

# Introduction to Arduino and the LCD Display – Hello, World!

Dr. Matthew Riehl

August 24, 2021

## 1 Introduction

For the next several weeks, we will be building circuits to investigate the properties of resistors, capacitors, and other electronic components. In order to do this, we first build a simple device that will be used to help us gather our data in later labs.

*Note that this is NOT a programming class (no previous programming experience is needed); the code is provided and can be used without editing (although you are encouraged to play with the code to extend the functionality of the device...). Neither is this a circuits class; we will not be exploring the details of how the Arduino and other components work. Of course, we hope that this taste of programming and bread-boarding inspire you to take additional courses in computer science or electronics...*

This week, we will set up the Arduino to send a message to the LCD display. You will wire the LCD to the Arduino with the breadboard and jumper wires and use your computer to compile the code that will control the display. This code will be sent to the Arduino which will execute the code until it is turned off or until the code is replaced with a different set of instructions. Once the code has been stored on the Arduino, it can be disconnected from the computer and powered by any appropriate source.

In later labs, we will modify the circuit to gather data which will be displayed on the LCD screen. This has the benefit of allowing you to see what is going on inside the circuit to some extent. The downside is that any changes you make to your circuit or experiment will have to be coded into the program and loaded onto the Arduino.

## 2 Materials

- Computer with Arduino IDE installed ([arduino.cc/en/software](https://arduino.cc/en/software)<sup>1</sup>)
- Arduino Uno board and power cable
- Half-size breadboard
- LCD display module
- 220  $\Omega$  resistor ( $\Omega$  is Ohm)
- 10 k $\Omega$  potentiometer
- Jumper wires

---

<sup>1</sup>for this exercise, Arduino IDE 1.8.15 was used

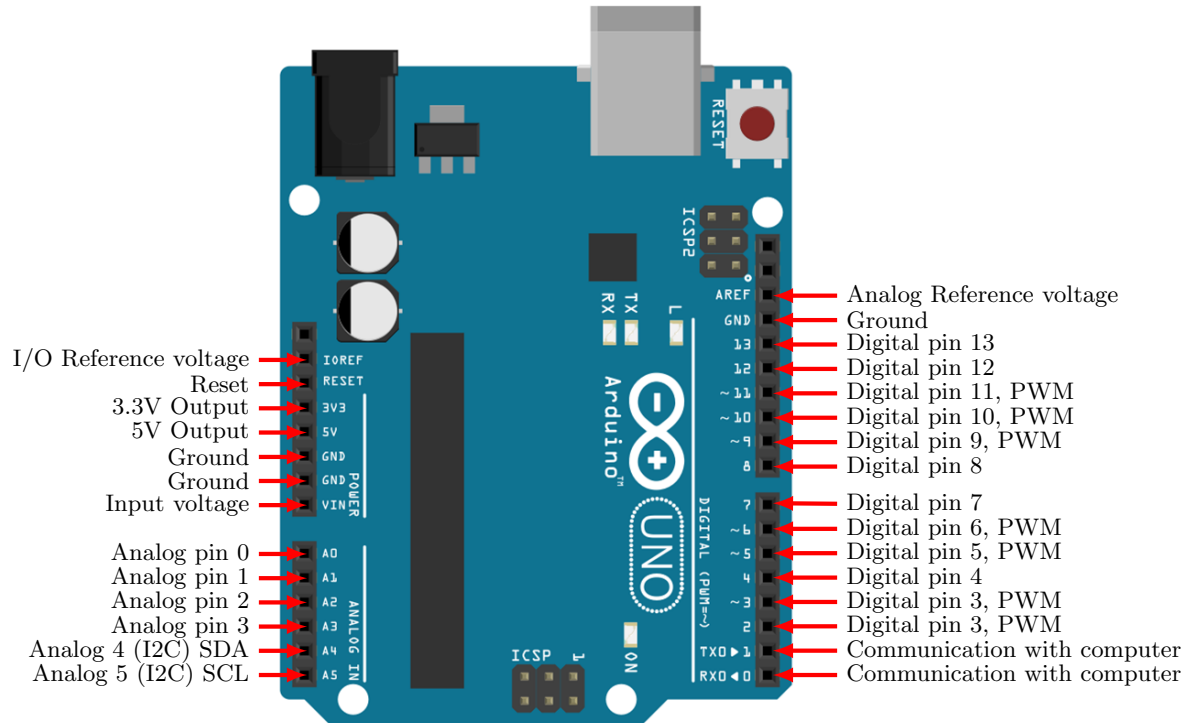


Figure 1: Pin-out Diagram for the Arduino Uno. PWM is Pulse Width Modulation

### 3 Wiring the LCD to the Arduino

We will build circuits by copying diagrams given in the laboratory handout. In some cases, the functions of pins will be described, but mostly we will leave the details to the engineers and programmers who designed the boards and accessories. Pin-out diagrams of the Arduino Uno (Figure 1) and the LCD board (Figure 2) are shown.

Some pins will not be used for our projects.

The wiring diagram is shown in Figure 3. It is useful to use the color of wire suggested in the diagram, because it will be easier to troubleshoot if the board does not work. It is suggested that you wire the breadboard first (black, red, and blue wires) and then connect the Arduino with final wires. If you have some experience with the Arduino, you might notice that the LCD wiring in this project is a bit different than what you have seen online. The wiring shown here is more clear and the code has been changed to reflect the changes.

You will need to find a  $220\Omega$  resistor. The resistance of each resistor is coded on the body of the resistor by means of a series of colored bands. Most resistors have either 4 bands or 5 bands. See figure 5 for a description of the resistors and the colors used and figure 6 for an example of how to read the bands.

### 4 Code for the Arduino

The code for Arduino is written in the computer language "C++", which is one of the most common languages in use. The code that will be used today is given below. Note that the text that occurs between `/*` and `*/` are comments, not part of the actual code. Also, any line that begins with `//` is a comment. If you delete all of the comments, the code is quite small.

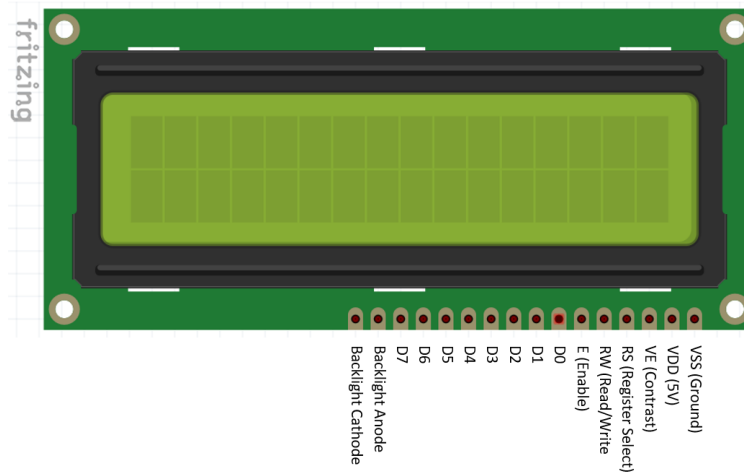


Figure 2: Pin-out Diagram for the LCD

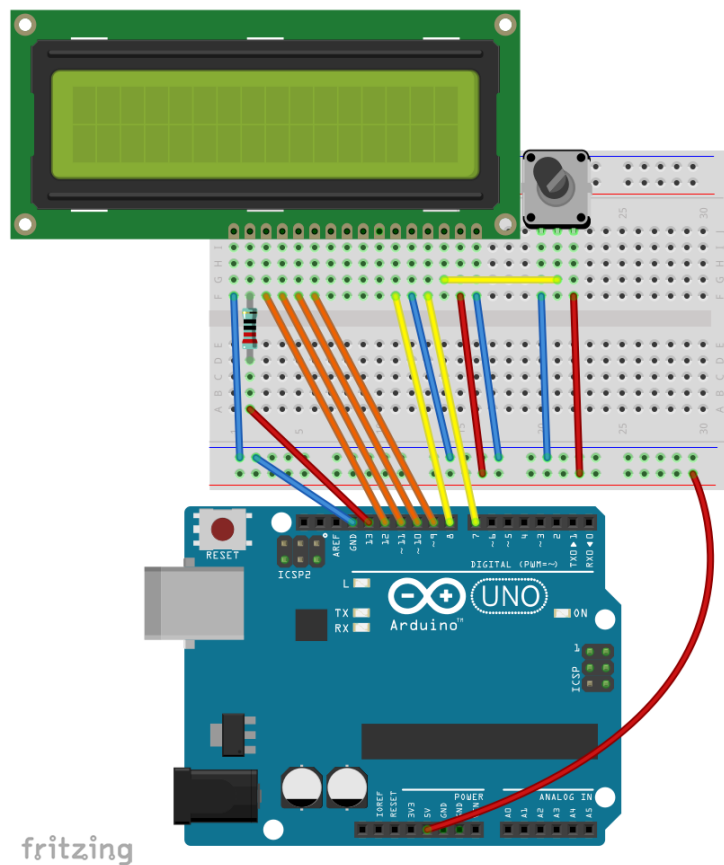


Figure 3: Hello World Wiring Diagram

This code is in the file Arduino\_Basic\_Hello\_World.ino. The file will open in the Arduino IDE. The file Arduino\_Hello\_World.ino demonstrates a number of commands available for the LCD.

```
/*
```

```
LiquidCrystal Library - Hello World
```

```
This code is included with the files for Arduino Lab Hello World by  
Matthew Riehl and is in the public domain. I have only modified the  
code to better support the laboratory exercise.
```

```
Demonstrates the use a 16x2 LCD display. The LiquidCrystal  
library works with all LCD displays that are compatible with the  
Hitachi HD44780 driver. There are many of them out there, and you  
can usually tell them by the 16-pin interface.
```

```
This sketch prints "Hello World!" to the LCD  
and shows the number of seconds since reset.
```

```
Note that the pin assignments are NOT the same as found on many Arduino  
pages. They have been moved around to make the wiring more intuitive  
and easier to troubleshoot.
```

```
The circuit:
```

```
* LCD RS pin to digital pin 7  
* LCD Enable pin to digital pin 8  
* LCD D4 pin to digital pin 9  
* LCD D5 pin to digital pin 10  
* LCD D6 pin to digital pin 11  
* LCD D7 pin to digital pin 12  
* LCD Backlight (pin 15 on LCD) to pin 13 on Arduino (might need to add a 220 ohm series resistor)  
* LCD R/W pin to ground  
* LCD VSS pin to ground  
* LCD VCC pin to 5V  
* 10K potentiostat:  
* ends to +5V and ground  
* wiper to LCD VO pin (pin 3)
```

```
Library originally added 18 Apr 2008  
by David A. Mellis  
library modified 5 Jul 2009  
by Limor Fried (http://www.ladyada.net)  
example added 9 Jul 2009  
by Tom Igoe  
modified 22 Nov 2010  
by Tom Igoe  
modified 7 Nov 2016  
by Arturo Guadalupi
```

```
This example code is in the public domain.
```

```
http://www.arduino.cc/en/Tutorial/LiquidCrystalHelloWorld
```

```

*/

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

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define backlightPin 13          // This allows the user to turn the back light on and off.

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  pinMode(backlightPin, OUTPUT);
  digitalWrite(backlightPin, HIGH);
  lcd.print("Hello, World!");      // Change this to something cool!
}

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);
}

```

## 5 Hello World Program in the Arduino IDE

Open the Arduino IDE, identified by the logo shown in Figure 4. Open the **File>Examples>01.Basics>Blink** and wait for the code to appear in a new window. Read through the comments which explain what the program does and which parts of the code control the output.



Figure 4: Arduino Logo

Connect the Arduino to the computer with the cable. Under the Tools menu in the IDE, find the Board Manager and select your board (Arduino Uno) and then find the Port menu and ensure that the IDE is connected to the Arduino Uno port. Under the Sketch menu, choose Upload and wait until the routine is over. If no errors are encountered, a small LED on the Arduino will be blinking on and off in one second intervals. Now change the timing on the blinking and upload your new program to see the new blink pattern. This sketch does not use the LCD, which will remain off.

Color	1 <sup>st</sup> Band	2 <sup>nd</sup> Band	3 <sup>rd</sup> Band	Multiplier	Tolerance
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	±1%
Red	2	2	2	100Ω	±2%
Orange	3	3	3	1kΩ	
Yellow	4	4	4	10kΩ	
Green	5	5	5	100kΩ	±0.5%
Blue	6	6	6	1MΩ	±0.25%
Violet	7	7	7	10MΩ	±0.10%
Grey	8	8	8	100MΩ	±0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	±5%
Silver				0.01Ω	±10%

Figure 5: Color banding codes for resistors

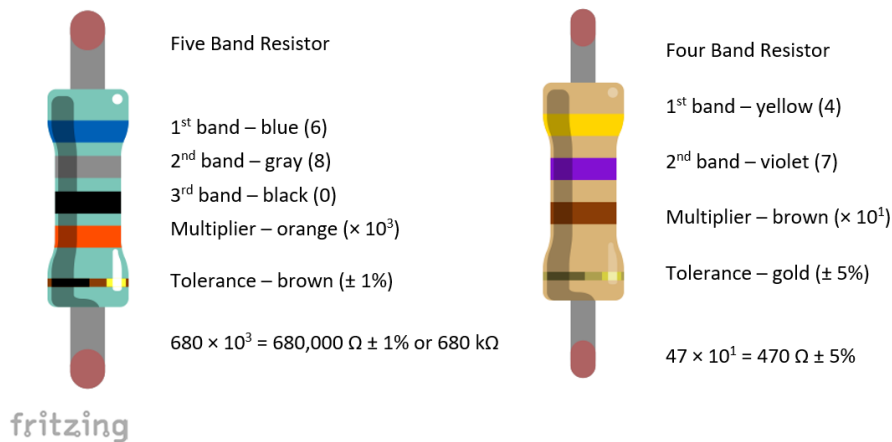


Figure 6: Examples of a 5-band and 4-band resistor

Open the file `Arduino_Basic_Hello_World` by double clicking icon. It will open in a new Arduino IDE window. It should be identical to the code above. Upload this sketch as it is and ensure that it is working. The LCD should display a message on the first line and be counting seconds on the second line. It may be necessary to change the setting on the potentiometer in order to see the screen with proper contrast.

Now, the file `Arduino_Hello_World.ino` can be opened and uploaded to the Arduino. Take some time to read the code and try to modify it to see what it can do. More details on how the commands work can be found online (see the links in the comment section of the program)– this project is not intended to teach the details of programming.