KONGUNADU ARTS AND SCIENCE COLLEGE (AUTONOMOUS)

Re-accredited by NAAC with A+ Grade - 4th cycle,
College of Excellence – UGC
52nd Rank among Colleges in NIRF 2024
Coimbatore - 641 029, Tamil Nadu, India

DEPARTMENT OF COMPUTER SCIENCE (AIDED)

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

SEMESTER – _____

BONAFIDE CERTIFICATE

This is to certify that

_____ Programming Lab is a Bonafide Record of original Practical Work done by

NAME:
REGISTER NO:

Staff-incharge **Head of the Department**

Submitted for the End of Semester Practical Examination held on_____

Internal Examiner **External Examiner**

PYTHON PROGRAMMING AND IOT LAB

INDEX

3. Write a program to measure the distance using Ultrasonic Sensor and make LED blink using Arduino.

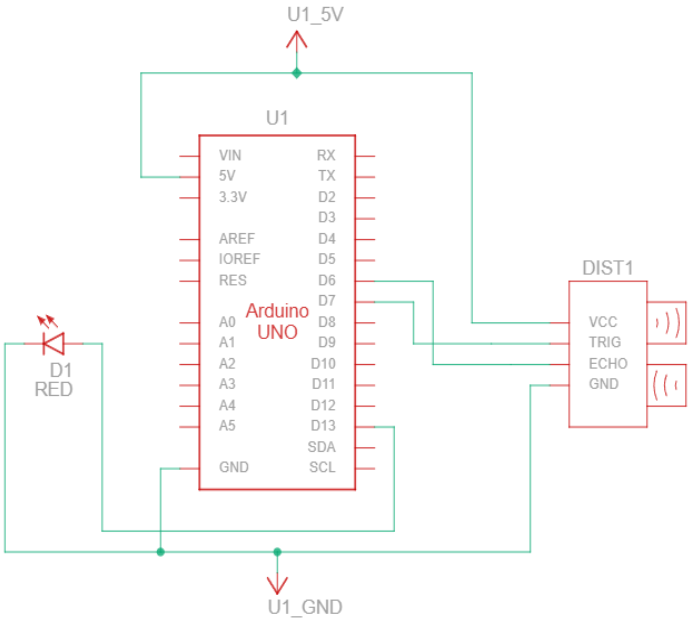
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- Ultrasonic Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- LED Bulb's Anode to pin 13 in Arduino UNO.
- LED Bulb's Cathode to GND in Arduino UNO.
- Sensor's VCC pin to 5V in Arduino UNO.
- Sensor's GND pin to GND in Arduino UNO.
- Sensor's TRIG pin to pin 7 in Arduino UNO.
- Sensor's ECHO pin to pin 6 in Arduino UNO.

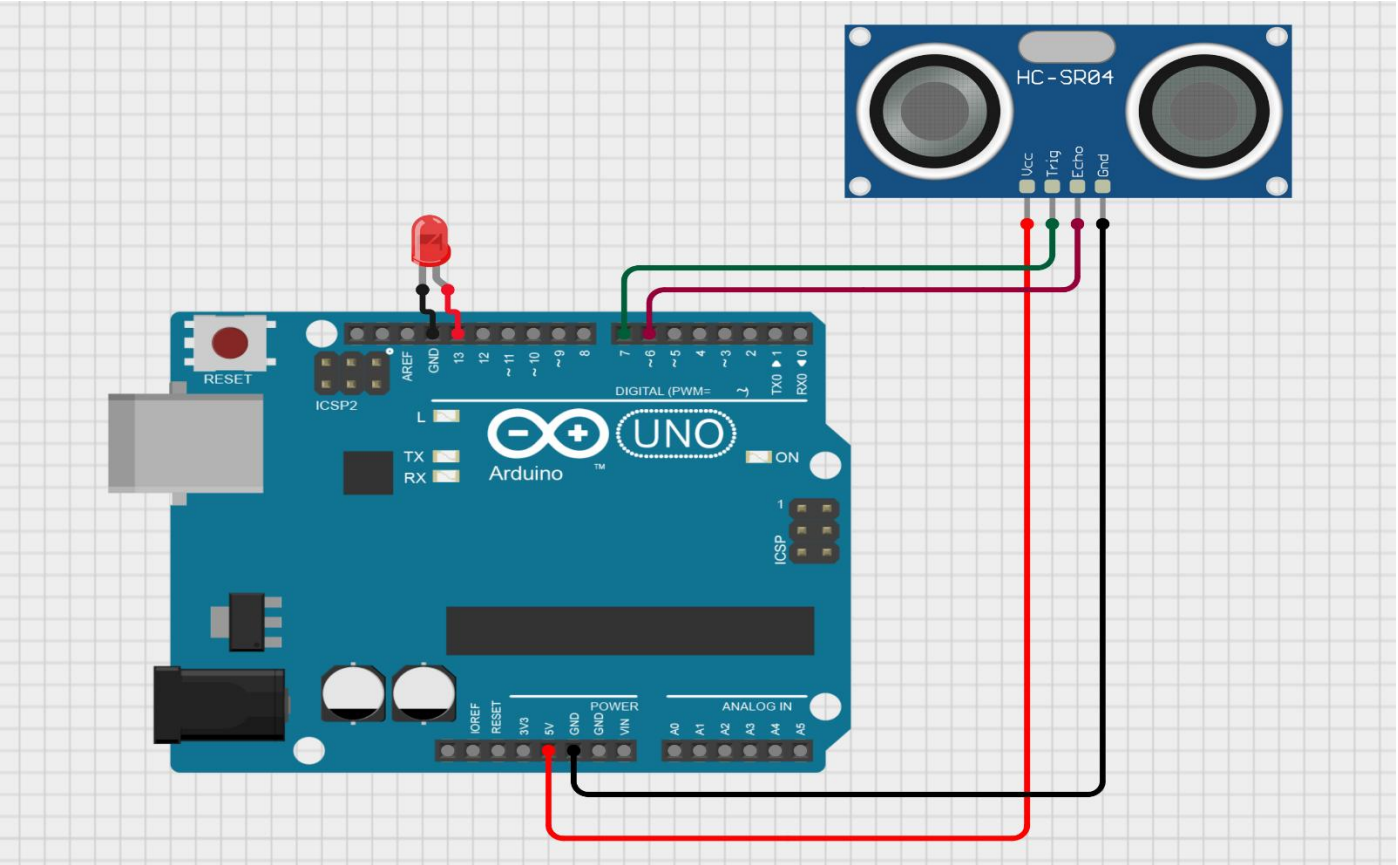
SCHEMATIC DIAGRAM:



CODING:

```
const int led = 13;
const int trigger = 7;
const int echo = 6;

void setup() {
  pinMode(led,OUTPUT);
  pinMode(trigger,OUTPUT);
  pinMode(echo,INPUT);
  Serial.begin(9600);
}

void loop() {
  long duration;
  long distance;

  digitalWrite(trigger,LOW);
  delayMicroseconds(2);
  digitalWrite(trigger,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger,LOW);

  duration = pulseIn(echo,HIGH);
  distance = ((duration/29)/2);

  Serial.print("Distance:");
  Serial.print(distance);
  Serial.println("cm");
```

CIRCUIT DIAGRAM:



OUTPUT:

Serial Monitor

```
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:78cm
Distance:77cm
Distance:77cm
Distance:70cm
Distance:62cm
Distance:56cm
Distance:43cm
Distance:59cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
Distance:89cm
```

```
if (distance <= 25)
{
   digitalWrite(led,HIGH);
   delay(500);
   digitalWrite(led,LOW);
   delay(500);
  }
  else
  {
   digitalWrite(led,LOW);
  }
  delay(200);
  }
```

5. Write a program to turn ON and OFF an LED connected to an Arduino board using python.

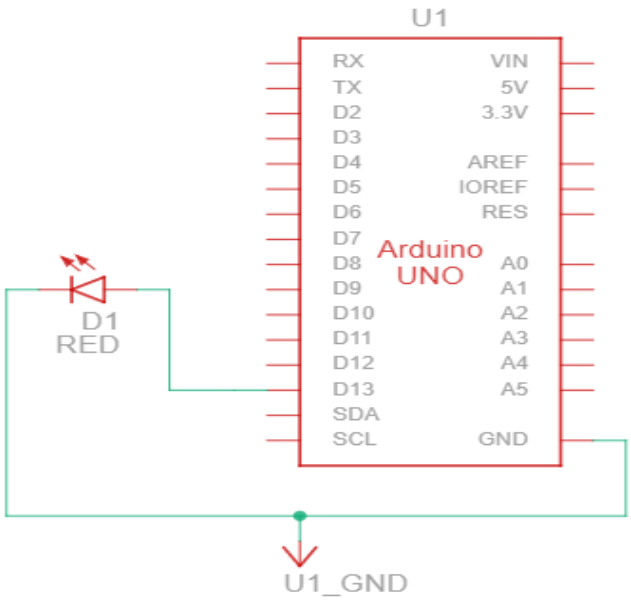
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- USB Cable.

CONNECTIONS:

- LED Bulb's Anode to pin 13 in Arduino UNO.
- LED Bulb's Cathode to GND in Arduino UNO.

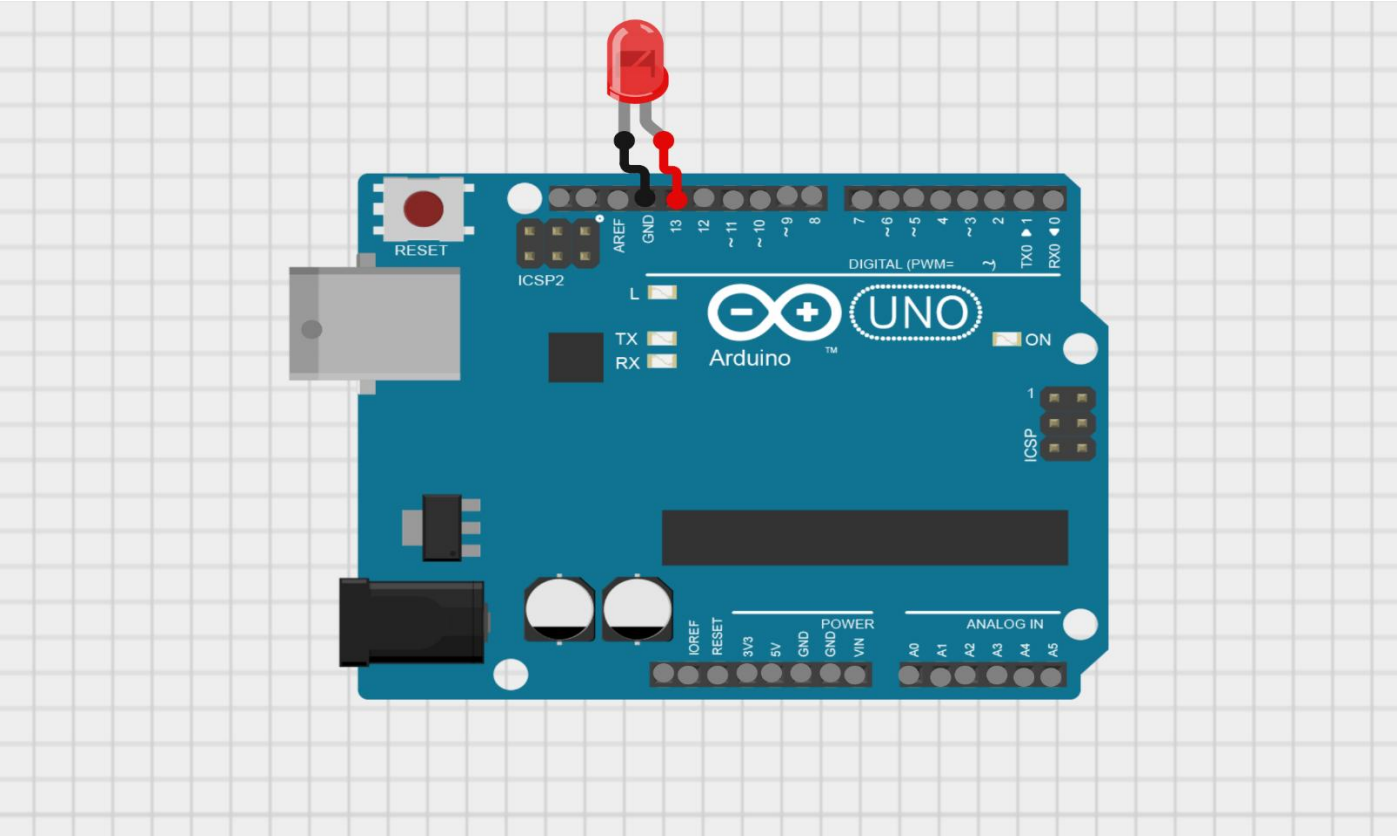
SCHEMATIC DIAGRAM:



CODING:

```
const int led = 13;

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    digitalWrite(led, Serial.read() == '1' ? HIGH : LOW);
  }
}
```

CIRCUIT DIAGRAM:



OUTPUT:

```
IDLE Shell 3.11.5
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\HARI\IOT lab problems\led_pyhton.py
Type '1' to on LED '0' to off LED  and 'q' to exit : 1
LED ON
Type '1' to on LED '0' to off LED  and 'q' to exit : 0
LED OFF
Type '1' to on LED '0' to off LED  and 'q' to exit : 2
Invalid command....
Type '1' to on LED '0' to off LED  and 'q' to exit : 1
LED ON
Type '1' to on LED '0' to off LED  and 'q' to exit : 0
LED OFF
Type '1' to on LED '0' to off LED  and 'q' to exit : q
Program Terminated....
>>>
```

PYTHON CODE:

```
import serial

import time


a = serial.Serial('COM3', 9600)

time.sleep(2)


while True:

   cmd = input("Type '1' to on LED '0' to off LED  and 'q' to exit : ").strip().lower()

   if cmd == '1':

      a.write(b'1')

      print(“LED ON”)

   elif cmd == '0':

      a.write(b'0')

      print(“LED OFF”)

   elif cmd == 'q':

      print(“Program Terminated.”)

      break

   else:

      print("Invalid command.")
```

6. Write a program to interface PUSH button/Digital Sensor (IR) with Arduino and turn ON LED when PUSH button is pressed or at a Sensor detection.

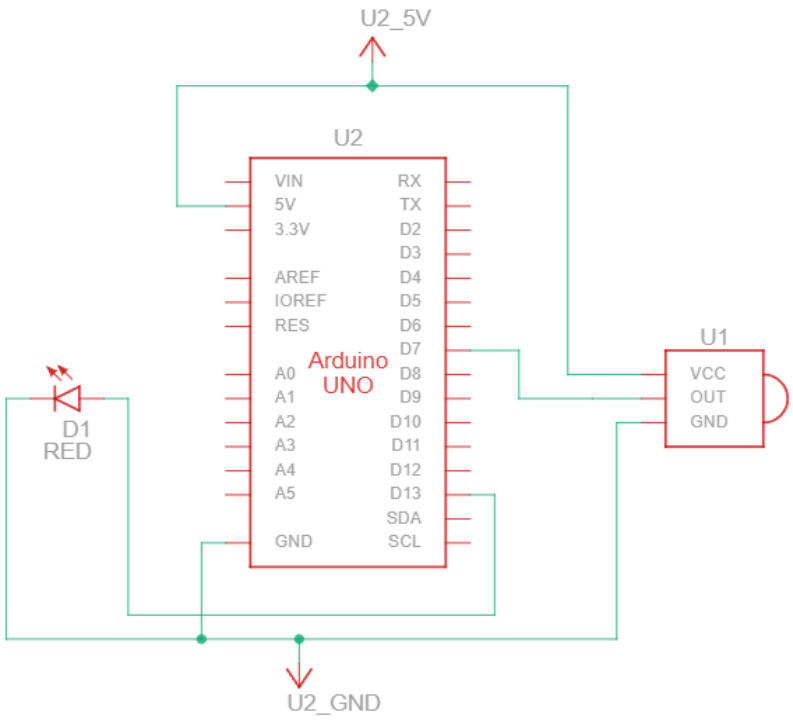
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- InfraRed Module.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- LED Bulb's Anode to pin 13 in Arduino UNO.
- LED Bulb's Cathode to GND in Arduino UNO.
- Module's VCC pin to 5V in Arduino UNO.
- Module's GND pin to GND in Arduino UNO.
- Module's OUT pin to pin 7 in Arduino UNO.

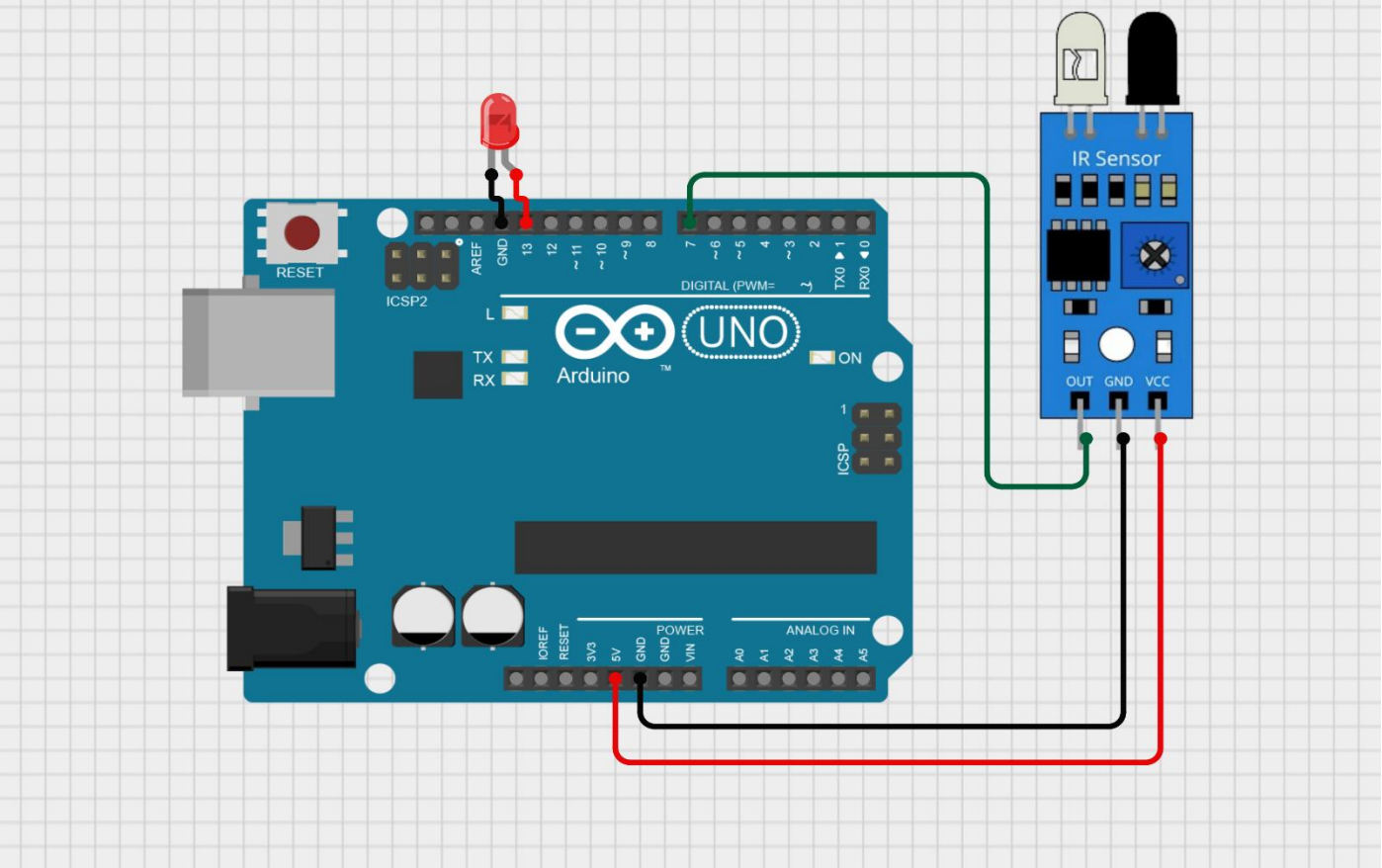
SCHEMATIC DIAGRAM:



CODING:

```
const int led = 13;

const int ir = 7;


void setup(){

  pinMode(led,OUTPUT);

  pinMode(ir,INPUT);

  Serial.begin(9600);

}


void loop(){

  int x = digitalRead(ir);

  Serial.println(x);

  delay(300);


  if(x==1)

  {

    digitalWrite(led,LOW);

  }

  else

  {

    digitalWrite(led,HIGH);

  }

}
```

CIRCUIT DIAGRAM:



OUTPUT:

7. Write a program to interface Bluetooth with Arduino and turn LED ON/OFF when '1'/'0' is received from the smartphone using Bluetooth.

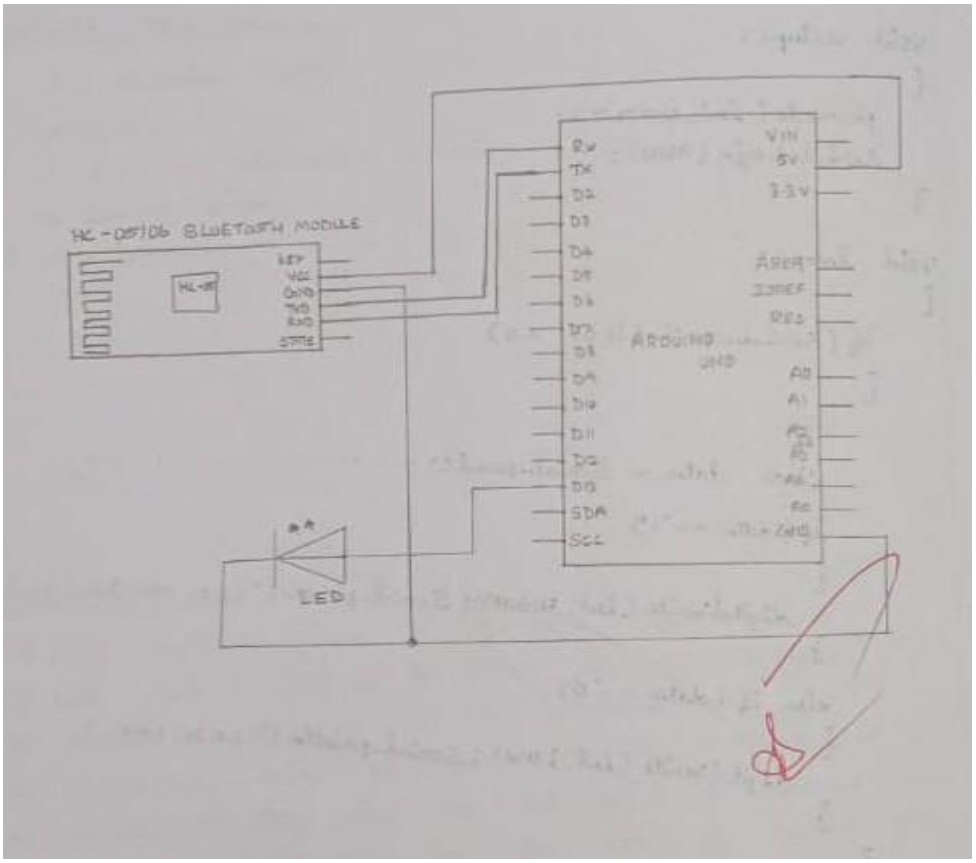
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- HC-05 Bluetooth Module.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- LED Bulb's Anode to pin 13 in Arduino UNO.
- LED Bulb's Cathode to GND in Arduino UNO.
- Module's VCC pin to 5V in Arduino UNO.
- Module's GND pin to GND in Arduino UNO.
- Module's TXD pin to D0(TX) pin in Arduino UNO.
- Module's RXD pin to D1(RX) pin in Arduino UNO.

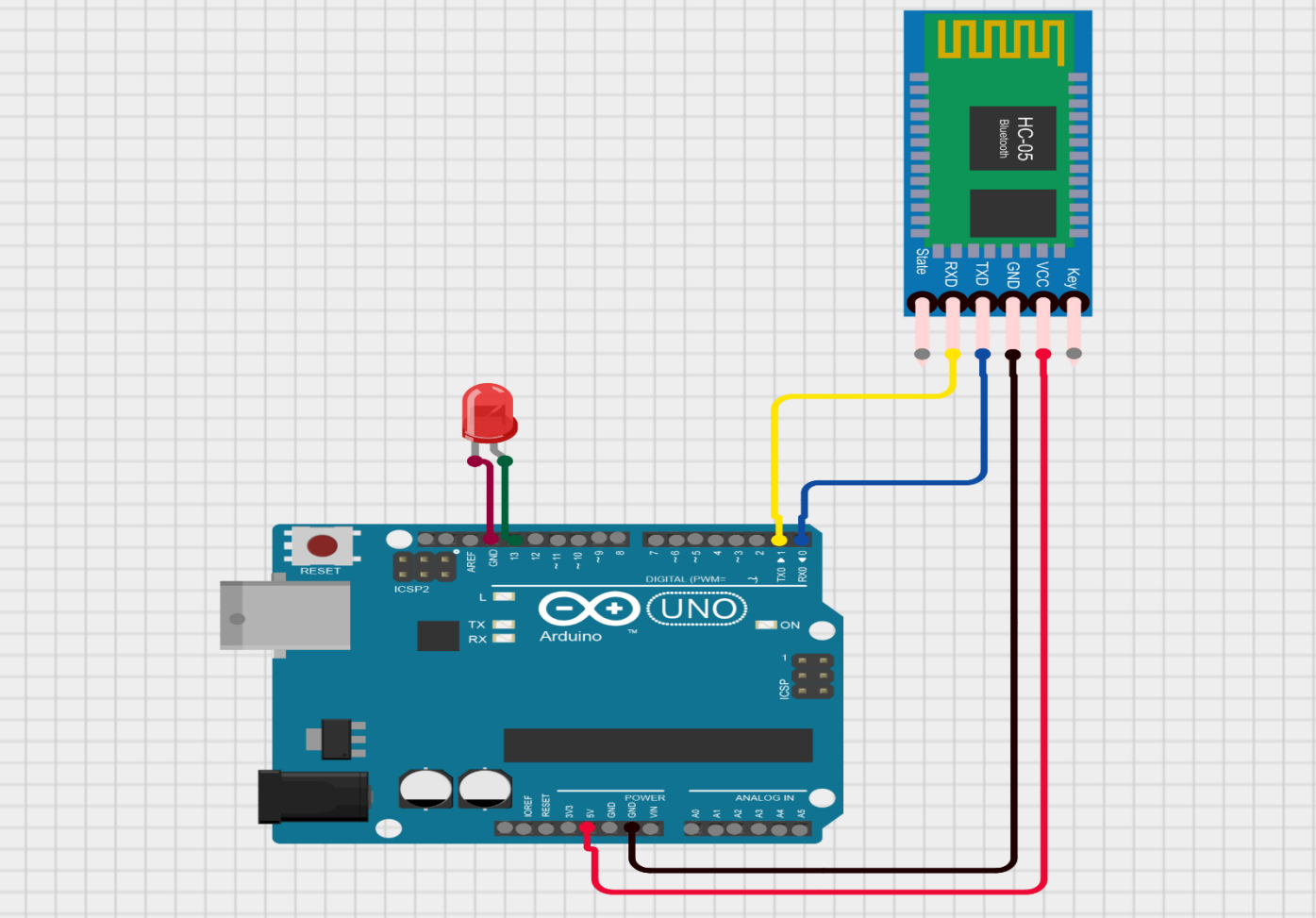
SCHEMATIC DIAGRAM:



CODING:

```
const int led = 13;


void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}


void loop() {
  if (Serial.available() > 0) {
    char data = Serial.read();


    if (data == '1') {
      digitalWrite(led, HIGH);
      Serial.println(“LED ON.”);
    }
    else if (data == '0') {
      digitalWrite(led, LOW);
      Serial.println(“LED OFF.”);
    }
  }
}
```

CIRCUIT DIAGRAM:



OUTPUT:

8. Write a program to measure the Soil Moisture using Arduino and Soil Moisture Sensor.

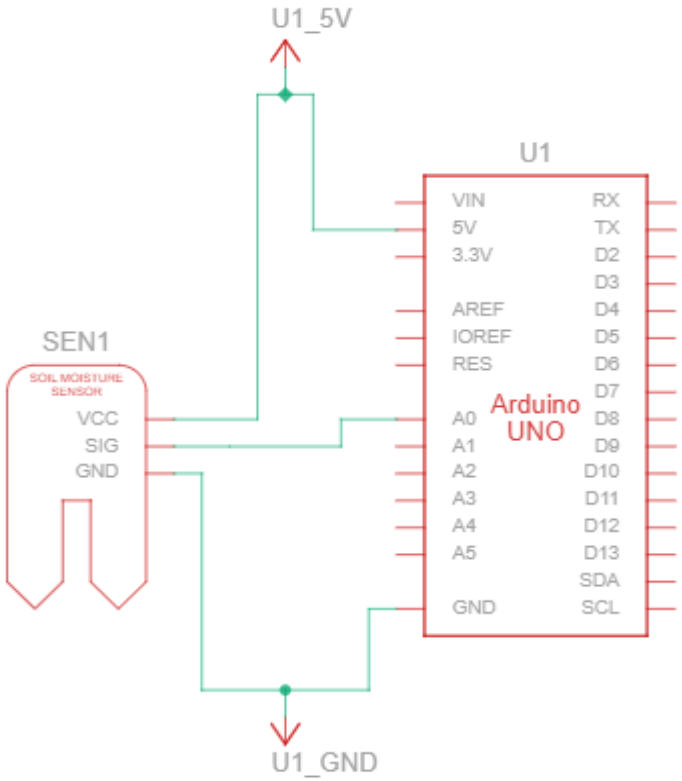
COMPONENTS REQUIRED:

- Arduino UNO.
- Soil Moisture Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- Sensor's VCC pin to 5V in Arduino UNO.
- Sensor's GND pin to GND in Arduino UNO.
- Sensor's SIGNAL pin to pin A0 in Arduino UNO.

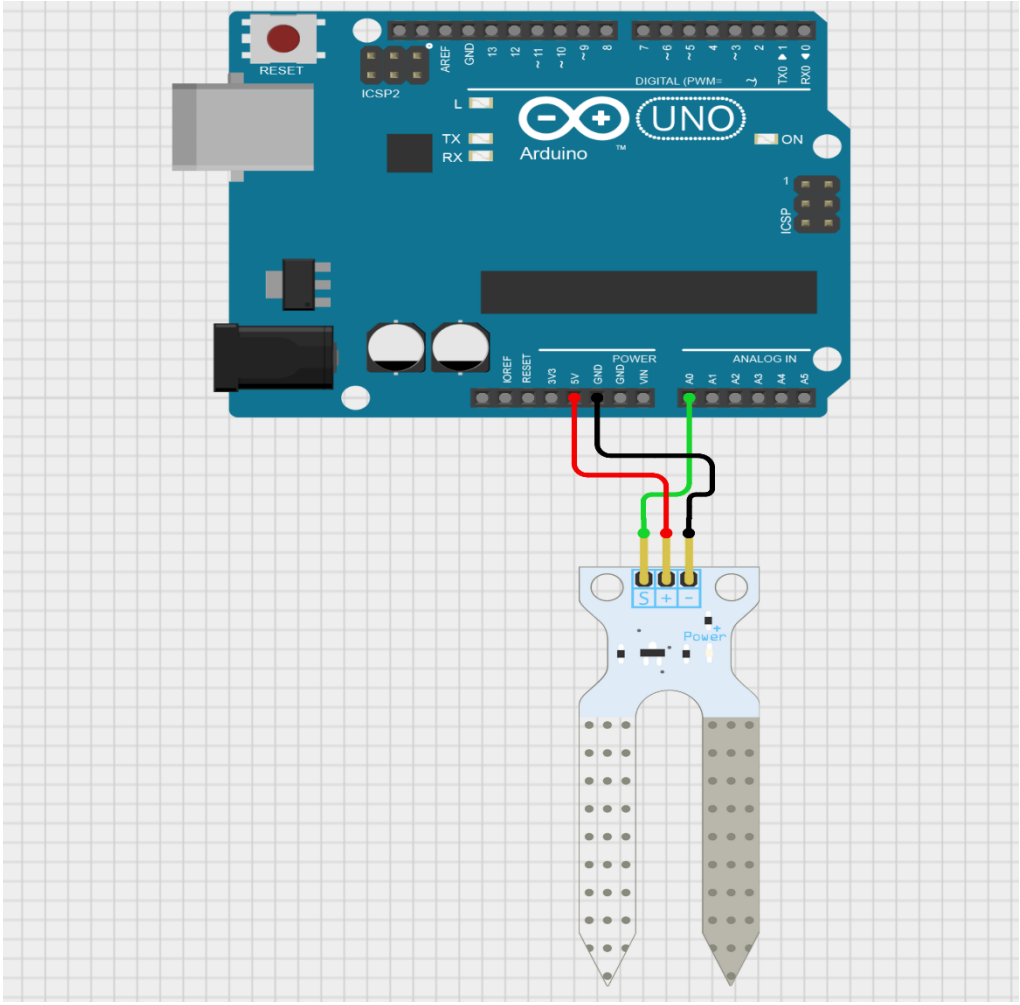
SCHEMATIC DIAGRAM:



CODING:

```
const int soilmoisture = A0;


void setup() {
  Serial.begin(9600);
  pinMode(soilmoisture, INPUT);
}


void loop() {
  int sensorValue = analogRead(soilmoisture);


  if (sensorValue < 0 || sensorValue > 1023) {
    Serial.println("Error: Sensor value out of range!");
  } else {
    int moisturePercent = map(sensorValue, 1023, 0, 0, 100);


    Serial.print("Soil Moisture: ");
    Serial.print(moisturePercent);
    Serial.println(" %");
  }


  delay(1000);
}
```

CIRCUIT DIAGRAM:



OUTPUT:

Serial Monitor

```
Soil Moisture: 100 %
Soil Moisture: 100 %
Soil Moisture: 100 %
Soil Moisture: 100 %
Soil Moisture: 100 %
Soil Moisture: 100 %
Soil Moisture: 17 %
Soil Moisture: 82 %
Soil Moisture: 14 %
Soil Moisture: 28 %
Soil Moisture: 61 %
Soil Moisture: 99 %
Soil Moisture: 99 %
Soil Moisture: 79 %
Soil Moisture: 38 %
Soil Moisture: 16 %
Soil Moisture: 14 %
Soil Moisture: 82 %
Soil Moisture: 99 %
Soil Moisture: 14 %
Soil Moisture: 43 %
Soil Moisture: 38 %
Soil Moisture: 38 %
```

9. Write a program to use DHT-11 sensor to measure the Temperature and Humidity using Arduino using Python.

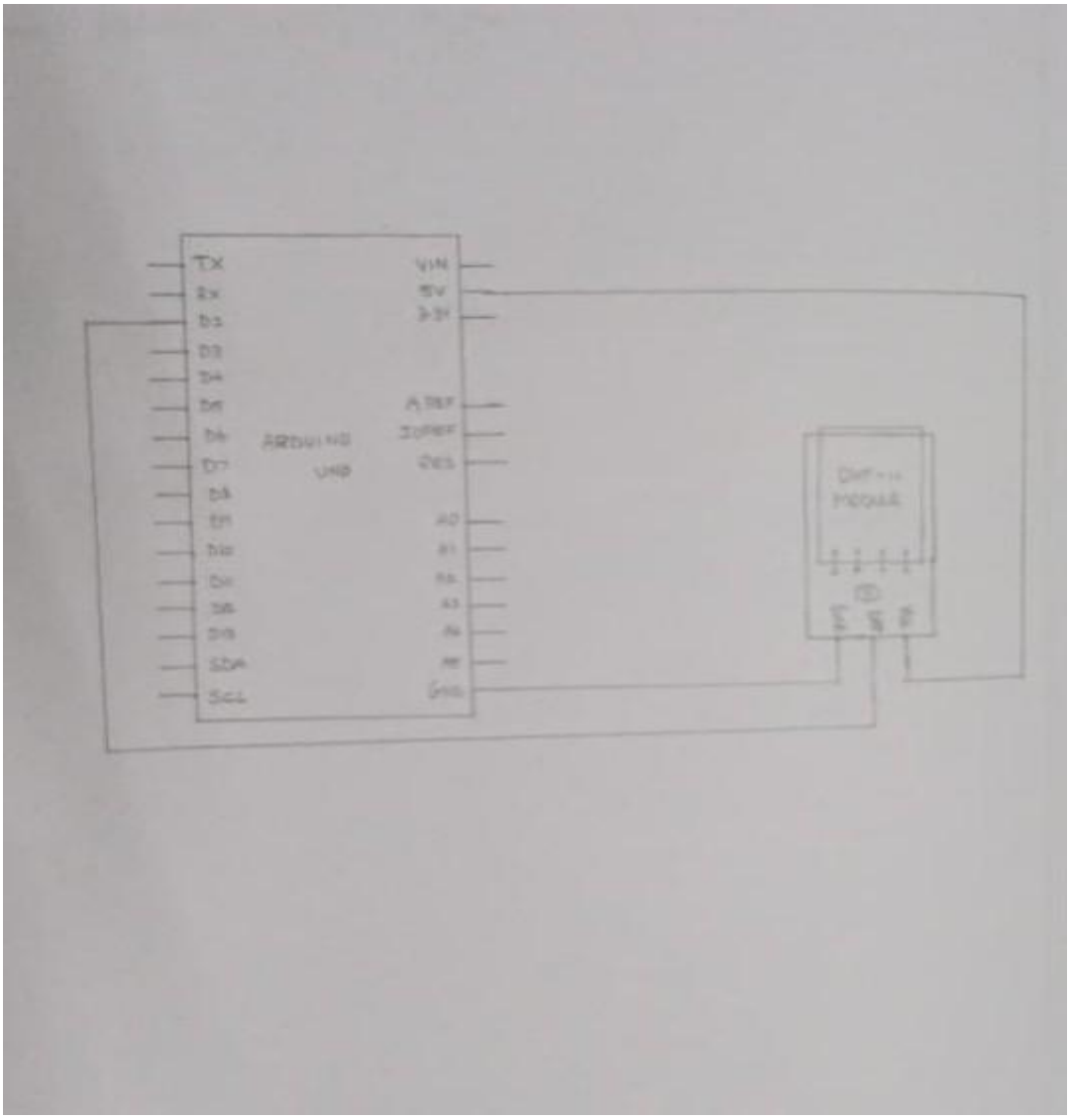
COMPONENTS REQUIRED:

- Arduino UNO.
- DHT-11 Module.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- Module's VCC pin to 5V in Arduino UNO.
- Module's GND pin to GND in Arduino UNO.
- Module's OUT pin to pin 2 in Arduino UNO.

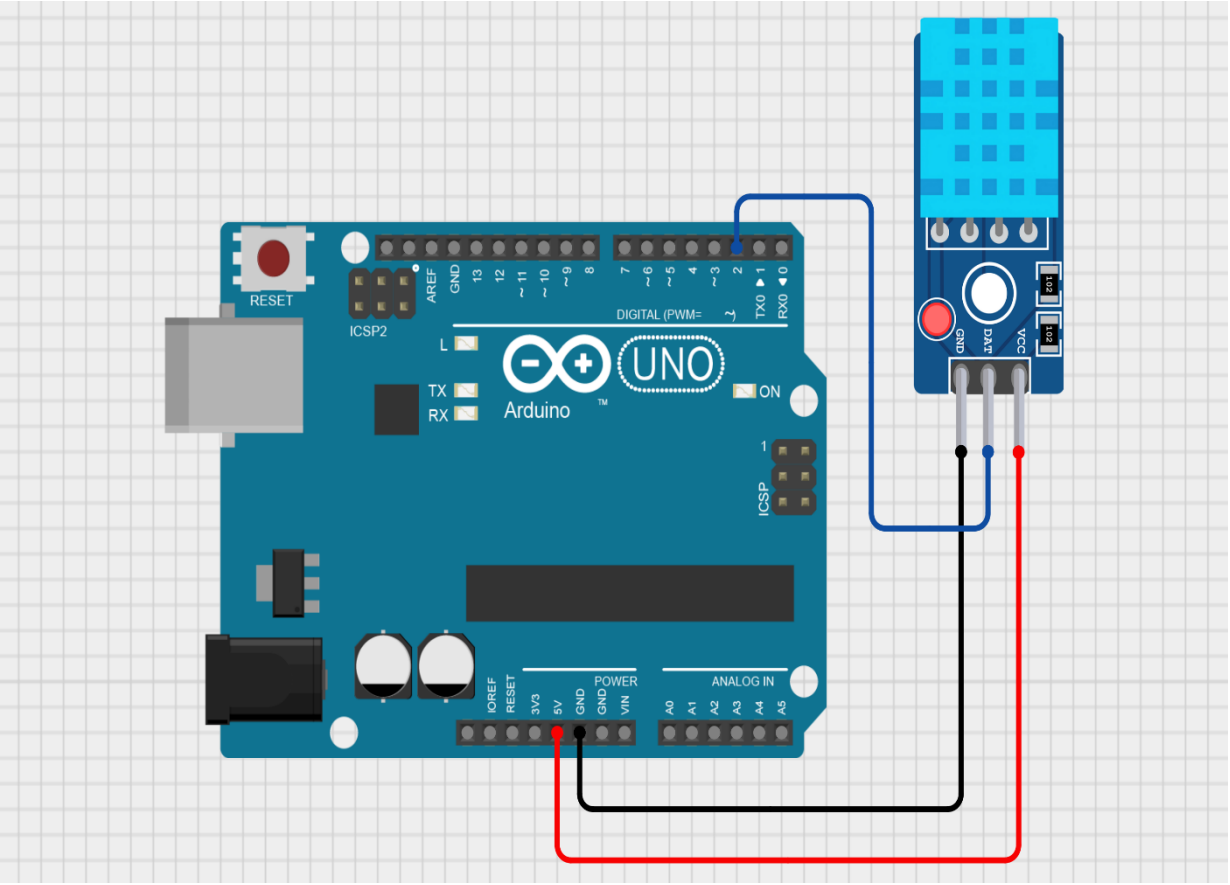
SCHEMATIC DIAGRAM:



CODING:

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000);

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();



  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" % \t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" *C");
}
```

CIRCUIT DIAGRAM:



OUTPUT:

PYTHON CODE:

```
import serial
import time

arduino = serial.Serial(port='COM3',9600)
time.sleep(2)

print("Reading data from Arduino...")

while True:
    try:
        data = arduino.readline().decode().strip()
        if data:
            if "," in data:
                temperature, humidity = data.split(",")
                print(f"Temperature: {temperature} °C, Humidity: {humidity} %")
            else:
                print("Arduino says:", data)
    except Exception as e:
        print("Error:", e)
```

10. Write a program to measure Distance using Ultrasonic Sensor and Arduino.

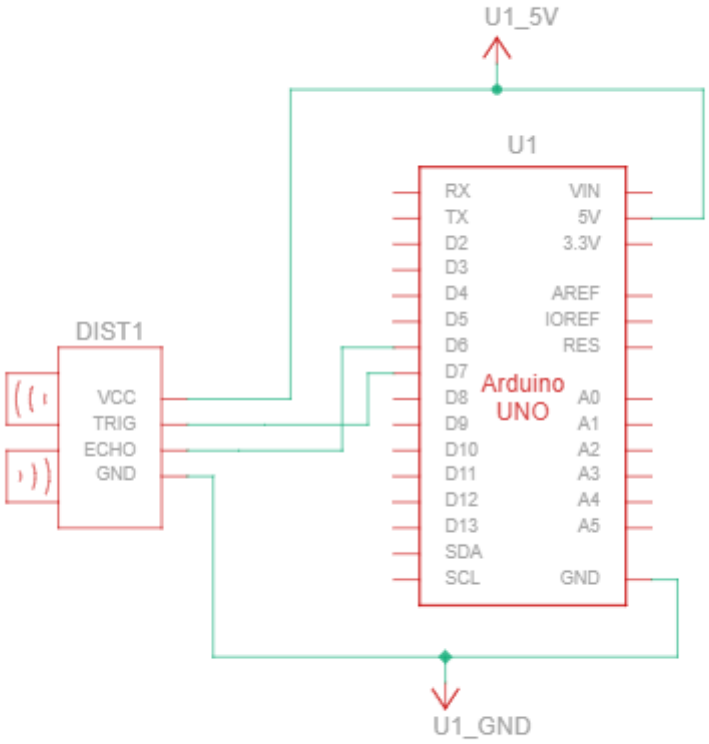
COMPONENTS REQUIRED:

- Arduino UNO.
- Ultrasonic Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- Sensor's VCC pin to 5V in Arduino UNO.
- Sensor's GND pin to GND in Arduino UNO.
- Sensor's TRIG pin to pin 7 in Arduino UNO.
- Sensor's ECHO pin to pin 6 in Arduino UNO.

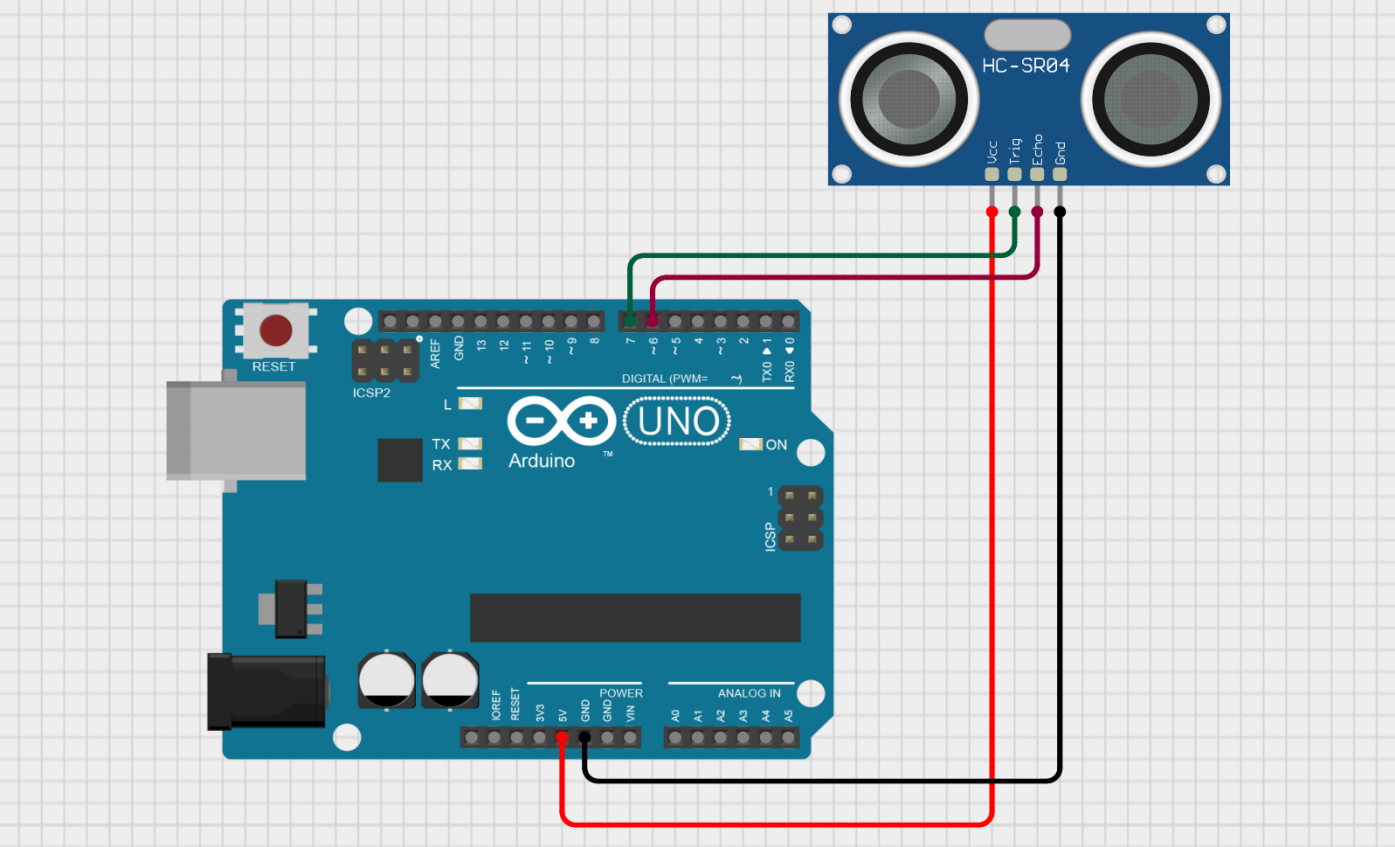
SCHEMATIC DIAGRAM:



CODING:

```
const int trigger = 7;
const int echo = 6;

void setup() {
  pinMode(trigger,OUTPUT);
  pinMode(echo,INPUT);
  Serial.begin(9600);
}

void loop() {
  long duration;
  long distance;

  digitalWrite(trigger,LOW);
  delayMicroseconds(2);
  digitalWrite(trigger,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger,LOW);

  duration = pulseIn(echo,HIGH);
  distance = ((duration/29)/2);
  Serial.print("Distance:");
  Serial.print(distance);
  Serial.println("cm");
  delay(200);
}
```

CIRCUIT DIAGRAM:



OUTPUT:

Serial Monitor

```
Distance:93cm
Distance:171cm
Distance:71cm
Distance:86cm
Distance:184cm
Distance:336cm
Distance:336cm
Distance:336cm
Distance:199cm
Distance:256cm
Distance:127cm
Distance:336cm
Distance:193cm
Distance:221cm
Distance:171cm
Distance:223cm
Distance:262cm
Distance:164cm
Distance:213cm
Distance:259cm
Distance:163cm
Distance:217cm
Distance:259cm
Distance:132cm
Distance:175cm
Distance:158cm
Distance:179cm
```

11.Detect the vibration of an object using Arduino.

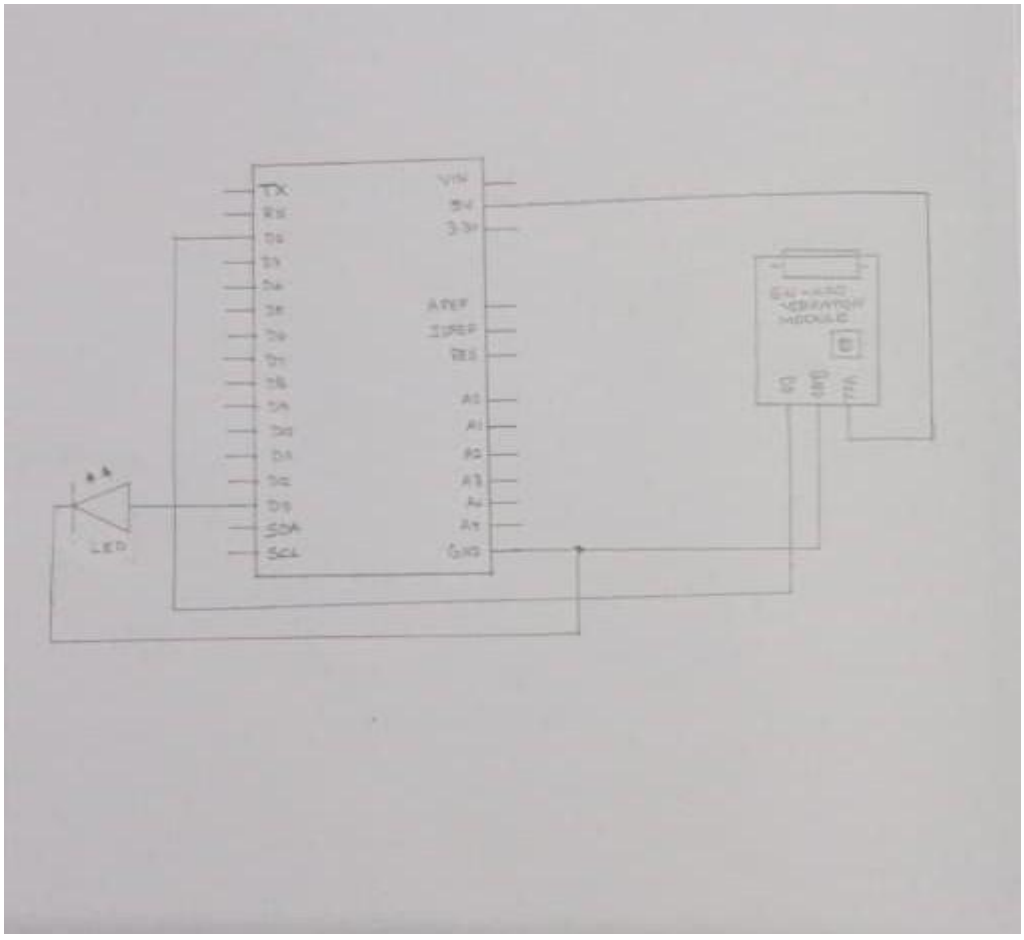
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- Vibration Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- LED Bulb's Anode to pin 13 in Arduino UNO.
- LED Bulb's Cathode to GND in Arduino UNO.
- Sensor's VCC pin to 5V in Arduino UNO.
- Sensor's GND pin to GND in Arduino UNO.
- Sensor's OUT pin to pin 2 in Arduino UNO.

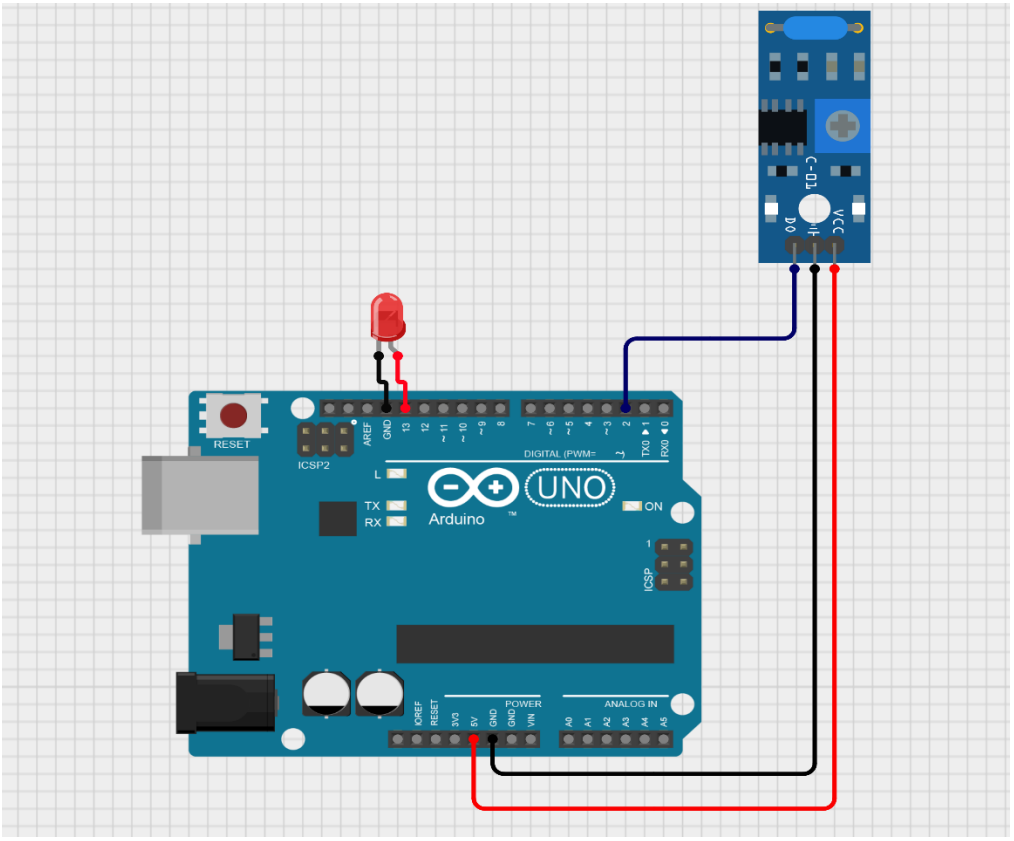
SCHEMATIC DIAGRAM:



CODING:

```
const int vibrationSensor = 2;
const int led = 13;

 void setup() {
 pinMode(vibrationSensor, INPUT);
 pinMode(led, OUTPUT);
 Serial.begin(9600);
 Serial.println("Vibration Monitoring Started...");
}

void loop() {
 int sensorValue = digitalRead(vibrationSensor);
 if (sensorValue != HIGH && sensorValue != LOW) {
  Serial.println(" ⚠️ Error: Invalid sensor reading!");
  digitalWrite(led, LOW);
  delay(500);
  return;
 }

 if (sensorValue == HIGH) {
  digitalWrite(led, HIGH);
  Serial.println(" ✅ Vibration Detected.");
 } else {
  digitalWrite(led, LOW);
  Serial.println(" ⛔ No Vibration Detected.");
 }
 delay(100);
}
```

CIRCUIT DIAGRAM:



OUTPUT:

12.Detect the object using IR sensor.

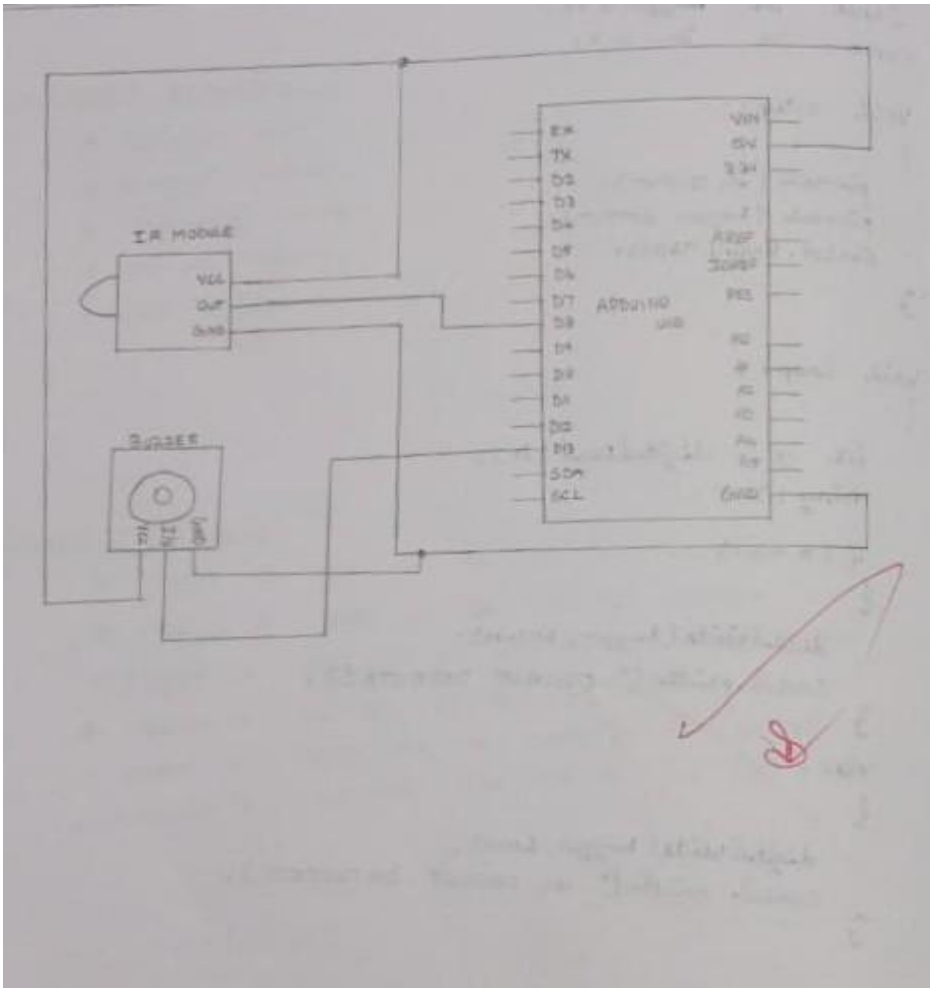
COMPONENTS REQUIRED:

- Arduino UNO.
- Buzzer Module.
- InfraRed Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- Buzzer Module’s VCC to pin 5V in Arduino UNO.
- Buzzer Module’s GND to GND in Arduino UNO.
- Buzzer Module’s I/O to pin 13 in Arduino UNO.
- Sensor’s VCC pin to 5V in Arduino UNO.
- Sensor’s GND pin to GND in Arduino UNO.
- Sensor’s OUT pin to pin 8 in Arduino UNO.

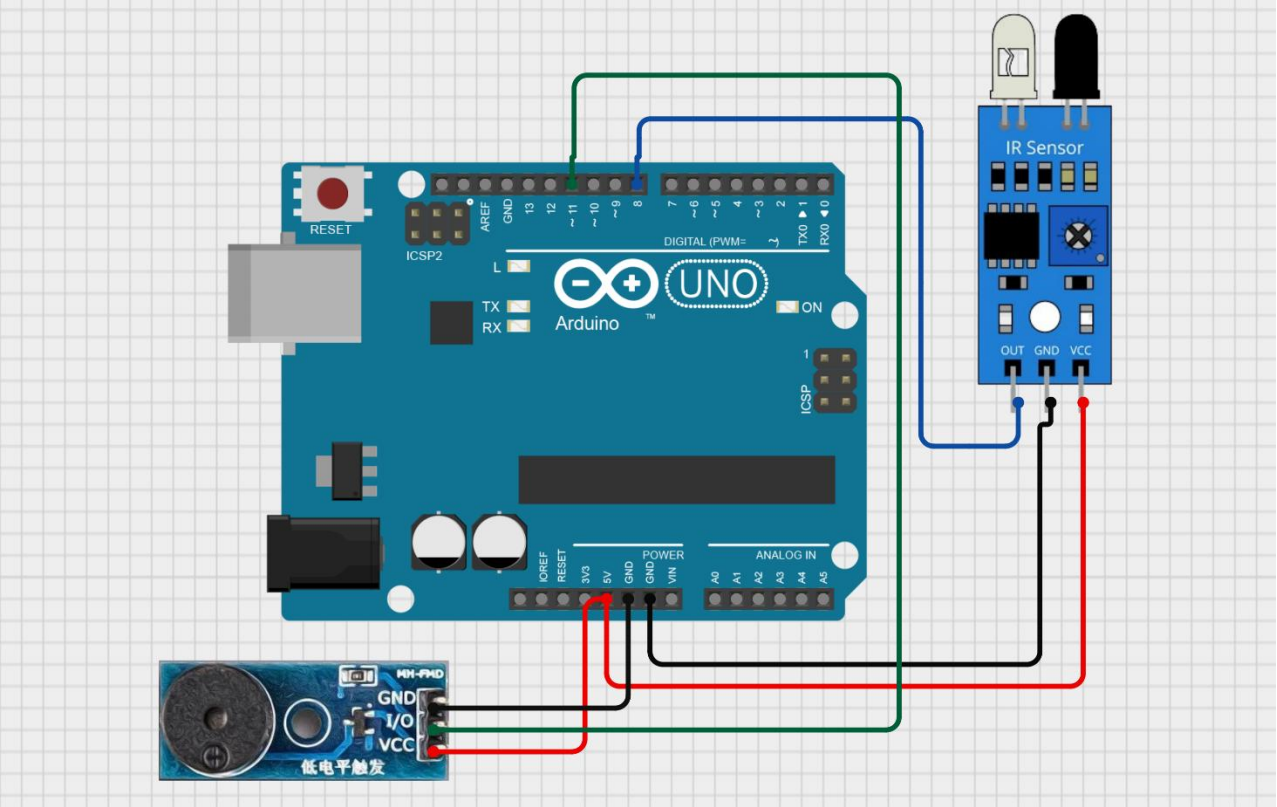
SCHEMATIC DIAGRAM:



CODING:

```
const int led = 13;
const int ir = 8;

void setup(){
  pinMode(led,OUTPUT);
  pinMode(ir,INPUT);
  Serial.begin(9600);
}

void loop(){
  int x = digitalRead(ir);
  delay(300);

  if(x==1)
  {
    digitalWrite(led,LOW);
    Serial.println("OBJECT DETECTED");
  }
  else
  {
    digitalWrite(led,HIGH);
    Serial.println("NO OBJECT DETECTED");
  }
}
```

CIRCUIT DIAGRAM:



OUTPUT:

Serial Monitor

```
OBJECT DETECTED
OBJECT DETECTED
OBJECT DETECTED
NO OBJECT DETECTED
NO OBJECT DETECTED
NO OBJECT DETECTED
NO OBJECT DETECTED
NO OBJECT DETECTED
NO OBJECT DETECTED
NO OBJECT DETECTED
OBJECT DETECTED
OBJECT DETECTED
OBJECT DETECTED
```

13.Sense a Finger when it is placed on Board using Arduino.

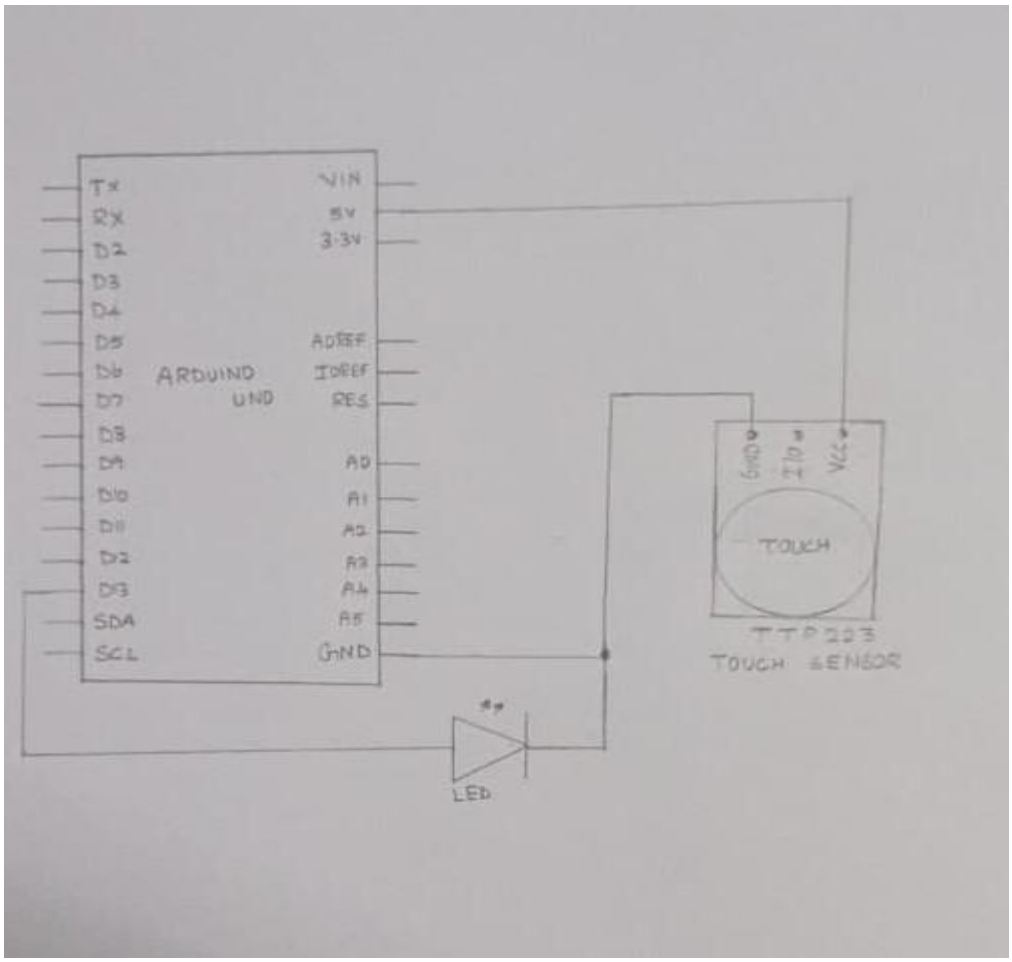
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- TTP 223 Touch Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- LED Bulb's Anode to pin 13 in Arduino UNO.
- LED Bulb's Cathode to GND in Arduino UNO.
- Sensor's VCC pin to 5V in Arduino UNO.
- Sensor's GND pin to GND in Arduino UNO.
- Sensor's OUT pin to pin 2 in Arduino UNO.

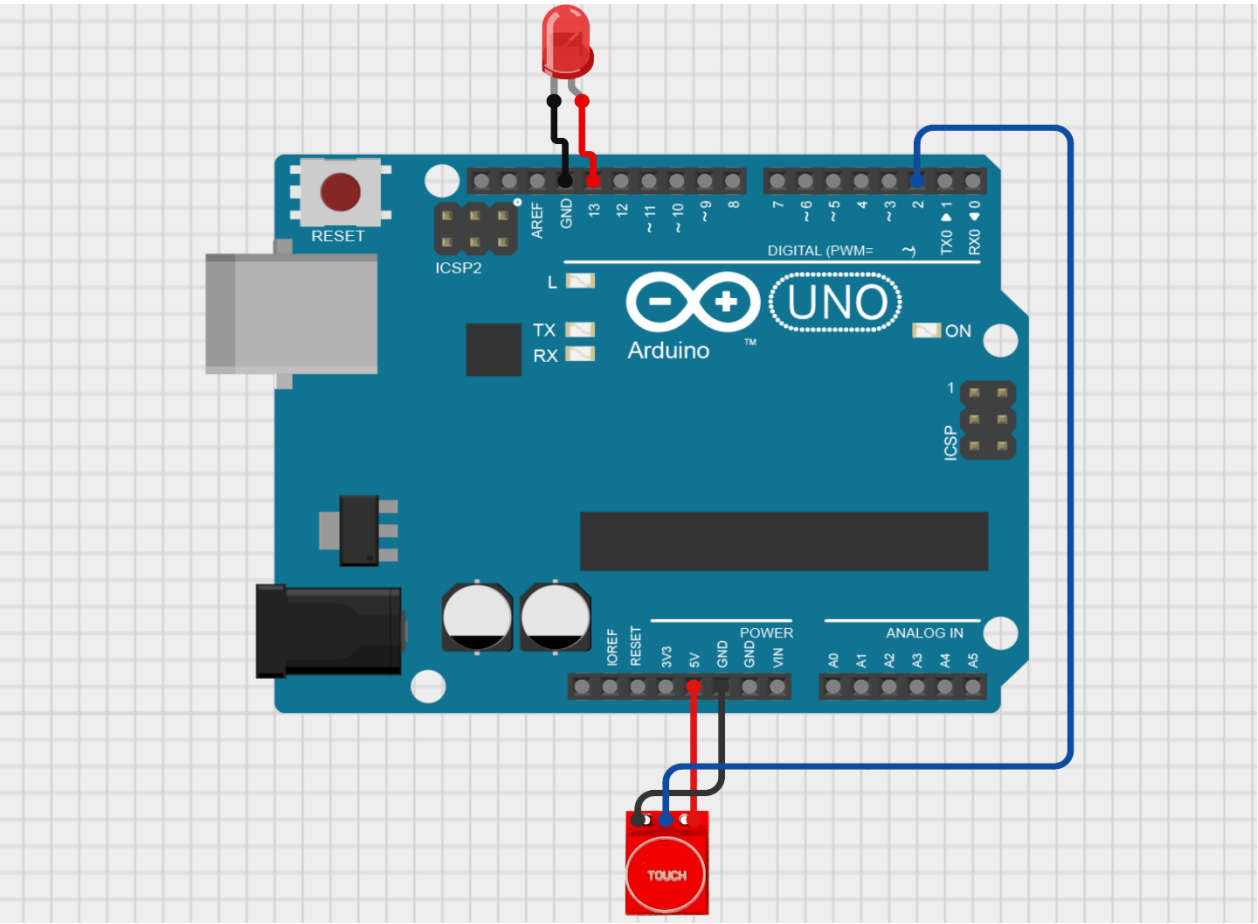
SCHEMATIC DIAGRAM:



CODING:

```
const int touchPin = 2;
const int ledPin = 13;

void setup() {
  pinMode(touchPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int touchState = digitalRead(touchPin);
  Serial.println(touchState);

  if (touchState == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }

  delay(100);
}
```

CIRCUIT DIAGRAM:



OUTPUT:

14. Write a program for Temperature Notification using Arduino.

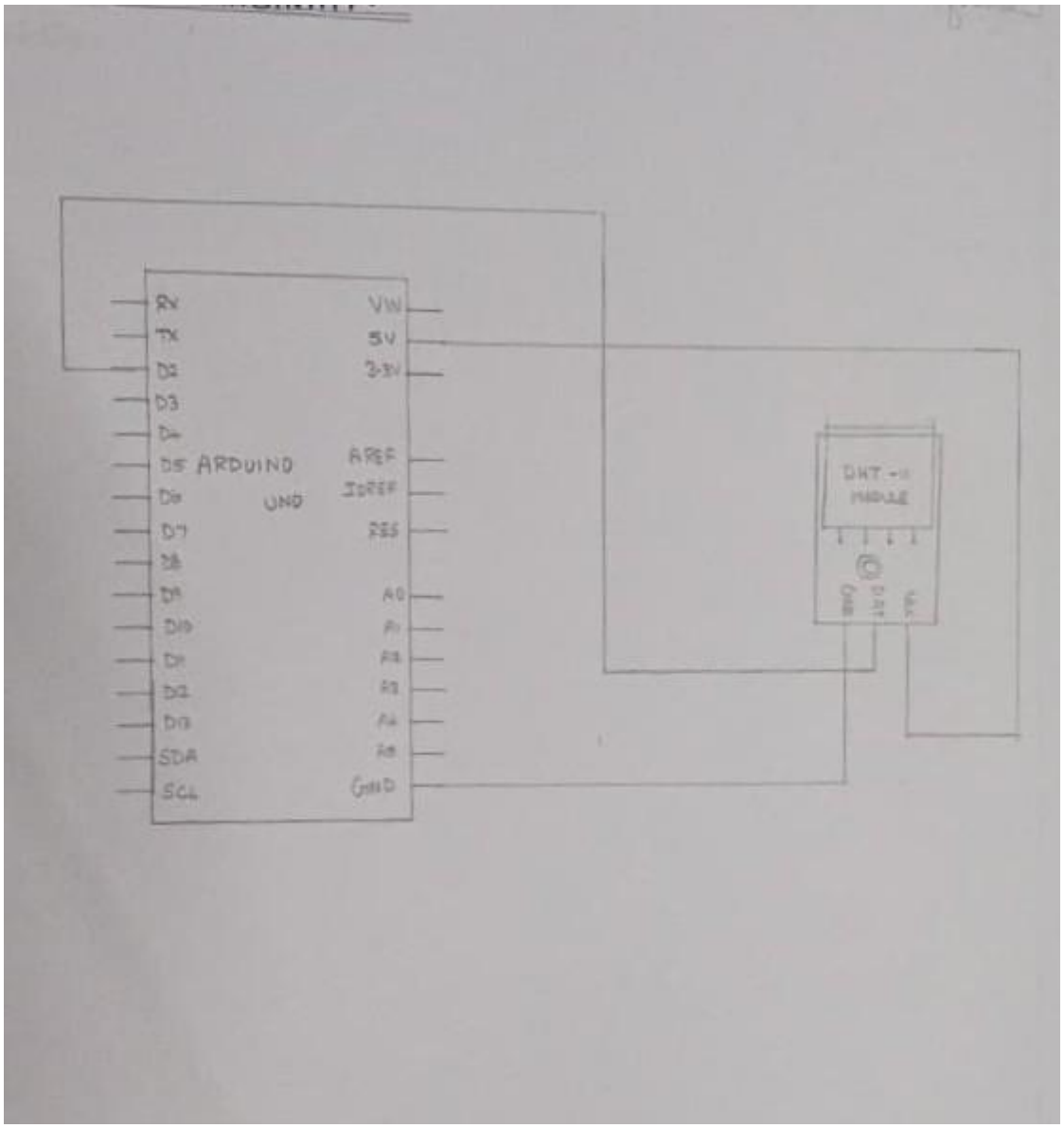
COMPONENTS REQUIRED:

- Arduino UNO.
- DHT-11 Sensor.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- Sensor’s VCC pin to 5V in Arduino UNO.
- Sensor’s GND pin to GND in Arduino UNO.
- Sensor’s DAT pin to pin 2 in Arduino UNO.

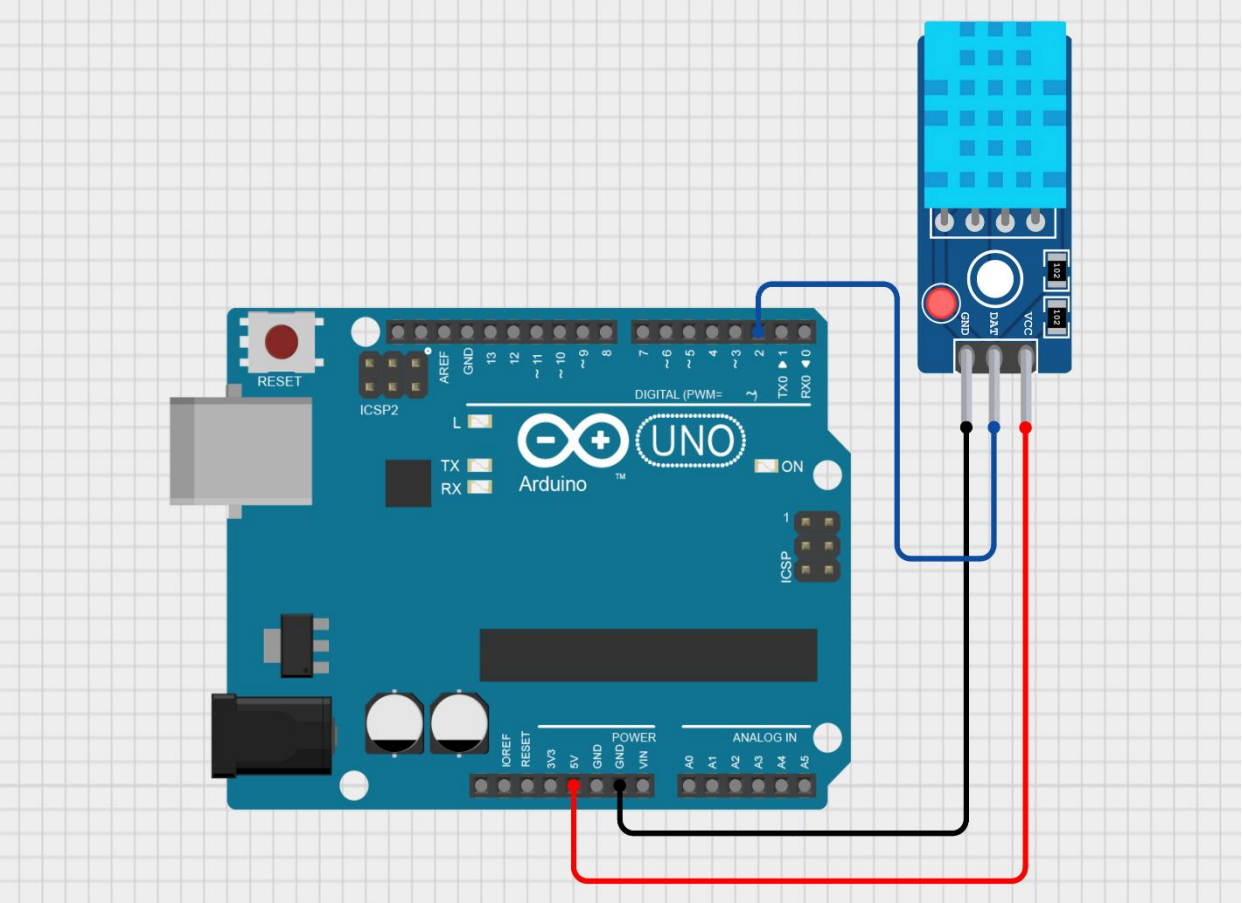
SCHEMATIC DIAGRAM:



CODING:

```
#include <DHT.h>

#define DHTPIN 2

#define DHTTYPE DHT11

#define TEMP_THRESHOLD 30


DHT dht(DHTPIN, DHTTYPE);


void setup()

{

  Serial.begin(9600);

  dht.begin();

}


void loop()

{

  delay(2000);

  float temperature = dht.readTemperature();

  if (isnan(temperature))

  {

    Serial.println("Failed to read from DHT sensor!");

    return;

  }

  Serial.print("Temperature: ");

  Serial.print(temperature);

  Serial.println(" °C");
```

CIRCUIT DIAGRAM:



OUTPUT:

```
if (temperature > TEMP_THRESHOLD)
  {
    Serial.println("Temperature exceeds threshold!");


    }
}
```

15.Use LDR to vary the Light Intensity of LED using Arduino.

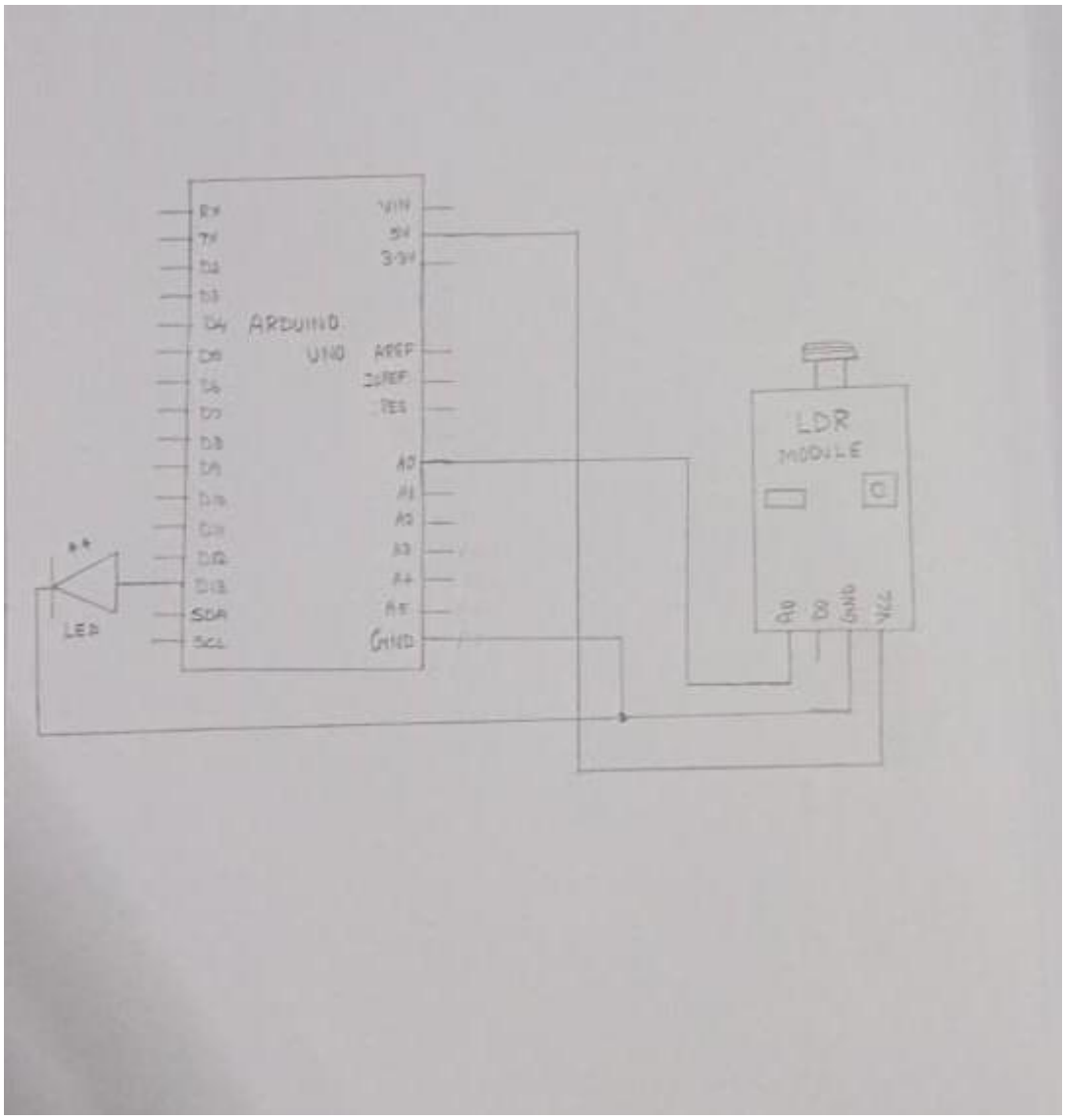
COMPONENTS REQUIRED:

- Arduino UNO.
- LED.
- LDR.
- Jumper Wires.
- USB Cable.

CONNECTIONS:

- LDR Module's VCC pin to 5V in Arduino UNO.
- LDR Module's GND pin to GND in Arduino UNO.
- LDR Module's A0 pin to A0 pin in Arduino UNO.

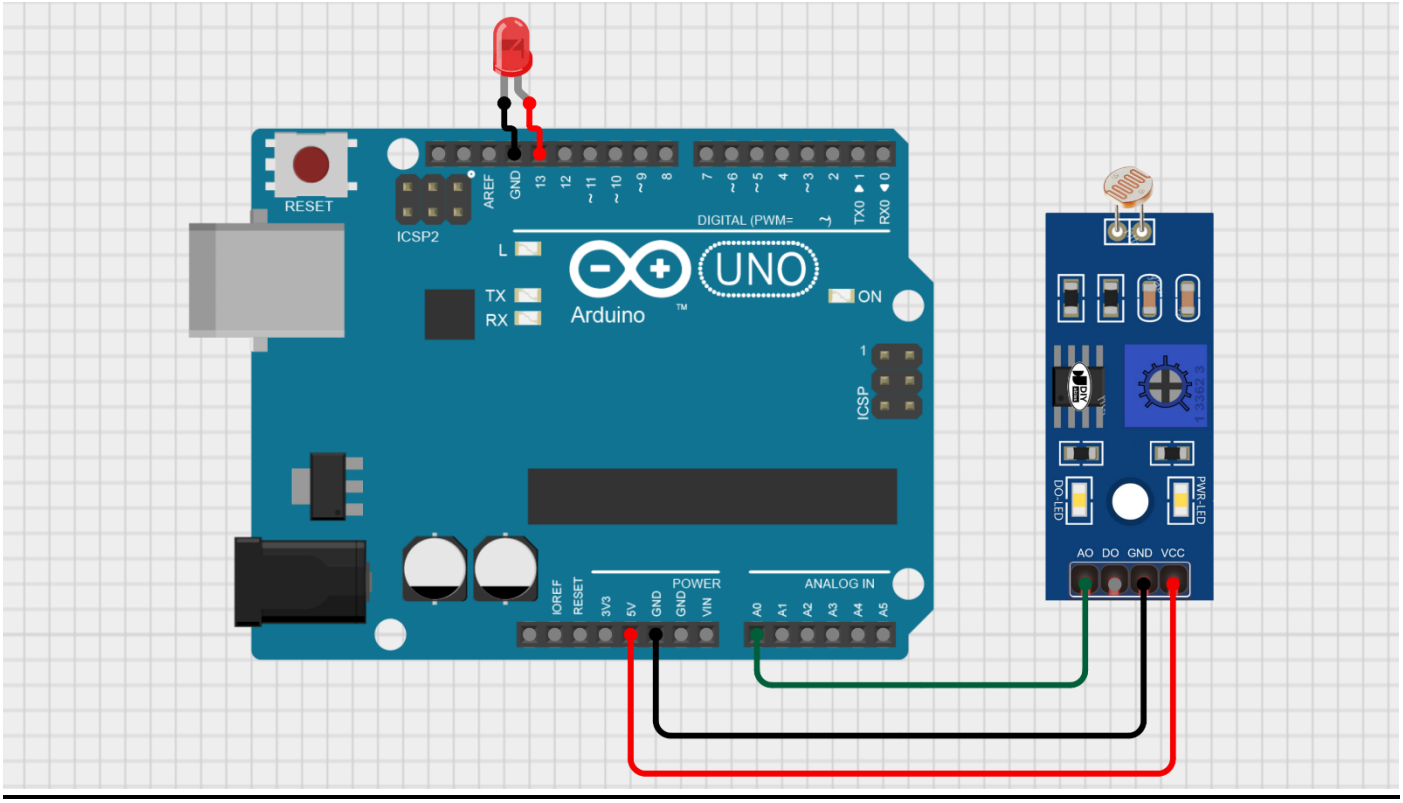
SCHEMATIC DIAGRAM:



CODING:

```
const int ldrPin = A0;
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int ldrValue = analogRead(ldrPin);
  int ledBrightness = map(ldrValue, 0, 1023, 0, 255);
  analogWrite(ledPin, ledBrightness);

  Serial.print("Light Intensity: ");
  Serial.print(ldrValue);
  Serial.println(" lx");
  delay(100);
}
```

CIRCUIT DIAGRAM:



OUTPUT: