



Interfacing 20x4 LCD with Arduino

by [apoorveinstein](#) on January 6, 2015

Table of Contents

Interfacing 20x4 LCD with Arduino	1
Intro: Interfacing 20x4 LCD with Arduino	2
Step 1: Material Required	2
Step 2: LCD-Pinout	3
Step 3: Schematic	4
Step 4: Code	6
Step 5: That's All	7
Related Instructables	7
Advertisements	8
Comments	8

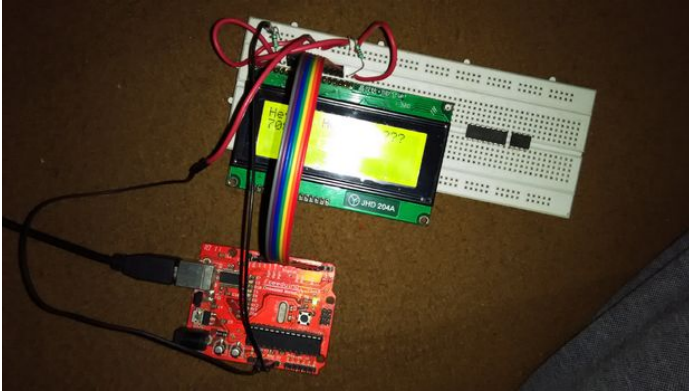
Author:apoorveinstein

An electronics enthusiast... #Complexity simplified

Intro: Interfacing 20x4 LCD with Arduino

A liquid-crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly.

Here, in this i'ble we're going to use a monochromatic 20x4 alphanumeric LCD. 20x4 means that 20 characters can be displayed in each of the 4 rows of the 20x4 LCD, thus a total of 80 characters can be displayed at any instance of time.



Step 1: Material Required

List of requirements is as follows:

1. Arduino/or any other clone
2. 20x4 LCD
3. Breadboard
4. 1k, 220R Resistors
5. Connecting wires

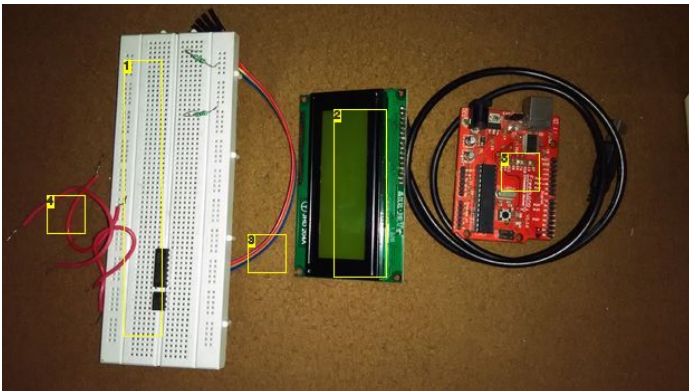


Image Notes

1. Breadboard, had few ICs lying around.
2. 20x4 LCD
3. Connecting wires
4. Some more connecting wires
5. Freeduino

Step 2: LCD-Pinout

The LCD we are using has 16 pins. Description of each pin is as follows:

Pin No Symbol Level Description

1 VSS 0V Ground

2 VDD 5V Supply Voltage for logic

3 VO (Variable) Operating voltage for LCD

4 RS H/L H: DATA, L: Instruction code

5 R/W H/L H: Read(MPU?Module) L: Write(MPU?Module)

6 EH H->L Chip enable signal

7 DB0 H/L Data bus line

8 DB1 H/L Data bus line

9 DB2 H/L Data bus line

10 DB3 H/L Data bus line

11 DB4 H/L Data bus line

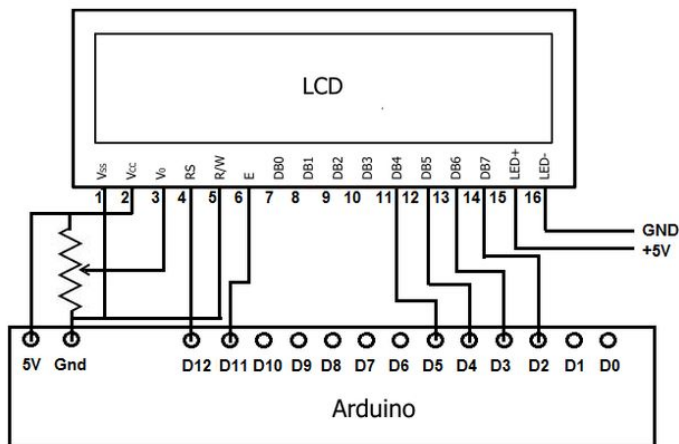
12 DB5 H/L Data bus line

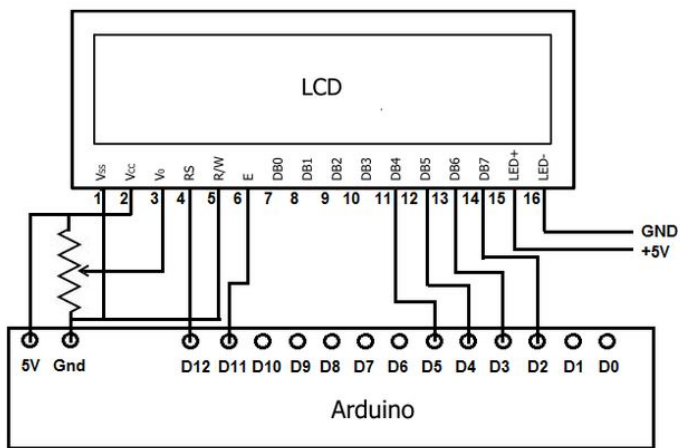
13 DB6 H/L Data bus line

14 DB7 H/L Data bus line

15 A 5V LED +

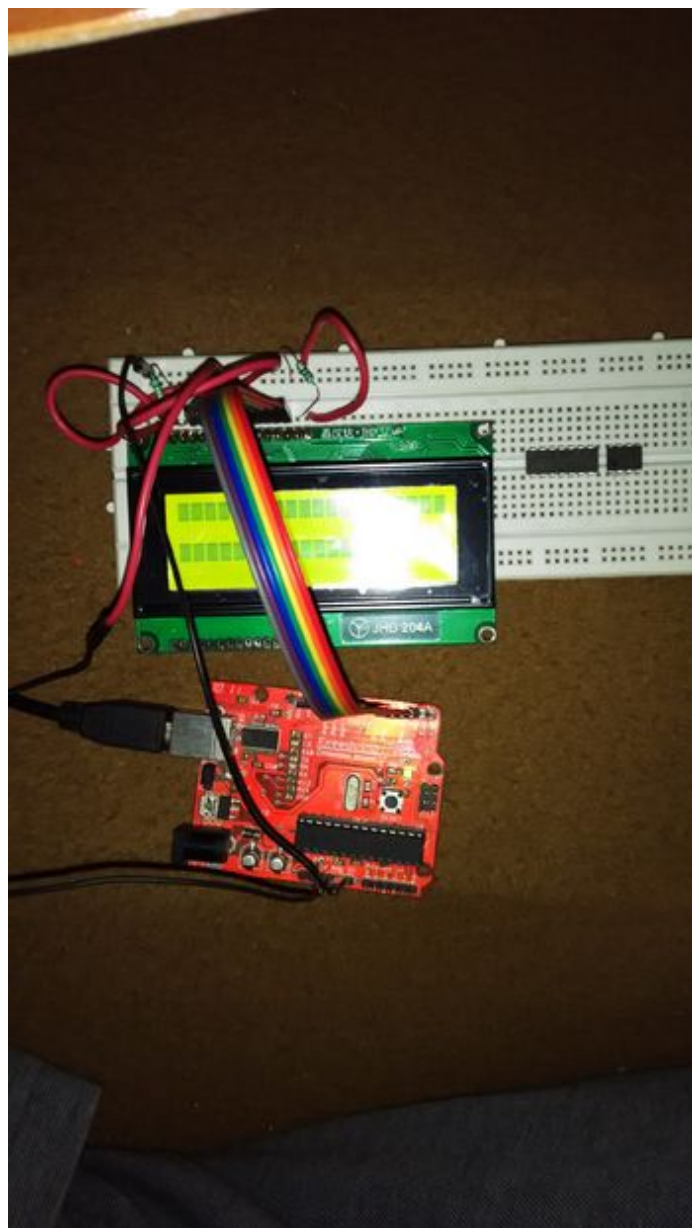
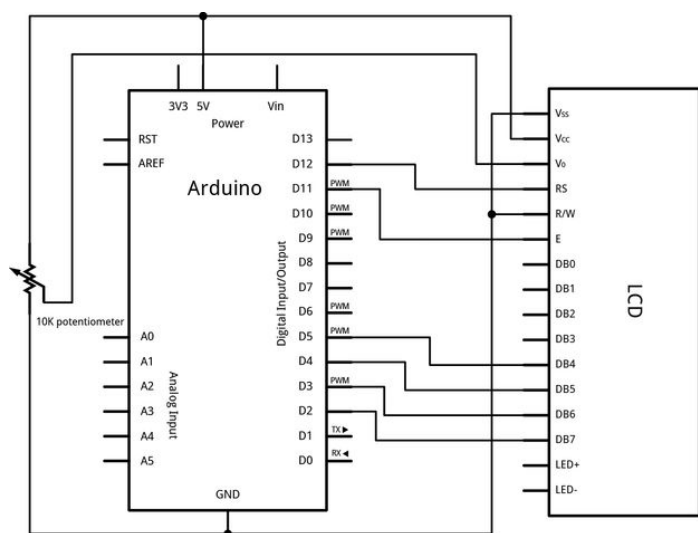
16 K 0V LED-

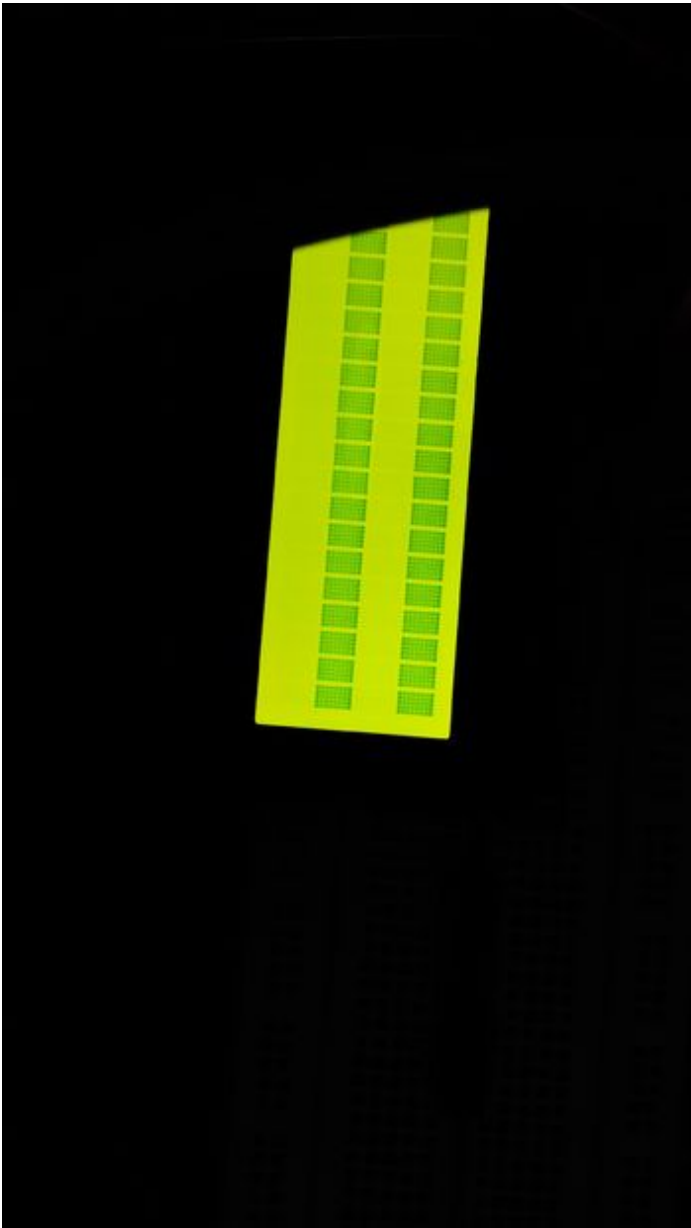




Step 3: Schematic

1. Study the schematic carefully.
2. Place your LCD on the bread board.
3. Make connections as per the schematic.
4. Instead of the potentiometer, you can use a 1k resistor and connect Pin 3 of LCD to Vcc via the resistor.
5. Carefully check whether all the connections are tight and correct.
6. Power up your Arduino via USB and check whether the LCD lights up. If yes, proceed.





Step 4: Code

There are two options that you can follow, either you can build up your own code, if you know how to or you can use the example code from the built-in Arduino library "**LiquidCrystal.h**".

I have used the **HelloWorld** example from the library. You can find this example in Arduino IDE following the path: File->Examples->LiquidCrystal->HelloWorld.

Now connect the Arduino to any of the USB ports and upload the code.

Once the code is uploaded, you must be able to see "hello, world" printed on the LCD.

```
File Edit Sketch Tools Help
HelloWorld / Arduino 1.8.5

#include <LiquidCrystal.h>

// This example code is in the public domain.
// http://www.arduino.cc/en/Reference/LiquidCrystal

// Include the library code:
#include <LiquidCrystal.h>

// Initialize the library with the number of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // Set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD:
  lcd.print("hello, world!");
}

void loop() {
  // Set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // Print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

Step 5: That's All

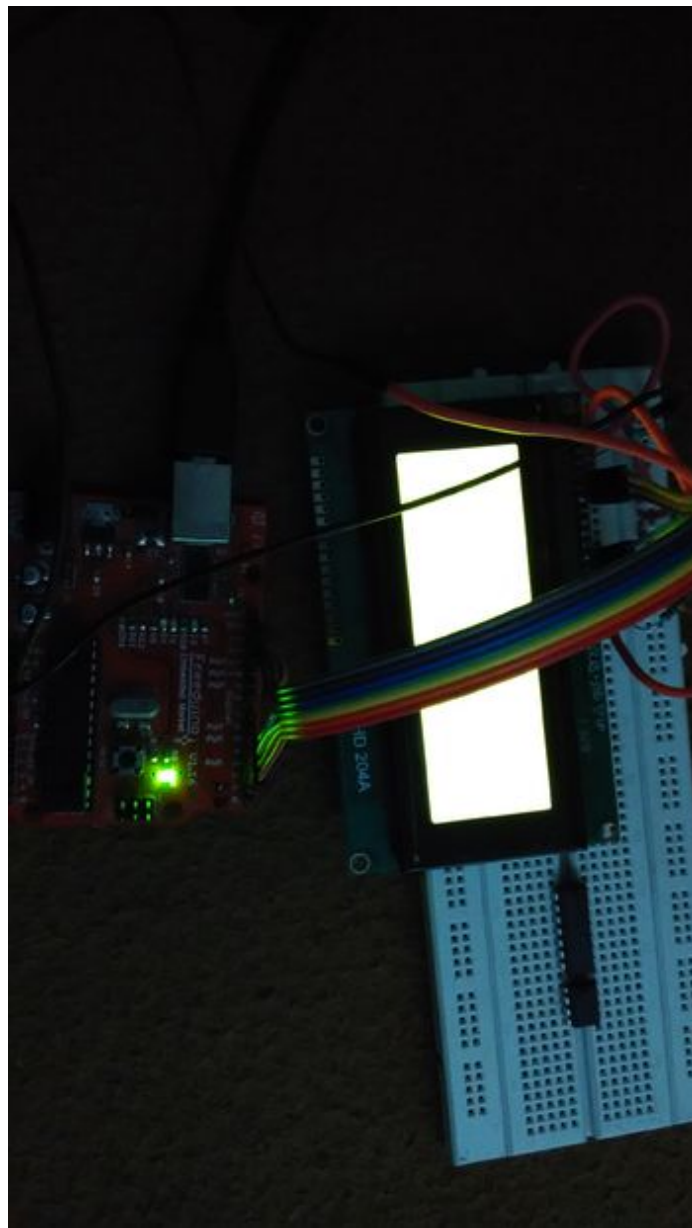


Image Notes

1. Hey!!! How R U??

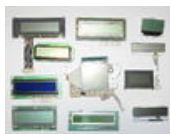
Related Instructables



Using PCF8574 backpacks with LCDs and Arduino by tronixlabs



Arduino Thermostat with 2x DS18B20 i2c 4x16 Display, 2 RGB LED's and 3 Relay's by Guxtdu



Salvaging Liquid Crystal Displays (LCDs) by Josehf Murchison



LCD display I2C adapter for Arduino with PCF8574A by uge02



Tutorial: Módulo Serial para Display LCD con Arduino by ElectroCrea



Arduino ultrasonic range finder, with I2C lcd display! by Jan_Henrik

Comments