

# Challenge 1 - Blink

In this tutorial you'll be getting an Arduino to flash an LED on and off. It is considered the "Hello World" application when using the Arduino platform.

This is what your Arduino circuit will do at the end of the process.

This challenge will be split in various sections, like the later ones. The first stage is a functional demo to test to see if the code and circuit work correctly.

## Theory

### Electricity

<https://learn.sparkfun.com/tutorials/alternating-current-ac-vs-direct-current-dc>

<https://learn.sparkfun.com/tutorials/what-is-electricity>

<https://learn.sparkfun.com/tutorials/electric-power>

### Batteries

<https://learn.sparkfun.com/tutorials/what-is-a-battery>

<https://learn.sparkfun.com/tutorials/battery-technologies>

### Onboard LED

Each Arduino board has an built-in onboard LED to be used for testing purposes and simple feedback. On the Arduino Uno, this LED is attached to pin 13 as it is on most Arduino boards. A few differ however.

For a list of the pins used on Arduino boards, look under Circuit on this page:

<https://www.arduino.cc/en/tutorial/blink>

If you're using an Arduino Compatible board, then the pin allocation may be different. You will need to check the reference guide for your particular board. For instance, if you have a Adafruit Feather HUZZAH ESP8266 board, you can view the [pinout guide here](#), and notice that the onboard LED is on pin 0.

## Arduino Style Guide

Writing code is difficult. Reading code can be even more confusing, especially if it's not written well. One of the many skills of a programmer is to be able to write clear code, not just functional code.

Part of writing clean code is using clear variable names and writing comments.

Use this site as a starting point for the Arduino Style Guide:

<https://www.arduino.cc/en/Reference/StyleGuide>

## Resources Needed

Hardware:

- Arduino Uno
- LED
- 220Ω Resistor
- Jumper wires

Software:

- [Arduino IDE](#)
- Fritzing

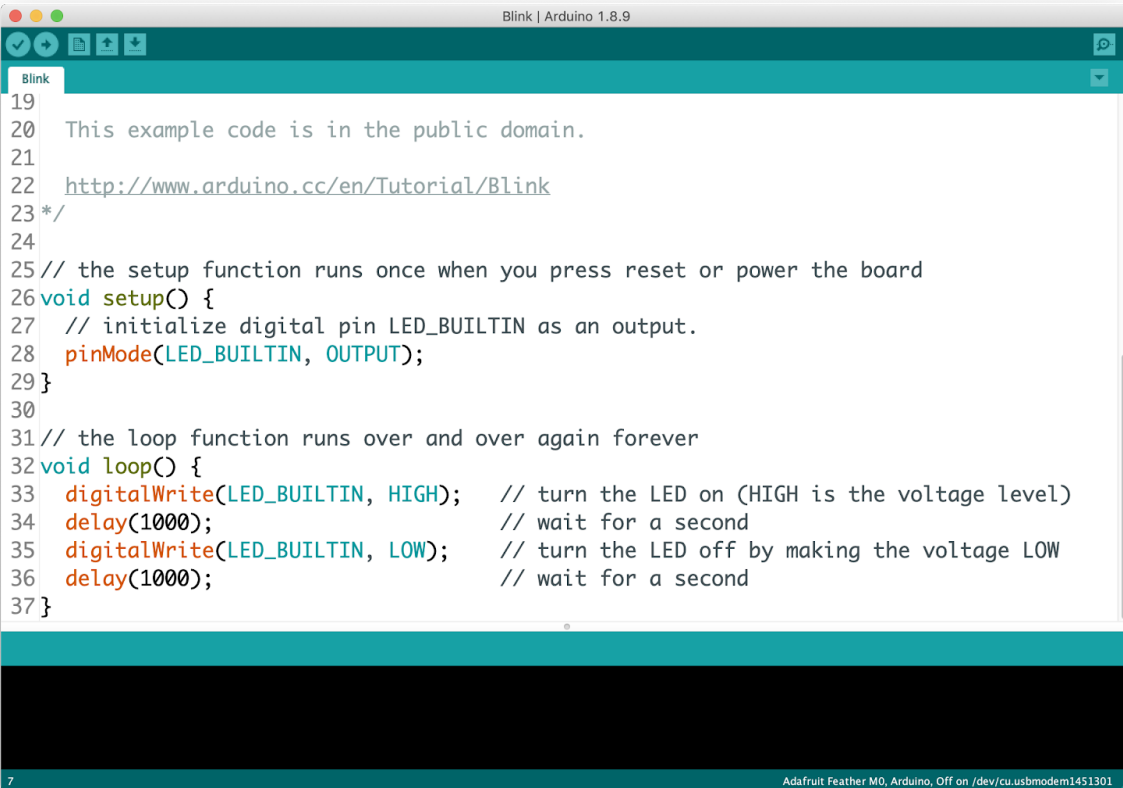


# Version 1

## Circuit

This version only requires the use of an Arduino with no additional components.

## Code

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.9". The code editor shows the following text:

```
19
20 This example code is in the public domain.
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
```

The status bar at the bottom indicates "7" and "Adafruit Feather M0, Arduino, Off on /dev/cu.usbmodem1451301".

This code example can be found in the Arduino IDE under File -> Examples -> 01. Basics -> Blink

```
/*
```

## Blink

Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED\_BUILTIN is set to the correct LED pin independent of which board is used.

If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:

<https://www.arduino.cc/en/Main/Products>

modified 8 May 2014

by Scott Fitzgerald

modified 2 Sep 2016

by Arturo Guadalupi

modified 8 Sep 2016

by Colby Newman

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/Blink>

\*/

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin LED\_BUILTIN as an output.

pinMode(LED\_BUILTIN, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

```

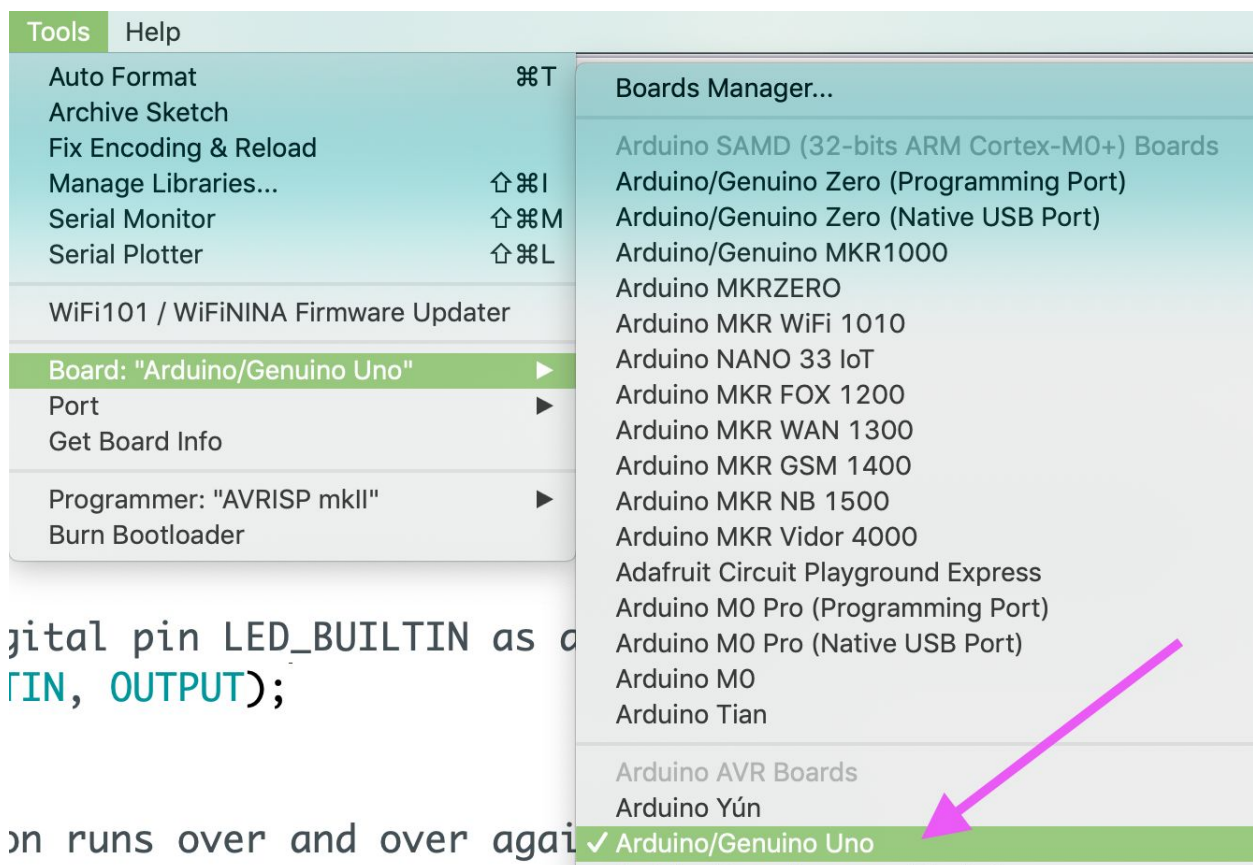
    digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the
voltage LOW
    delay(1000);                      // wait for a second
}

```

## Process

Load the code from the Examples menu.

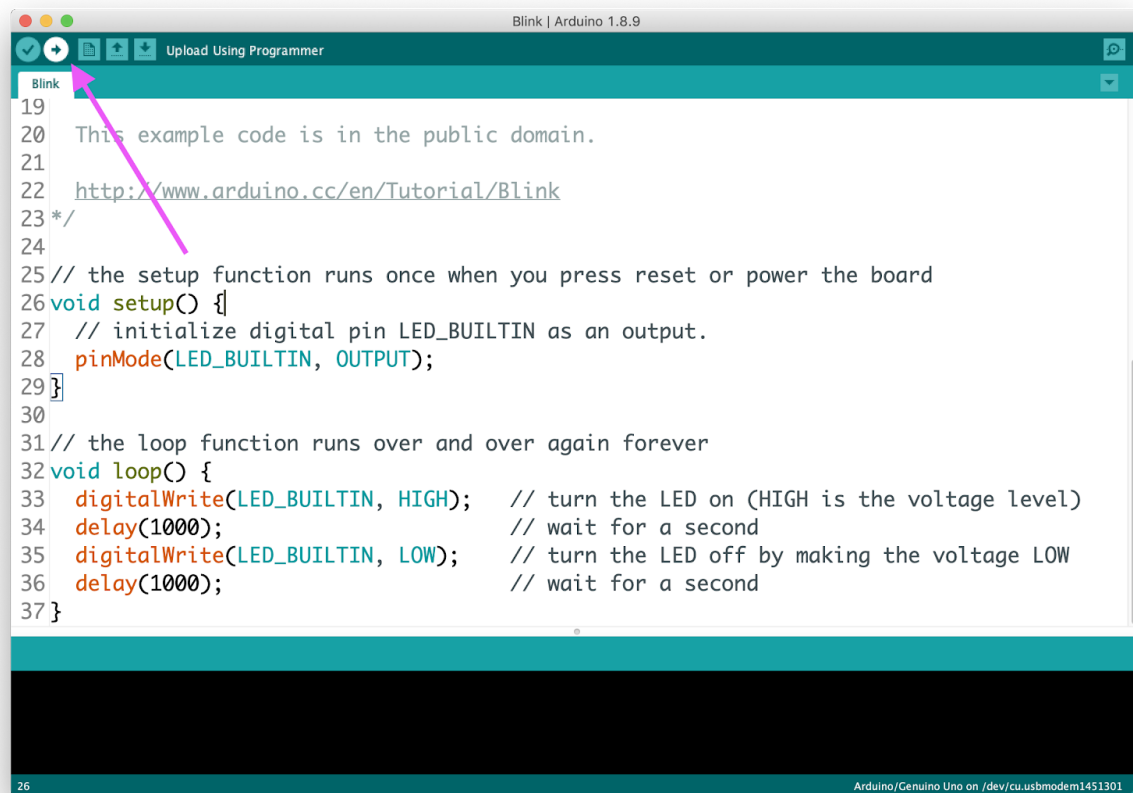
Then select the correct board in the Tools -> Board menu.



Then, select the correct port in the Tools - Port menu. If the board has been detected by the Arduino IDE, you'll see it shown against the correct Serial Port. Under Windows, these appear as COM1, COM2 etc.



Finally, press the upload button to publish the code to the board.

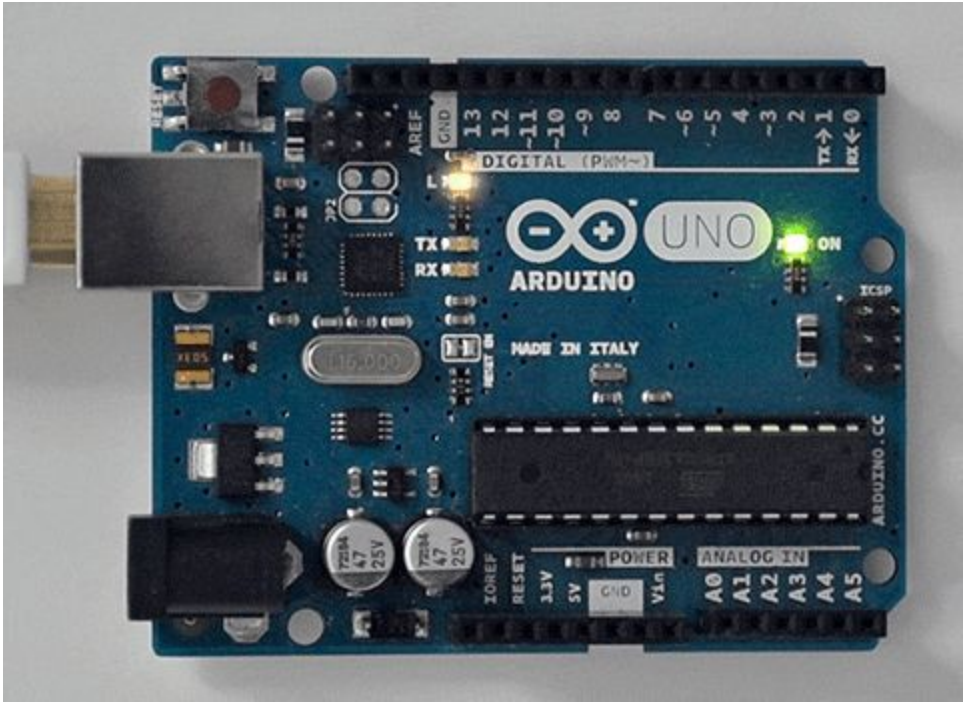


```
19
20 This example code is in the public domain.
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
```

26 Arduino/Genuino Uno on /dev/cu.usbmodem1451301

The onboard LED should be flashing on for a second, and then off for a second as can be seen below.



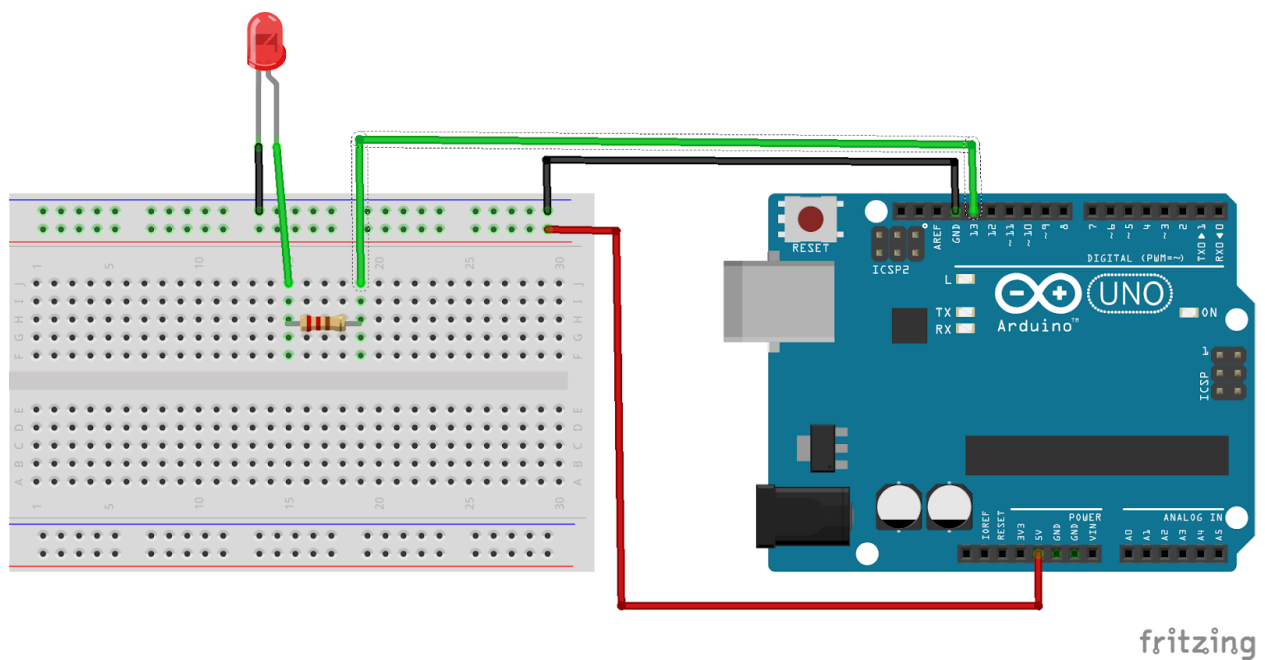


## Version 2

Now that you have the onboard LED flashing, it's now time to extend your circuit by adding in an external LED and connecting it using wires and a resistor.

### Circuit

To expand this circuit, you'll use a breadboard. If you're not familiar with breadboards, click [this link](#).



[Original Fritzing File.](#)

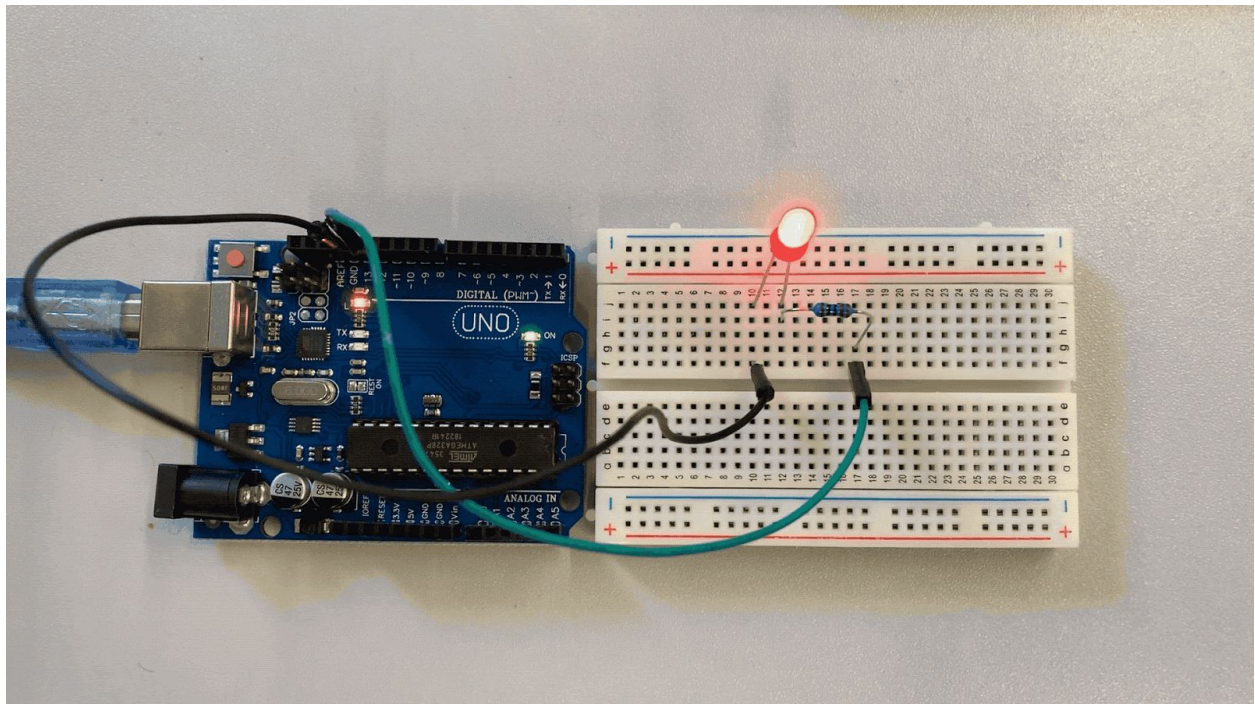
**Note** You may notice that the 5v power rail on the breadboard is connected to the Arduino's 5v pin however is not being used. This connection is not critical for this circuit, however will be used in later challenges.

The green wire in the circuit is connected from Pin 13 to one end of the resistor and then to the longer pin on the LED.

# Process

There is no additional code needed for this version. Power the Arduino and you should see the external LED now flashing alongside the onboard LED.

The circuit and output should look similar to this:



You'll notice that both the external LED and the onboard LED are blinking together. This is due to the onboard LED being attached to pin 13 on the Arduino Uno.

## Challenge Extension

Modify the code and wiring to change the LED from pin 13 to pin 9.

### Circuit

Change the wiring so that the cable coming from the resistor connects to pin 9 instead of 13.

### Code

The bulk of the code will stay the same, however you'll need to instruct it to use pin 9, instead of LED\_BUILTIN. LED\_BUILTIN is a keyword for pin 13 on the Arduino Uno.

To learn how to change the output pin, read the following reference guide for the functions used.

<https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>

<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>