Home / Programming / Built-in Examples / Blink

Blink

Turn an LED on and off every second.

Last revision • 10/03/2024

This example shows the simplest thing you can do with an Arduino to see physical output: it blinks the on-board LED.

Hardware Required

- Arduino Board

optional

- LED
- 220 ohm resistor

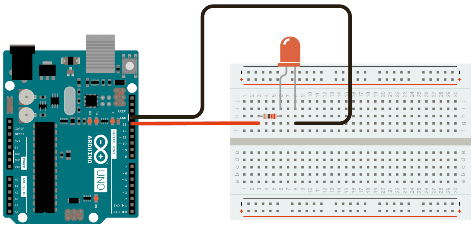
Circuit

This example uses the built-in LED that most Arduino boards have. This LED is connected to a digital pin and its number may vary from board type to board type. To make your life easier, we have a constant that is specified in every board descriptor file. This constant is *LED_BUILTIN* and allows you to control the built-in LED easily. Here is the correspondence between the constant and the digital pin.

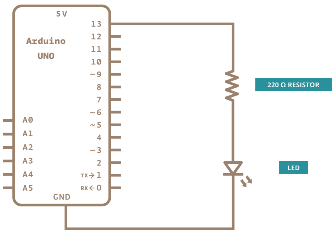
- D13 - 101
- D13 - Due
- D1 - Gemma
- D13 - Intel Edison
- D13 - Intel Galileo Gen2
- D13 - Leonardo and Micro
- D13 - LilyPad
- D13 - LilyPad USB
- D13 - MEGA2560
- D13 - Mini
- D6 - MKR1000
- D13 - Nano
- D13 - Pro
- D13 - Pro Mini
- D13 - UNO
- D13 - Yún
- D13 - Zero

If you want to light an external LED with this sketch, you need to build this circuit, where you connect one end of the resistor to the digital pin correspondent to the *LED_BUILTIN* constant. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the GND. In the diagram below we show an UNO board that has D13 as the LED_BUILTIN value.

The resistor is essential for safe operation as it limits the current flowing through the LED, preventing damage to both the LED and the Arduino's output pin. You can choose the resistor value based on the desired current using Ohm's Law (V = IR) where V is the voltage of your board (5V or 3.3V) minus the forward voltage for the LED you are using (typical for red would be 1.8 to 2.2 volts). In this case, using a 220-ohm resistor with an Arduino UNO R3 (a 5V board) limits the current to a safe level for both the LED and the Arduino pin. Adjusting the resistor value allows you to control the LED's brightness while ensuring safe operation. For 5V boards you can expect the LED to be visible to a resistor value of up to 1K Ohm.



Schematic



Code

After you build the circuit plug your Arduino board into your computer, start the Arduino Software (IDE) and enter the code below. You may also load it from the menu File/Examples/01.Basics/Blink . The first thing you do is to initialize LED_BUILTIN pin as an output pin with the line

`pinMode(LED_BUILTIN, OUTPUT);`

In the main loop, you turn the LED on with the line:

`digitalWrite(LED_BUILTIN, HIGH);`

This supplies 5 volts to the LED anode. That creates a voltage difference across the pins of the LED, and lights it up. Then you turn it off with the line:

`digitalWrite(LED_BUILTIN, LOW);`

That takes the LED_BUILTIN pin back to 0 volts, and turns the LED off. In between the on and the off, you want enough time for a person to see the change, so the `delay()` commands tell the board to do nothing for 1000 milliseconds, or one second. When you use the `delay()` command, nothing else happens for that amount of time. Once you've understood the basic examples, check out the BlinkWithoutDelay example to learn how to create a delay while doing other things.

Once you've understood this example, check out the DigitalReadSerial example to learn how read a switch connected to the board.

```
/*
  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://docs.arduino.cc/hardware/

  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.

  https://docs.arduino.cc/built-in-examples/basics/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
}

```

See Also

Learn more

You can find more basic tutorials in the built-in examples section.

You can also explore the language reference, a detailed collection of the Arduino programming language.

Last revision 2015/07/28 by SM