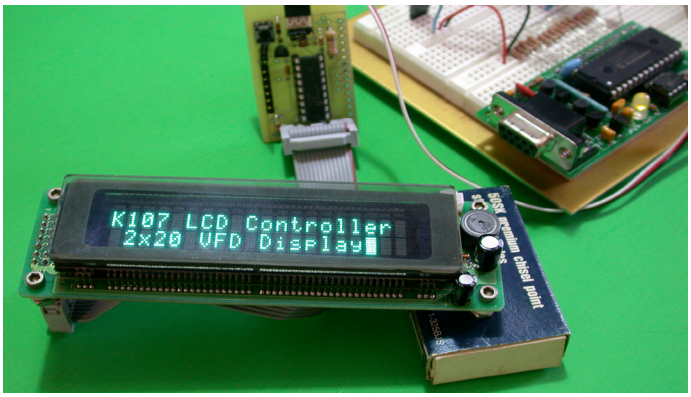
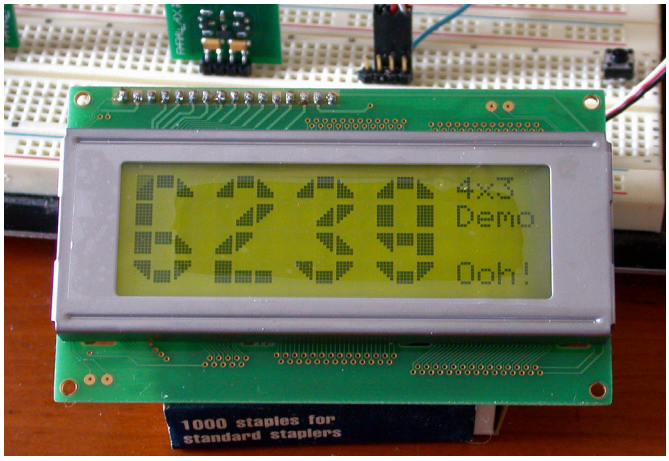
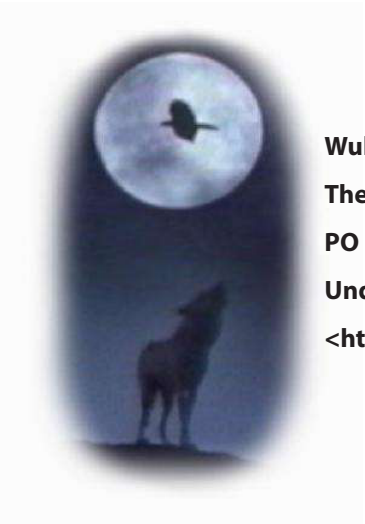
LCD117 Command Set

Brian Riley (N1BQ)

Revised: 5/12/2013







Wulfden at Hawk's Mountain

The Shoppe at Wulfden

PO Box 188

Underhill Center, VT 05490

<http:/www.wulfden.org>

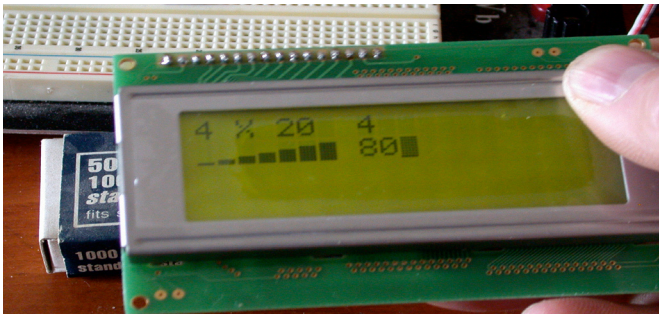
Current for Peter Anderson's LCD117 Rev D Firmware

Controller Chip Differences

There are various PH Anderson chips the LCD 106, 107, 108, 117 and 118in circulaation. The basic command set is the same for all three controller chips. The #107 adds software control of the backlight and 4 general purpose outputs and one beep output to the #106. The newer #117 adds a large character set display, in two different widths and a custom boot text display to the #107. The #117 only has two general outputs. The #108 and #118 chips have identical command sets to the #107 and #117.

Note In the LCD Command list that follows, commands shared among all the controller chips are simply listed, those specific to only one or two of the chips will have the chip number in parentheses next to the command title.

Note When started (unless configured otherwise), the Anderson controller will briefly display its configuration settings for its geometry, cursor style and backlight intensity. This takes a finite period of time. If your application displays important information at startup, it should allow at least a three second pause for the display to initialize before attempting to write to the display.



Caution As previously stated, the Anderson chip has a 64 byte buffer. Generally speaking, typical LCD screen displays will rarely tax this limit, but display updates with cursor positioning strings might over run the buffer. But this could happen if you were to, say, try to do a time display to tenths of seconds in real time. You need to consider this when programming the display outputs.

?a	set cursor to home position
?b	destructive backspace
?f	clear screen and leave cursor at home position
?g	beep (50 ms of 500 hz from J5-2)
?h	non-destructive backspace
?i	forward cursor
?j	down cursor
?k	up cursor
?l	clear current line, cursor at start of line
?m	carriage return, cursor to start of current line
?n	CRLF, cursor at start of next line, line cleared
?t	tab, advance one tab position
??	display a "?"

Note In late 2008 Peter Anderson released verison 117C, in 9600 and 19200 baud. This fimware is significantly faster than earlier versions. This somewhat lessens the concerns of the prior paragraph, but does not eliminate them altogether.

Note Any software command to the Anderson controller chip that causes a write to the EEPROM must be followed by a pause of approximately 100 milliseconds (e.g. in PBASIC **PAUSE 100**) This would include commands to set Tabs, Display geometry, startup screen, etc.

Basic LCD Commands

A partial list of various commands is in the chart to above. Note that all commands are prefaced with

a **"?"** and the commands are case sensitive. The cursor may be set to any position using the x and y commands;

```
?y0
?x15
```

Note that the line number, follows the '**y**' command and the column number, consisting of two digits, follows the '**x**' command. Thus, outputting the string;

```
"?y1?x10Hello?n"
```

positions the cursor at column 10 of line 1 and then prints "Hello", followed by a newline which also clears the next line.

Note that the line and column numbers begins with 0. Thus, for a 4 X 20 LCD, valid lines are 0 - 3 and valid columns are 00 - 19. Any of eight user defined characters may be displayed using the digits 0 - 7. Thus;

```
"?5?2"
```

causes user defined character 5 followed by user character 2 to be displayed on the LCD. Defining the user characters is discussed below.

Configuration Commands

A number of parameters are stored in the processors EEPROM. This includes the geometry of the LCD, the type of cursor, the number of spaces in a tab and the eight user defined characters. The settings of all of these are displayed briefly on the LCD when the processor boots.

The default is a 20 X 4 LCD, a tab size of 4, a full blinking cursor (3), backlight on ($B0) and the eight special characters consisting of a single horizontal line, two horizontal lines, etc.

All of these parameters may be modified. Note that when the modifications are made, the new values are written to the processor's EEPROM. Thus, the user defined characters and the geometry of the LCD need only be modified one time.

Set LCD Geometry

Setting the geometry defines the configuration of the interfacing LCD.

```
"?Gyxx"
```

For example;

```
"?G216"
```

indicates the LCD is a 2X16 configuration.

Note that appropriate configurations include
2X16 2X20 2X24,
2X40 4X16 4X20.
The configuration is important for the processor to properly position the cursor after each text character is displayed and in executing such commands as backspace, up cursor, down cursor, new line, etc.

Set Tab

The tab size may be set;

```
"?s5"
```

In this case, the tab size is set to five. Valid values of the tab are 1 - 8. When this command is received by the processor, the new tab size is written to EEPROM and this value is used thereafter when executing the **?t** command.

When executing the **"?t"** command, the cursor is advanced, and any characters in its path are overwritten with a space. For example, if the cursor is in column 3 and the tab size is 5, the cursor will advance to column 5. Anything in columns 3 and 4 will be replaced with spaces.

Backlight Intensity (107, 108, 117, 118)

The intensity of an LED backlight on the associated LCD may be adjusted;

```
"?B80"
```

Note that the digits following the **"?B"** are two digit hexadecimal, ranging from **00** (back light off) to **FF** (back light full on). This command causes an output to change duty cycle over the range of 0 to 255/256. The output may be used to switch a power transistor which then controls the average backlight current.

Note with most displays you will not see any noticeable increase in intensity beyond $80.

Set Cursor Style

The style of the cursor may be set using the ?c command.

```
"?c3"
```

where the number is in the range of 0-3. A 0 configures as no cursor, a 2 as a non blinking cursor and a 3 as a blinking cursor. As with the set geometry and set tab, the style of the cursor is saved to EEPROM.

User Defined Characters

User defined characters may be defined using the **?D** command;

```
"?D300000000001f1f1f"
```

The number after the **"?D"** is the number associated with the user defined character, in this case user defined character 3. This is then followed by the eight data bytes expressed in two digit hexadecimal. Note that the hexadecimal letters must be lower case. In this example, lines 0, 1, 2, 3 and 4 consist of no pixels and lines 5, 6 and 7 consist of all five pixels. Thus, when user defined character 3 is displayed using the command **"?3"**, a character consisting of the lower three lines will be displayed.

Each user defined character is saved in EEPROM. The design provides a 64 byte serial receive buffer. However, be careful. If one is defining all eight user defined characters, this involves sending 19 * 8 or 152 characters. Writing each of the eight bytes to EEPROM requires 15 ms or more and thus, one can easily over run the buffer. Rather, provide a one second delay after defining each character.

Direct Control of the LCD

Commands may be directly passed to the LCD using the ?! command;

```
"?!01"
```

In this example the command **01** is sent directly to the LCD which clears the LCD.

A word of caution. With all other commands the program keeps track of the current cursor position. This is not done with commands sent directly to the LCD using the **"?!"** command. Thus, if the user configures the LCD such that the cursor is located at some point, subsequent line feeds and similar will not work correctly, since the controller will not know the current cursor location.

Clearly, if the **"?!"** command is used to place the LCD in a mode such that the cursor is decremented with each character or the display itself is scrolled, the subsequent operation of the LCD which assumes an incrementing cursor and a fixed display will give unpredictable results.

General Purpose Outputs (107, 108, 117, 118)

The LCD #107/117 provides four/two general purpose TTL outputs. Any of these outputs, (4, 5, 6 & 7 for the 107/117 chip and 4 &5 for the 108/118 chip), may be brought high or low using the **"?H#"** and **"?L#"** commands.

"?H4?L7"

The above brings output 4 high and output 7 low. Note that outputs 0, 1, 2 and 3 are not valid.
On power up, all outputs are at a high impedance (configured as inputs). As each output is addressed, it is taken out of the high impedance state. The initial high impedance state permits the user to use either pull up or pull down resistors to avoid "bounce" when the processor is powered up.The current (source or sink) by any output should be limited to 15 mA.

*** Note *** - While OUT6 and OUT7 have outlets on the Rev 3 board. They will not be available if the board is wired for LCD 108/118.

Big Number Mode (117, 118)

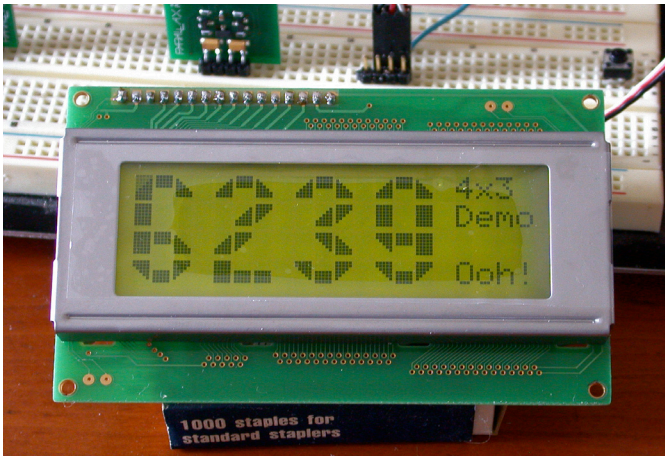
The LCD #117 design provides both three and four block wide characters. This command will be ignored unless the display geometry command is set to a 4 line mode (4x16 or 4x20) The big number mode is entered;

"?>3" or **"?>4"**

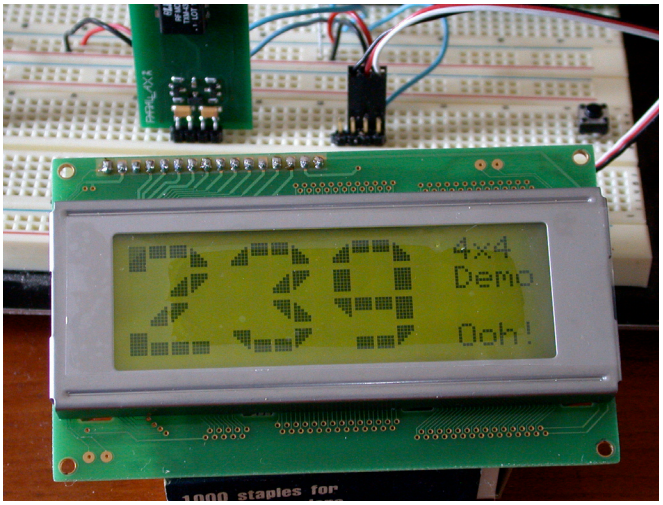
On receipt of either of the **"?>3"** or **"?>4"** commands, the processor will download to the LCD the special character set required to display the big numbers. These are probably different from any user defined characters which the user may have defined. Thus, don't attempt to display any previously defined special characters after having invoked the "big number" format. You will need to reload these characters if they are needed.

Caution The 64 byte buffer limit needs to be considered here. In 'BigNum' mode the display is sending 16 screen blocks (4x3 plus 4 blanks in **"?>3"** mode) or 20 screen blocks (4x4 plus 4 blanks in **"?>4"** mode) to the actual display. So when considering the timing think 16 or 20 characters NOT just a single character.

Note The improved code of 117C made some improvements here. I had a loop displaying Big-Nums tha required a dleay of 200 milliseconds to keep the display stright. I was able to cut the delay t 30 milliseconds!



In the **"?>3"** mode, as each numeric digit is received, it will erase anything appearing in the three column wide by four high space and display the number in the big number format. In addition, it will clear the column to the right of the character. The cursor will then be located at the top row for display of the next digit. Note that for the three block wide mode, this is four columns to the right of the upper left of the previous character. Thus, each digit uses four columns, three for the display of the digit and one for the spacing between digits. Display of the minus sign, colon and decimal point use two columns; one for the display of the symbol plus one space.



The **"?>4"** mode is similar, except that each digit is four wide plus one space.

There is no intelligent "wrapping" provision for the "big number" mode. Thus, if there are only two columns remaining before the end of the LCD and you attempt to display a "7", part of the "7" will appear on the right side of the LCD, and the rest on the beginning of the display.

The big number mode may be exited at any time:

"?<"

This may be used to display ordinary text along with the "big number" characters. For example, in one of the photos, the message "4X4 Demo" appears. This might be done;

"?y0?x164X4**"** ' row 0, beginning at col 16
"?y1?x16 Demo**"** ' row 1, beginning at col 16

Note that the **"?<"** command simply exits the "big number" mode. It does not download the user defined characters that might be used for graphics. If one later wishes to use the user defined characters;

"?R" ' restore the user defined characters.

Again, note that the characters associated with the display of the "big number" characters are not compatible with the user defined characters one might use for plotting and thus, "big numbers" and graphics type applications using specially defined characters cannot coexist on the display.

Custom Boot Text Screen(117, 118)

The customized text screen may be an advertisement which is displayed on boot or a template which may be displayed at any time. It is set using the **?C** command, followed by the row, followed by the twenty characters which are to appear on the row. For example, the following sequence defines the custom screen shown in the photo.

"?C0abcdefghijklmnopqrst"
"?C1ABCDEFGHIJKLMNOPQRST"
"?C201234567890123456789"
"?C398765432109876543210"

Note that this custom screen may be displayed at any time;

"?*"

Boot Screen Selection (117, 118)

The screen which is displayed on boot may be specified using the **"?S#"** command followed by a number, 0, 1 or 2 will display, on boot

"?S0" ' blank screen
"?S1" ' the configuration settings
"?S2" ' the user custom text screen