LCD Display

Task: Show a text on an LCD Display.

Overview:

An LCD is short for **Liquid Crystal Display** . When current is applied to this special kind of crystal, it turns opaque. This is used in a lot of calculators, watches, and many simple displays. Adding an LCD to your project will make it super portable and allow you to integrate up to 32 characters (16 x 2) of information into your Arduino project.

The LCD has the well-known Hitachi HD44780 with parallel control. It is soldered on a small circuit with a PCF8574 that handles I2C communication. This PCF8574 is just a 8 bit I/O extender through TWI or I2C protocol such as the one in a AVR microcontroller and of course the Arduino that is used during our workshops.

Datasheet of PCF8574 from NXP [http://www.nxp.com/documents/data_sheet/PCF8574.pdf]

It means also that all the control of the cursor and character control has to be made in the software embedded in the master controller. Concerning the package, it's only a few millimeters thicker than the LCD alone. There is a useful 4 pins connector for I2C (and another one if you need to connect something else on the bus), and an adjustable for the contrast control. Just connect the ground and the voltage (5V) to the Arduino then the data SDA on ANALOG 4 and clock SCL on ANALOG 5. In fact these pins are also digital for the TWI hardware inside the micro-controller of Arduino. You can't connect the I2C bus elsewhere, or you have to develop a software I2C library that consumes resources.

BEWARE !

If you intend to use an Arduino Mega (or any controller with a different chip than ATmega8 ou 168 et 328 as standard Arduinos), you may check what pins have the SDA and SCL signals.

Pin mapping of the ATmega2560 for Arduino Mega [http://arduino.cc/en/Hacking/PinMapping2560]

Thus, some extension shield will be useless because they don't connect the I2C pins of the Arduino Mega.

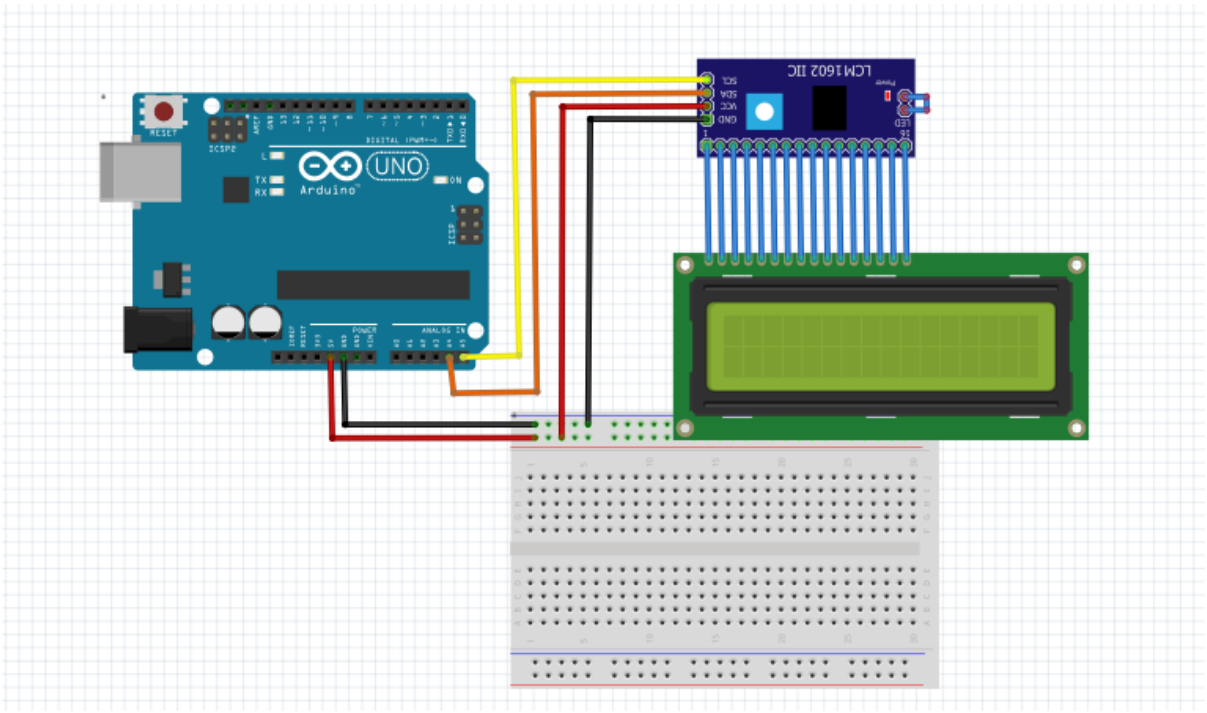
Arduino LiquidCrystal Library

The Arduino community has a library called LiquidCrystal_I2C that simplifies the controls and signals to the LCD. To use this, you have to download the file and add the library file in the Arduino library. In the Arduino IDE, add this line to the very first line of code:

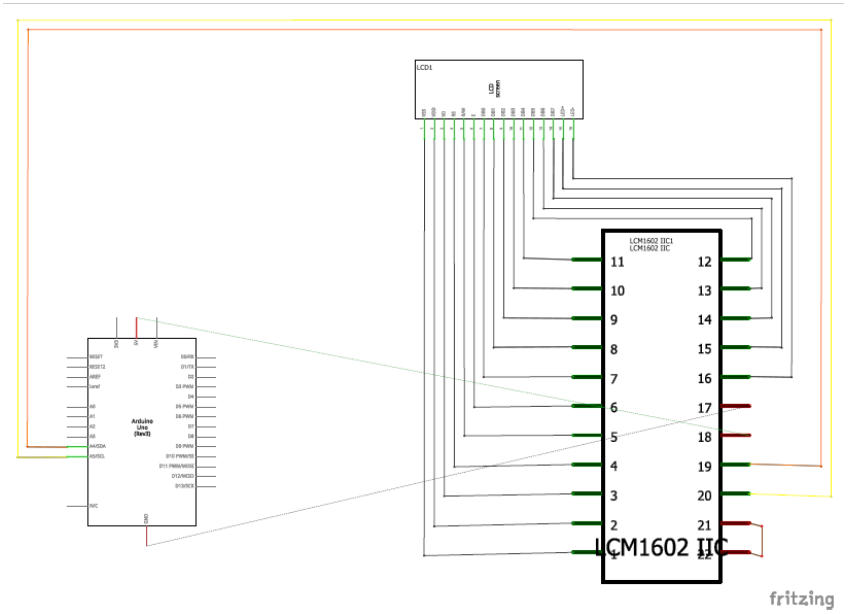
#include <LiquidCrystal_I2C.h>

Setup:

Diagram



Schematic

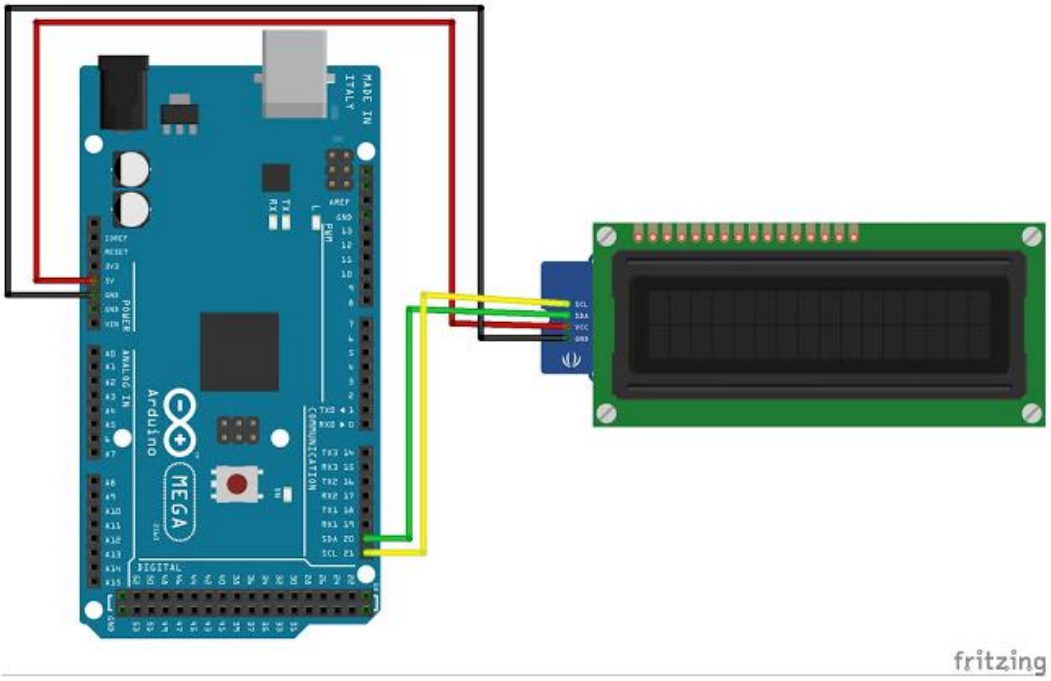


Coding:


lcd_test | Arduino 1.8.5
File Edit Sketch Tools Help
lcd_test
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
// initialize the LCD
lcd.begin();
// Turn on the blacklight
lcd.setBacklight((uint8_t)1);
// First row
lcd.print("Assalamualaikum");
// Second row
lcd.setCursor(0,1);
lcd.print("Dunia");
}
void loop()
{
// Do nothing here...
}
Done Saving.
2080 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM8

For Arduino Mega

Schematic diagram



1. Connect the Black jumper cable from the GND pin on the LCD to the GND pin on the MEGA
2. Connect the Red jumper cable from then VCC pin on the LCD to the 5V pin on the MEGA
3. Connect the Green jumper cable from the SDA pin on the LCD to the SDA pin on the MEGA
4. Connect the Yellow jumper cable from the SCL pin on the LCD to the SCL pin on the MEGA

Arduino Sketch:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F,16,2);  // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()
{
 lcd.init();                  // initialize the lcd
 lcd.init();
 // Print a message to the LCD.
 lcd.backlight();
```

```
  lcd.setCursor(3,0);
  lcd.print("Hello, world!");
  lcd.setCursor(2,1);
  lcd.print("Ywrobot Arduino!");
   lcd.setCursor(0,2);
  lcd.print("Arduino LCM IIC 2004");
   lcd.setCursor(2,3);
  lcd.print("Power By Ec-yuan!");
}



void loop()
{
}
```