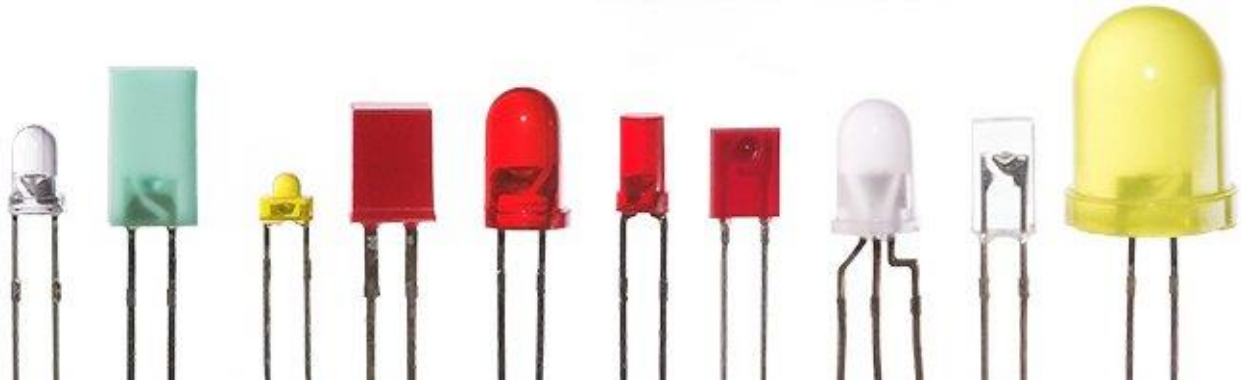


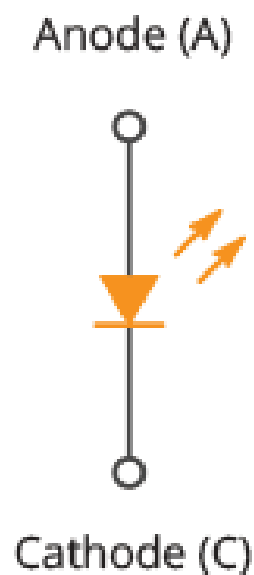
LIGHT EMITTING DIODE (LED)



LEDs are everywhere – in our phones, in our cars, and even in our homes. Whenever an electronic device lights up, there is a good chance that an LED is behind it.

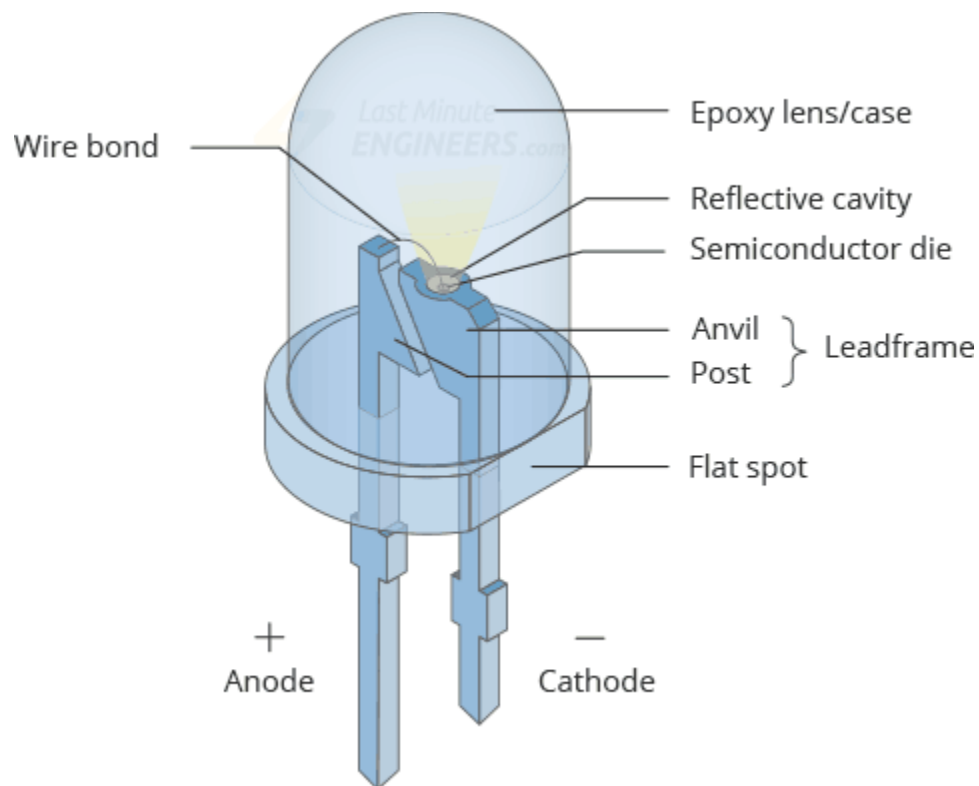
LEDs are like tiny lightbulbs. Low energy consumption, small size, rapid switching and long lifespan makes them ideal for mobile devices and other low-power applications.

LED stands for Light Emitting Diode. They are a special type of diode that convert electrical energy into light. They have very similar electrical characteristics to a normal PN junction diode. That's why the symbol of LED is similar to the normal PN junction diode except that it contains arrows pointing away from the diode indicating that light is being emitted by the diode.



LED CONSTRUCTION

LEDs are so common, they come in a huge variety of shapes, sizes and colors. The LEDs you are most likely to use are the standard through hole LEDs with two legs. Following figure shows the parts of it.



The construction of an LED is very different from an ordinary diode. The PN junction of an LED is surrounded by a transparent, rigid plastic epoxy resin shell.

The shell is constructed in such a way that photons of light emitted by the junction are focused upward through the domed top of the LED, which itself acts like a lens. This is why the emitted light appears brightest on top of the LED.

Just as in an ordinary diode, the positive side of the LED is called the Anode, while the negative side of the LED is called the Cathode. The cathode is usually indicated by having a shorter lead than the anode. Not only this, the outside of the plastic case typically has a flat spot or notch which can also indicate the cathode side of the LED.

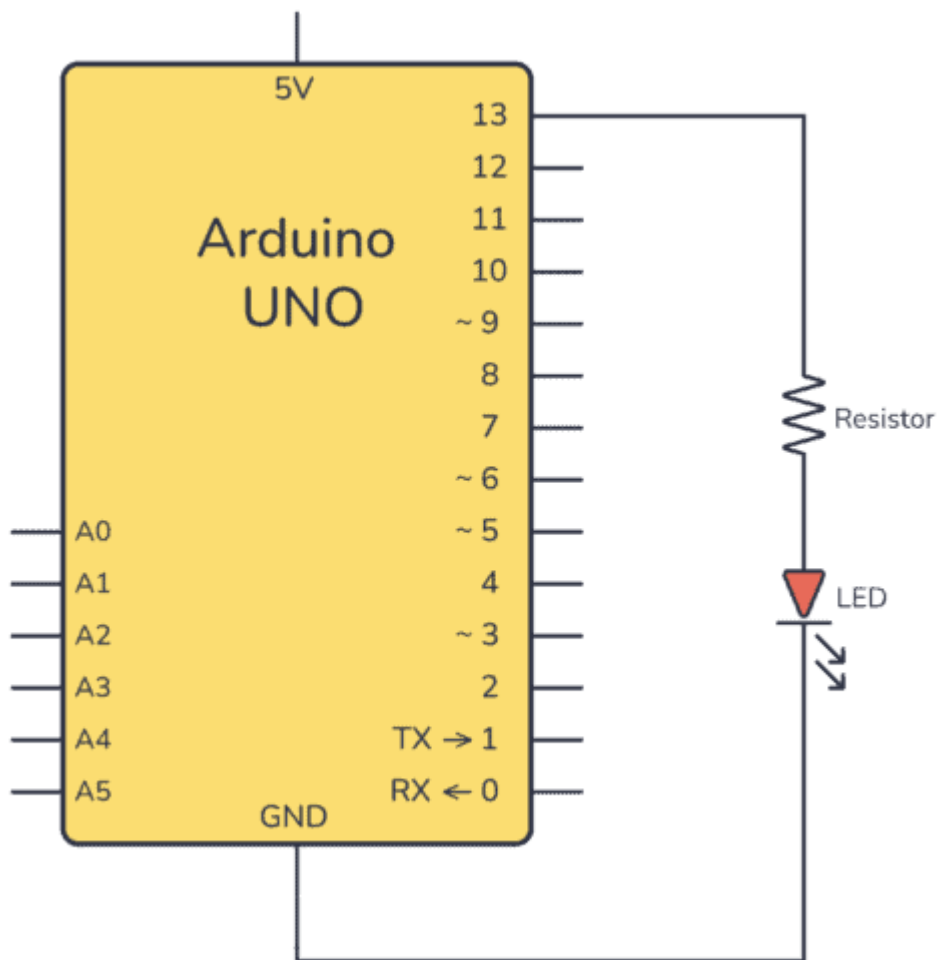
The Arduino blink LED circuit is a simple circuit that works great for starting to learn Arduino. Both the code and the connections are straightforward so that you can understand it with little to no background.

In this quickstart guide, you'll learn how to connect an LED to an Arduino board and make it blink.

PARTS NEEDED

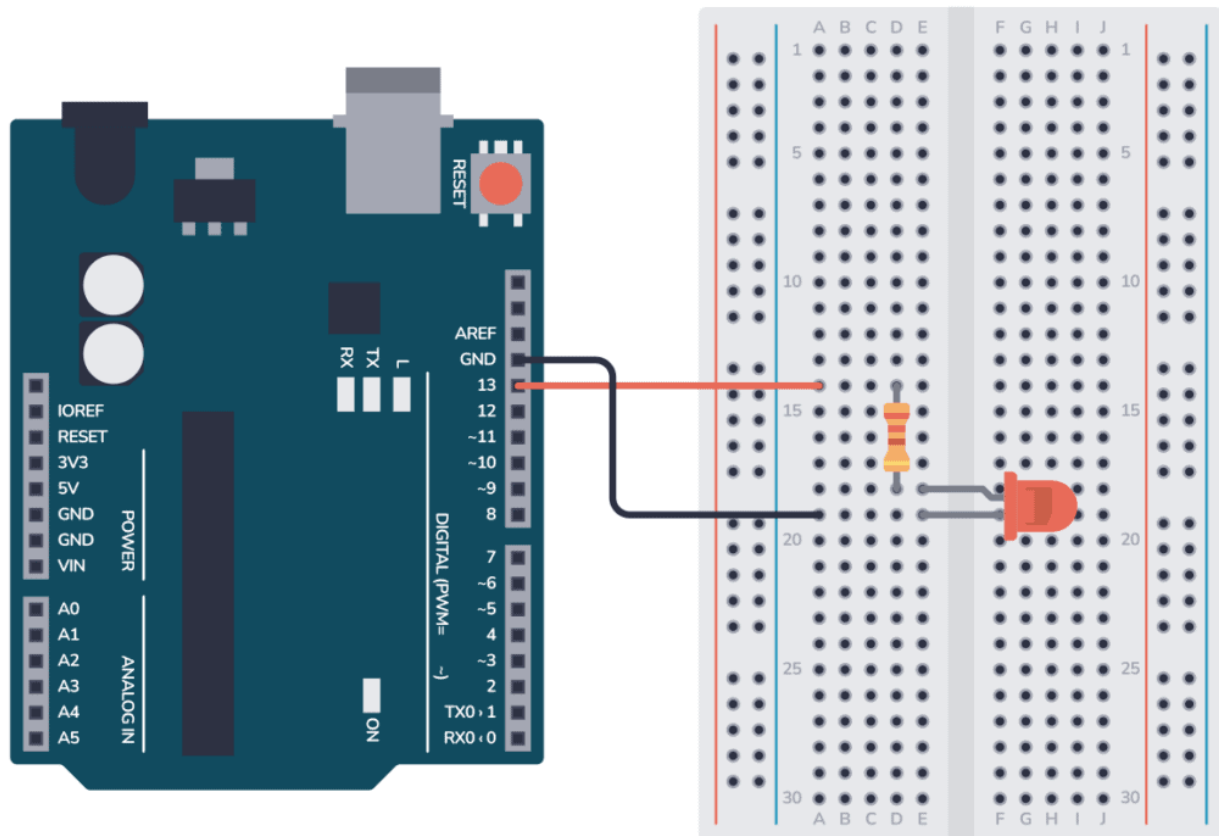
- Arduino Uno
- Breadboard (and some breadboard wires)
- Light-Emitting Diode (LED) (Most LEDs will work)
- Resistor (220 Ω)
- Arduino Blink LED Circuit

To connect an LED to an Arduino, you need a resistor in series with the LED. This is to limit how much current the LED pulls out of the Arduino pin. The value isn't crucial but should be between 220 Ω and 1000 Ω .



CONNECTING ON A BREADBOARD

Here's how you can connect the LED and the resistor to the Arduino by using a breadboard and a couple of cables:



ARDUINO BLINK LED CODE

All Arduino code is structured around the two main functions `setup()` and `loop()`.

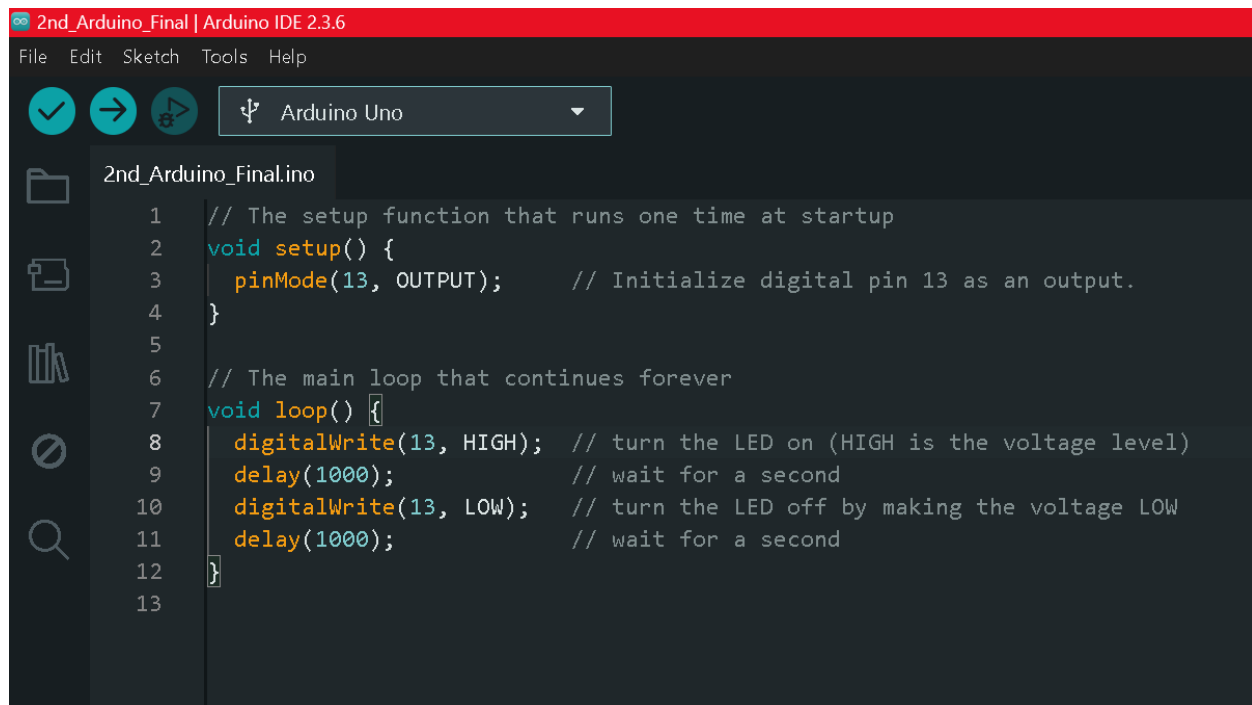
The `setup()` function runs only once when the Arduino board starts up. It is used for initializing variables, pins, and other settings.

The `loop()` function runs repeatedly after the `setup()` function has been executed. Whatever code is inside this function will be executed over and over again in an endless loop until the Arduino is powered off or reset.

Inside `setup()`, you need to configure pin 13 as an output.

Inside `loop()`, you need to set pin 13 HIGH, wait for a second, turn it low, then wait for another second.

Check out the complete code:



```
2nd_Arduino_Final | Arduino IDE 2.3.6
File Edit Sketch Tools Help

[Checkmark] [Next] [Previous] [USB] Arduino Uno

2nd_Arduino_Final.ino
1 // The setup function that runs one time at startup
2 void setup() {
3     pinMode(13, OUTPUT);    // Initialize digital pin 13 as an output.
4 }
5
6 // The main loop that continues forever
7 void loop() {
8     digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
9     delay(1000);            // wait for a second
10    digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
11    delay(1000);            // wait for a second
12 }
13
```

HOW THE CODE WORKS

Inside the `setup()` function there is only one line: `pinMode(13, OUTPUT);` This line sets pin 13 as an output so that we can use it to turn the LED on or off.

Inside the `loop()` function there are four lines:

- `digitalWrite(13, HIGH);` This line turns on the LED connected to pin 13. HIGH sets the voltage of the pin to the logic HIGH level (usually 5V on most Arduino boards), which turns on the LED.
- `delay(1000);` This line adds a delay of 1000 milliseconds (1 second). It means the LED will remain on for one second before moving on to the next line of code.
- `digitalWrite(13, LOW);` This line turns off the LED by setting the voltage level of pin 13 to LOW (0V).
- `delay(1000);` This line adds another 1-second delay. So after the LED is turned off, the program waits for one second.

After this, the program goes back to the beginning of the `loop()` function where it turns the LED on again, and the process repeats.

This code results in a LED (connected to pin 13) blinking on and off repeatedly, with each state (on and off) lasting for one second.