

Make Your Own Blink

Before You Start

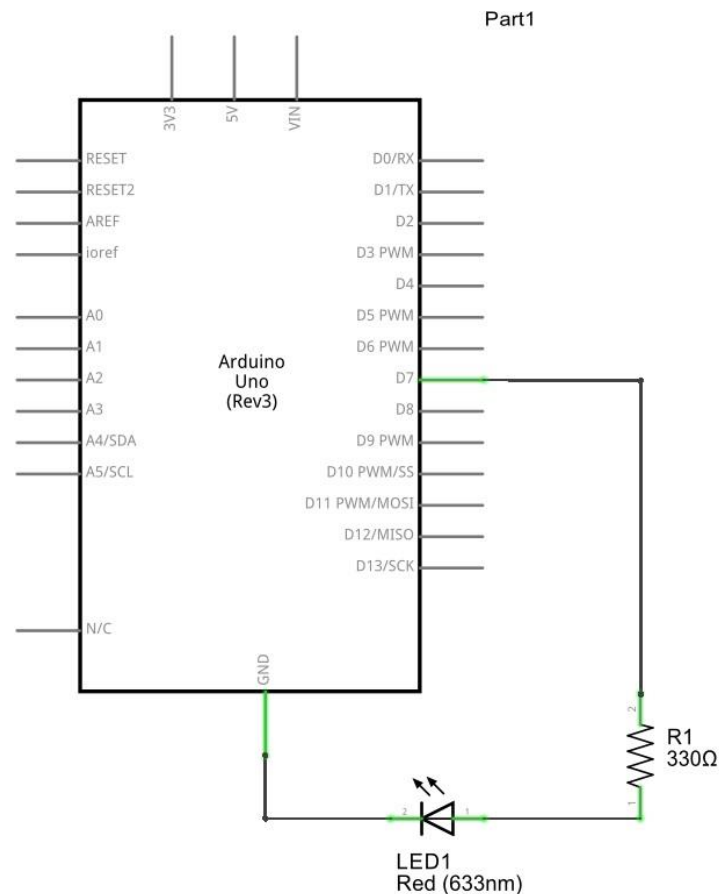
You should have gone through the “Getting Started” page on the Arduino Website.

- Windows <https://www.arduino.cc/en/Guide/Windows>
- Mac <https://www.arduino.cc/en/Guide/MacOSX>
- Linux is almost the same, and tutorials can be found on the Arduino getting started page.

This page guides you through making sure that you can connect your board, troubleshooting, and uploading your first sketch. Don't worry if you didn't completely understand everything that was said, we will go over the very basics in a little more detail in this tutorial!

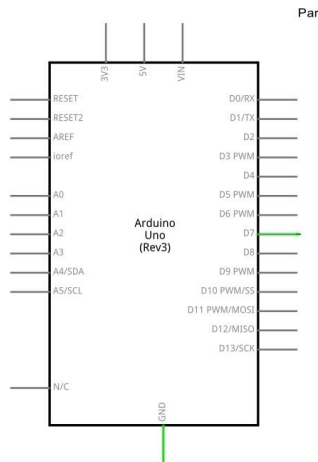
The Circuit

Let's take a look at the circuit schematic for our project below, and then break it down.



THE ARDUINO UNO

The following component is the Arduino UNO. Each of the pins is labelled as it appears on the Arduino (e.g. D7 corresponds to D7 on the board, A0 corresponds to A0 on the board). One of the more notable pins is the GND pin, which stands for “ground”. This is the common GND connection for many of your projects. Current flows FROM a pin, TO ground.



THE RESISTOR

The next component is a resistor. The resistor is a device that resists current flow. We put resistors in front of our LED's to make sure that too much current doesn't flow through the LED. If the resistor was removed, and the LED was connected by itself, too much current would flow, and the LED would burn out.

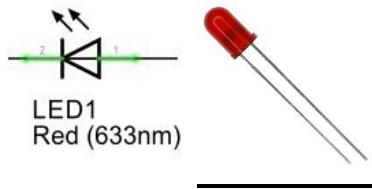


Resistors have units in Ω (ohms). This is a unit just like the metre or gram, except it is the unit of resistance, as opposed to distance or mass respectively. The greater the value of the resistor, the more that it resists the flow of current.

The resistor has four colored bands, and these bands give you information about the value and accuracy of the resistor. Look online to find out how to read resistor color codes.

THE LED

LED is an acronym that stands for “Light Emitting Diode”. True to its name, the LED is a type of diode that emits light when voltage is applied. It is important to always remember the resistor that comes before your LED!



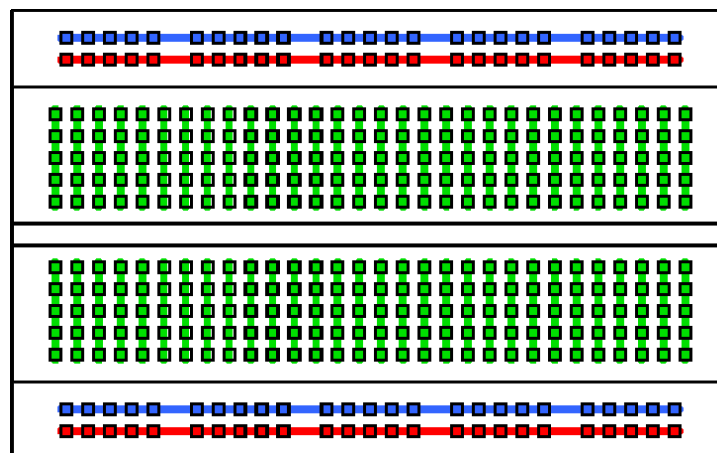
Resistors do not have a forward or backward side. No matter which way you place them in a circuit, they perform the same. However LED's have a polarity. This means that you must insert them correctly into the circuit for them to work correctly. An easy way to remember which way to insert the LED is to think of the triangle as an arrow, pointing in the same direction as current flow. (This is the same direction as the path towards GND).

LED's have one lead that is longer than the other. The longer lead is the positive lead, and the shorter one is negative (Anode and Cathode respectively). Ensure that you understand how to orient an LED, otherwise it may burn out and not work!

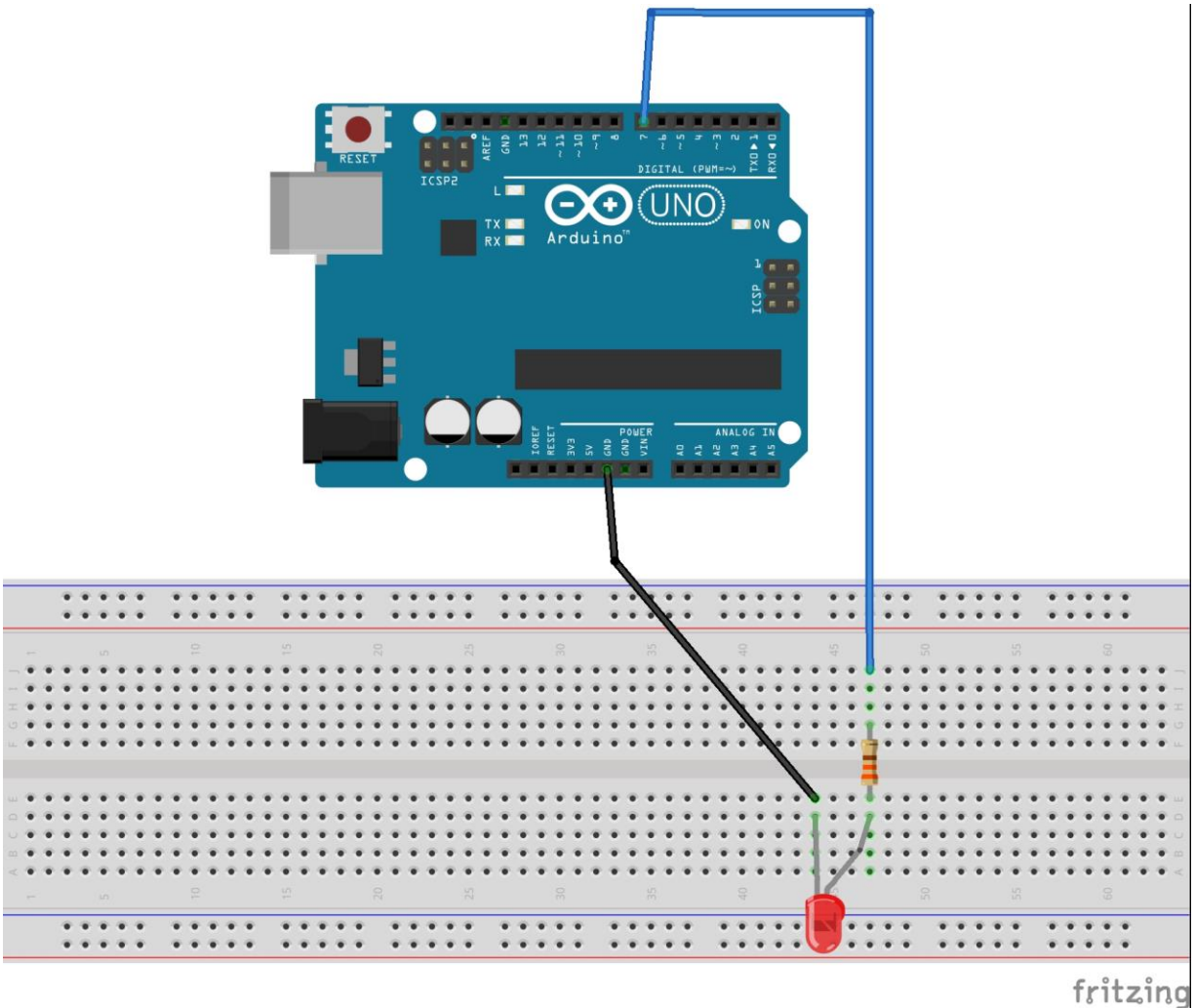
The Breadboard

A breadboard is a board that has holes to insert wires into, with the holes connected in the arrangement below. It is very useful for prototyping circuits, and trying something out before building a final version.

The common way to connect the breadboard is to connect ground (GND) to the blue bus, and your positive voltage to the red bus. The other holes, which are connected in the manner noted by the green lines, are used for inserting your components and connecting components to power or ground via a wire.



The breadboard layout for this circuit appears below.



Try and connect the circuit as show below. Pay careful attention to the specific holes on the breadboard that are used. Notice that the components connect nicely to each other using the green busses from the breadboard layout diagram.

The Code

Building a circuit is only half of any Arduino project. The other half consists of programming the Arduino to control the circuit you built. First, let's take a look at the code we will use for this project. The comments in the code break it down step by step.

```
1. void setup() {  
2.   // This is the area that you set up your program. This  
3.   // includes starting things up on the microcontroller,  
4.   // and making sure that your peripherals are working  
5.   // correctly.
```

```
6.  
7.  // On the line below, we tell the Arduino to make pin 7  
8.  // OUTPUT. This means we can change the value of  
9.  // the pin between 1 and 0.  
10. pinMode(7, OUTPUT);  
11. }  
12.  
13. void loop() {  
14.  // Once your Arduino is finished preparing from the setup()  
15.  // function above, it runs through the code below over and  
16.  // over forever.  
17.  
18.  // First, we write a 1, or HIGH to pin 7. This turns  
19.  // the LED on  
20.  digitalWrite(7, HIGH);  
21.  
22.  // Next, we wait 500 milliseconds, or half a second.  
23.  delay(500);  
24.  
25.  // Next, we write a LOW, or 0 to pin 7. This turns the  
26.  // LED off  
27.  digitalWrite(7, LOW);  
28.  
29.  // Then we wait another half a second  
30.  delay(500);  
31.  
32.  // After this point, our program goes back to the  
33.  // beginning of loop() and starts over. Our LED will  
34.  // blink forever!  
35. }
```

Connect your Arduino to your computer, and open the Arduino IDE. Ensure that your Arduino is connected. Next, enter the code above (you don't need to include the comments) into your Arduino IDE.

Click verify (the check mark) to ensure that your code is error free. If everything looks good, then upload the code to the Arduino.

The Result

If everything is connected correctly, and the code executes successfully, you should see the LED blink on an off, completing one cycle per second. Congratulations, you built your first real Arduino project!