IEEE WIE

Presents

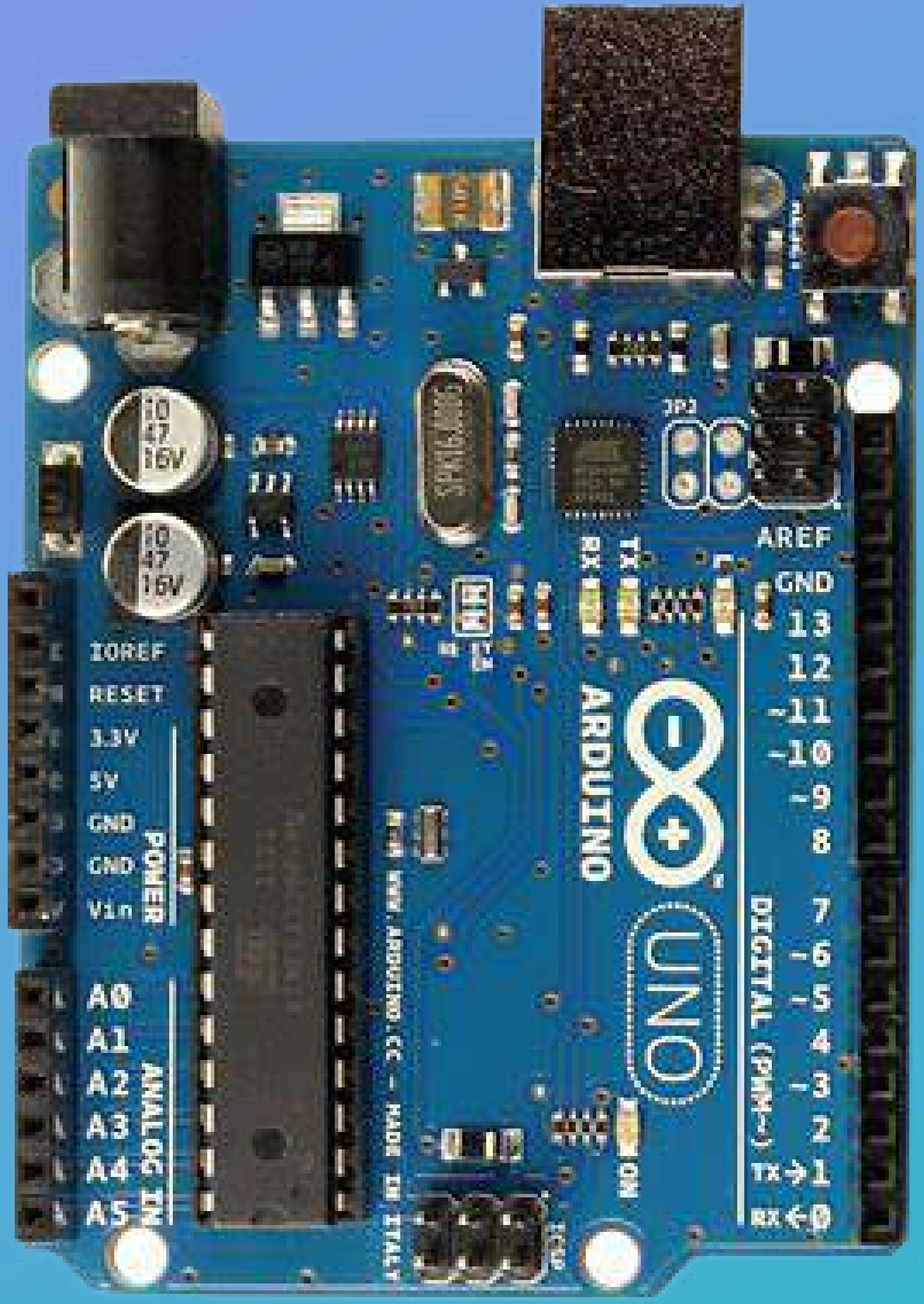
TECHNOVA 1.0

ARDUINO & SENSORS

Coordinators: Ananya Ghosh & Sadhana

Arduino- Introduction

Arduino is an **open-source electronics prototyping platform** based on flexible, easy-to use hardware and software. Arduino senses the environment by reading data from various buttons, components and sensors. We will focus on **UNO model**. You can tell your board what to do by sending a set of instructions to the **microcontroller** on the board. To do so you use the **Arduino programming language** (based on Wiring), and the **Arduino Software** (IDE), based on Processing. Applications: **IoT applications, wearable, 3D printing, and embedded environments**.



Arduino UNO

POWERED BY

USB cable connected to the Computer

CRYSTAL OCSILLATOR

The crystal oscillator helps Arduino in dealing with time issues.

3.3 V

Supply 3.3 output volt

5.5 V

•Supply 5 output volt

GND- GROUND

Can be used to ground your circuit.

ANALOG PINS A0-A5

Read signal from an analog sensor and convert it into a digital value that can be read by the microprocessor.

MICROCONTROLLER

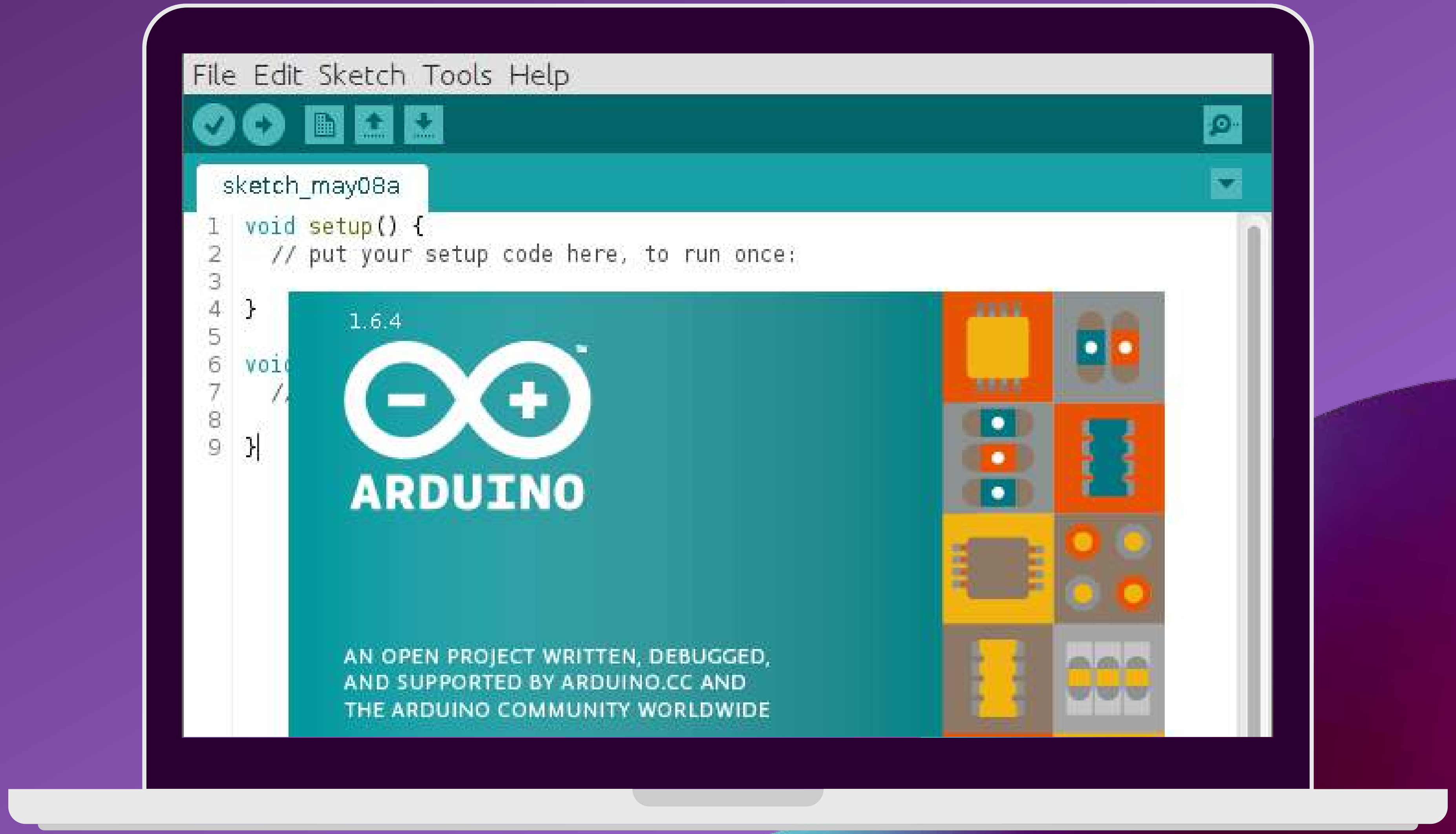
Brain of your board

DIGITAL PINS 0-13

The Arduino UNO board has 14 digital I/O pins

Working on IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.



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

}

voi
  /

}
```

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

Tinkercad

https://www.tinkercad.com

FEATURE 1

Tinkercad Circuits allows anyone to virtually create and program Arduino projects without the need for physical hardware.

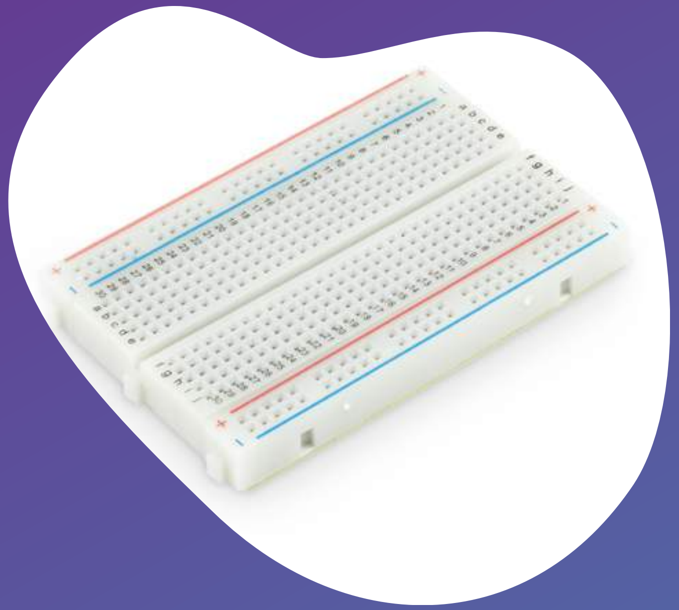
FEATURE 2

There's everything from LEDs to integrated circuits (ICs), and even a few instrument tools.

FEATURE 3

The programming area is a simplified integrated development environment (IDE) that makes programming the Arduino very straightforward.

IMPORTANT COMPONENTS

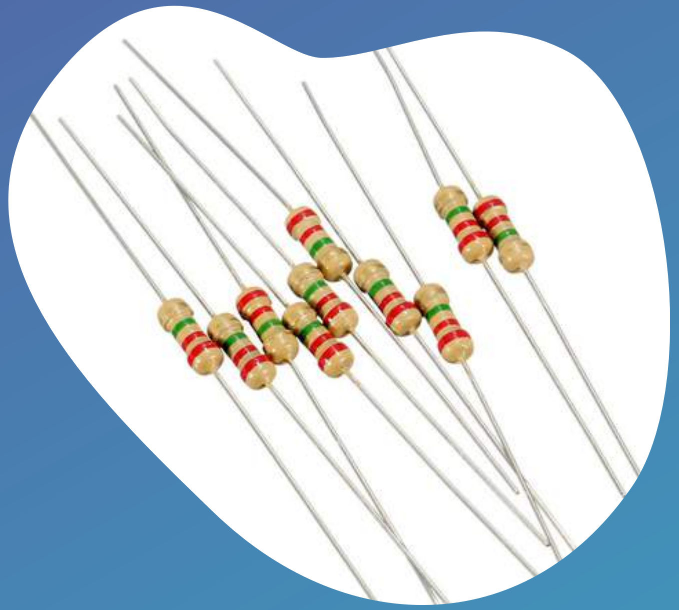


BREADBOARD

A breadboard, is a construction base for prototyping of electronics.

JUMPER WIRES

Jumper wires have connector pins at each end, allowing them to be used to connect two points to each other without soldering.



RESISTORS

Resistors to limit the amount of current going to certain components in the circuit

Code-Along Projects

- **Blinking LED:**

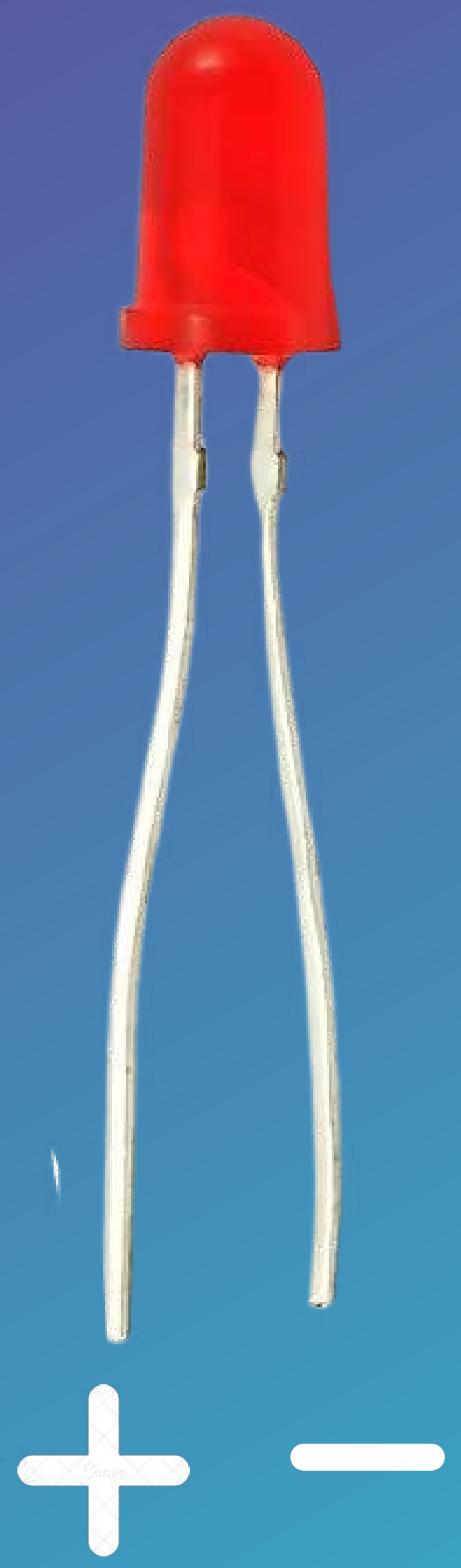
  Single LED Blinking

  Button controlled blinking

- **Automatic Switch on –off LDR**

Blinking Single LED

- LED stands for **Light Emitting Diode.**
- Connect the **long leg of the LED** (positive leg, called the anode) to a **220 Ohm resistor** and then to **digital pin 8**.
- Connect the **short leg** (negative leg, called the cathode) directly to **ground GND**.
- Turning the pin on will light up the LED, turning it off will turn the LED off.
- The resistor is necessary to protect the LED from too much current -- it will burn out without one.



Code

```
 //definition digital 8 pins as pin to control the LED
int ledPin=8;
// put your setup code here, to run once
//void setup(): This is run by the Arduino once every time it starts. This is where you can configure variables and anything your Arduino needs to run.
void setup()
{
 pinMode(ledPin,OUTPUT);
 //Set the digital 8 port mode, OUTPUT: Output mode
 //pinMode(8, OUTPUT): This tells the Arduino to use this pin as an output, without this line, the Arduino would not know what to do with each pin.
 //This only needs to be configured once per pin, and you only need to configure pins you're intending to use.
}
```

// put your main code here, to run repeatedly

//void loop(): Any code inside this loop is repeatedly run over and over again, until the the Arduino is turned off.

//This can make larger projects more complex, but it works amazingly well for simple projects.

void loop()

{

digitalWrite(ledPin,HIGH); //HIGH is set to about 5V PIN8 // turn LED on

//digitalWrite(8, HIGH): This is used to set the pin HIGH or LOW -- ON or OFF.

//Just like a light switch, when the pin is HIGH, the LED will be on.

//When the pin is LOW, the LED will be off. Inside the brackets, you need to specify some additional information for this to work correctly.

//Additional information is known as parameters or arguments.

//The first (7) is the pin number. If you have connected your LED to a different pin, for example, you would change this from seven to another number.

//The second parameter has to be HIGH or LOW, which specifies if the LED should be turned on or off.

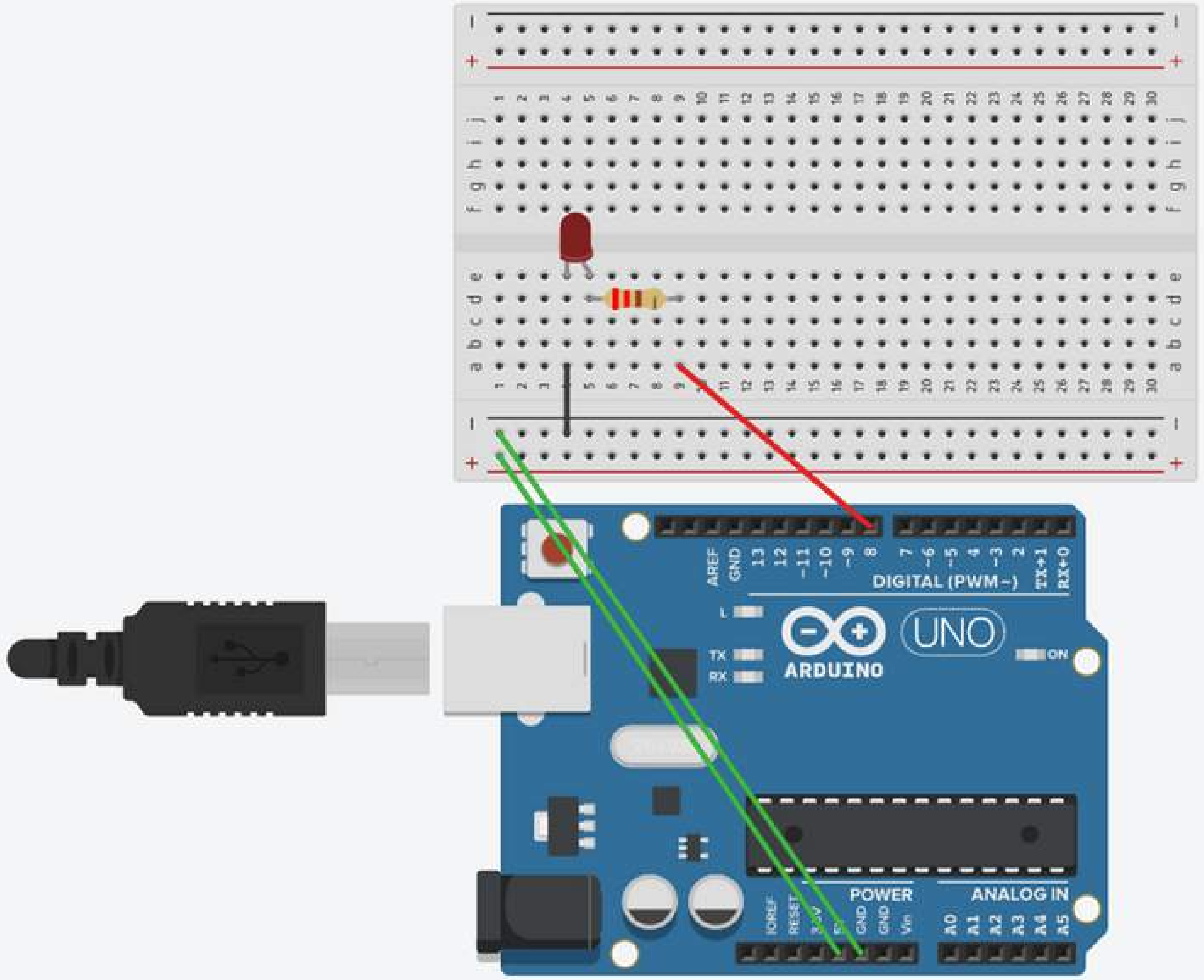
```
delay(1000); //Set the delay time, 1000 = 1S //wait 1 second
```

//delay(1000): The tells the Arduino to wait for a specified amount of time in milliseconds.
1000 milliseconds is equal to one second, so this will make the Arduino wait for once second.

```
 digitalWrite(ledPin,LOW); //LOW is set to about 5V PIN8 // turn LED off
 delay(1000); //Set the delay time, 1000 = 1S //wait 1 second
}
```

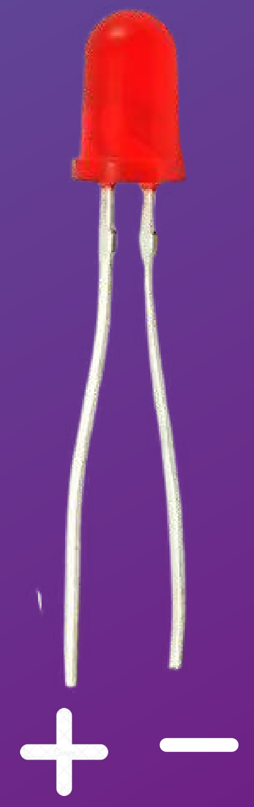
//Once the LED has been turned on for one second, the Arduino then runs the same code, only it proceeds to turn the LED off and wait another second.
Once this process has finished, the loop starts again, and the LED is once again turned on.

//https://www.tinkercad.com/things/2MwHJmZ3dPc-copy-of-lesson-1-blinking-led/editel


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

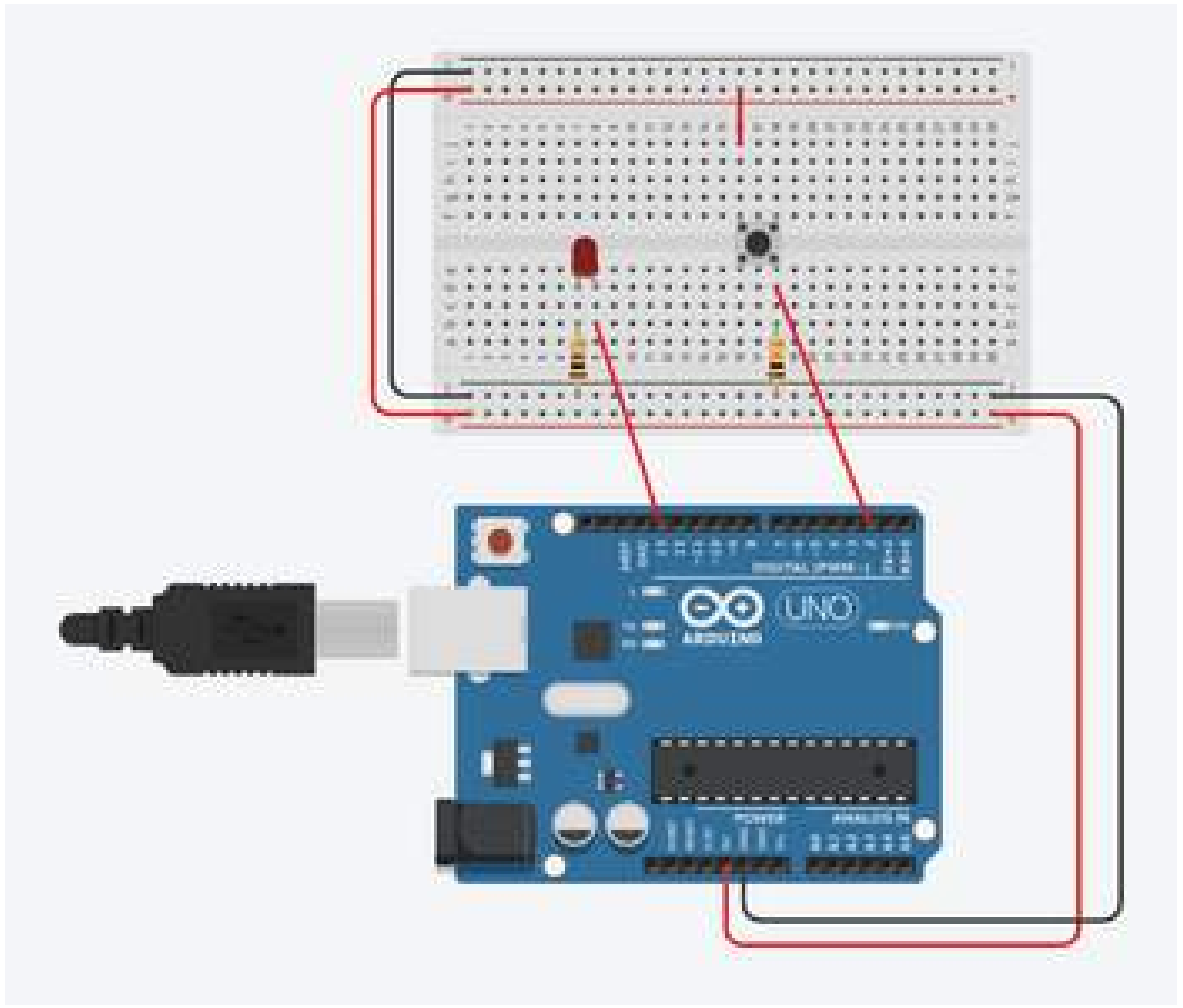
Blinking Single LED with Button

- Connect the button so that is bridges the channel in the middle of the breadboard.
- Connect the top right leg to Pin 2.
- Connect the bottom right leg to resistor and then to ground.
- Connect the bottom left leg to 5V. When the button isn't pressed, the Arduino detects ground (pin 2 > resistor > ground).
- When you press the button, 5V is connected to ground. Arduino pin 2 can detect this change, as pin 2 has now changed from ground to 5V;



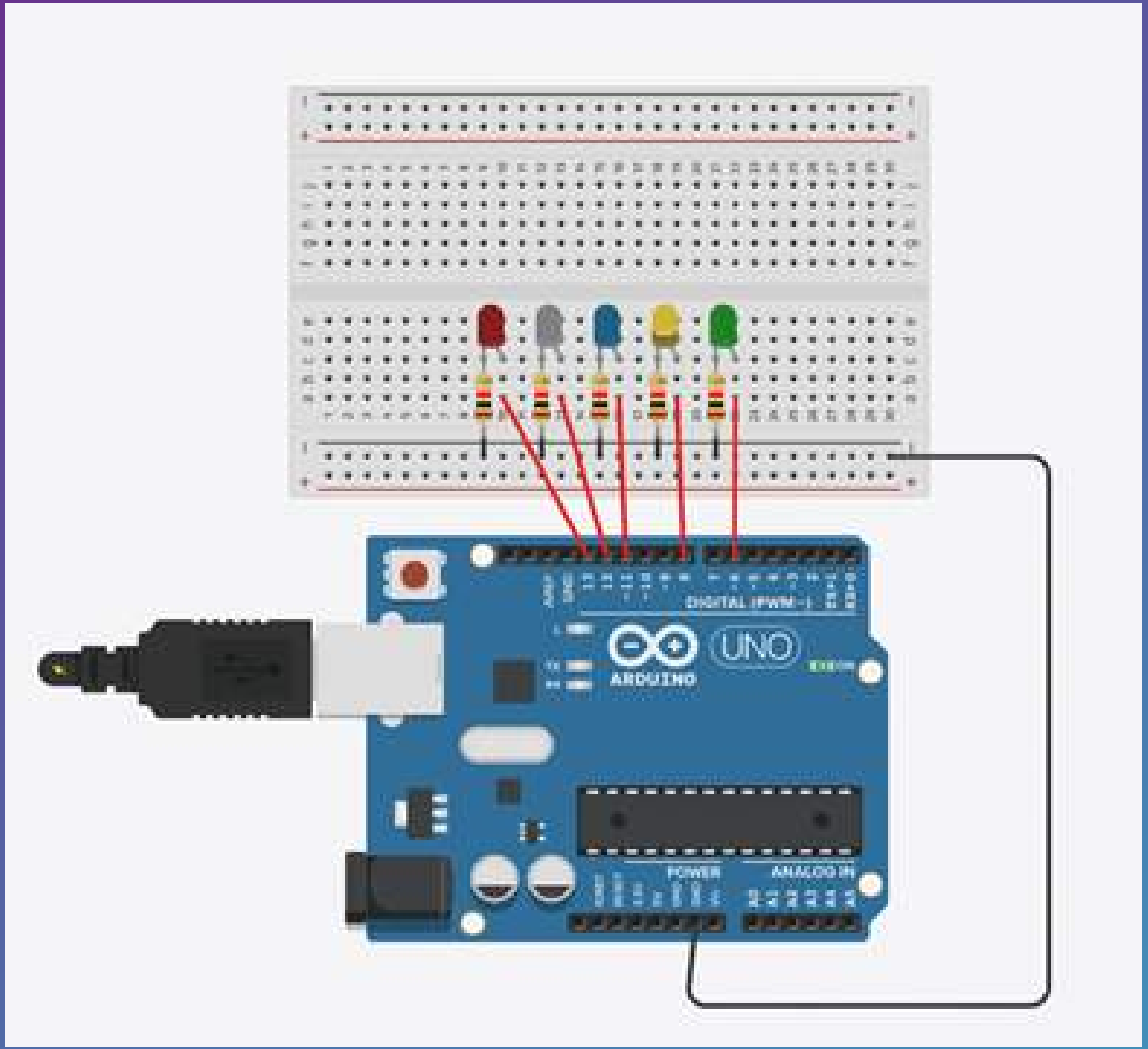
Code

```
 int pin = 0;
void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
}
void loop()
{
  pin = digitalRead(2);
  if(pin == HIGH)
  {
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
  }
  delay(15);
}
```



https://www.tinkercad.com/things/6DnEYNJj14D-copy-of-push-button-arduino-led/editel

Blinking Multiple LED



https://www.tinkercad.com/things/jaENecueYb2-copy-of-multiple-led-blinking/editel

Code

```
void setup()
{
 pinMode(13,OUTPUT);
 // RED LED
 pinMode(12,OUTPUT);
 // WHITE LED
 pinMode(11,OUTPUT);
 // BLUE LED
 pinMode(8,OUTPUT);
 // YELLOW LED
 pinMode(6,OUTPUT);
 // GREEN LED
}
```

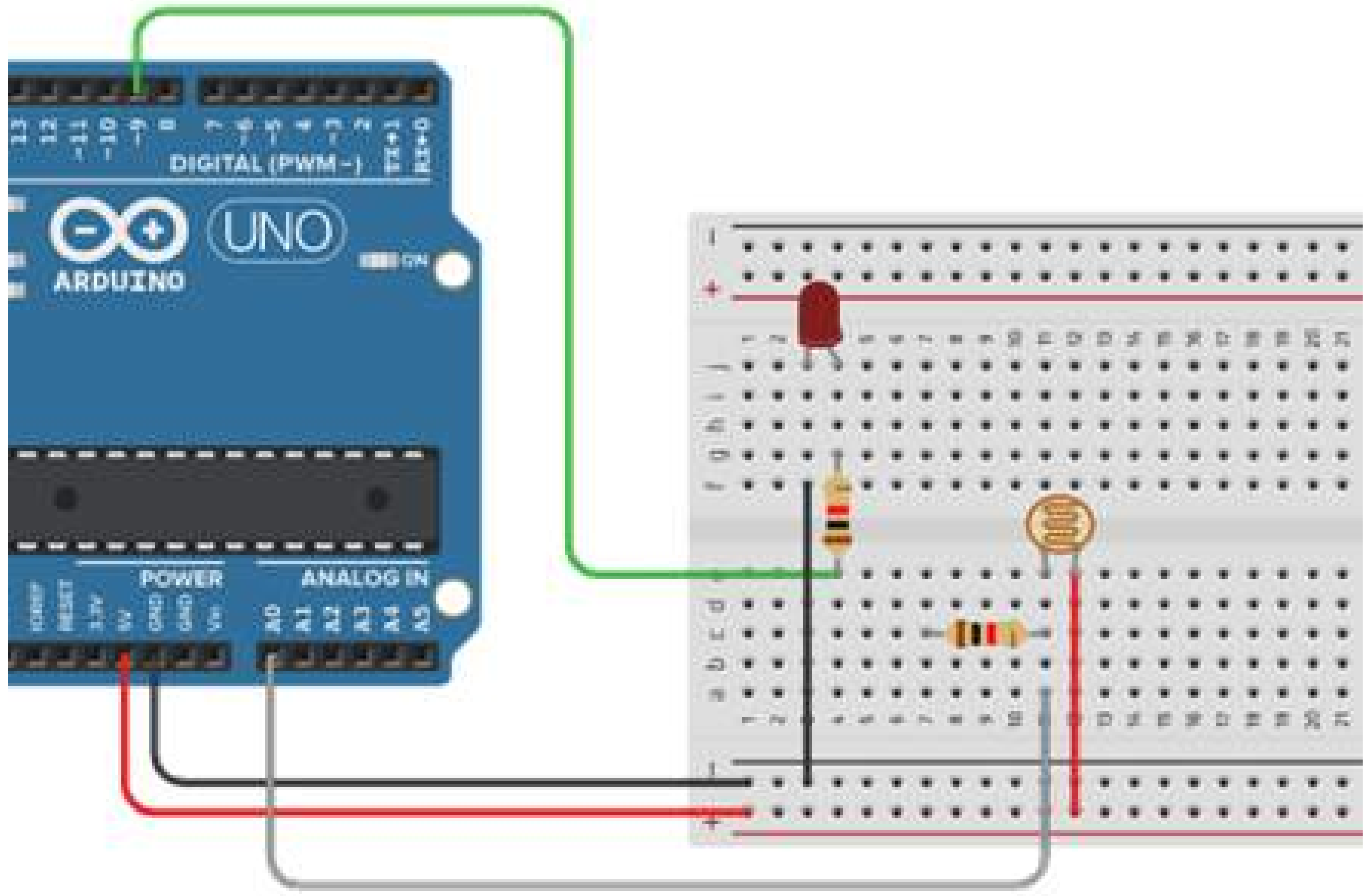
```
void loop()
{
 digitalWrite(13, HIGH);
 delay(1000);
 digitalWrite(13, LOW);
 delay(1000);
 digitalWrite(12, HIGH);
 delay(1000);
 digitalWrite(12, LOW);
 delay(1000);
 digitalWrite(11, HIGH);
 delay(1000);
 digitalWrite(11, LOW);
 delay(1000);
 digitalWrite(8, HIGH);
 delay(1000);

digitalWrite(8, LOW);
 delay(1000);
 digitalWrite(6, HIGH);
 delay(1000);

digitalWrite(6, LOW);
  delay(1000);  //  Wait
for 1000 millisecond(s)
}
```

Automatic Light Switch

An **LDR sensor** (Light Dependent Resistor) is a device that is used to **detect light.** It detects the light intensity. We use it to control the lights, when there is dark it detects light intensity and blows the lights. As it mainly used in mobiles for auto-brightness and smart street lights.

Code

```
int sensorPin = A0;
int ledPin = 9;
void setup()
{
Serial.begin(9600);
pinMode(sensorPin, INPUT);
pinMode(ledPin, OUTPUT);
}
void loop()
{
intsensorValue = analogRead(sensorPin);
if( sensorValue <= 300 ) // Change the value as per your requirement
{
digitalWrite(ledPin, HIGH);
Serial.print("LED ON ");
Serial.println(sensorValue);
delay(100);
}
else
{
digitalWrite(ledPin, LOW);
Serial.println("LED OFF");
}
}
```

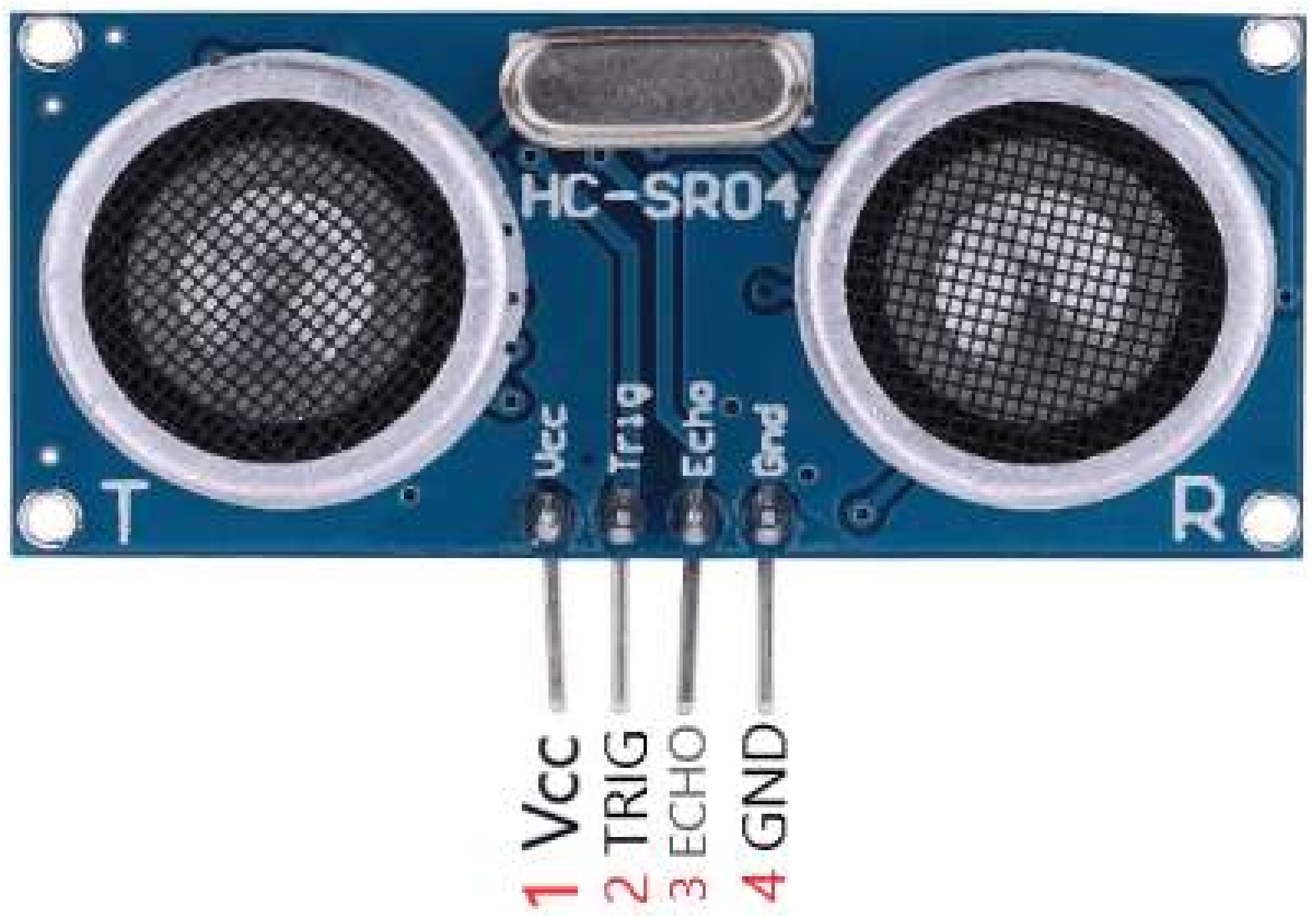


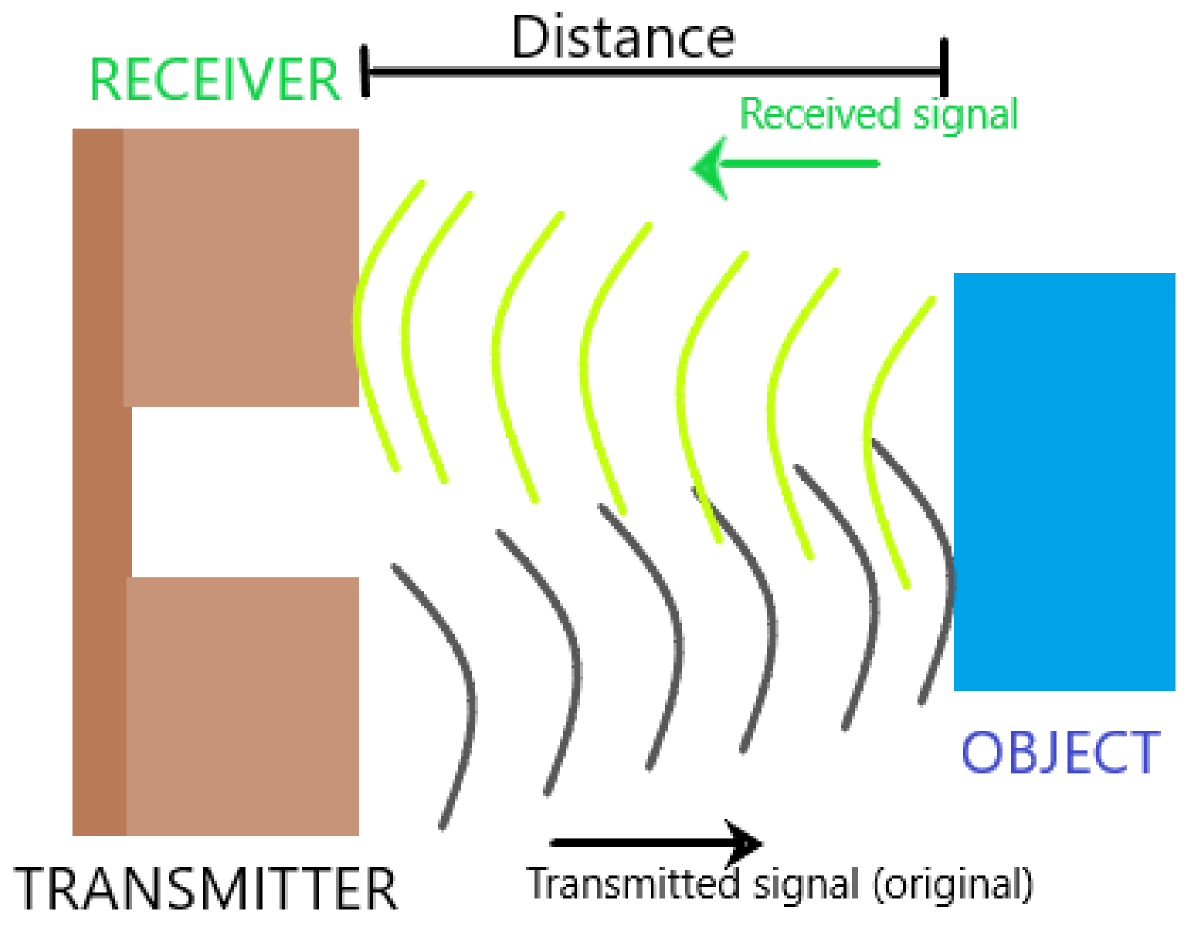
IMPORTANT SENSORS

ULTRASONIC SENSORS

- The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object.
- We will set the TRIG pin to HIGH for some time (about 3 to 100 microseconds).
- As soon the TRIG pin is LOW, the Ultrasonic sensor sends the pulses and sets the ECHO pin to HIGH.
- When the sensor gets the reflected pulses, it sets the ECHO pin to LOW.
- We need to measure the time for which the ECHO pin was HIGH.
- We will use the PulseIn() function to read the time from the output of the ECHO pin. It will wait for the specified pin to go HIGH and LOW. The function would return the timing at the end.
- The TRIG pin is set LOW for 4 microseconds and then HIGH for 15 microseconds.
- The timing will be calculated in microseconds.

Pins & Explaination





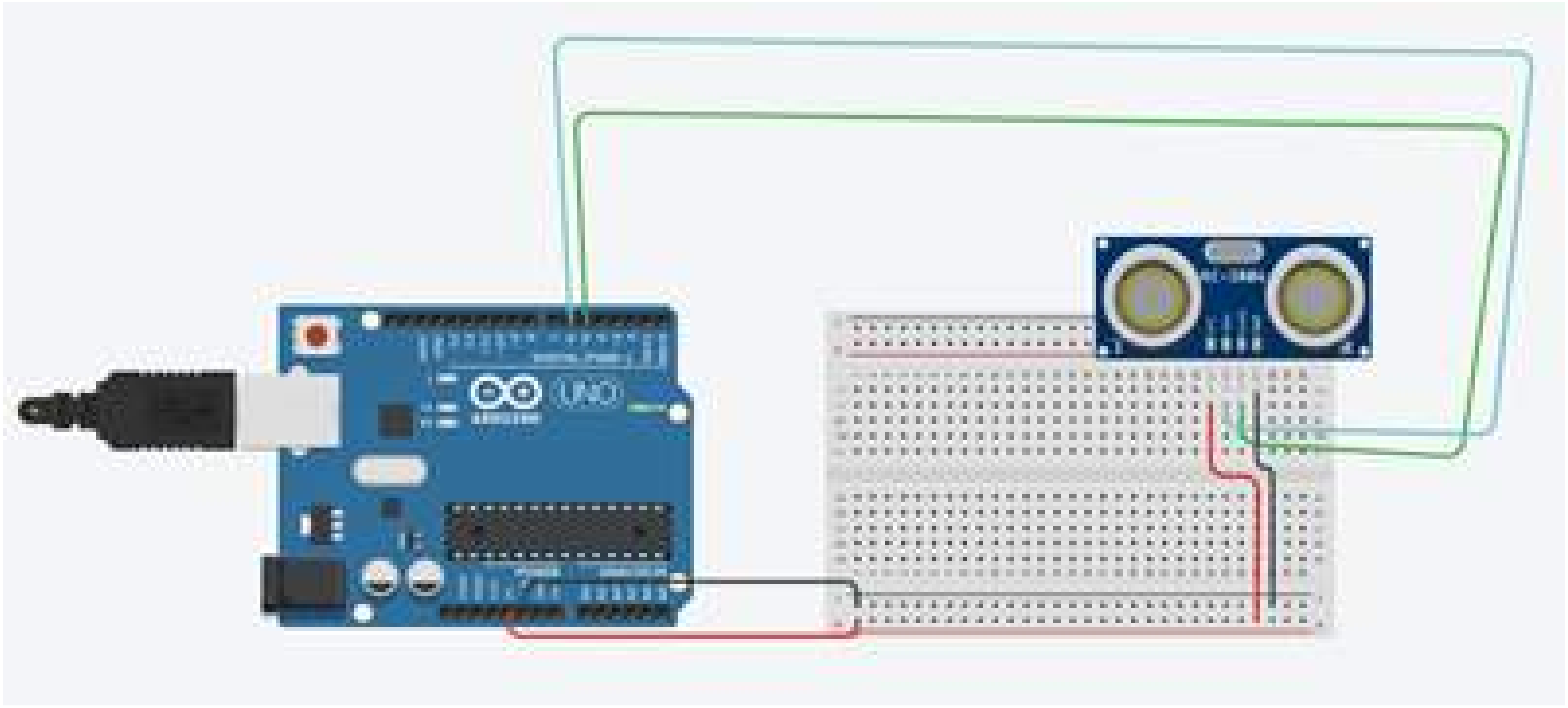
- Connect the VCC pin of HC-SRO4 to 5V of the Arduino board.
- Connect the GND pin of HC-SRO4 to GND of the Arduino board.
- Connect the TRIG pin of HC-SRO4 to pin 6 of the Arduino board.
- Connect the ECHO pin of HC-SRO4 to pin 5 of the Arduino board.

Code

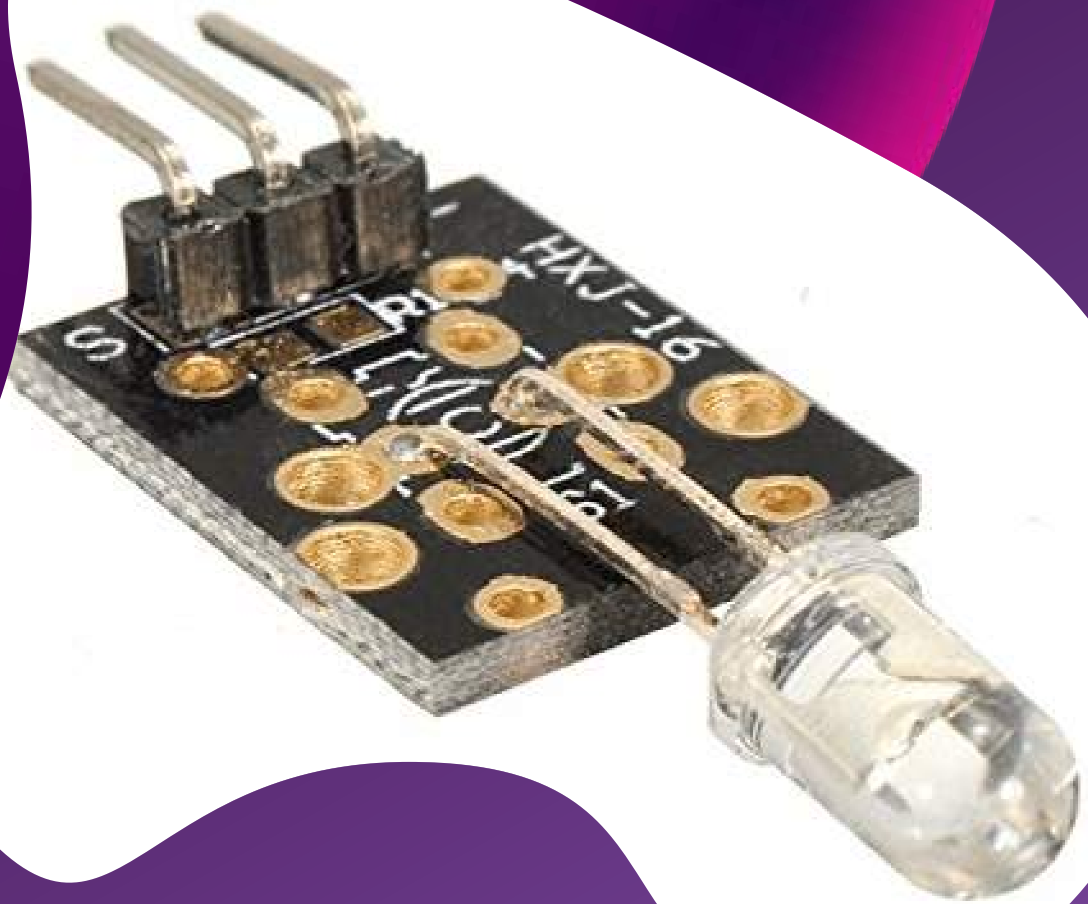
```
#define ECHOpin 5
// it defines the ECHO pin of the sensor to pin 5 of Arduino
#define TRIGpin 6
// we have defined the variable
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement
void setup()
{
  pinMode(TRIGpin, OUTPUT); // It sets the ECHO pin as OUTPUT
  pinMode(ECHOpin, INPUT); // It sets the TRIG pin as INPUT
  Serial.begin(9600); // // Serial Communication at the rate of 9600 bps
  Serial.println("Test of the Ultrasonic Sensor HC-SR04");
// It will appear on Serial Monitor
```

```
Serial.println("with the Arduino UNO R3 board");
}
void loop()
{
  // It first sets the TRIG pin at LOW for 2 microseconds
  digitalWrite(TRIGpin, LOW);
  delayMicroseconds(4);
  // It now sets TRIG pin at HIGH for 15 microseconds
  digitalWrite(TRIGpin, HIGH);
  delayMicroseconds(15);
  digitalWrite(TRIGpin, LOW);
  // It will read the ECHO pin and will return the time
  duration = pulseIn(ECHOpin, HIGH);
  // distance formula
  distance = duration*(0.034/2); // (speed in microseconds)
   // Speed of sound wave (340 m/s)divided by 2 (forward and backward
bounce)
  // To display the distance on Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm"); //specified unit of distance
}
```

https://www.tinkercad.com/things/3X1bMfnT7fb-copy-of-lcd-ultrasonic-sensor/editel


UNO

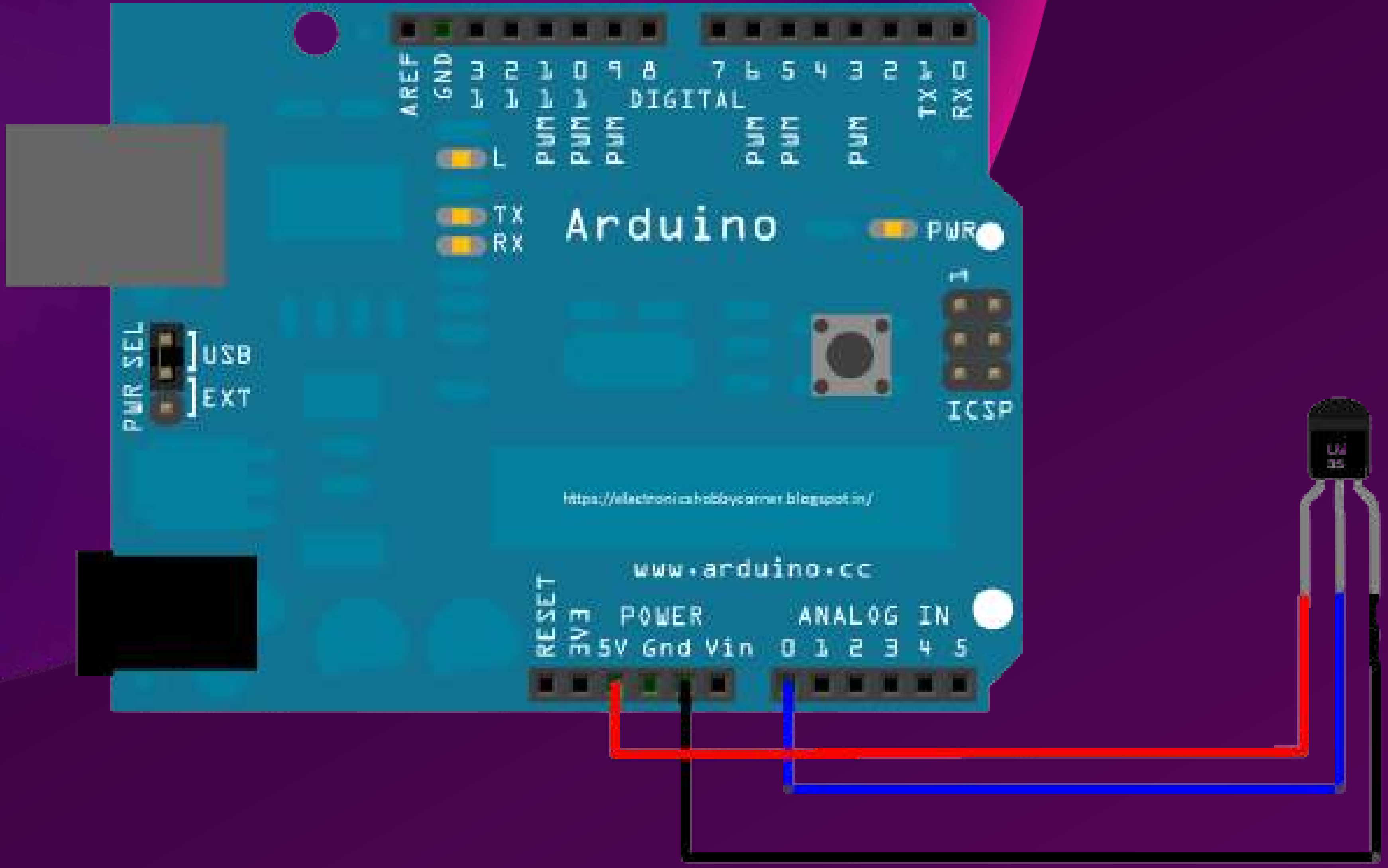
Infrared emission sensor

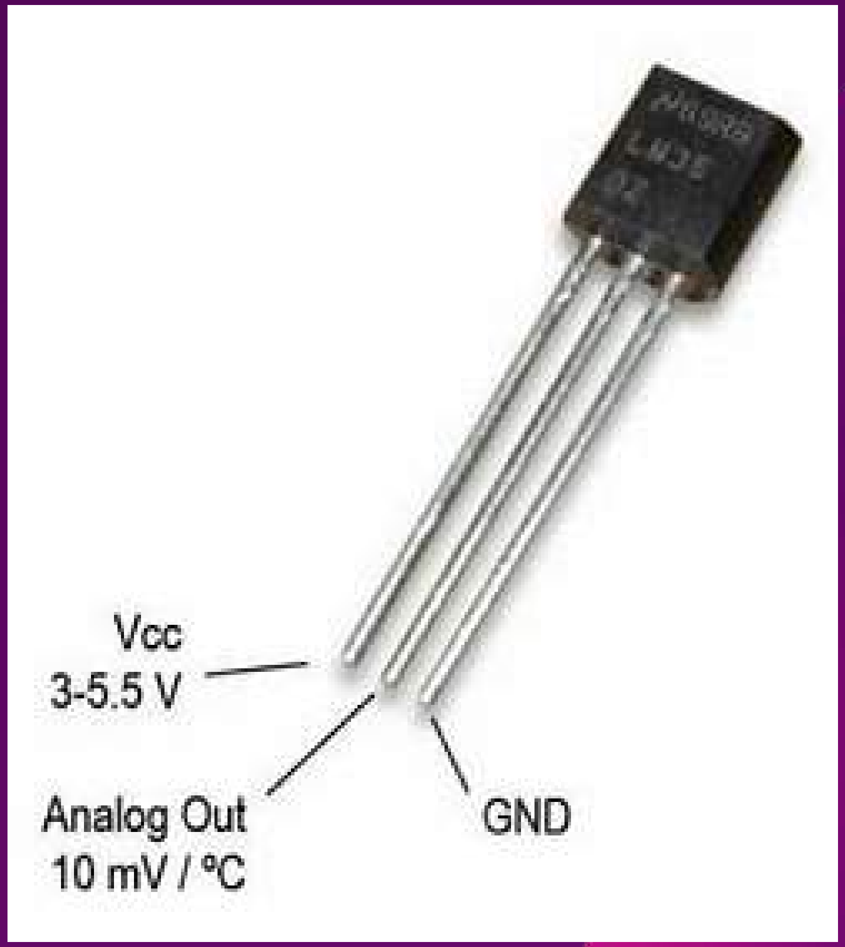


An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion

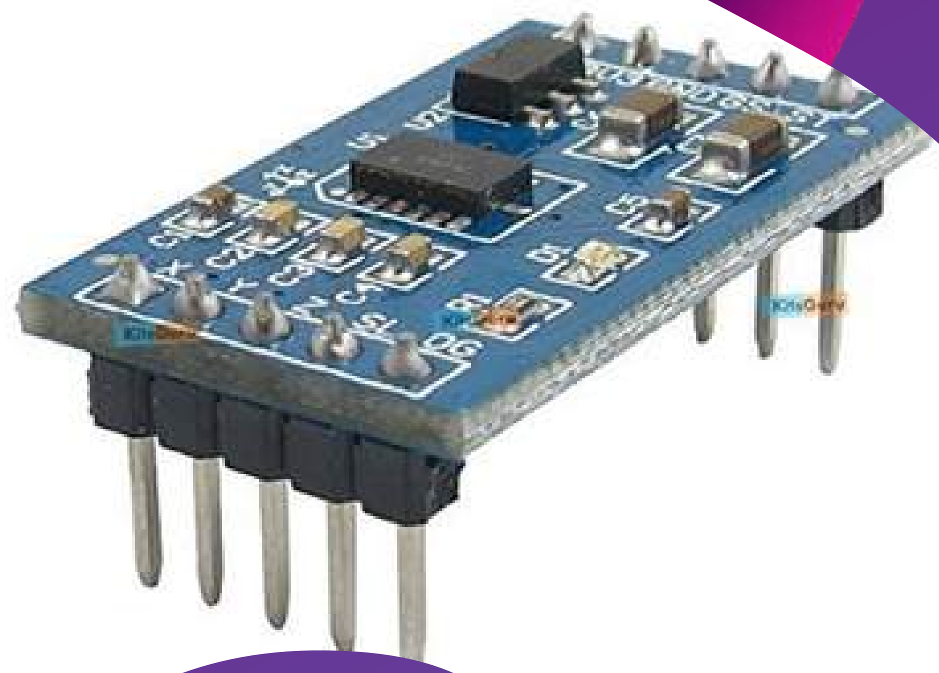
LM35

- LM35 which is a temperature sensor . LM35 is an analog temperature sensor.





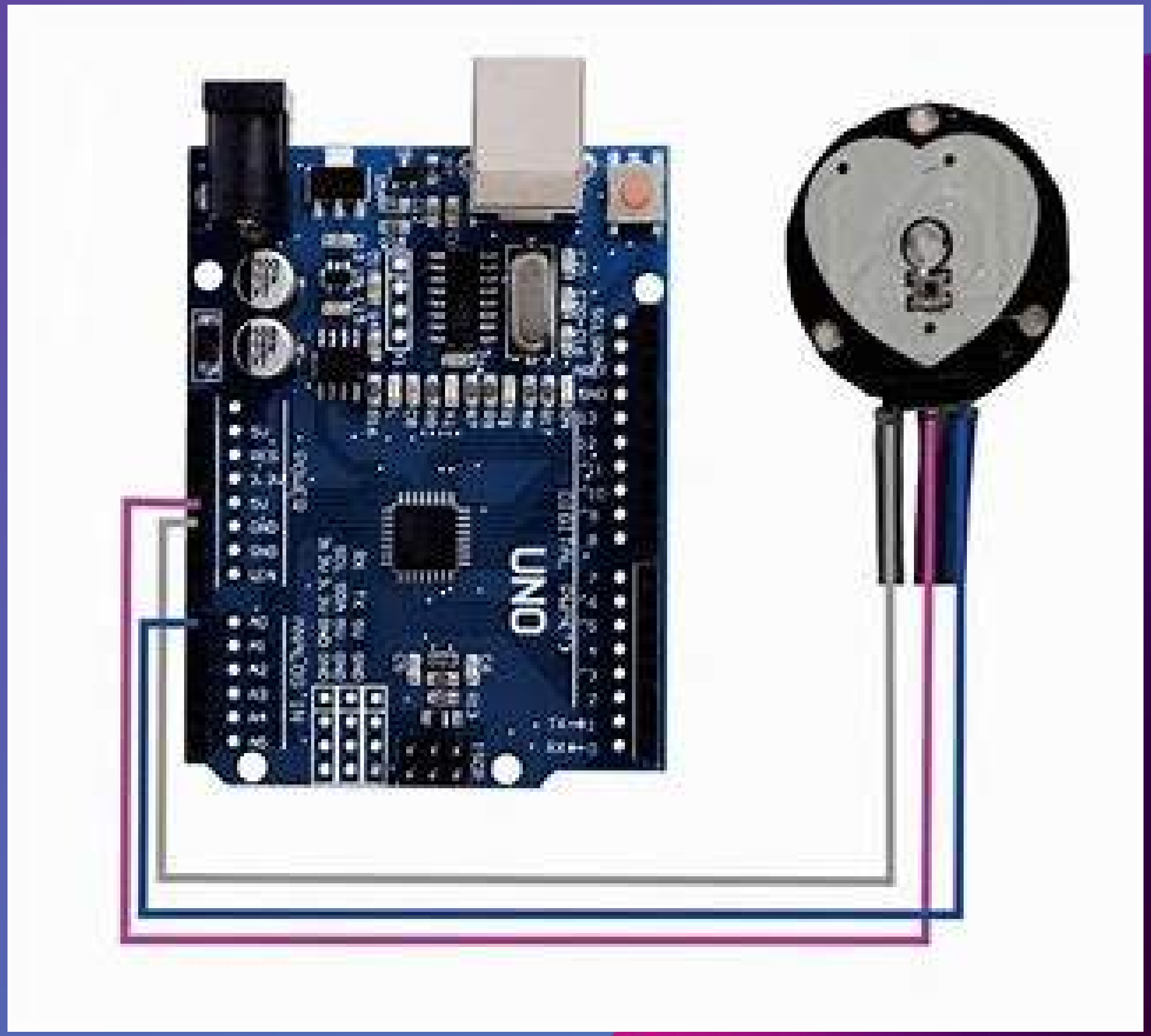
Temperature Sensor



The accelerometer is the device capable of detecting changes in motion in the form of acceleration. It can also measure the vibration of a structure. The ADXL3xx outputs the acceleration on each axis as an analog voltage between 0 and 5 volts. To read this, all you need is the analogRead() function.

Accelerometer sensor

The pulse sensor module has a light which helps in measuring the pulse rate. When we place the finger on the pulse sensor, the light reflected will change based on the volume of blood inside the capillary blood vessels. During a heartbeat, the volume inside the capillary blood vessels will be high.



Pulse sensor

pH sensor

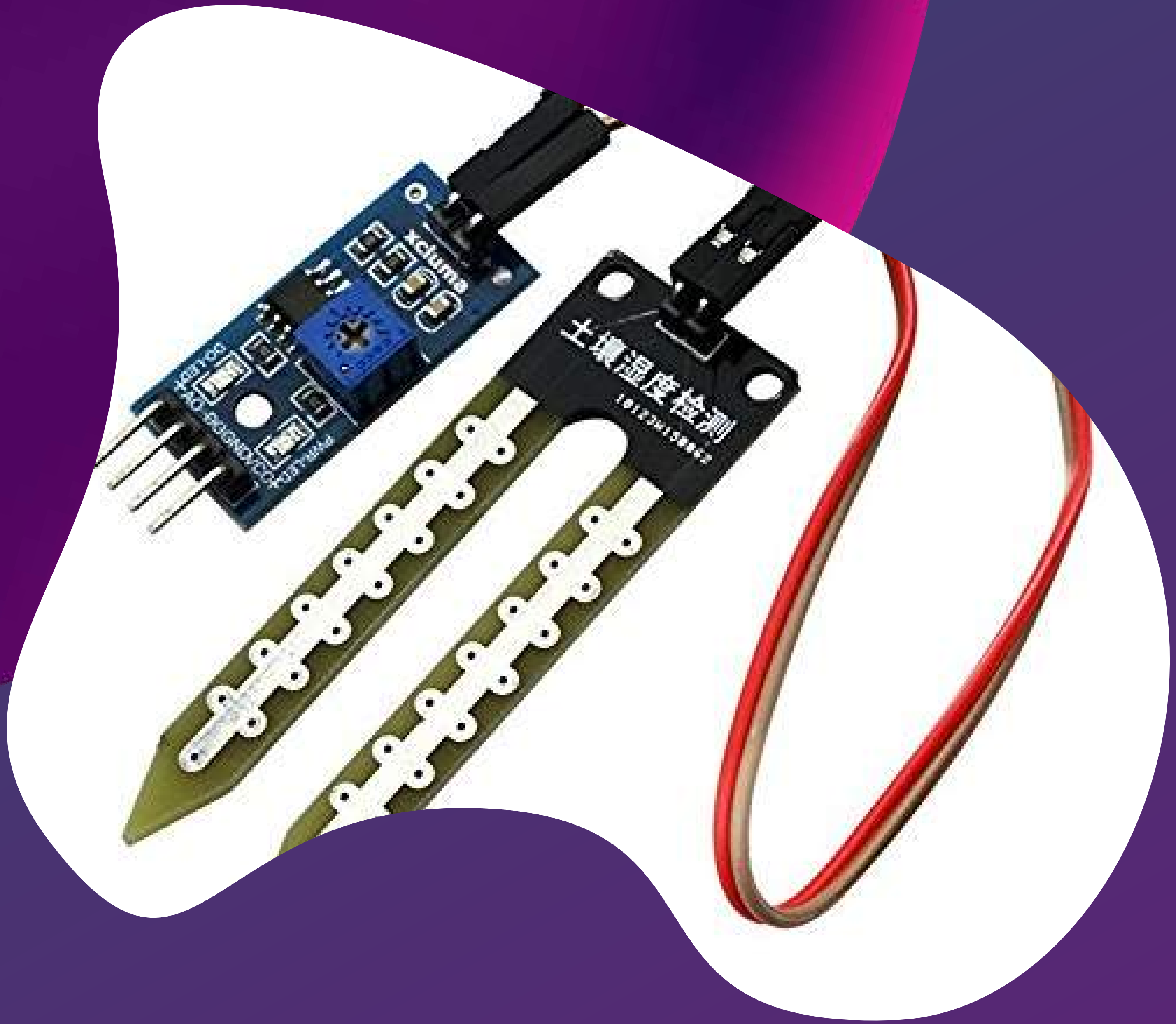
- Sensor measures the potential difference between two electrodes: a reference electrode (silver / silver chloride) and a glass electrode that is sensitive to hydrogen ion.
- we can deduce the equation to convert the measured voltage to pH. The general formula would be y = mx + b.
- The code consists of taking 10 samples of the analogue input A0, ordering them and discarding the highest and the lowest and calculating the mean with the six remaining samples by converting this value to voltage in the variable pHVol
- then using the equation that we have calculated with the pH reference values we convert ***pHVol*** to ***pHValue*** and send it to the serial port to see it in the serial monitor.

Gas Sensor

- **MQ-9**

- This gas sensor has high sensitity to Carbon Monoxide, Methane and LPG. The sensor could be used to detect different gases contains CO and combustible gases, it is with low cost and suitable for different application.

- The sensitivity of the sensor can be adjusted by using the potentiometer.

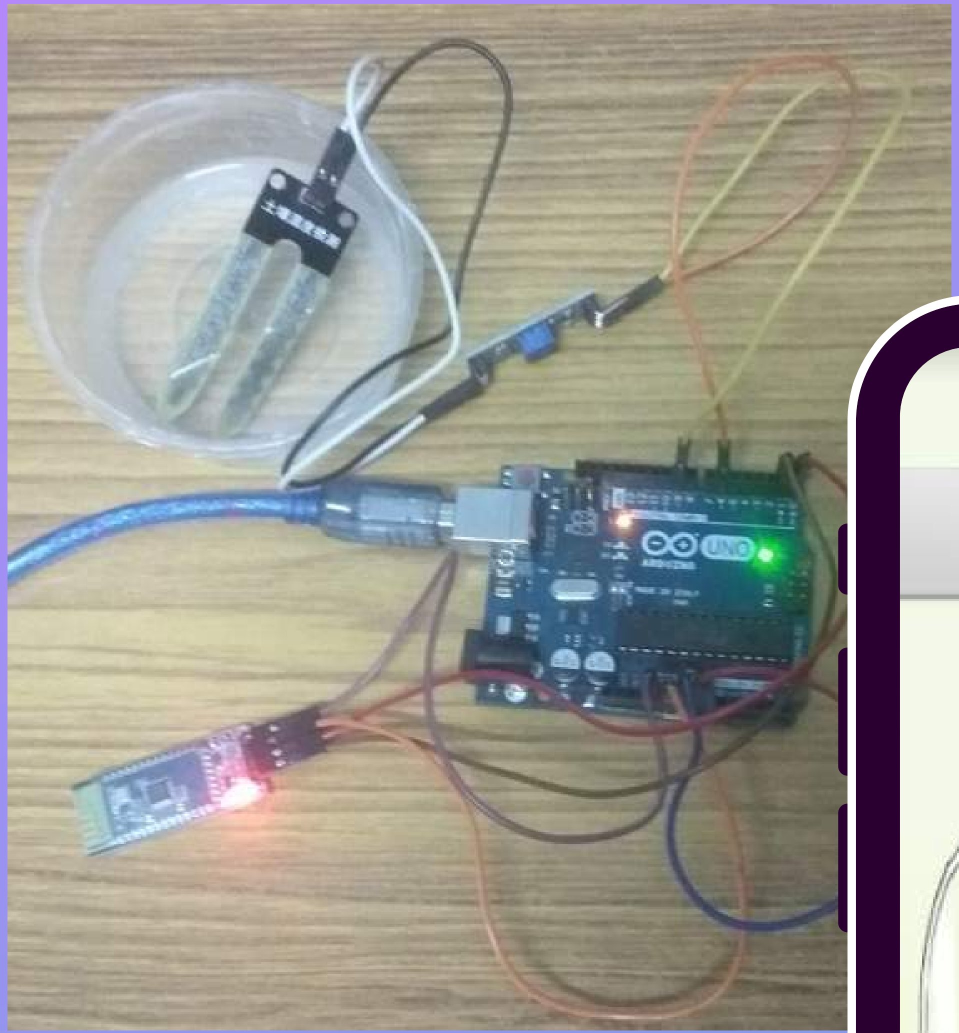


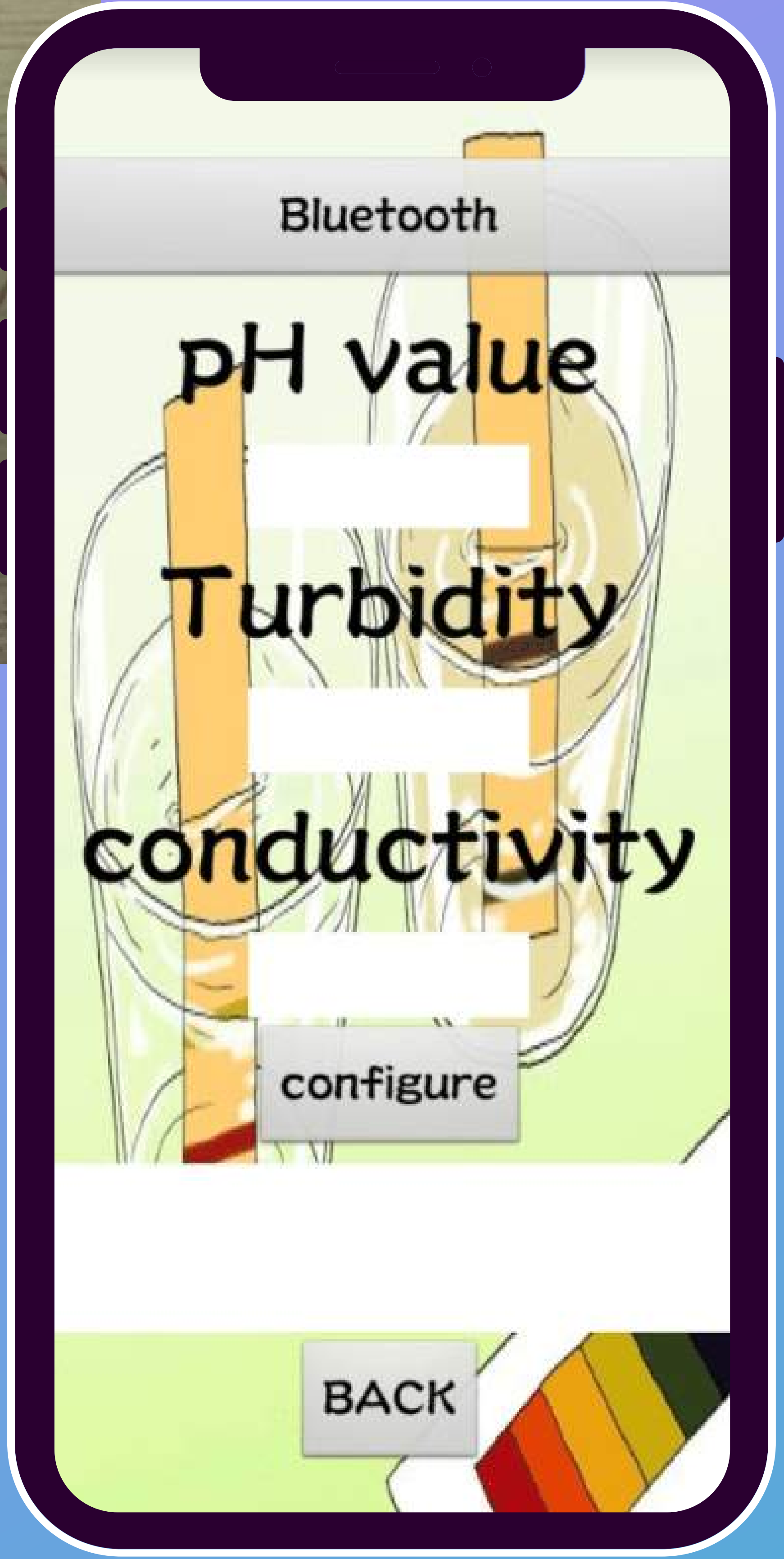
The Soil Moisture Sensor measures soil moisture grace to the changes in electrical conductivity of the earth (soil resistance increases with drought). The electrical resistance is measured between the two electrodes of the sensor. code allows to light up a LED if the sensorValue is smaller than the limit. Otherwise, the LED stays off.

Soil moisture sensor

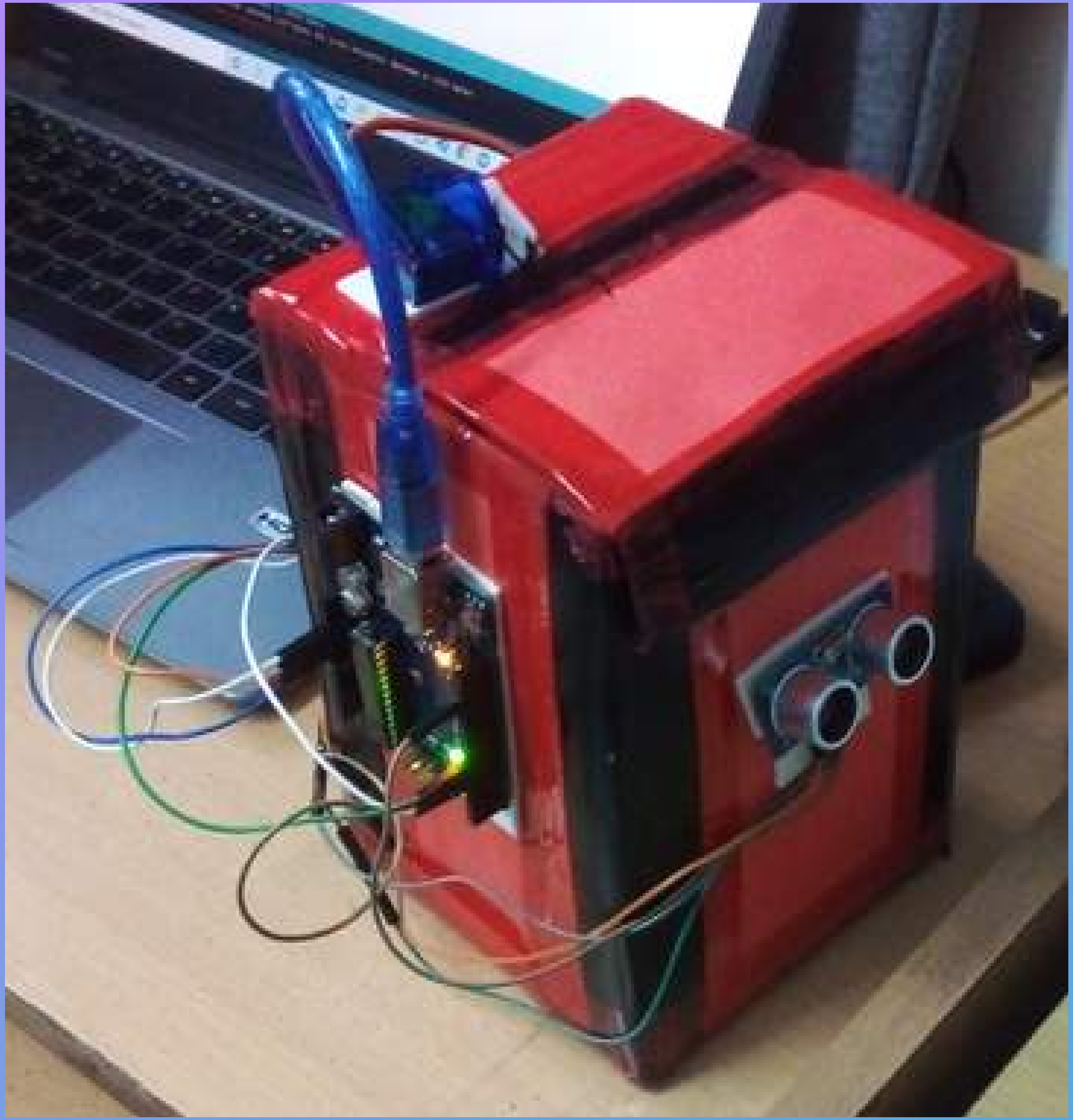
Flame Sensor

A flame sensor works by detecting the presence of a flame within the furnace. The sensor is a short length of thin metallic rod that creates a small current of electricity in order to confirm there is fire burning within the furnace.

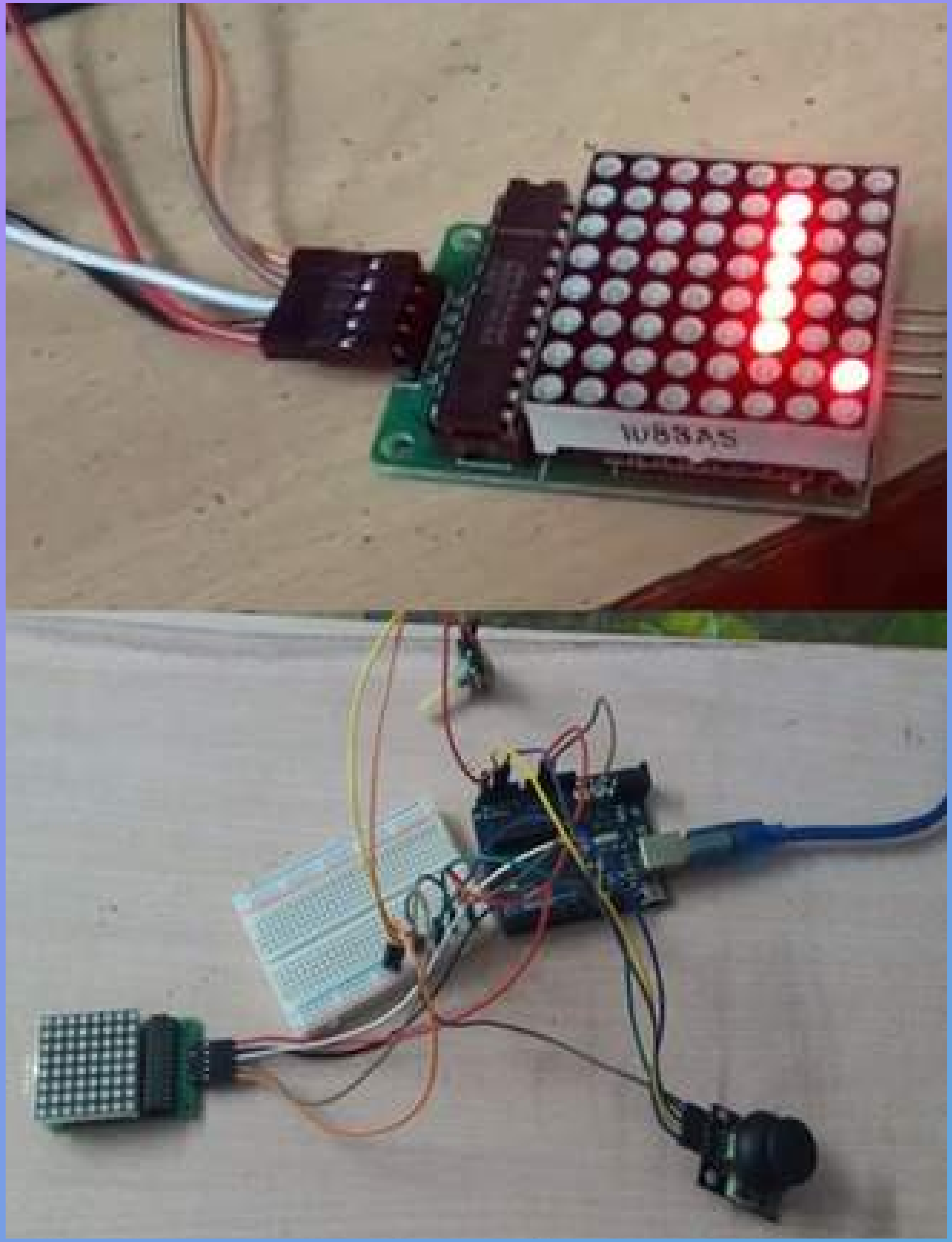




Water Quality Checking app



Smart Dustbin



Snake Game

Thank You