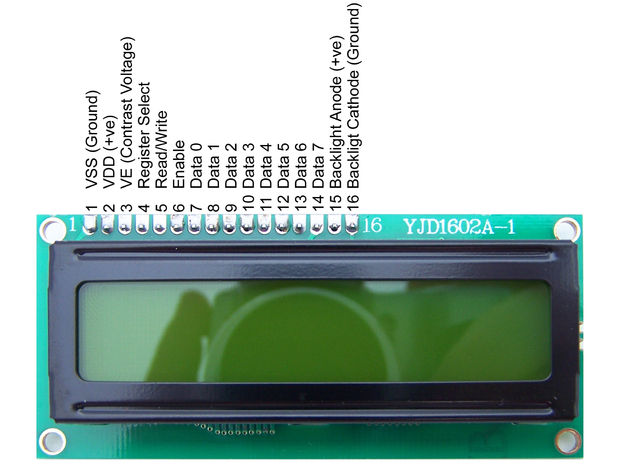
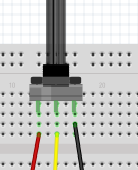
This is just a simple instructional on using the Mega 2560 with a generic LCD shield.

Below is the pinout for the generic lcd shield. <http://cdn.instructables.com/FMC/PZRT/G8LWOHW9/FMCPZRTG8LWOHW9.MEDIUM.jpg>



You can use any pins you really want to for the data of course, but for mine I used pins (4,5,6,7) on the Arduino connected to (Data 4, Data 5, Data 6, Data 7) respectfully on the LCD Shield. (VSS, Backlight Cathode, Read/Write) are connected to ground, (VDD and Backlight Anode) connect to 5V, Register Select is connected to pin 8, and Enable is connected to pin 9 on the Arduino.

VE(Contrast Voltage) proved to be the most important pin when setting up. It connects to the center rail of a 10kohm potentiometer. It is very important to have this calibrated correctly or text will be too dark or too light to read. Pictured below, VE will connect to yellow. Black and Red are ground and 5V respectively.



Now to Programming.

//In the Arduino program window you must remember to include the library for the shield, it is within the default library //download so you shouldn’t need to manually download it.

#include <LiquidCrystal.h>

// and then initialize the library with the numbers of the interface pins

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

// You’ll notice pins 8 and 9 are not connected on the shield as they are not required for a 16x2 display(to my //knowledge).

void setup() {

// finally set up the LCD's number of columns and rows:

lcd.begin(16, 2); // For a 16x2 LCD shield, there are other standards, but this is what I have.

// Print a message to the LCD.

lcd.print("Hello, world!");

//To replace the text printed above, such as in a loop, you’ll need to set the start position of the text, such that

lcd.setCursor(0,0); //Will set the text on the top line in the corner.

//and then print your new text, taking care to use all 16 characters (including spaces) to overwrite previous text.

lcd.print("You may wonder ");

// Printing to the second line, again setCursor but this time to (0,1)

lcd.setCursor(0,1);

lcd.print("who I am? ");

// This is basically all you are required to do, to overwrite you will need to change cursor position to whatever line you //want to overwrite. It’s a bit tedious, but it’s pretty simple code. Remember to put in delays or it will go by too quickly //to read. One other useful command is lcd.clear(); that should wipe anything you have on your LCD shield.

}

void loop();{

}

Please be aware of capitalization as word likes to auto capitalize some things. “setCursor” must have the C capital or Arduino will not recognize it.