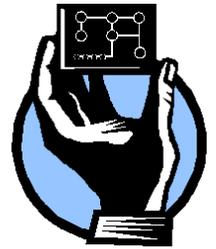


Arduino Blink Challenge



Student Resource: Circuit Basics

◆ Simple Circuit

A simple circuit consists of three minimum elements that are required to complete a functioning electric circuit: a source of electricity (battery), a path or conductor on which electricity flows (wire) and an electrical resistor (lamp) which is any device that requires electricity to operate. The illustration below shows a simple circuit containing, one battery, two wires, and a bulb. The flow of electricity is caused by excess electrons on the negative end of the battery being attracted to flow toward the positive end, or terminal, of the battery. When the simple circuit is complete, electrons flow from the negative terminal through the wire conductor, then through the bulb (lighting it up), and finally back to the positive terminal - in a continual flow.

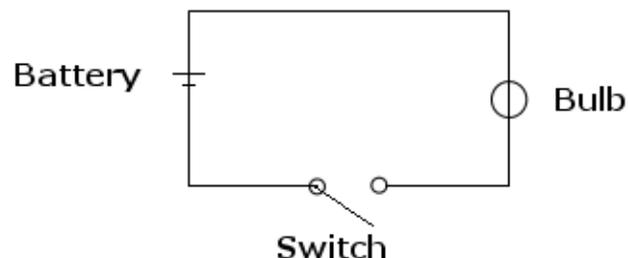
Simple Circuit



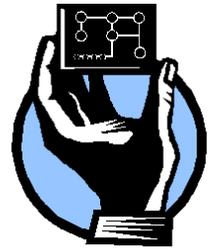
◆ Schematic Diagram of a Simple Circuit

The following is a schematic diagram of the simple circuit showing the electronic symbols for the battery, switch, and bulb.

Schematic Diagram of a Simple Circuit



Arduino Blink Challenge



Student Worksheet:

◆ Engineering Teamwork and Planning

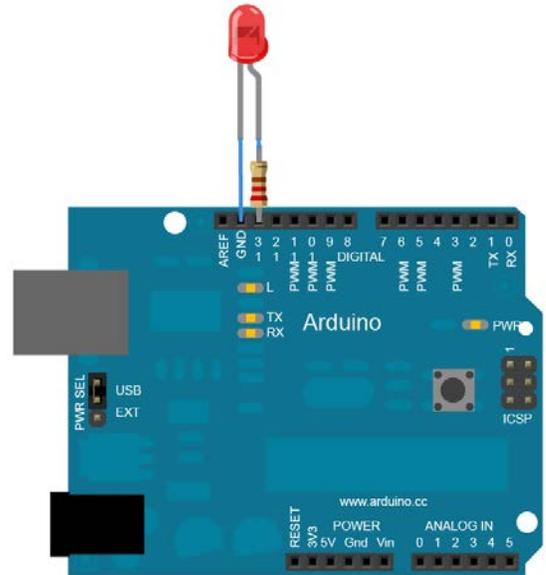
You are part of a team of engineers given the challenge of using Arduino to make a light blink on for 5 seconds and off for 2. The instructions and code below will help you see how to program the Arduino to blink for one second on and one second off.

◆ Research Phase

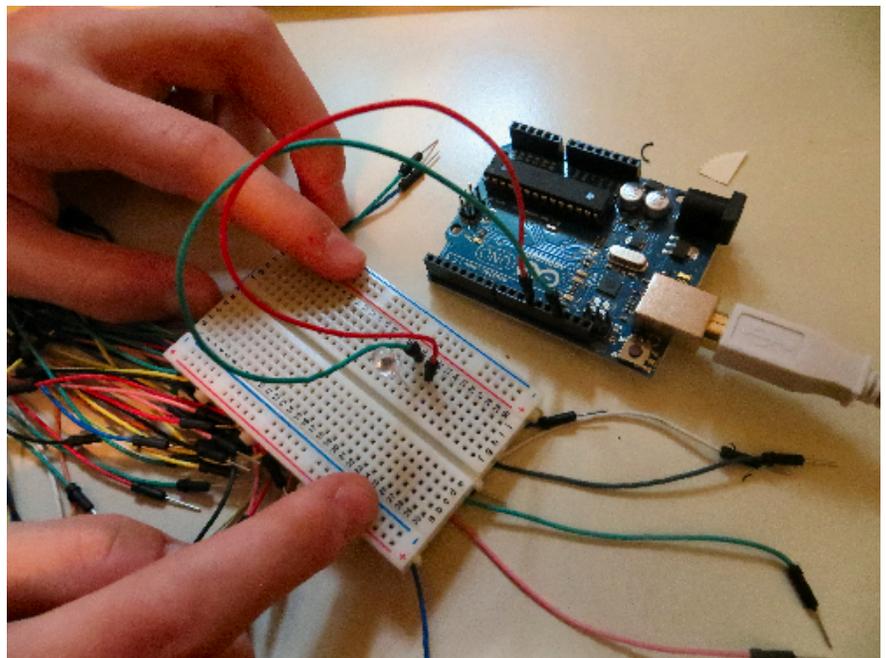
Read the materials provided to you by your teacher. If you have access to the internet ahead of the activity, explore the Arduino website and become familiar with the logic of the programming code.

◆ Building the Circuit

To build the circuit, attach a 220-ohm resistor to pin 13. Then attach the long leg of an LED (the positive leg, called the anode) to the resistor. Attach the short leg (the negative leg, called the cathode) to ground. Then plug your Arduino board into your computer, start the Arduino program, and enter the code below. Note: Most Arduino boards already have an LED attached to pin 13 on the board itself. If you run this example with no hardware attached, you should see that LED blink.

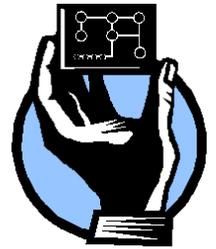


You may also set up your blinking light using a breadboard as in the image above. In this case, you'll use the connectors to link the Arduino to the breadboard and make a complete circuit by having another connector going back to Arduino. You would need to include a separate light on the breadboard too. You can do this lesson either way...with or without the breadboard -- it just depends on what materials you have provided to you and if your team wishes to gain some experience on the breadboard.



Note: Some content and images on this page are derived from Arduino.cc via their Arduino getting started guide (<http://arduino.cc/en/Guide/HomePage>)

Arduino Blink Challenge



Student Worksheet: (continued)

◆ Schematic

The illustration or schematic on the right shows how the circuit for the light works in the Arduino.

◆ Code

In the program below, the first thing you do is to initialize pin 13 as an output pin with the line
`pinMode(13, OUTPUT);`

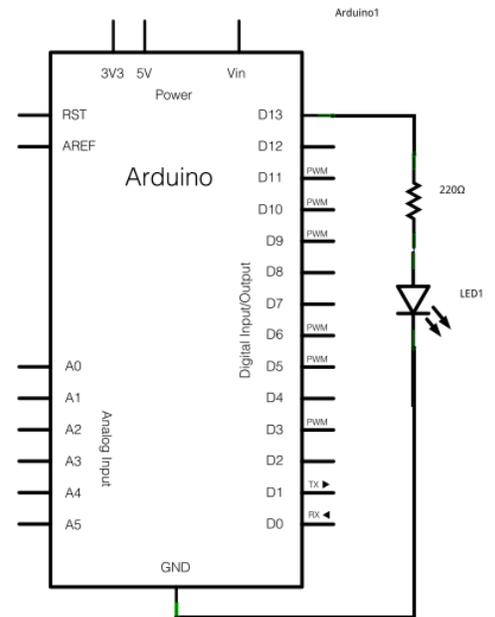
In the main loop, you turn the LED on with the line:
`digitalWrite(13, HIGH);`

This supplies 5 volts to pin 13. That creates a voltage difference across the pins of the LED, and lights it up.

Then you turn it off with the line:

```
digitalWrite(13, LOW);
```

That takes pin 13 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 Arduino to do nothing for 1000 milliseconds, or one second. When you use the `delay()` command, nothing else happens for that amount of time.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second,
  repeatedly.

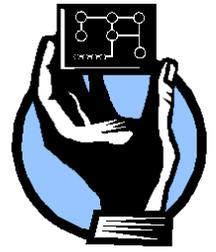
  This example code is in the public domain.
  */

// 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 Blink Challenge



Student Worksheet:

◆ Reflection

Complete the reflection questions below:

1. What challenges did you have, if any, to programming the Arduino to blink on and off at 1 second intervals? How did you resolve any challenges you encountered?

2. Were you able to adjust the code to change the intervals to 5 seconds on and 2 seconds off?

3. Do you think that this activity was more rewarding to do as a team, or would you have preferred to work alone on it? Why?

4. What do you think about the Arduino? Was it a good way for you and your team to explore basic computer programming?

5. How complicated or different do you think code would be to provide instructions to a cell phone to play a particular ringtone?

6. Do you think that stoplights use computer programming to provide a pattern for lights changing at intersections? How do you think stoplights were coordinated prior to computer technology?