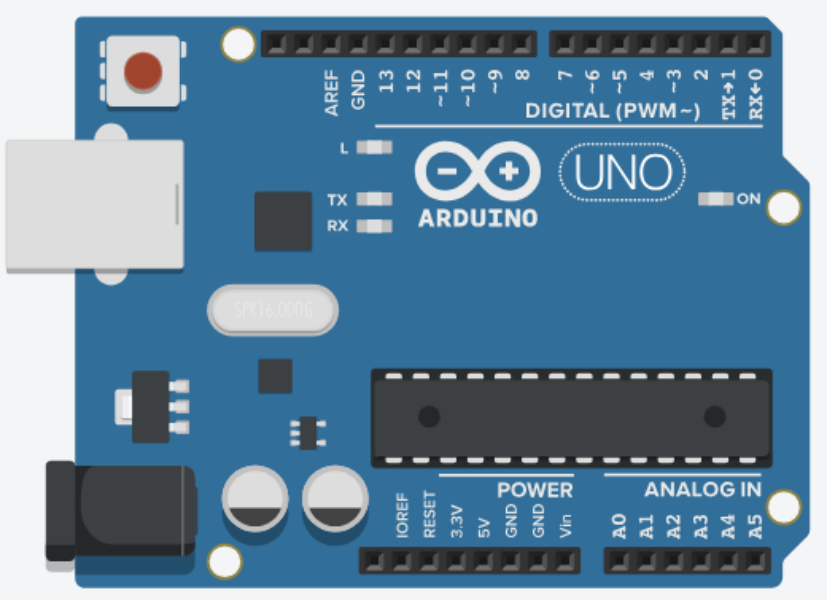
Tutorial 1: Digital Output

In this tutorial, we will learn how to light up a single LED using an Arduino microcontroller. This tutorial is designed for students who are new to Arduino and want to learn the basics of working with electronic components.

Hardware Required:

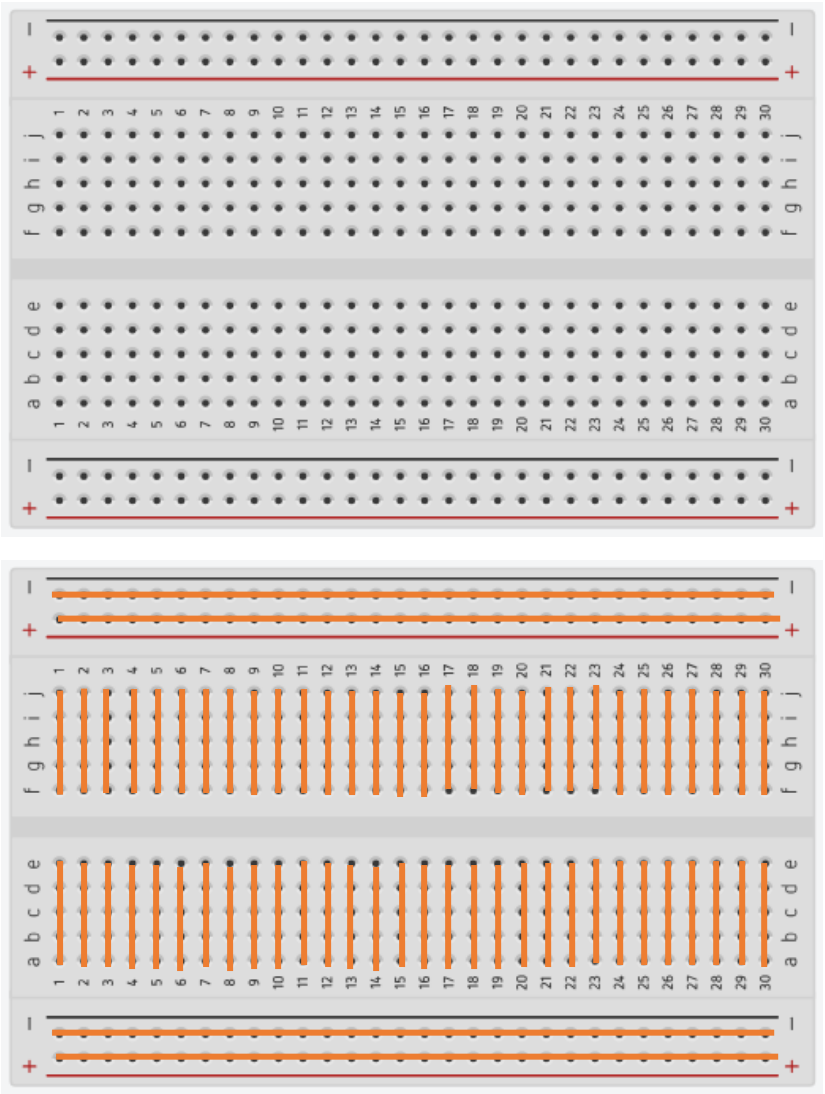
1. Arduino board (e.g., Arduino Uno)



Arduino UNO known as microcontrollers board. It contains 6 analogue inputs, 14 digital input/output pins (six of which can be used as PWM outputs), a USB port, a power jack, and a reset button. It comes with everything needed to support the microcontroller; to get started, just plug in a USB cable, an AC-to-DC adapter, or a battery.

General-Purpose Input Output (GPIO) is a digital pin of an IC. It can be used as input or output for interfacing devices. If we want to read the switch's state, sensor data, exp: then we need to configure it as input. And if we want to control the LED brightness, motor rotation, show text on display, exp: then we need to configure it as output.

2. Breadboard



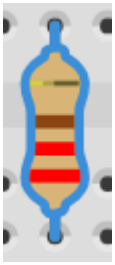
A breadboard is used to build and test circuits quickly before finalizing any circuit design. The holes on the top of the bread board are connected by metal strips that run underneath the board. The arrangement of the metal strips is illustrated above. Noted that the remaining holes are connected vertically, the top and bottom rows of holes are connected horizontally.

3. LED (Light Emitting Diode)



LEDs are diodes which are electronic devices that only allow current to go through them in one direction. This means that LEDs (and other diodes) have polarity which are positive (+) and negative (-) sides to them. For an LED to work it needs to be connected to a voltage source with the correct side. The voltage supply side of the diode is the positive (+) side, this is called the anode. The negative side is called the cathode.

4. Resistor (220 ohms)



A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor.

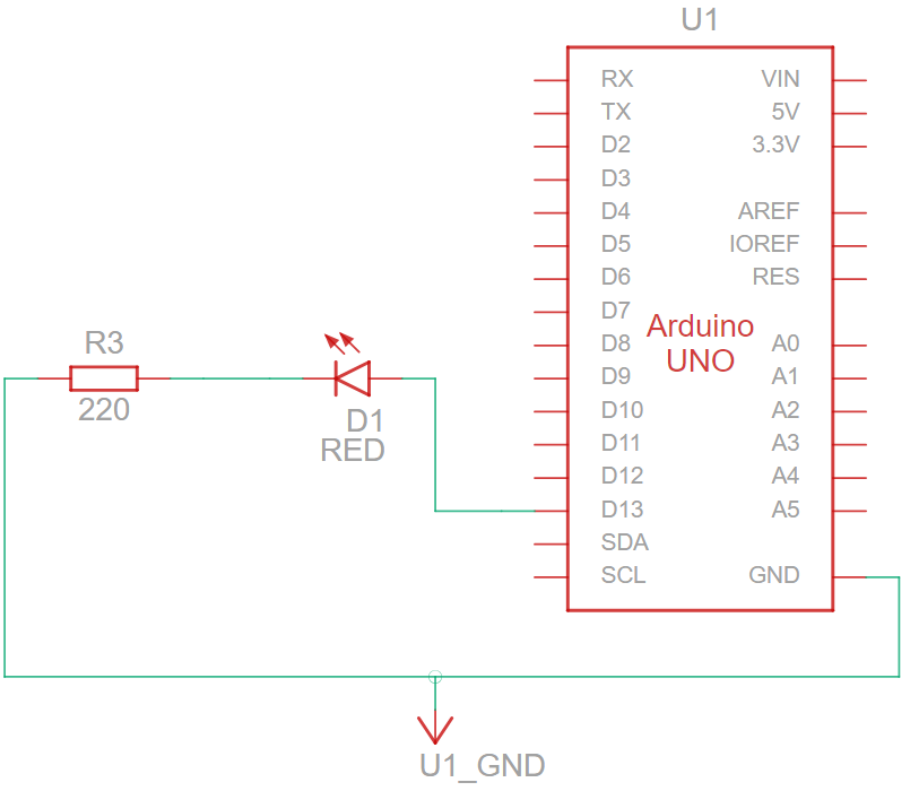
5. Jumper wires



Jumper wires are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering. You can use jumper wires to modify a circuit or diagnose problems in a circuit.

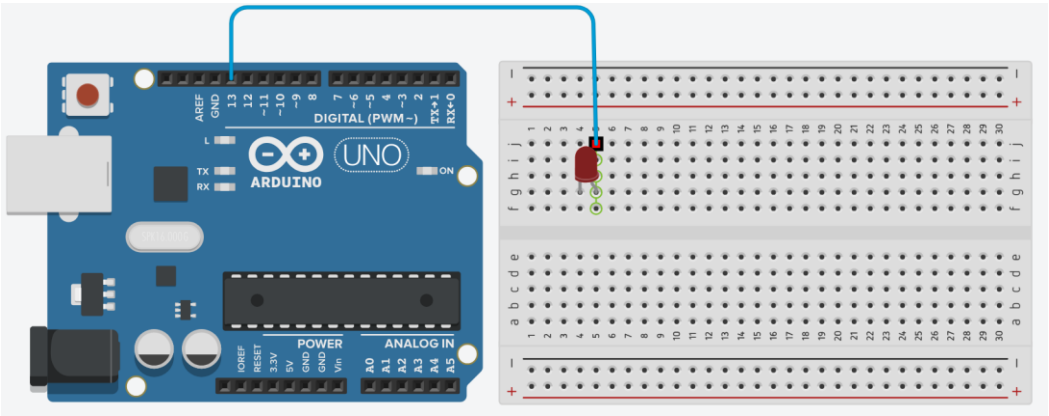
Step 1: Setting up the Circuit.

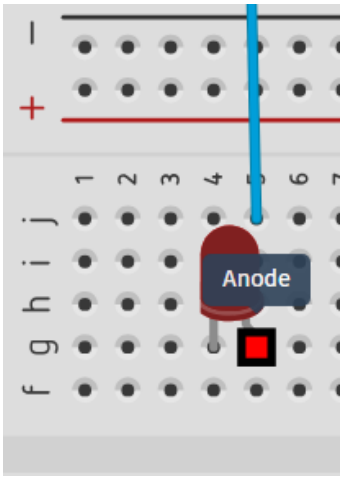
For this activity, we will recreate this circuit on a breadboard and Arduino UNO. We will go through step-by-step process of connecting the circuit for this activity only.



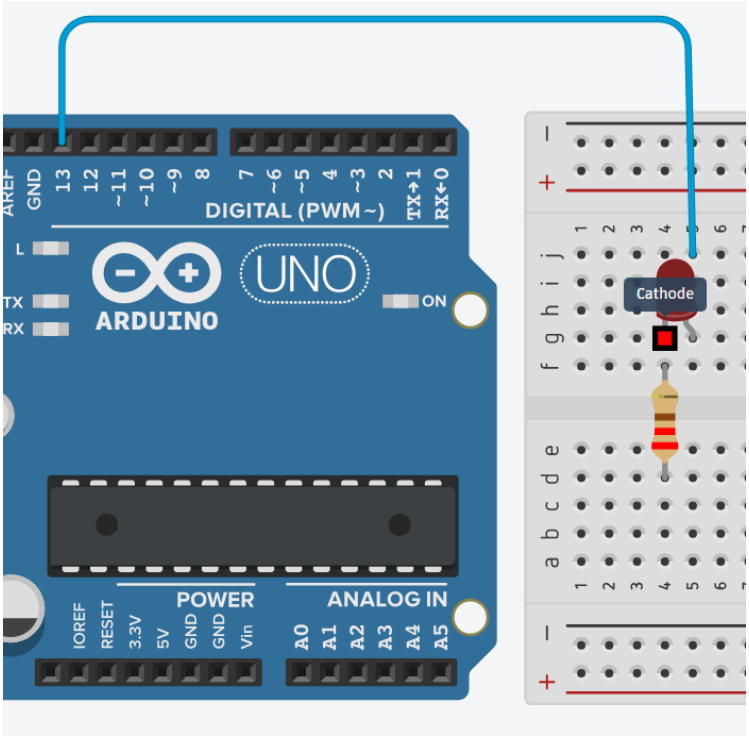
To begin, let's set up the circuit on the breadboard:

1. Connect the longer leg (anode) of the LED to digital pin 13 on the Arduino.

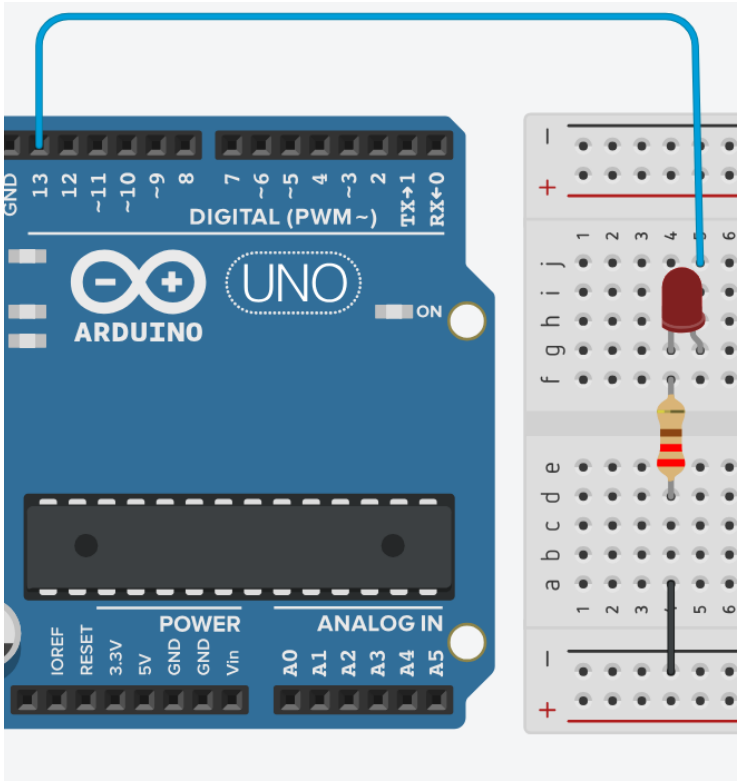



Anode

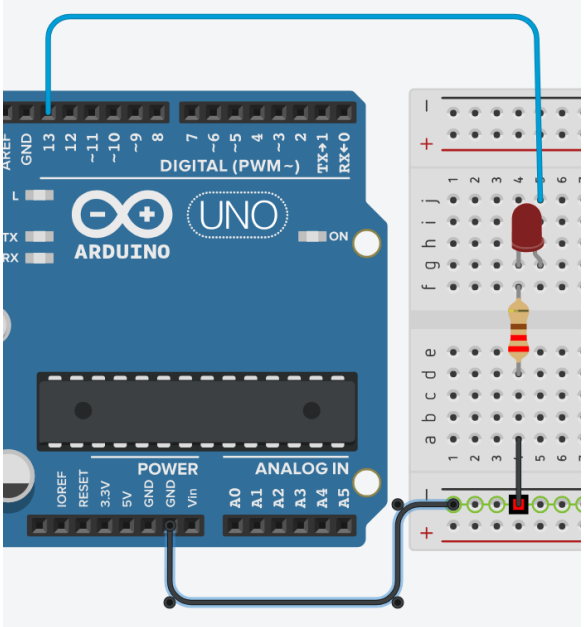
2. Connect one end of the resistor to the cathode of the LED.



3. Connect the other end of the resistor to the breadboard's ground rail (GND).



4. Connect the GND (ground) pin of the Arduino to the breadboard's ground rail (GND).



Step 2: Writing the Code

Now, let's write the Arduino code to control the LED:

1. Launch the Arduino IDE on your computer.
2. Select the appropriate board and port from the "Tools" menu.
3. We start by defining a constant variable named *RED_LED* before *void setup ( )* and assign it the value of 13.

```
// Define the pin connected to the LED
const int RED_LED= 13;

void setup() {
  // put your setup code here, to run once:

}
```

This represents the pin number to which the LED is connected. You can put any name that you like. However, it is best to put a name that is easily understood.

4. In the *setup( )* function, we set the *RED_LED* as an output using the *pinMode( )* function.

```
void setup() {
// put your setup code here, to run once:
  // Set the LED pin as an output
   pinMode(RED_LED, OUTPUT);
}
```

5. Inside *the loop( )* function, we turn on the LED by setting the RED_LED to HIGH using *digitalWrite( )*.

```
void loop() {
  // put your main code here, to run repeatedly:
  // Turn on the LED
 digitalWrite(RED_LED, HIGH);
}
```

The final code should look like this:

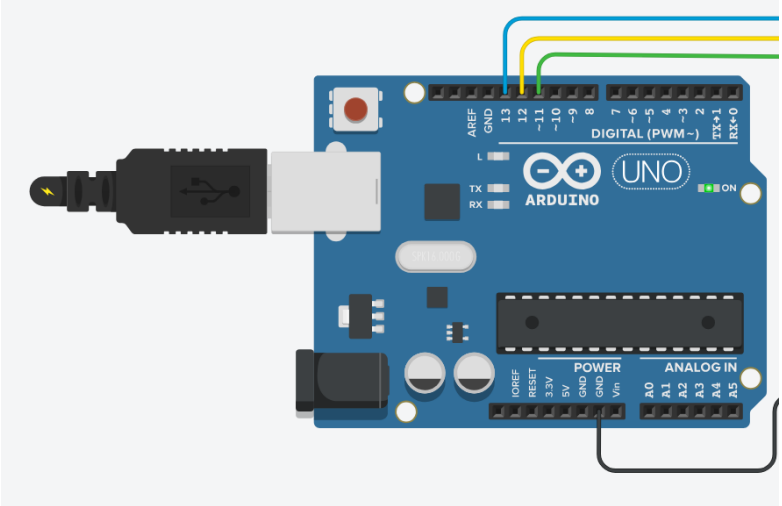
```
// Define the pin connected to the LED
const int RED_LED= 13;

void setup() {
// put your setup code here, to run once:
 // Set the LED pin as an output
  pinMode(RED_LED, OUTPUT);
}

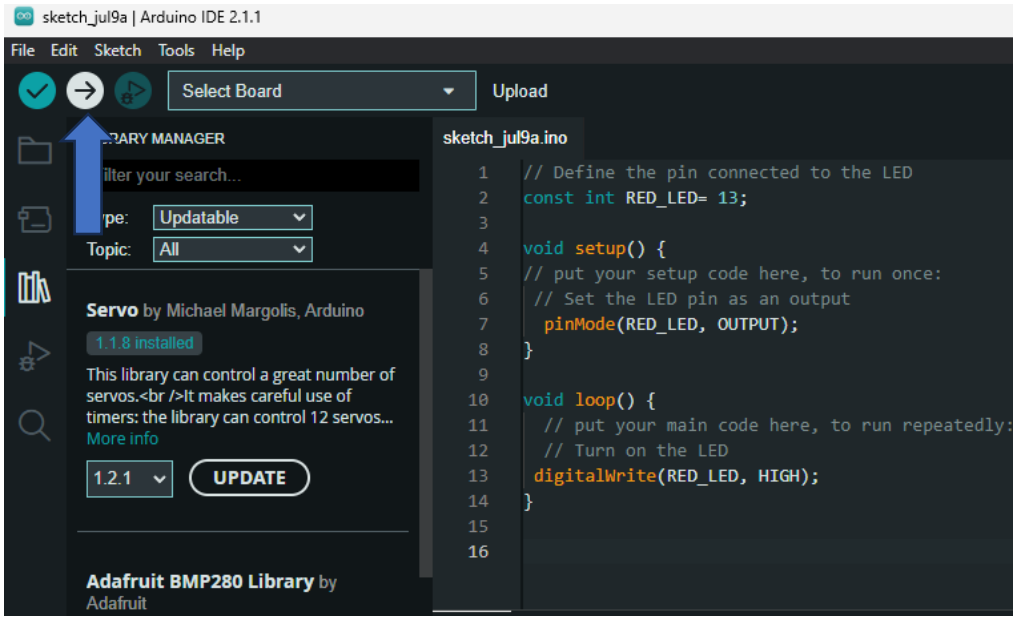
void loop() {
  // put your main code here, to run repeatedly:
  // Turn on the LED
 digitalWrite(RED_LED, HIGH);
}
```

Step 3: Uploading the Code

1. Connect the Arduino board to your computer using a USB cable.



2. Click on the "Upload" button in the Arduino IDE to compile and upload the code to the Arduino board.



3. Once the upload is complete, the Arduino board will execute the code and the LED will start lighting up.

Congratulations! You have successfully completed the tutorial on lighting up a single LED using Arduino. You learned how to set up the circuit, write the code, and upload it to the Arduino board. Experiment with different delays and modify the code to create various LED blinking patterns. This is just the beginning of your journey into the exciting world of Arduino and electronics!

Exercise

Try lighting up 3 different LEDs separately.

Note: For STEMbot, See the LEDs' label on the board to know which Arduino pin they are connected to

