal 80 (decimal 128) added to it.

Serout 0, N9600, (\$1b, \$80 + 1) ‘ Move cursor to the first character on current line  
Serout 0, N9600, (\$1b, \$80 + 8) ‘ Move cursor to the 8<sup>th</sup> character on current line

**Scroll Down** ESC + (1b 2b)

The Scroll Down command moves each line on the display down one position. The cursor is then positioned to the first character of the first line.

Serout 0, N9600, (\$1b, \$2b) ‘ Scroll down one line

**Scroll Up** ESC - (1b 2d)

The Scroll Up command moves each line on the display up one position. The cursor is then positioned to the first character of the second or fourth line, depending on the display format. See the Set Number of Chars and Set Number of Lines commands for more information on setting the display format.

Serout 0, N9600, (\$1b, \$2d) ‘ Scroll up one line

**Blinking Cursor** ESC B (1b 42)

Normally the cursor position is not indicated on the display. A new character just shows up at the next position. A blinking block cursor can be displayed by executing this Blinking Cursor command.

Serout 0, N9600, (\$1b, \$42) ‘ Turn on blinking cursor

**Underline Cursor** ESC U (1b 55)

Normally the cursor position is not indicated on the display. A new character just shows up at the next position. An underline cursor can be displayed by executing this Underline Cursor

command.

Serout 0, N9600, (\$1b, \$55) ‘ Turn on underline cursor

### **Cursor Off** ESC F (1b 46)

Executing this command turns off any previously selected blinking or underline cursor. New characters will show up at the next position on the display with no cursor shown.

Serout 0, N9600, (\$1b, \$46) ‘ Turn off blinking or underline cursor

### **Repeat** ESC R *n c* (1b 52 *n c*)

The Repeat command will show the character represented by *c* above on the display *n* times. *n* can be any number between 0 and the number of characters on the line. *c* can be any character. This command can be useful to blank out a single line on the display by sending 16 or 20 spaces, for example.

Serout 0, N9600, (\$1b, \$52, 10, “X”) ‘ Repeat the character X 10 times

### **Set Up Horizontal Bar Graph** ESC H *n* (1b 48 *n*)

The SLCD includes commands to automatically display horizontal (on one line) bar graphs, either going from left to right, right to left, or both at the same time. A volt meter reading or stereo level display can be shown using these commands, for example.

The bar graph commands use custom characters on the display to create the necessary bar characters. The display allows up to 8 custom characters, the first 5 of which are used for the horizontal bar graph. These characters are loaded each time a Set Up Horizontal Bar Graph command is executed, replacing any previously loaded custom characters. One of these Set Up commands must be executed before a Display Horizontal Bar Graph command is executed.

The value *n*, above, specifies the maximum number of bars in the bar graph. Three bars fit into each character position on the display. The maximum value for *n* is 120 bars. 120 bars uses 40 character positions. Two 24-bar bar graphs fit on a 16 character line. A bar graph 60 bars long will fit onto a single line of a 20 character display. The SLCD will allow you to set a maximum length that is longer than will fit on one line so care should be exercised when setting the length.

Multiple horizontal bar graphs may be put onto the display at once using a Display Horizontal Bar Graph command. However, only one type of vertical or horizontal bar graph may appear on the display at one time.

Serout 0, N9600, (\$1b, \$48, 30)    ‘ Set up horizontal bar graph 30 bars long

**Display Horizontal Bar Graph From Left** ESC >  $n$  (1b 3e  $n$ )

This command will display a horizontal bar graph  $n$  bars long going from left to right on the display. The cursor should be positioned to the left of where the bar graph is desired to be displayed before executing this command. After the bar graph is displayed, the cursor is positioned at the right of the graph. This allows another bar graph to be positioned right next to it.

See the command Set Up Horizontal Bar Graph for more information about displaying bar graphs.

Serout 0, N9600, (\$1b, \$3e, 10)    ‘ Display 10 bar horizontal bar graph

**Display Horizontal Bar Graph From Right** ESC <  $n$  (1b 3c  $n$ )

This command will display a horizontal bar graph  $n$  bars long going from right to left on the display. The cursor should be positioned to the left of where the bar graph is desired to be displayed before executing this command. After the bar graph is displayed, the cursor is positioned at the right of the graph. This allows another bar graph to be positioned right next to it.

See the command Set Up Horizontal Bar Graph for more information about displaying bar graphs.

Serout 0, N9600, (\$1b, \$3c, 10)    ‘ Display 10 bar horizontal bar graph

**Set Up Vertical Bar Graph Going From Top Down** ESC V  $n$  (1b 56  $n$ )

The SLCD includes commands to automatically display vertical (up and down) bar graphs, either going from the top down or the bottom up. A volt meter reading or graphic EQ display can be shown using these commands, for example.

The bar graph commands use custom characters on the display to create the necessary bar characters. The display allows up to 8 custom characters, all of which are used for the vertical bar graph. These characters are loaded each time a Set Up Vertical Bar Graph command is executed, replacing any previously loaded custom characters. One of these Set Up commands must be executed before a Display Vertical Bar Graph command is executed.

The value  $n$ , above, specifies the maximum number of bars in the bar graph. 8 bars fit into each

character position on the display. The maximum value for  $n$  is 32 bars. 32 bars uses 4 character positions (lines).

Multiple vertical bar graphs may be put onto the display at once using the Display Vertical Bar Graph command. However, only one type of vertical or horizontal bar graph may appear on the display at one time.

Serout 0, N9600, (\$1b, \$56, 32)     ‘ Set up vertical bar graph 32 bars high going down

### **Set Up Vertical Bar Graph Going From Bottom Up** ESC ^ $n$ (1b 5e $n$ )

The SLCD includes commands to automatically display vertical (up and down) bar graphs, either going from the top down or the bottom up. A volt meter reading or graphic EQ display can be shown using these commands, for example.

The bar graph commands use custom characters on the display to create the necessary bar characters. The display allows up to 8 custom characters, all of which are used for the vertical bar graph. These characters are loaded each time a Set Up Vertical Bar Graph command is executed, replacing any previously loaded custom characters. One of these Set Up commands must be executed before a Display Vertical Bar Graph command is executed.

The value  $n$ , above, specifies the maximum number of bars in the bar graph. 8 bars fit into each character position on the display. The maximum value for  $n$  is 32 bars. 32 bars uses 4 character positions (lines).

Multiple vertical bar graphs may be put onto the display at once using the Display Vertical Bar Graph command. However, only one type of vertical or horizontal bar graph may appear on the display at one time.

Serout 0, N9600, (\$1b, \$5e, 32)     ‘ Set up vertical bar graph 32 bars high going up

### **Display Vertical Bar Graph** ESC | $n$ (1b 7c $n$ )

This command will display a vertical bar graph  $n$  bars high on the display going either from the top down or the bottom up, whichever was set up last using a Set Up Vertical Bar Graph command. The cursor should be positioned at the top of where the bar graph is desired to be displayed before executing this command. After the bar graph is displayed, the cursor is positioned at the top right of the graph. This allows another bar graph to be positioned right next to it.

See the commands to Set Up Vertical Bar Graphs for more information about displaying bar graphs.

Serout 0, N9600, (\$1b, \$7c, 10)      ‘ Display 10 bar vertical bar graph

### **Display String** ESC S *n* (1b 53 *n*)

The SLCD includes 240 characters of non-volatile EEPROM memory that can be used to store message strings that can then be displayed to the screen using the Display String command. These strings are retained in EEPROM memory, even if the power is turned off. *n* above specifies the starting address of the string in the EEPROM memory, starting with address 0. Each string in EEPROM memory must end with a null character (hexadecimal or decimal 0). Strings may contain any command other than another Display String, Write String or Reset.

A special string begins at address 0 which is used as a splash screen that appears when the display is first turned on. If no splash screen is desired, address 0 must contain the null (hexadecimal or decimal 0) end of string character. The next string would then start at address 1.

Strings are written to the EEPROM memory using the Write String (ESC w) command.

Serout 0, N9600, (\$1b, \$53, 0)      ‘ Display string starting from position 0

### **Write String** ESC w *n s* (1b 77 *n s*)

The SLCD includes 240 characters of non-volatile EEPROM memory that can be used to store message strings that can then be displayed to the screen using the Display String command. These strings are retained in EEPROM memory, even if the power is turned off. *n* above specifies the starting address of the string in the EEPROM memory, starting with address 0. Each string in EEPROM memory must end with a null character (hexadecimal or decimal 0). Strings may contain any command other than another Display String, Write String or Reset.

A special string begins at address 0 which is used as a splash screen that appears when the display is first turned on. If no splash screen is desired, address 0 must contain the null (hexadecimal or decimal 0) end of string character. The next string would then start at address 1.

It can take up to 6 milliseconds to write each character to the internal EEPROM. It is best not to send too many characters too quickly so that the internal buffer is not overrun.

Strings are displayed using the Display String (ESC S) command.

Serout 0, N9600, (\$1b, \$77, 0, “melabs.com”, 0)      ‘ Write splash screen to EEPROM

### **Set Carriage Return Line Feed Modes** ESC m *n* (1b 6d *n*)

The Set Carriage Return Line Feed Modes command can be used to cause a line feed to

automatically be added after a carriage return is received, or a carriage return to be added after a line feed is received. This makes the cursor move to the beginning of the next line if only a carriage return or a line feed is received - receiving both is not required. This command can also be used to specify what happens if the cursor is on the last line of the display and a line feed is received. By default, when a line feed is received with the cursor on the last line of the display, the cursor will move up to the first line. The Set Carriage Return Line Feed Modes command can be used to specify that the lines on the display are scrolled up one position if a line feed is received with the cursor on the last line. The cursor would then stay on the last line.

The table below shows what setting does what combinations of carriage return and line feed handling. If the mode is set to automatically add a line feed after a carriage return, any line feeds sent to the SLCD are ignored. If the mode is set to automatically add a carriage return after a line feed, any carriage returns sent to the SLCD are ignored. Both of these options may be selected at the same time.

The selected mode can be made permanent (saved to EEPROM) using the Write Defaults to EEPROM command. Otherwise the setting only lasts until the power is turned off.

Hexadecimal Code	Function
1b 6d 0 (default)	Carriage return and line feed operate separately. Line feed on last line does not scroll.
1b 6d 1	Line feed automatically follows carriage return. Line feed on last line does not scroll.
1b 6d 2	Carriage return automatically follows line feed. Line feed on last line does not scroll.
1b 6d 3	Line feed automatically follows carriage return and carriage return automatically follows line feed. Line feed on last line does not scroll.
1b 6d 4	Carriage return and line feed operate separately. Line feed scrolls when on last line.
1b 6d 5	Line feed automatically follows carriage return. Line feed scrolls when on last line.
1b 6d 6	Carriage return automatically follows line feed. Line feed scrolls when on last line.
1b 6d 7	Line feed automatically follows carriage return and carriage return automatically follows line feed. Line feed scrolls when on last line.

Serout 0, N9600, (\$1b, \$6d, 1)

‘ Set carriage return to automatically do a line feed

Serout 0, N9600, (\$1b, \$6d, 4)

‘ Set line feed to scroll display when on last line

### **Set Number of Chars** ESC *c n* (1b 63 *n*)

Set Number of Chars is one of 2 commands to set the display format, the other being Set Number of Lines. This command tells the SLCD how many characters are on each line for use with the character positioning and scrolling commands. The default for *n* is 20 which is usually adequate for either a 16 or 20 character per line display. Scrolling can be sped up a bit for a 16 character wide display by setting the number of characters per line to 16. It can also be set to 8, 12, 24 or 40 to match whatever size display is used with the SLCD controller.

The value can be made permanent (saved to EEPROM) using the Write Defaults to EEPROM command. Otherwise the setting only lasts until the power is turned off.

Serout 0, N9600, (\$1b, \$63, 16)      ‘ Set number of chars per line to 16

### **Set Number of Lines** ESC *l n* (1b 6c *n*)

Set Number of Lines is one of 2 commands to set the display format, the other being Set Number of Chars. This command tells the SLCD how many lines the display has for use with the scrolling commands. The default for *n* is 4 which usually works for either a 2 or 4 line display. Scrolling can be sped up a bit for a 2 line display by setting the number of lines to 2. This also ensures the cursor is set to the second line instead of the invisible fourth line after scrolling up.

The value can be made permanent (saved to EEPROM) using the Write Defaults to EEPROM command. Otherwise the setting only lasts until the power is turned off.

Serout 0, N9600, (\$1b, \$6c, 2)      ‘ Set number of lines to 2

### **Set I<sup>2</sup>C Address** ESC *i n* (1b 69 *n*)

The default base I<sup>2</sup>C address for the SLCD is hexadecimal 50. This address may be changed to any other even, non-zero value using the Set I<sup>2</sup>C Address command with *n*, above, specifying the new address. The address change will take place immediately and the display will stop responding to the old address.

The value can be made permanent (saved to EEPROM) using the Write Defaults to EEPROM command. Otherwise the setting only lasts until the power is turned off.

Jumpers JP1 and JP2 control the addition of an offset to the base I<sup>2</sup>C address. If both jumpers are open, there is no offset and the address is simply the base address. If both jumpers are shorted, for example, then an offset of 6 is added to the base address. The table below shows what the offsets would be for the 4 jumper combinations. Note that these same jumpers are also used to select the baud rate.

I <sup>2</sup> C Address	JP1	JP2
Base + 0 (default)	OPEN	OPEN
Base + 2	SHORTED	OPEN
Base + 4	OPEN	SHORTED
Base + 6	SHORTED	SHORTED

Serialout 0, N9600, (\$1b, \$69, \$60)    ‘ Set base I<sup>2</sup>C address to hexadecimal 60

### Write Defaults to EEPROM ESC d w (1b 64 77)

The values for the I<sup>2</sup>C address, number of characters per line, number of lines on the display, cursor settings, carriage return and line feed modes, and backlight brightness can all be changed using commands, as shown above. These changes remain in effect until the power is turned off. The changes can be made permanent (saved to EEPROM) using the Write Defaults to EEPROM command. The Write Defaults to EEPROM command makes all the current values of these settings the default so the next time the power is turned on, the SLCD uses the values set in the commands above. It can take up to 36 milliseconds to complete the write. It is best not to send too many additional characters during this time so that the internal buffer is not overrun.

This command requires 2 bytes as an extra security measure to ensure the defaults are not changed inadvertently.

Serialout 0, N9600, (\$1b, \$64, \$77)    ‘ Write defaults to EEPROM

### Reset Display ESC z (1b 7a)

This command will reset the display, reloading the original custom character set and resetting all the options to their defaults stored in EEPROM. It takes about 50 milliseconds for this process. Any characters sent to the display during this period are ignored.

Serialout 0, N9600, (\$1b, \$7a)    ‘ Reset display  
 Pause 50                            ‘ Wait 50 milliseconds

### Using Other LCDs

The melabs Serial LCD board is available separately, i.e. without a display, and can be connected to other displays that utilize a Hitachi HD44780 or equivalent controller IC. The HD44780 is the most commonly used controller on character LCDs.

The melabs SLCD includes two 16-pin connector areas to connect to the 2 most commonly used LCD interfaces. These are a 16-pin single-row connection and 8 by 2-row connection. The

SLCD is intended to be mounted on the back of the LCD to one of these 16-pin connections. Make sure the LCD you intend to use with the melabs SLCD board matches the pinouts of the SLCD connectors, as shown in the table below. The last 2 pins of the 16 are for the backlight LED. These are not absolutely required and a display without a backlight having a 14-pin connector may also be used.

J4, J5:

Pin	Name	Function
1	Vss	Ground
2	Vdd	+5 volts
3	Vo	Contrast pot
4	RS	Register select
5	R/W	Read / write
6	E	Chip enable
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7
15	A	LED +
16	K	LED -

### Assembly Instructions

1. Break 4 of the pins from the 20-pin single-row header strip. If using the I<sup>2</sup>C interface, solder these 4 pins to the top of the SLCD board (soldering from the back) into the area labeled J2. If using the RS232-style interface, break 1 more pin off the 4 and solder the remaining 3 pins to the top of the SLCD board (soldering from the back) into the area labeled J1.
2. If the LCD has a single-row connector on it, solder the remaining 16 pins to the back of the SLCD board (soldering from the top) into the area labeled J4. If the LCD has a 2-row connector on it, break the remaining 16-pin strip into two 8-pin strips. Solder each of these strips to the back of the SLCD board (soldering from the top) into the area labeled J5.
3. Using double-sided foam, stick the SLCD board to the back of the LCD while aligning the pins previously soldered to the SLCD with the mating holes in the LCD.

4. Solder the 16 pins into the LCD.

5. Power up the board and set the contrast using POT1 as outlined in a previous section.

If you have questions regarding the melabs Serial LCD, our contact information is provided below.

microEngineering Labs, Inc.  
Box 60039  
Colorado Springs CO 80960-0039  
USA  
(719) 520-5323  
(719) 520-1867 fax  
<http://www.melabs.com>  
email:[support@melabs.com](mailto:support@melabs.com)