

Modern Device LCD117 Serial LCD Board Kit Instructions

instructions version 0.4 2/6/09

Preface

It is possible to drive an LCD with an Arduino (or other microcontroller) without using additional interface circuitry. This usually requires 7 connection wires to the LCD and a few components. More importantly, the programming is a little tricky with a number of numerical parameters of which the programmer must be aware.

The Peter Anderson LCD 117 chip reduces the necessary connections to 3 leads; +5volts, Ground, plus one serial data wire. It also eliminates the more tedious programming chores and frees up the microcontroller (and its pins) for other tasks. This provides an easy, trouble-free interface to any LCD with a HD44780 compatible driver chip. All programming requires some trial and error, but the LCD board and kit will get you going a lot faster than a "raw" LCD.

The industry has standardized LCD character displays on the Hitachi HD44780 LCD controller chip. Avoid anything that is not "HD44780 compatible". The LCD 117 kit works with this standard. It's fairly safe to buy almost anything on the market however. If the display has driver chips, the odds are they are compatible, and will work well with the LCD117 board.

Before you begin

The LCD117 board has been designed to accommodate the physical form factor of the vast majority (but not all) LCD character displays. It can even be made to work with the more rare pinouts, by adding some wires to pins and doing a little pin remapping.

Parts List

Semiconductors

- 1 phanderson.com LCD 107 / 117 programmed PIC chip
- 1 IRL 530 N channel MOSFET or TIP120 NPN transistor (TIP 120 provided with kit)
- 1 400x diode for reverse polarity protection

Capacitors

- 1 .1 ufd (104) 25 volt monolithic ceramic capacitor

Resistors

- 1 10 k ohm 1/4 watt resistor (brown-blk-orange)
- 1 330 ohm 1/4 watt resistor (orange-orange-brown)
- 1 5 k ohm trimpot for LCD contrast control

RBL resistor choices

- 1 27 ohm 1/4 watt resistor (red-violet-black)
- 1 15 ohm 1/4 watt resistor (brown-green-black)
- 1 spare resistor lead for shunt if needed varies by display, check the LCD display datasheet

Hardware

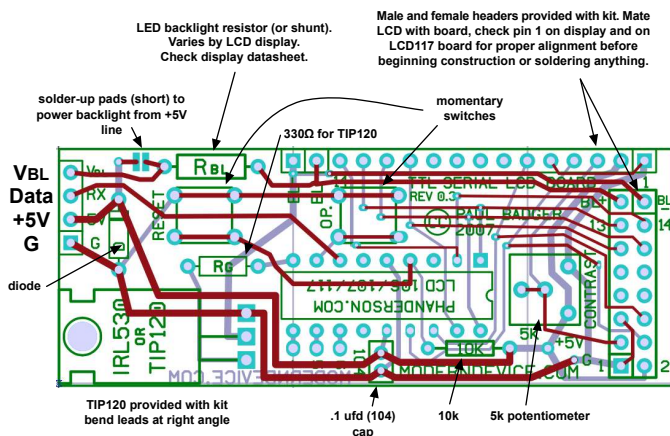
- 2 momentary switches
- 1 16 position male header pins
- 1 16 position female header pins
- 1 2 x 8 position female header pins
- 1 4 position right-angle male header pins

Check the datasheet of the display that you have, or if lacking a datasheet and pinout, just locate pin 1, pin 14 and the backlight pins. Sometimes the backlight pins are marked "A" & "K" for anode (BL+) and cathode (BL-). Ascertain that the backlight pins are pins 15 and 16 on the LCD and that their polarity matches the board layout. While the majority of LCD's have pin layouts (pinouts for short) that match the board, the position of the backlight pins can vary on some rare LCD's, so make sure they are in the correct position in case some ad-lib pin swapping is in order to get the pins lined up correctly.

Cut the male and female LCD header pins to size, if needed and mate the pins with the LCD and pcb **before soldering anything**. Check to make sure that pin 1 on the pcb and lines up with pin 1 of the LCD before proceeding. Make sure you understand which direction the pins should be facing. Check the backlight pins for correct polarity.

For some LCD's it will be necessary to build the LCD117 with the parts facing the LCD circuit board, instead of the more common method shown in the photos. Figuring this out in advance will definitely make your life easier.

Once you have arrived at a game plan for mating the board to LCD, and doublecheck the pinouts, put the headers aside and begin construction with the small parts first.



Construction

As with most printed circuit boards, assemble the smallest parts first. This includes the resistors, capacitor and diode and the solder bridge to the left of R_BL. You may wish to leave R_BL (backlight resistor) off to test different values with your particular display backlight.

Next solder in the momentary switches and TIP120. The TIP120 transistor should have the leads bent at a right angle, just where the leads change thickness. Solder only one lead first to make sure the transistor is seated, flip the board and check the seating, then reheat the joint and apply pressure if necessary. Finish by soldering in the other leads. Solder in the potentiometer.

The PIC chip for the kit is provided with a socket. It makes replacing the chip easy, in the rare circumstance that the chip should get damaged, by an overvoltage say. Feel free to use it or not as desired.

If only the two diagonal corner pins of the socket/chip are soldered down first, it may be inspected to make sure that the socket is well seated against the PCB. If it is not well seated, simply reheat the solder joints while applying pressure with your index finger on the socket/chip on the other side or the board. It also often helps to bend over two or more diagonal corner pins, to hold the chip/socket in while you solder.

Solder in the four-pin right angle header. Because most people will not use the V_BL pin, feel free to clip it off and just make it a three-pin header if you wish.

Finally, solder in your male and female headers according to your established plan. Solder the header onto the LCD and you're ready to go.

Testing

Power up the board and adjust the potentiometer clockwise until you see a flashing cursor. If you don't see the cursor, check:

- solder shunt next to R_BL not soldered
- no R_BL resistor
- power leads reversed, or no 5 volts present
- reverse polarity protection diode in backwards
- chip in backwards
- short circuit somewhere
- not all pins in socket correctly
- check solder joints on all pins, reheat if necessary

Push the RESET switch and you should see a startup screen displaying the custom characters and the display geometry for which the LCD117 is currently configured. The backlight will come on for a second or two, then go off.

The chip defaults to 4 x 20 so you won't see a correct startup screen on a 2 x 16 LCD until you send it the correct geometry parameters in software, which the chip will then save.

The "OP." switch activates an "operation test" that will present the display's character set, scrolling across the screen. It is activated in conjunction with the reset button. Press and hold reset, press OP while holding reset, release reset, release OP. The backlight will stay on during Op Test.

Both the backlight and the screen geometry are under software control, so you will have to send the string "?Bff" to the board to turn the backlight on to full brightness permanently. This gets stored in EEPROM so you only need to do it once. See the documents at moderndevice.com and phanderson.com for programming with the display.

The board will require different R_BL resistors for different LCD displays. Consult the datasheet on your particular display, if you have one, for the maximum current that is recommended for the backlight. Note that all LCD's do not have backlights and some have high-voltage electroluminescent backlights that cannot be driven by this board.

Calculating R_BL

Here's how to calculate the R_BL resistor. Check the datasheet for the backlight supply voltage and current. In our example case this is 3.3V and 60 mA. With a 5 volt supply, the difference between 5V and 3.3V is 1.7 V. The 1.7 volts will appear across the R_BL, the backlight current-limiting resistor.

Because R_BL is in series (connected end to end) with the backlight LED's, the current through the resistor is the same as the current through the LED's. (The current through every element in a series current is the same.)

Now we have two out of the three variables in Ohm's Law and we can solve for the resistance.

Ohm's Law states: $V=IR$ where V is volts, I is current in amps, R is resistance in ohms.

so the current-limiting resistor is 1.7 volts / .06 amps = 28.3 ohms. Any resistor close to 28 ohms (e.g. 27 ohms) will work fine, as will 33 ohms, 30 ohms, 25 ohms, or 22 ohms.

You can also use a multimeter, set to the 200 mA current range, to check your LCD's backlight current draw. Be careful with this though, because there is a fuse inside the meter you can blow, if you touch the probes to a voltage source, such as +5V and ground, with no current limiting resistor in the circuit. The trick is to set up the circuit, then complete the circuit with the meter probes. (In series with

the circuit.) This could be accomplished by leaving opening the solder shunt for the backlight, for example, and completing the circuit with your meter probes. Just start with a larger resistor, such as 47 ohms, insert it into the RBL holes without soldering, and read the current on your meter, work your way down with smaller resistors until you get to the datasheet current.

Some LCD's have resistors built into them, and a shunt can be used for the RBL resistor, others require a resistor. If you are using an LCD with no datasheet, you'll have to judge by the backlight brightness or current. You are going to be safe most of the time with 20 mA through the backlight, but the display may not look very bright. Proceed to smaller resistors until you just achieve what looks like a reasonable backlight brightness.

Pin Connctions on LCD header

1	GND
2	5V
3	CNTR
4	RS
5	R/W
6	E
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7
15	BL+ (A)
16	BL- (K)

Modern Device / PH Anderson's Serial LCD Driver Commands

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, Basic Atom and any other microcontrollers that are capable of 9600 bps serial transmission.

Commands are case sensitive! Some commands require a following pause. Sending certain commands too quickly will overrun the serial buffer on the LCD117 (PIC) chip and result in the chip resetting to the startup screen. If this happens, try inserting some short delays into your code. The chip recently has a new revision and some delays can be much shorter than listed here.

Command	Suggested Delay	Arduino Example
send text		<code>Serial.print("Hello World!");</code>
?a		<code>Serial.print("?a");</code>
?b		<code>Serial.print("?b"); // backspace</code>
?c#		<code>("?c0"); // turn cursor off"</code>
?f		<code>Serial.print("?f");</code>
?g		<code>Serial.print("?g");</code>
?h		<code>Serial.print("?h");</code>
?i		<code>Serial.print("?i");</code>
?j		<code>Serial.print("?j");</code>
?k		<code>Serial.print("?k");</code>
?l		<code>Serial.print("?l");</code>
?m		<code>Serial.print("?m");</code>
?n		<code>Serial.print("?n");</code>
?s#	100ms	<code>Serial.print("?s7"); // set tab to 7 spaces</code>
?t		<code>Serial.print("?t");</code>
?x##		<code>Serial.print("?x09"); // cursor to column 10</code>
?y#		<code>Serial.print("?y3"); // cursor to row 4</code>
??		<code>Serial.print("??");</code>
?!		<code>Serial.print("!01");</code>
?B##	100ms	<code>Serial.print("?BFF"); // backlight on full brightness</code>
?D#	100ms	<code>Serial.print("?D31F001F001F001F00"); //custom character 3, every other line black</code>
?#	5ms	<code>Serial.print("?3"); // print custom character 3</code>
?H#		<code>Serial.print("?H4"); // aux pin 4 HIGH</code>
?L#		<code>Serial.print("?L4"); // aux pin 4 LOW</code>
?G		<code>Serial.print("?G216"); // configure driver for 2 x 16 LCD</code>
Enhanced Commands		<code>Serial.print("?x01?y1?fHello World"); // cursor to beginning of line 1, clear screen, print "Hello World"</code>
?>#	100ms	<code>Serial.print(">4"); // enter big number mode,</code>

?G Configure for LCD geometry.
Supported formats: 2X16, 2X20, 2X24, 2X40, 4X16 and 4X20

Ganging up commands is OK,
except for commands requiring a following pause

Enhanced Commands

?># Enter BIG Number Mode (numerals only!)
represents number of characters displayed. (20x4 LCD's only)

?< "Exit BIG number mode

?C# Define custom boot screen line # = 0 - 3

?S# Display the configuration setting on boot
0 = display no screen on boot
2 = display user-defined text screen on boot

```
Serial.print("?G216");  
// configure driver for 2 x 16 LCD 
```

```
Serial.print("?x01?y1?fHello World");  
// cursor to beginning of line 1, clear screen,   
// print "Hello World" 
```

```
100ms  Serial.print("?>4"); // enter big number mode   
// 4 character option 
```

```
Serial.print("?<");  
100ms  Serial.print("?C0abcdefghijklmnopqrst"  
Serial.print("?S2); // custom boot scrn" 
```