



Arduino Lesson 1. Blink

Created by Simon Monk



```
01 Blink | Arduino 1.0.1
File Edit Sketch Tools Help

Blink
}

Blink
Turn on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}

Arduino IDE on COM4
```

Guide Contents

Guide Contents	2
Overview	3
Parts	4
Part	4
Qty	4
The 'L' LED	5
Loading the 'Blink' Example	6
Saving a Copy of 'Blink'	8
Uploading Blink to the Board	9
How 'Blink' Works	11
Blinking Faster	13
Other Things to Do	14

Overview

In this lesson, you will learn how program your Arduino to make the Arduino's built-in LED blink.



```
Arduino IDE | Arduino 1.0.1
File Edit Sketch Tools Help

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

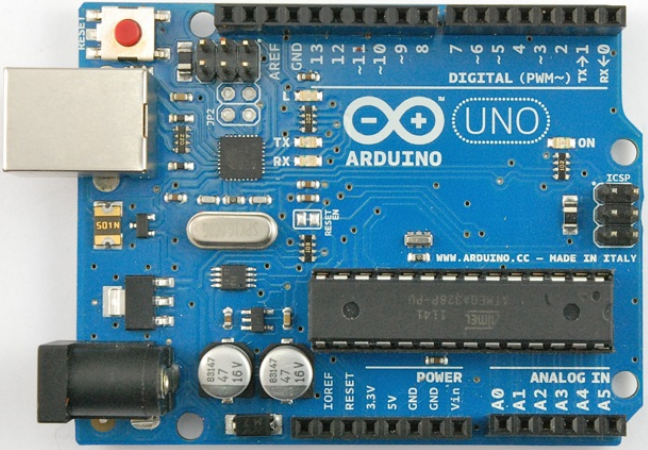

This example code is in the public domain.
*/
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
}

Arduino Uno on COM4
```

Parts

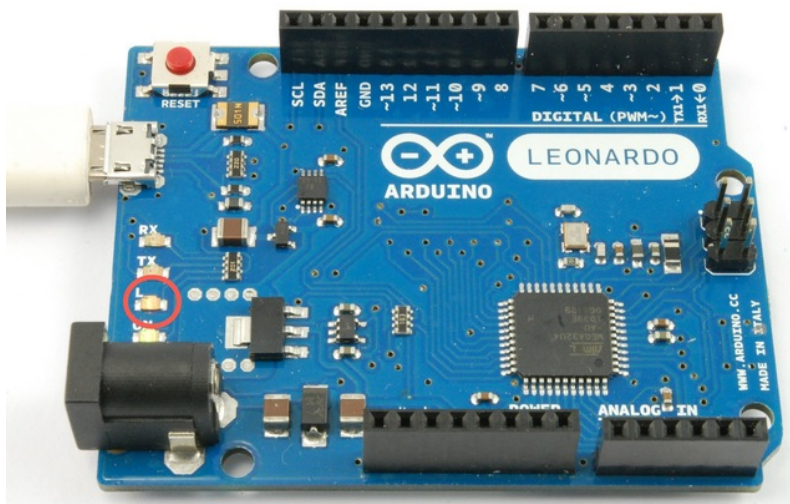
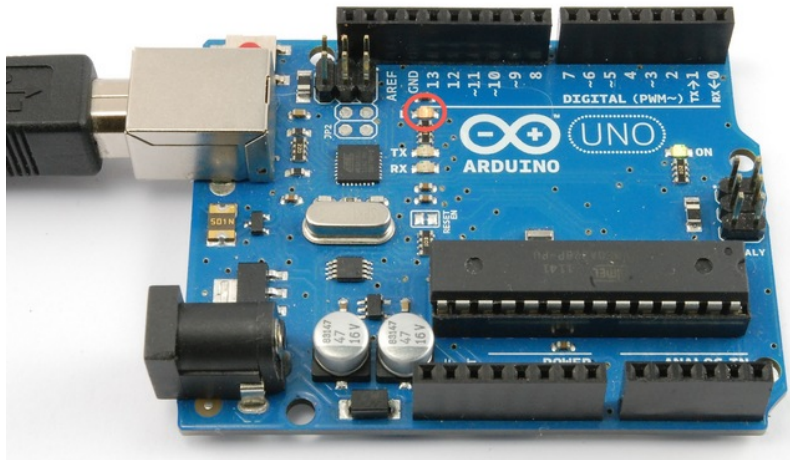
	Part	Qty
 <p>The image shows an Arduino Uno R3 board, which is a popular open-source microcontroller board. It features a blue PCB with a USB Type-B port, a DC power jack, a reset button, and a large ATmega328P microcontroller chip. The board is labeled with 'ARDUINO UNO' and 'MADE IN ITALY'.</p>	<p>Arduino Uno R3</p>	<p>1</p>
 <p>The image shows a USB lead cable, specifically a type A to B connector. It has a standard USB Type-A connector on one end and a USB Type-B connector on the other. The cable is black and appears to be made of braided shielding.</p>	<p>USB Lead - type A to B</p>	<p>1</p>

The 'L' LED

The Arduino has rows of connectors along both sides that are used to connect to electronic devices and plug-in 'shields' that allow the Arduino to do more.

However, the Arduino also has a single LED that you can control from your sketches. This LED is built onto the Arduino board and is often referred to as the 'L' LED as this is how it is labelled on the board.

The position of this LED is circled in red on the pictures of the Arduino Uno and Leonardo below.



Loading the 'Blink' Example

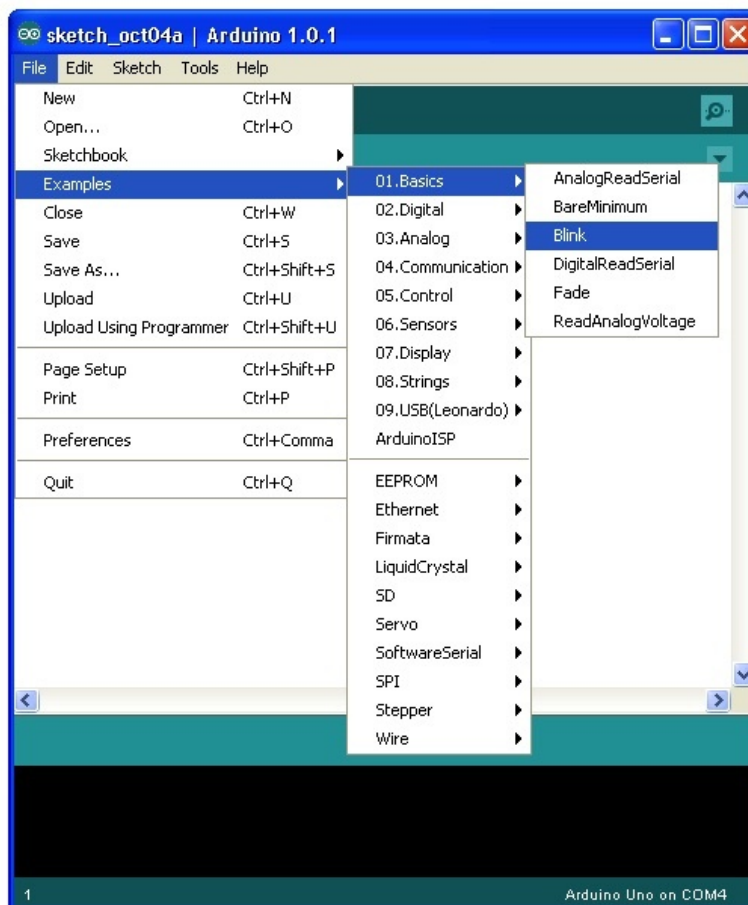
You may find that your Arduino board's 'L' LED already blinks when you connect it to a USB plug. This is because Arduino boards are generally shipped with the 'Blink' sketch pre-installed.

In this lesson, we will reprogram the Arduino with our own Blink sketch and then change the rate at which it blinks.

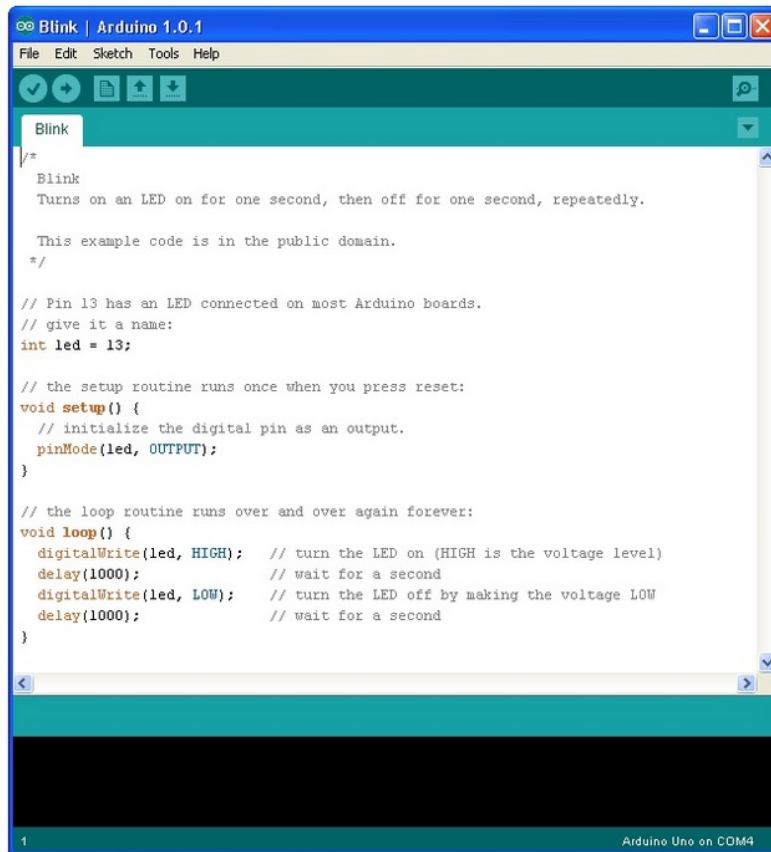
In Lesson 0, you setup your Arduino IDE and made sure that you could find the right serial port for it to connect to your Arduino board. The time has now come to put that connection to the test and program your Arduino board.

The Arduino IDE includes a large collection of example sketches that you can load up and use. This includes an example sketch for making the 'L' LED blink.

Load the 'Blink' sketch that you will find in the IDE's menu system under File → Examples → 01.Basics



When the sketch window opens, enlarge it so that you can see the whole of the sketch in the window.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and running. The main text area contains the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

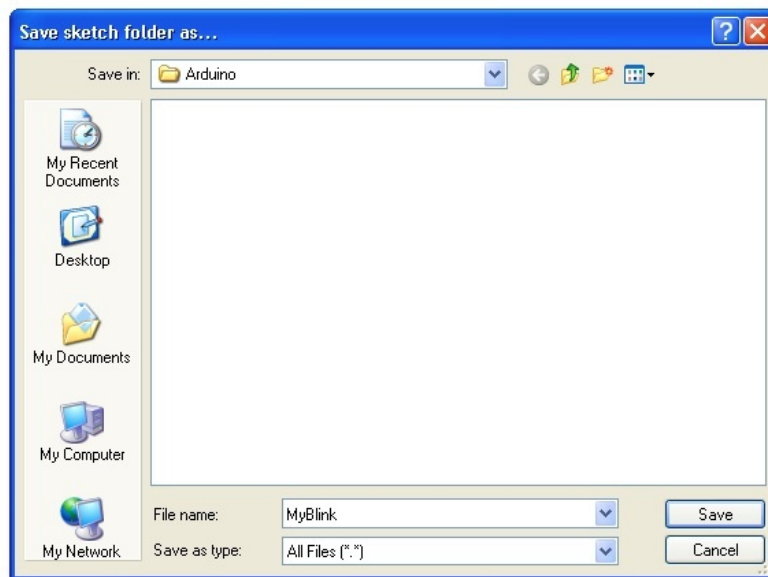
The status bar at the bottom indicates "1" on the left and "Arduino Uno on COM4" on the right.

Saving a Copy of 'Blink'

The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an Arduino board, but if you change them, you cannot save them as the same file.

We are going to change this sketch, so, the first thing you need to do is save your own copy that you can change however you like.

From the File menu on the Arduino IDE select the option 'Save As..' and then save the sketch with the name 'MyBlink'.



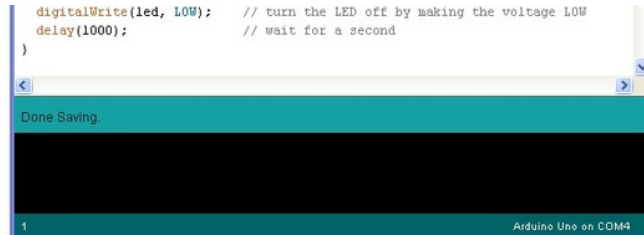
You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File → Sketchbook menu option.



Uploading Blink to the Board

Attach your Arduino board to your computer with the USB cable and check that the 'Board Type' and 'Serial Port' are set correctly. You may need to refer back to Lesson 0.

The Arduino IDE will show you the current settings for board at the bottom of the window.



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

```

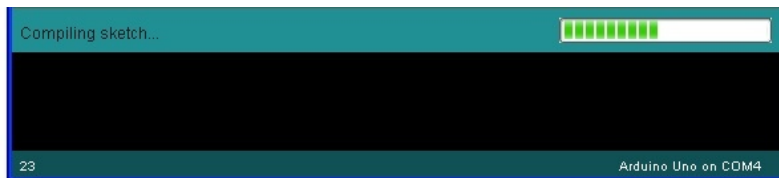
Done Saving.

1 Arduino Uno on COM4

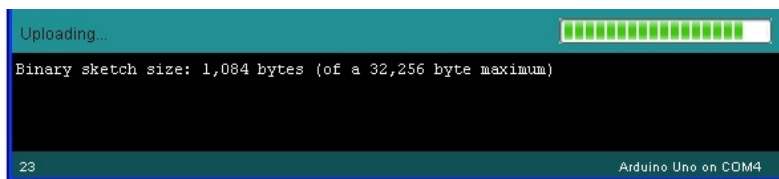
Click on the 'Upload' button. The second button from the left on the toolbar.



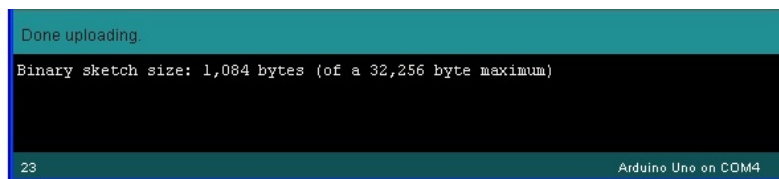
If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first it will say 'Compiling Sketch..'. This converts the sketch into a format suitable for uploading to the board.



Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Finally, the status will change to 'Done'.



The other message tells us that the sketch is using 1,084 bytes of the 32,256 bytes available. After the 'Compiling Sketch..' stage you could get the following error message:



```
Serial port 'COM4' not found. Did you select the right one from the Tools > Serial Port menu?  
at processing.app.SerialPort.exportApplet(SerialPort.java:1634)  
at processing.app.Sketch.exportApplet(Sketch.java:1634)  
at processing.app.Editor$DefaultExportHandler.run(Editor.java:2346)  
at java.lang.Thread.run(Thread.java:619)
```

23 Arduino Uno on COM4

The clue is at the top here, it probably means that your board is not connected at all, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.

If you get this, go back to Lesson 0 and check your installation.

Once the upload has completed, the board should restart and start blinking.

How 'Blink' Works

Here is the code for the Blink sketch.

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
*/  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

The first thing to note is that quite a lot of this sketch is what is called 'comments'. Comments are not actual program instructions, they are just comments about how the program works. They are there for our benefit, so that there is some explanation to accompany the sketch.

Everything between `/*` and `*/` at the top of the sketch is a block comment, that explains what the sketch is for.

There are also single line comments that start with `//` and everything up until the end of the line counts as being a comment.

The first actual line of code is:

```
int led = 13;
```

As the comment above explains, this is giving a name to the pin that the LED is attached to. This is 13 on most Arduinos, including the Uno and Leonardo.

Next, we have the 'setup' function. Again, as the comment says, this is run when the reset button is pressed. It is also run whenever the board resets for any reason, such as power first being applied to it, or after a sketch has been uploaded.

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

Every Arduino sketch must have a 'setup' function, and the part of it where you might want to add instructions of your own is between the { and the }.

In this case, there is just one command there, which, as the comment states tells the Arduino board that we are going to use the LED pin as an output.

It is also mandatory for a sketch to have a 'loop' function. Unlike the 'setup' function that only runs once, after a reset, the 'loop' function will, after it has finished running its commands, immediately start again.

```
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

Inside the loop function, the commands first of all turn the LED pin on (HIGH), then 'delay' for 1000 milliseconds (1 second), then turn the LED pin off and pause for another second.

Blinking Faster

You are now going to make your LED blink faster. As you might have guessed, the key to this lies in changing the parameter in () for the 'delay' command.

```
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

This delay period is in milliseconds, and so if you want the LED to blink twice as fast, change the value of 1000 to 500. This would then pause for half a second each delay rather than a whole second.

Upload the sketch again and you should see the LED start to flash more quickly.

Other Things to Do

Try changing the delay again to make the blinking even faster using 100 as the parameter to 'delay'.

About the Author.

Simon Monk is author of a number of books relating to Open Source Hardware. The following books written by Simon are available from Adafruit: [Programming Arduino](http://adafru.it/1019) (<http://adafru.it/1019>), [30 Arduino Projects for the Evil Genius](http://adafru.it/868) (<http://adafru.it/868>) and [Programming the Raspberry Pi](http://adafru.it/aM5) (<http://adafru.it/aM5>).