

Modern Device LCD117/ PH Anderson's Serial LCD Driver Commands

reference adopted from <http://phanderson.com/lcd106/lcd107.html>

The currently shipping chips seem to be LCD117 and LCD118. LCD 117 is a 9600 baud chip that is recommended for Arduino, Basic Stamp and most microcontrollers, which are capable of 9600 bps serial transmission.

Some commands require a following pause.

	Command / Comments	Suggested Pause	Example
	Send Text		<code>Serial.print("Hello World!");</code>
?a	Home Cursor		<code>Serial.print("?a");</code>
?b	Destructive Backspace		<code>Serial.print("?b"); // backspace</code>
?c#	Set Cursor Style: 0= none 2= blinking 3=underline		<code>("?c0"); // turn cursor off</code>
?f	Clear Screen		<code>Serial.print("?f");</code>
?g	Beep (requires speaker on pin 6)		<code>Serial.print("?g");</code>
?h	Backup Cursor (Non-destructive backspace)		<code>Serial.print("?h");</code>
?i	Forward cursor		<code>Serial.print("?i");</code>
?j	Up cursor		<code>Serial.print("?j");</code>
?k	Down cursor		<code>Serial.print("?k");</code>
?l	Clear cursor line		<code>Serial.print("?k");</code>
?m	Carriage Return		<code>Serial.print("?m");</code>
?n	CRLF, carriage return & line feed, cursor at start of next line, line cleared		<code>Serial.print("?n");</code>
?s#	Set tabs at # spaces	100 ms	<code>Serial.print("?s7"); // set tab to 7 spaces</code>
?t	Tab, advance one tab position		<code>Serial.print("?t");</code>
?x##	Position cursor on x column, (two characters are required), first column is column 0		<code>Serial.print("?x09"); // cursor to column 10</code>
?y#	Position cursor at y row, first row is row 0, one digit only (no leading zero)		<code>Serial.print("?y3"); // cursor to row 4</code>
??	Display a "?"		<code>Serial.print("??");</code>
?!	Send direct command to LCD		<code>Serial.print("?!01");</code>
?B	Backlight Intensity – sets PWM value, two hex digits req. (00 to FF)	100 ms	<code>Serial.print("?BFF"); // backlight on full brightness</code>
?D#	Define Character "?D#1A001A001A00" D# = character # 0-7 then 8 two character hex digits representing (5 bit values top to bottom)	100 ms	<code>Serial.print("?D31F001F001F001F00"); // custom character 3 // every other line black</code>
?#	Print a custom character numerals 0-7 are valid	5 ms	<code>Serial.print("?3"); // print custom character 3</code>
?H	High output on auxiliary digital pins: valid numbers are 4,5,6		<code>Serial.print("?H4"); // aux pin 4 HIGH</code>
?L	Low output on auxiliary digital pins: valid numbers are 4,5,6		<code>Serial.print("?L4"); // aux pin 4 LOW</code>
?G	Configure for LCD geometry. Supported formats: 2X16, 2X20, 2X24, 2X40, 4X16 and 4X20.		<code>Serial.print("?G216"); // configure driver for 2 x 16 LCD</code>
	Ganging up commands is OK, except for commands requiring a following pause		<code>Serial.print("?x01?y1?fHello World"); // cursor to beginning of line 1 // clear screen, print "Hello World"</code>
	Enhanced Commands		
?>#	Enter BIG Number Mode (numbers only!) ">#" X # = 3 or 4 , # represents number of characters displayed. (20x4 LCD's only)	100 ms	<code>Serial.print(">4"); // enter big number mode, 4 character option.</code>

?<	Exit BIG number mode "?<"		<code>Serial.print("?<");</code>
?C#	Define custom boot screen line # = 0 - 3	100 ms	<code>Serial.print("?C0abcdefghijklmnopqrst");</code>
?S#	"?S0" - display no screen on boot. "?S1" - display the configuration setting on boot "?S2" - display the user custom text screen on boot.		<code>Serial.print("?S2"); // custom boot screen</code>
?*	display boot screen at any time		<code>Serial.print("?*"); // show boot screen</code>