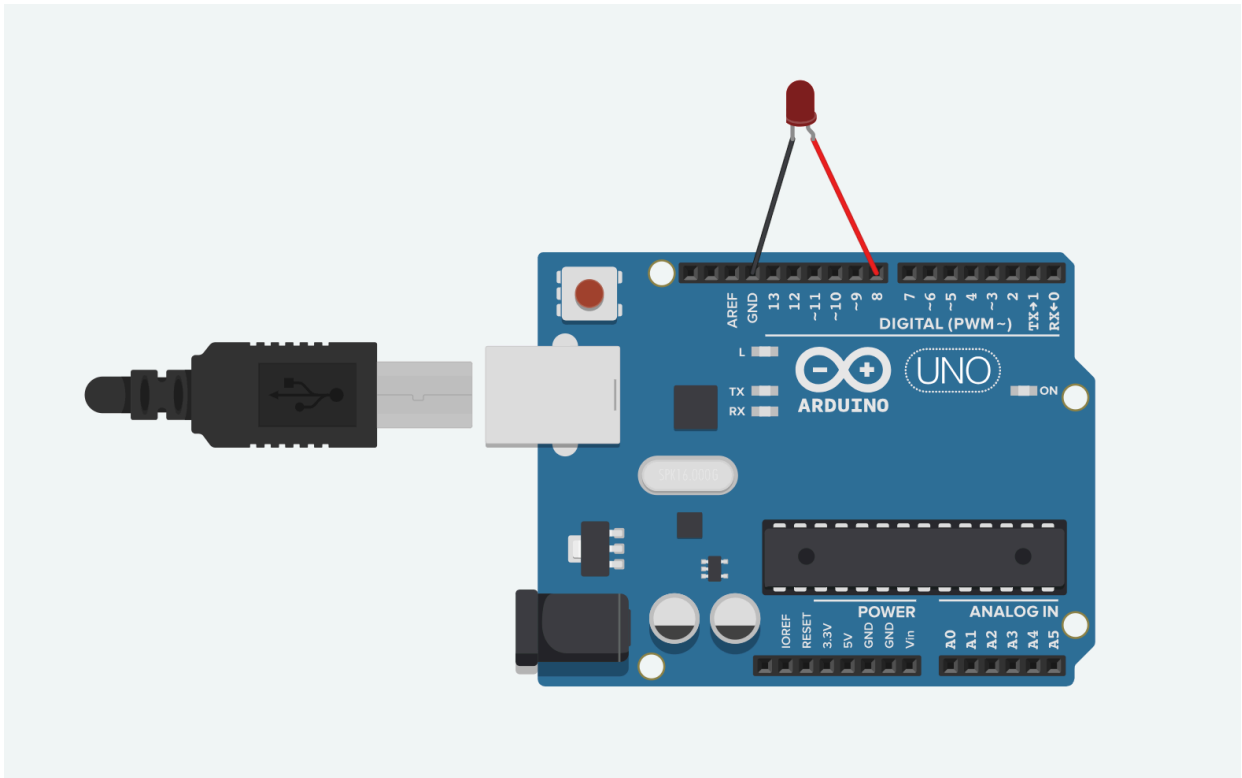


lot programs

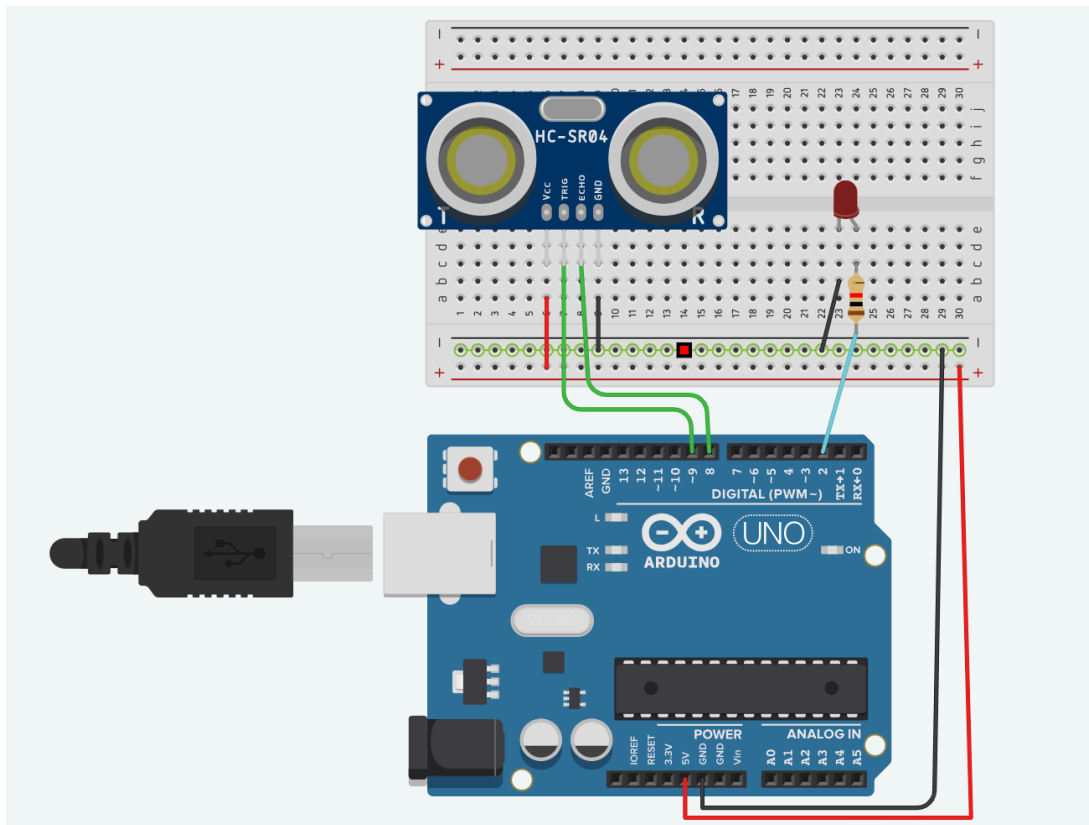
1. Experiments to switch on/off LEDs with the user's choice.



```
// C++ code
String opt;
void setup()
{
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}

void loop()
{
  Serial.println("Enter O to ON & F for OFF");
  opt=Serial.readString();
  if(opt=="O")
    digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  if(opt=="F")
    digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

2. Experiment with IR/Ultrasonic sensor and LED.



```
#define trigPin 9
#define echoPin 8
long duration;
int distance;

void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distance = duration * 0.034 / 2;
  if(distance <= 20)
  {
```

```

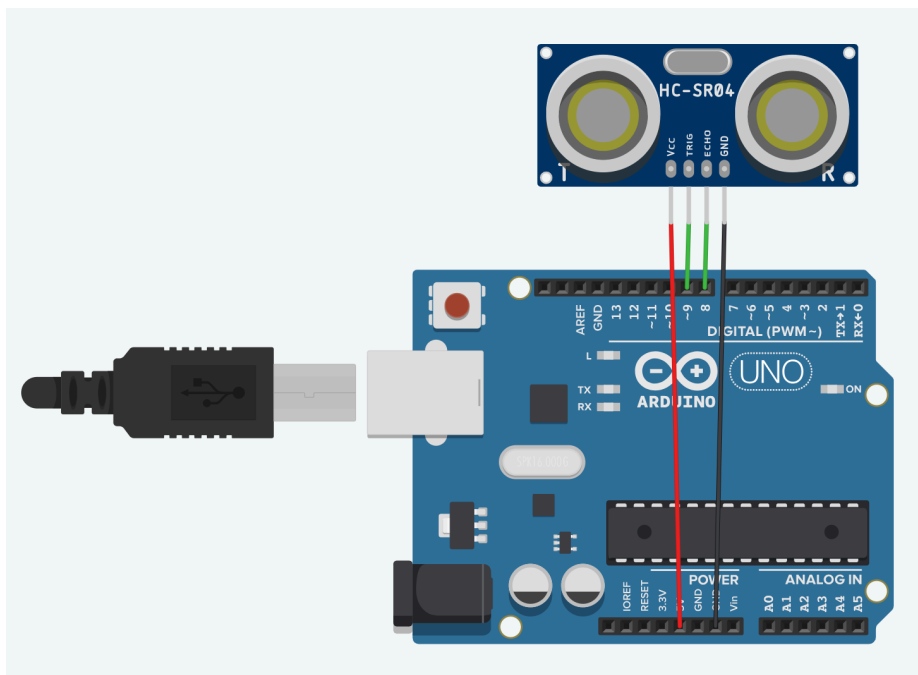
digitalWrite(2, HIGH);
Serial.print("Light is ON");

    delay(1000);
}
else{
    digitalWrite(2, LOW);
    Serial.print("Light is OFF");
}

    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
    delay(1000);
}

```

3. Experiment to measure the distance using ultrasonic sensors.



```

// C++ code
//
#define trigPin 9
#define echoPin 8
long duration;
int distance;

void setup()
{
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop()
{
    digitalWrite(trigPin, LOW);

```

```

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

duration = pulseIn(echoPin,HIGH);

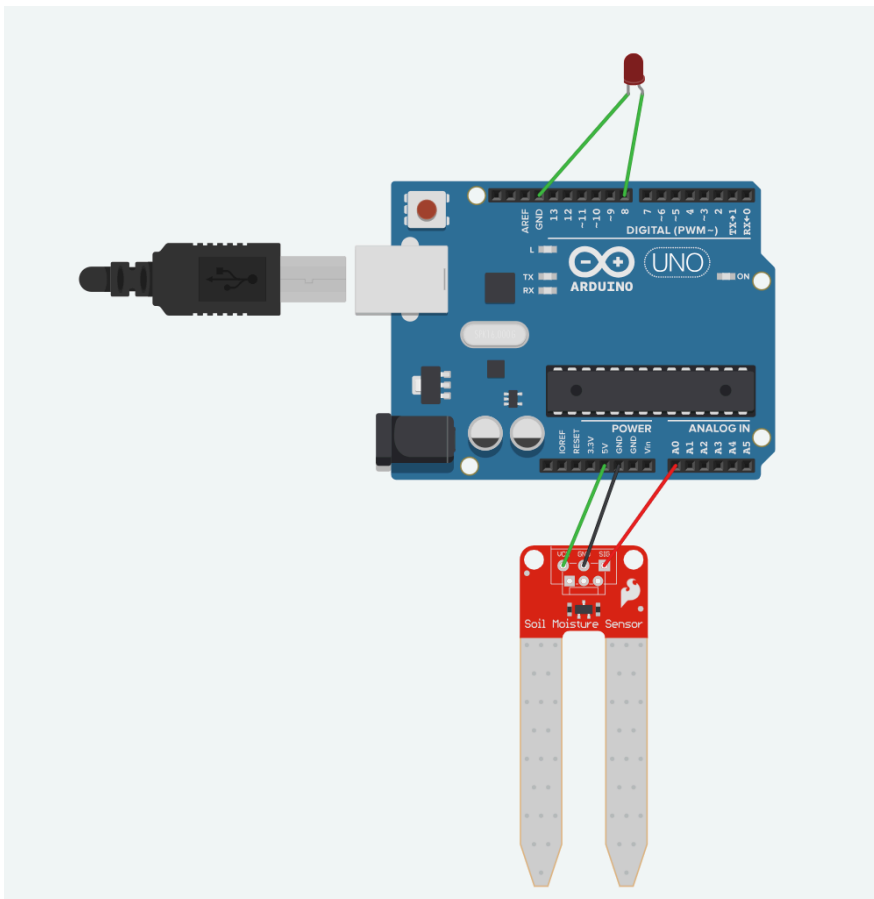
distance = duration * 0.034 / 2;

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

delay(1000);
}

```

4. Experiments to check moisture



```

// C++ code
//
void setup()
{
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}

void loop()
{
  int level;

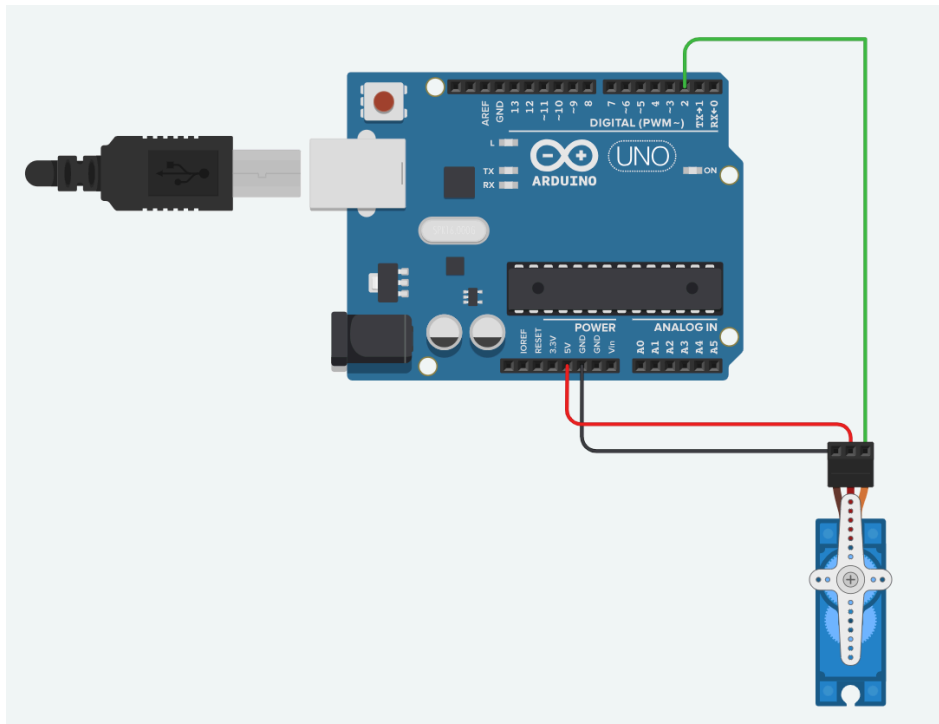
```

```

level=analogRead(0);
if(level<5){
    digitalWrite(8, HIGH);
    Serial.println("No Water, Switching on motor");
}
else
    digitalWrite(8, LOW);
Serial.println("Analog Level");
Serial.println(level);
delay(1000); // Wait for 1000 millisecond(s)
}

```

5. Experiment with servo motor.



```

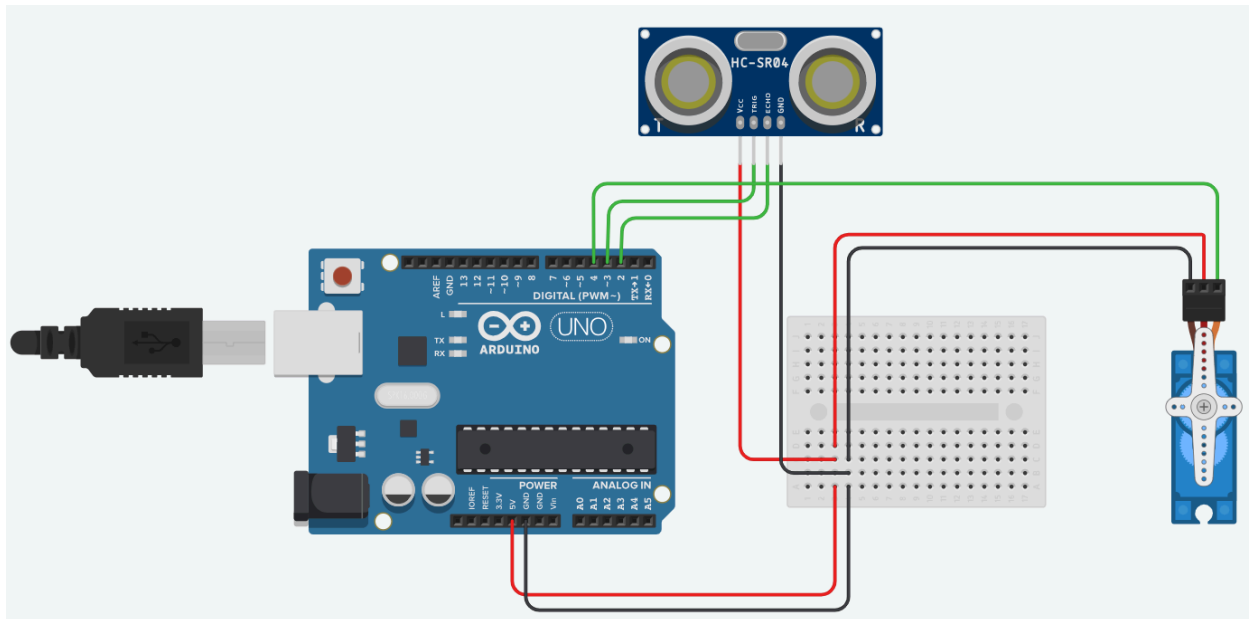
// C++ code
//
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(2);
}

void loop()
{
    for(int pos=0;pos<=180;pos++){
        myservo.write(pos);
        delay(15);
    }
    for(int pos=180;pos>=0;pos--){
        myservo.write(pos);
        delay(15);
    }
}

```

}

6. Experiment with IR/Ultrasonic sensor and servo motor.



```
#define trigPin 3
#define echoPin 2
#include <Servo.h>
long duration;
int distance;
Servo myservo;
void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  myservo.attach(4);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

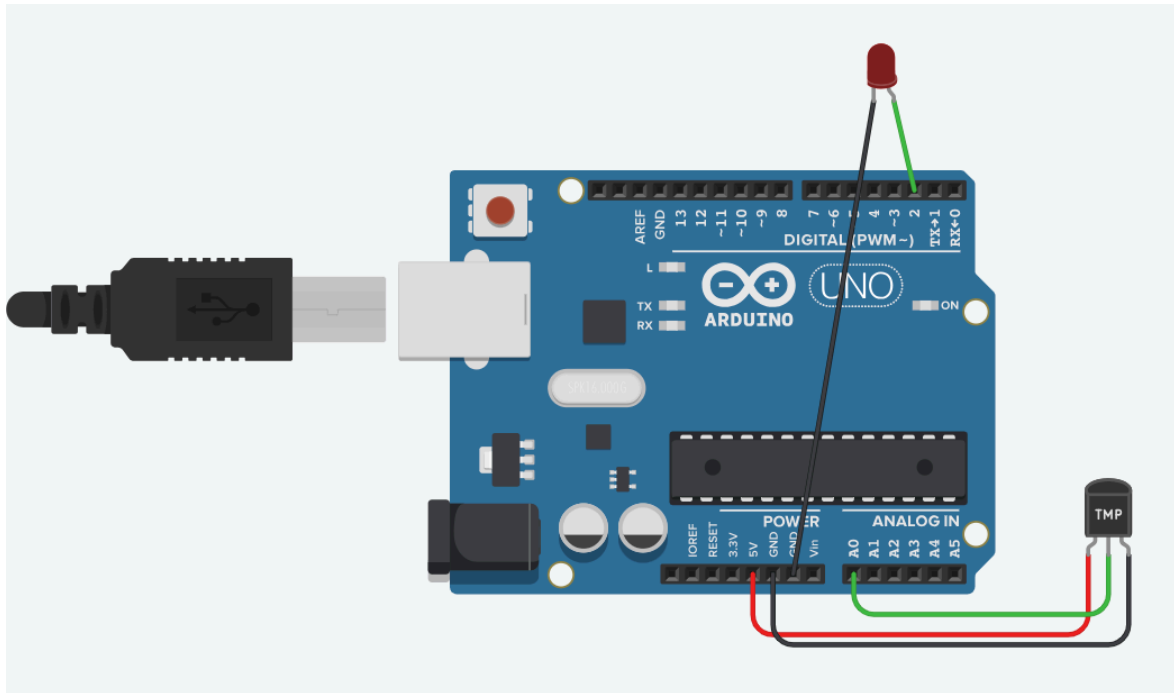
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  if(distance <= 30)
  {
    myservo.write(90);
```

```

    Serial.print("Servo rotate to 90D");
}
else{
    myservo.write(0);
    Serial.print("Servo rotate to 0D");
}
delay(1000);
}

```

7. Experiment to read the temperature and switch on the fan/LED.



```

const int tempPin = A0;
void setup()
{
    Serial.begin(9600);
    pinMode(2, OUTPUT);
}

void loop()
{
    int reading = analogRead(tempPin);
    //Convert reading to voltage
    float voltage = reading * 5.0/1023.0;
    //Convert voltage to temperature in Celsius
    float tempC = (voltage - 0.5) * 100.0;

    Serial.print("Temperature: ");
    Serial.print(tempC);
    Serial.println(" C");

    if(tempC > 30.0){
        digitalWrite(2, HIGH);
        Serial.print("Temperature more than 30C");
    }
}

```

8. Create a code to send email from the board. (gcr)

```
pi:~$ nano sendmimemail.py
```

9. IoT experiment to read the temperature and humidity and push it to IoT Cloud for visualization.

```
import sys
# import RPi.GPIO as GPIO
from time import sleep
# import Adafruit_DHT
import urllib3

def getSensorData():
    # RH,T=Adafruit_DHT.read_retry(Adafruit_DHT.DHT11,4)
    # return (str(RH),str(T))
    return (30,40) ## hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh hard coded

def main():

    print ('Starting')
    key='xxxxxxxxxxxxxxx' # Write API key from thingspeak
    baseURL=https://api.thingspeak.com/update?api\_key=%s'%key
    print ('*****')
    while True:
        try:
            RH,T=getSensorData()
            http = urllib3.PoolManager()
```



```

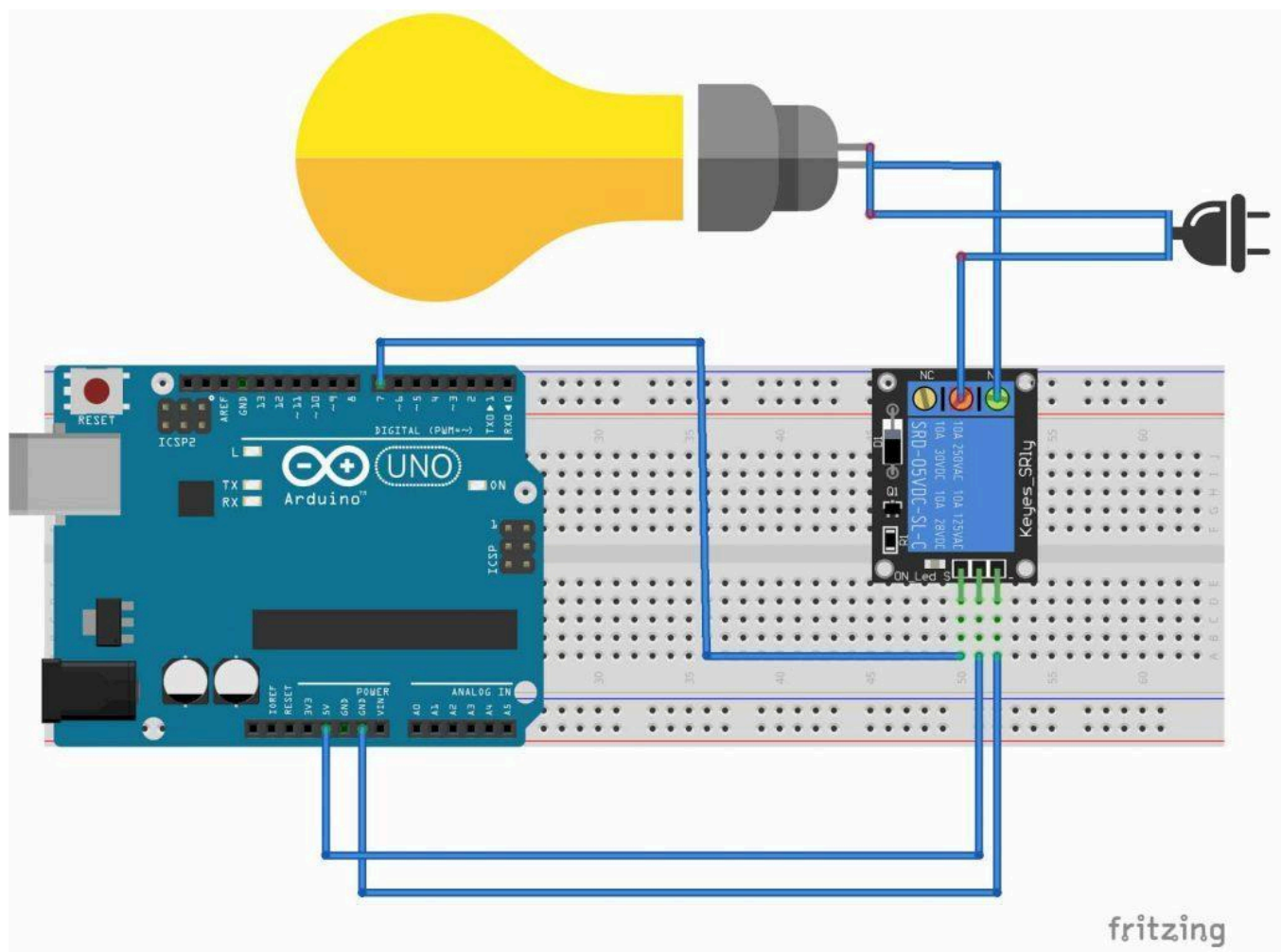
r = http.request('POST',url=baseURL+'&field1=%s &field2=%s'%(T,RH))
#req=urllib3.Request(url=baseURL+'&field1=%s &field2=%s'%(T,RH))
#f=urllib3.urlopen(req)
print(r.data)

#print (f.data())
print ("Humidity "+str(RH) +"%")
print ("Temperture "+str(T) +"C")

# f.close
sleep(15)
# GPIO.cleanup()
except:
    print (sys.exc_info()[0])
    break
#calling main method
main()

```

10. Experiment to switch on/off light using relay.



```
const int relayPin = 7; // Relay control pin
```

```
void setup() {  
    pinMode(relayPin, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(relayPin, HIGH); // Turn ON relay (light ON)  
    delay(3000);                // Wait 3 seconds  
    digitalWrite(relayPin, LOW); // Turn OFF relay (light OFF)  
    delay(3000);                // Wait 3 seconds  
}
```