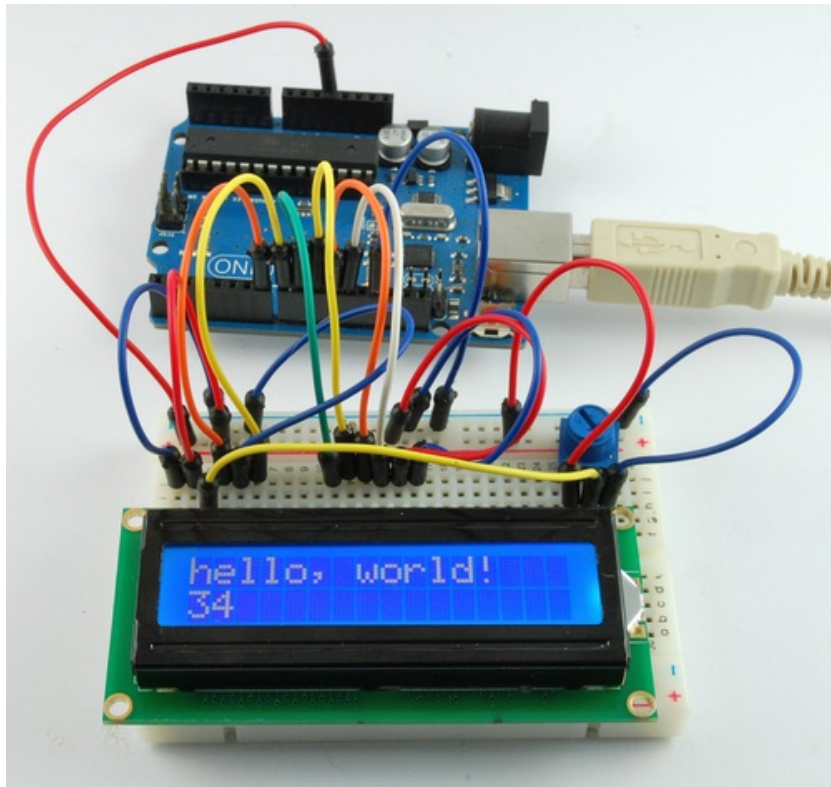


Arduino Lesson 11. LCD Displays - Part 1

Created by Simon Monk



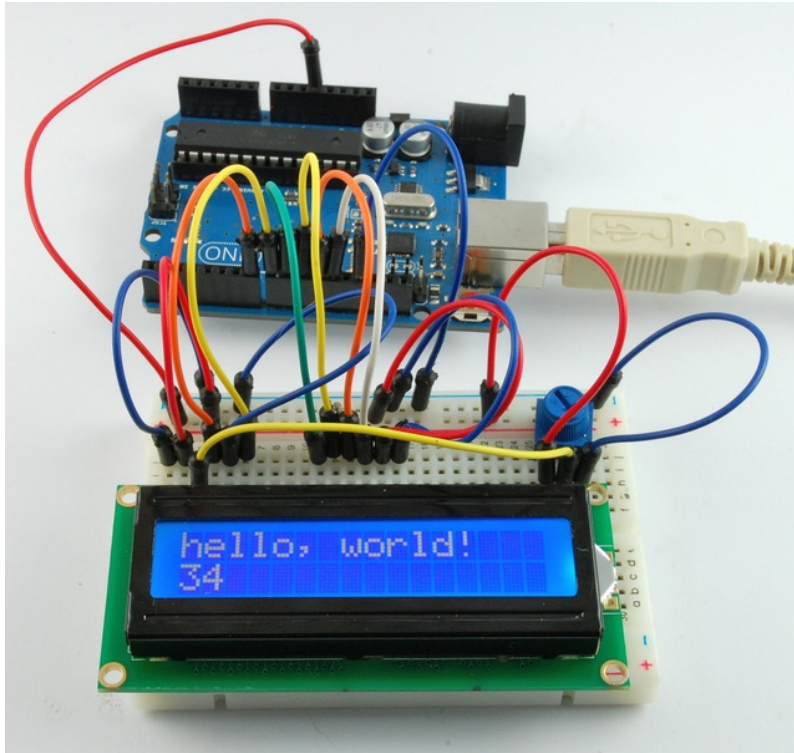
Last updated on 2013-06-11 08:30:45 PM EDT

Guide Contents

Guide Contents	2
Overview	3
Parts	4
Part	4
Qty	4
Breadboard Layout	6
Soldering Pins to the Display	7
Arduino Code	8
Other Things to Do	10

Overview

In this lesson, you will learn how to wire up and use an alphanumeric LCD display.

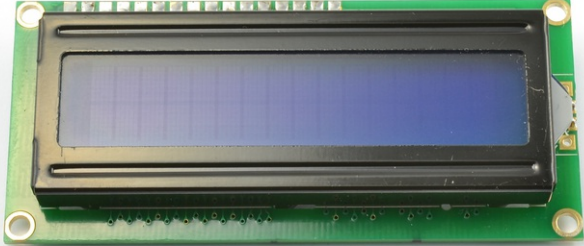
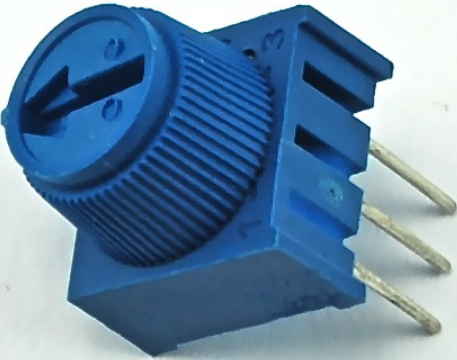
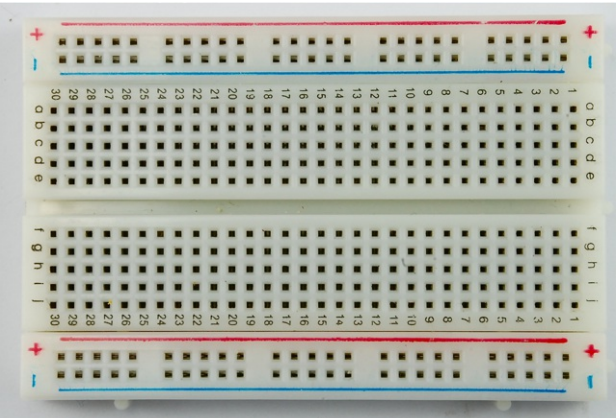


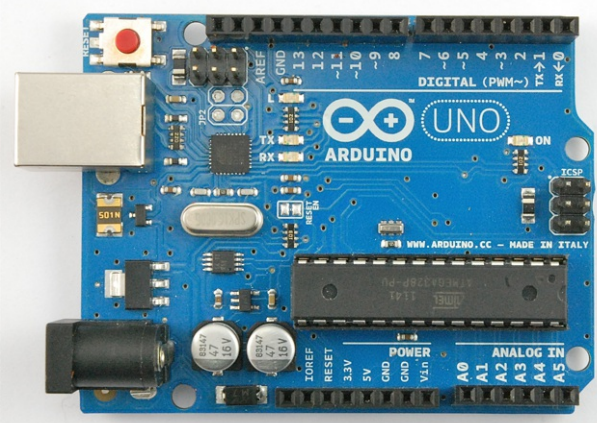
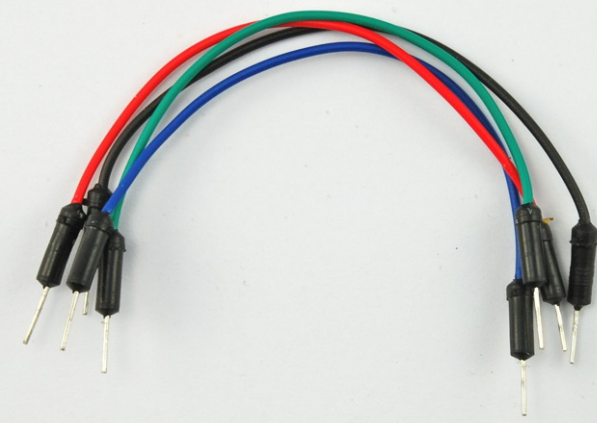
The display has an LED backlight and can display two rows with up to 16 characters on each row. You can see the rectangles for each character on the display and the pixels that make up each character. The display is just white on blue and is intended for showing text.

In this lesson, we will run the Arduino example program for the LCD library, but in the next lesson, we will get our display to show the temperature and light level, using sensors.

Parts

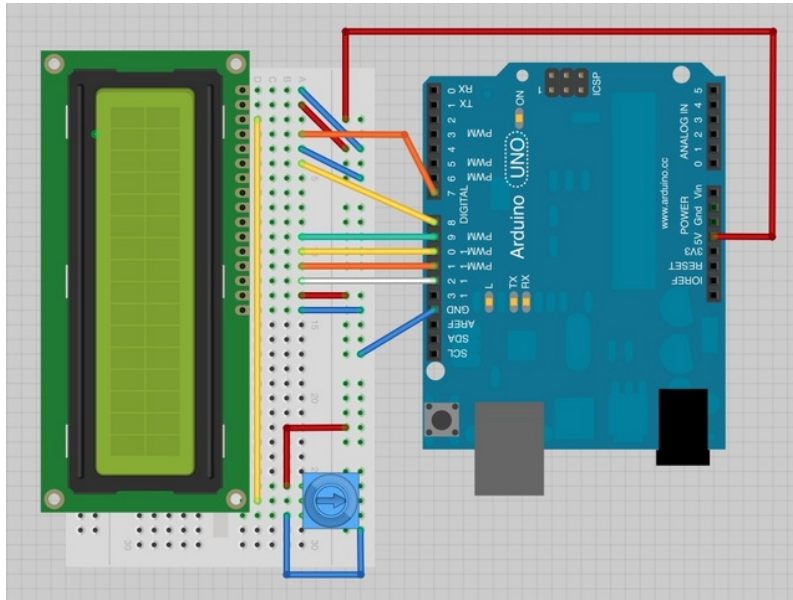
To build the project described in this lesson, you will need the following parts.

	Part	Qty
	LCD Display (16x2 characters)	1
	10 kΩ variable resistor (pot)	1
	Half-size Breadboard	1

	<p>Arduino Uno R3</p>	<p>1</p>
	<p>Jumper wire pack</p>	<p>1</p>

Breadboard Layout

The LCD display needs six Arduino pins, all set to be digital outputs. It also needs 5V and GND connections.

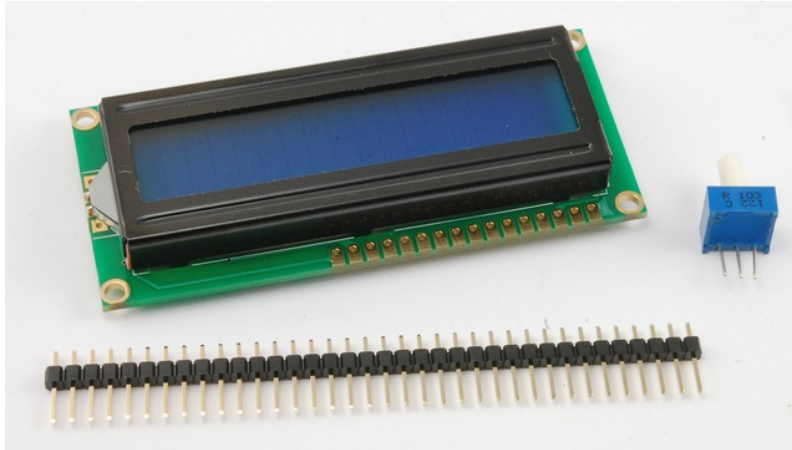


There are quite a few connections to be made. Lining up the display with the top of the breadboard helps to identify its pins without too much counting, especially if the breadboard has its rows numbered with row 1 as the top row of the board. Do not forget, the long yellow lead that links the slider of the pot to pin 3 of the display. The 'pot' is used to control the contrast of the display.

You may find that your display is supplied without header pins attached to it. If so, follow the instructions in the next section.

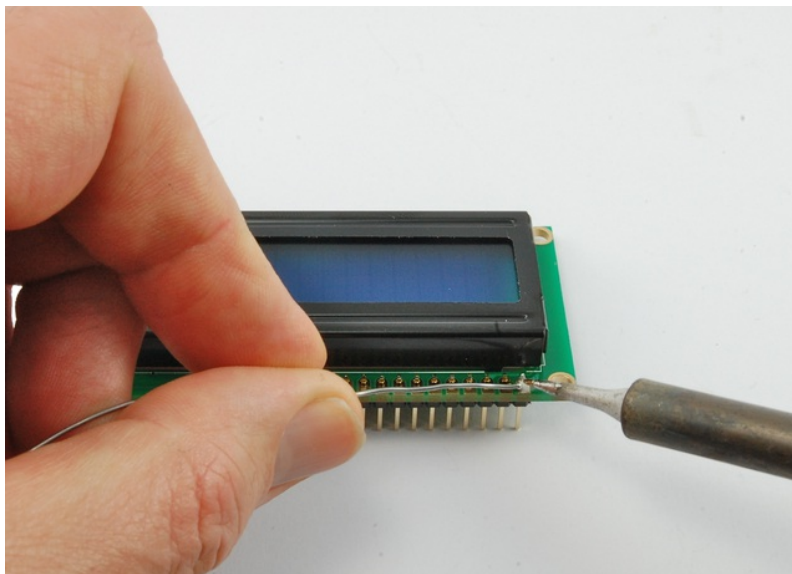
Soldering Pins to the Display

The display needs 16 pins, so if your header strip is longer than that then break it off to the right length.



Then put the length of 16 header pins into the solder tabs on the display and starting at one end, solder each of the pins in place. It can be easier to put the long end of the pins into the breadboard so that the header pins are held straight.

If you do not do this, then solder one pin in first and then get the pins in straight, melting the solder on the pin before making any adjustment.



Arduino Code

The Arduino IDE includes an example of using the LCD library which we will use. You can find this on the File menu under Examples → Liquid Crystal → HelloWorld.

This example uses different pins to the ones we use, so find the line of code below:

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

and change it to be:

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Upload the code to your Arduino board and you should see the message 'hello, world' displayed, followed by a number that counts up from zero.

The first thing of note in the sketch is the line:

```
#include <LiquidCrystal.h>
```

This tells Arduino that we wish to use the Liquid Crystal library.

Next we have the line that we had to modify. This defines which pins of the Arduino are to be connected to which pins of the display.

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

The arguments to this are as follows:

Display Pin Name	Display Pin Number	Arduino Pin (in this example)
RS	4	7
E	6	8
D4	11	9
D5	12	10
D6	13	11

D7	14	12
----	----	----

After uploading this code, make sure the backlight is lit up, and adjust the potentiometer all the way around until you see the text message

In the 'setup' function, we have two commands:

```
lcd.begin(16, 2);  
lcd.print("hello, world!");
```

The first tells the Liquid Crystal library how many columns and rows the display has. The second line displays the message that we see on the first line of the screen.

In the 'loop' function, we also have two commands:

```
lcd.setCursor(0, 1);  
lcd.print(millis()/1000);
```

The first sets the cursor position (where the next text will appear) to column 0 & row 1. Both column and row numbers start at 0 rather than 1.

The second line displays the number of milliseconds since the Arduino was reset.

Other Things to Do

Try pressing the Reset button on the Arduino, notice that the count goes back to 0.

Try moving the position where the count is displayed to near the middle of the second row of the display.

About the Author

Simon Monk is author of a number of books relating to Open Source Hardware. The following books written by Simon are available from Adafruit: [Programming Arduino](http://adafru.it/1019) (<http://adafru.it/1019>), [30 Arduino Projects for the Evil Genius](http://adafru.it/868) (<http://adafru.it/868>) and [Programming the Raspberry Pi](http://adafru.it/aM5) (<http://adafru.it/aM5>).