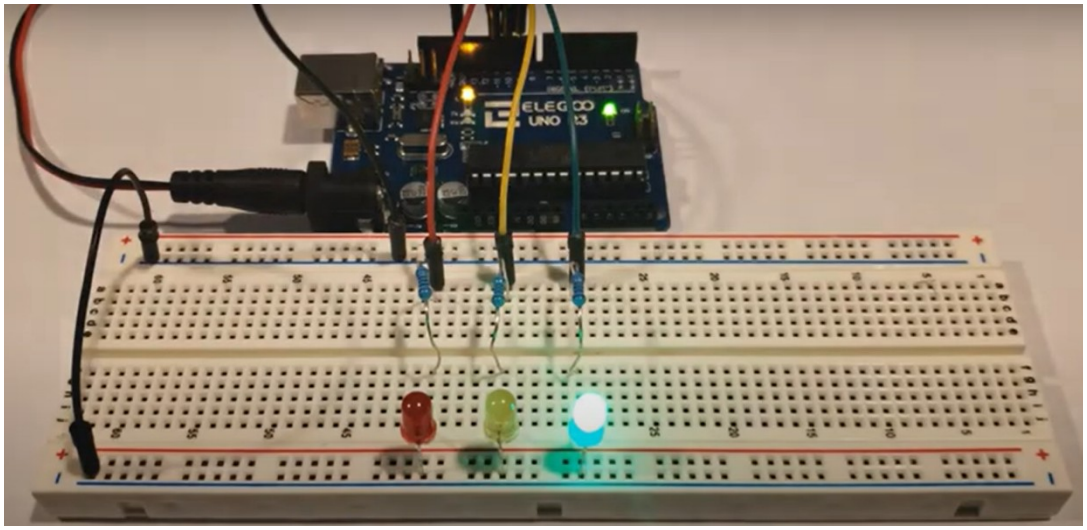
PROGRAM:

```
int led1 = 0;
int led2 = 1;
int led3 = 2;
int led4 = 3;
int led5 = 4;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  }
void loop() {
  digitalWrite(led1, HIGH);
   delay(80);
  digitalWrite(led1, LOW);
  digitalWrite(led2, HIGH);
  delay(80);
  digitalWrite(led2, LOW);
  digitalWrite(led3, HIGH);
  delay(80);
  digitalWrite(led3, LOW);
  digitalWrite(led4, HIGH);
  delay(80);
  digitalWrite(led4, LOW);
  digitalWrite(led5, HIGH);
  delay(80);
  digitalWrite(led5, LOW);
  delay(500);
  }
```

OUTPUT:

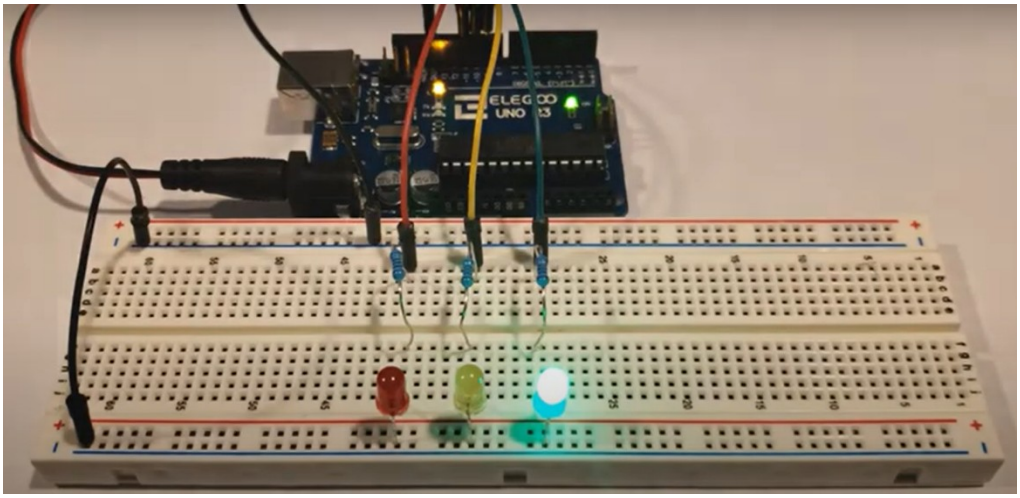


PROGRAM:

```
int LED1 = 0;
int LED2 = 1;
int LED3 = 2;
int LED4 = 3;
void setup(){
  Serial.begin(9600);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
 }
void loop(){
  while (Serial.available() == 0){
  }
  Int val = Serial.parseInt();
  if (val == "1"){
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    delay(2000);
    }
  if (val == "2"){
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    delay(2000);
   }
  if (val == "3"){
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
    delay(2000);
    }
  if (val == "4"){
```

```
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
     digitalWrite(LED3, LOW);
     digitalWrite(LED4, HIGH);
     delay(2000);
    }
  if (val > 4){
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
     digitalWrite(LED3, LOW);
     digitalWrite(LED4, LOW);
     Serial.println("The value is to high");
   }
  }
```

OUTPUT:

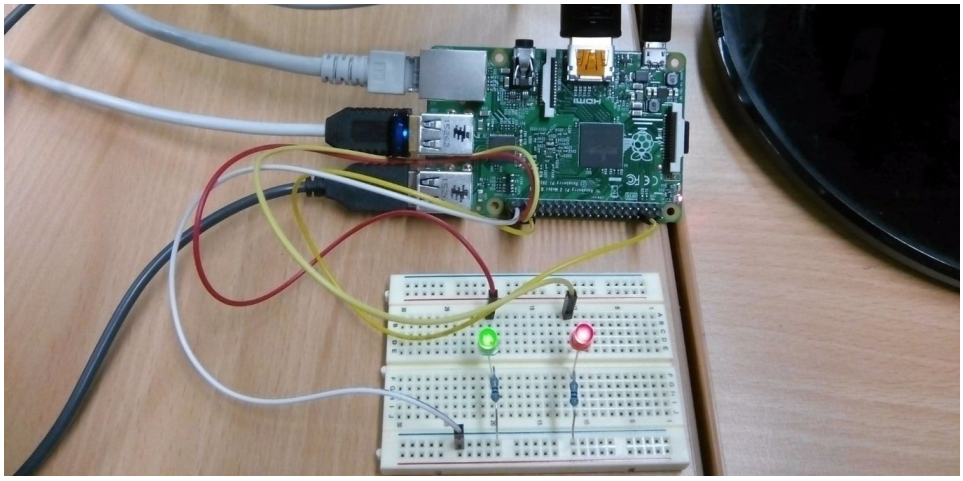


PROGRAM:

```
import RPi.GPIO as GPIO
import time
LED_PIN1 = 14
LED_PIN2 = 15
LED_PIN3 = 18
LED_PIN4 = 23
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN1, GPIO.OUT)
GPIO.setup(LED_PIN2, GPIO.OUT)
GPIO.setup(LED_PIN3, GPIO.OUT)
GPIO.setup(LED_PIN4, GPIO.OUT)
try:
    while True:
        val = int(input(“Enter value between 1 to 4 :::”))
        if(val == 1){
         GPIO.output(LED_PIN1, GPIO.HIGH)
         GPIO.output(LED_PIN2, GPIO.LOW)
         GPIO.output(LED_PIN3, GPIO.LOW)
         GPIO.output(LED_PIN4, GPIO.LOW)
         time.sleep(2)
        }
        if(val == 2){
         GPIO.output(LED_PIN1, GPIO.LOW)
         GPIO.output(LED_PIN2, GPIO.HIGH)
         GPIO.output(LED_PIN3, GPIO.LOW)
         GPIO.output(LED_PIN4, GPIO.LOW)
         time.sleep(2)
        }
        if(val == 3){
         GPIO.output(LED_PIN1, GPIO.LOW)
         GPIO.output(LED_PIN2, GPIO.LOW)
         GPIO.output(LED_PIN3, GPIO.HIGH)
         GPIO.output(LED_PIN4, GPIO.LOW)
         time.sleep(2)
        }
        if(val == 4){
         GPIO.output(LED_PIN1, GPIO.LOW)
```

```
        GPIO.output(LED_PIN2, GPIO.LOW)
        GPIO.output(LED_PIN3, GPIO.LOW)
        GPIO.output(LED_PIN4, GPIO.HIGH)
        time.sleep(2)
       }
       if(val > 4){
        GPIO.output(LED_PIN1, GPIO.LOW)
        GPIO.output(LED_PIN2, GPIO.LOW)
        GPIO.output(LED_PIN3, GPIO.LOW)
        GPIO.output(LED_PIN4, GPIO.LOW)
        time.sleep(2)
       }
except KeyboardInterrupt:
    GPIO.cleanup()
```

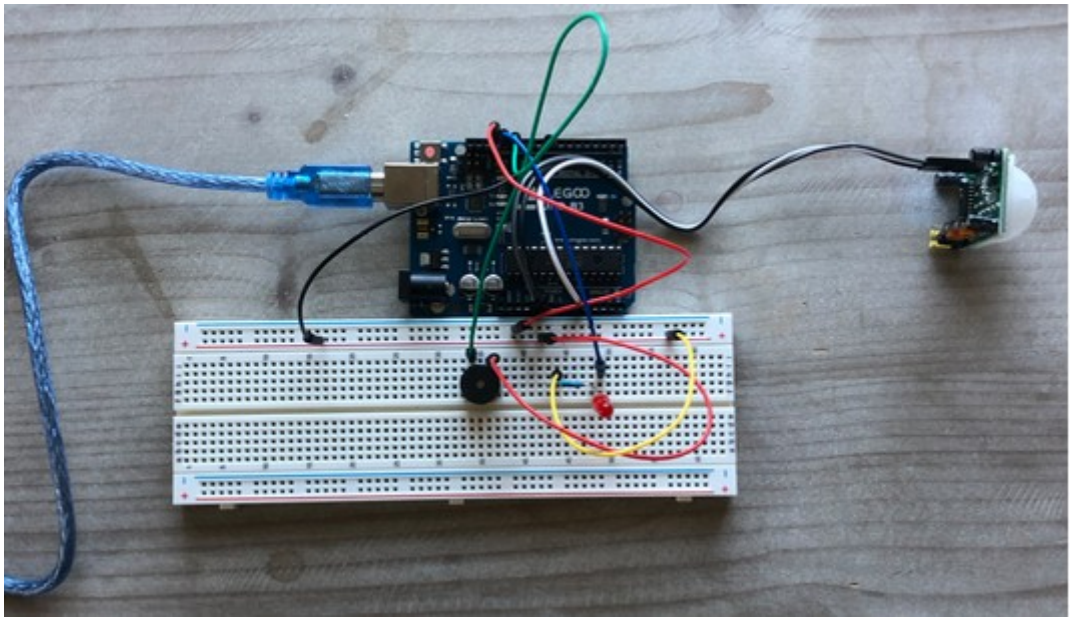
OUTPUT:



PROGRAM:

```
int pirPin = 2;
int buzzerPin = 8;
void setup() {
  pinMode(pirPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int motion = digitalRead(pirPin);
  if (motion == HIGH) {
    delay(1000);
    digitalWrite(buzzerPin, LOW);
  }
  Serial.println(motion);
  delay(100);
}
```

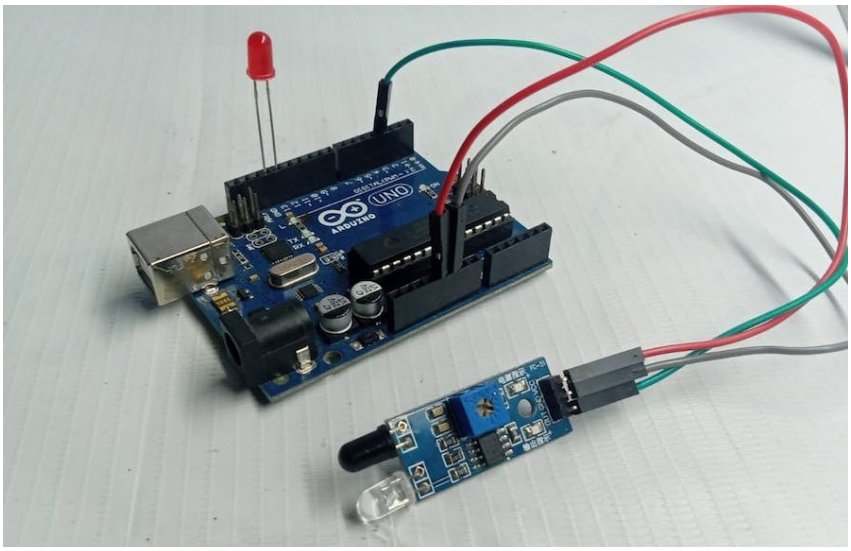
OUTPUT:



PROGRAM:

```
int pirPin = 2;
int ledPin = 8;
void setup() {
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600); }
void loop() {
  int motion = digitalRead(pirPin);
  if (motion == HIGH) {
      digitalWrite(ledPin, HIGH);
      delay(200);
      digitalWrite(ledPin, LOW);
      delay(200);  }
  Serial.println(motion);
  delay(100);
}
```

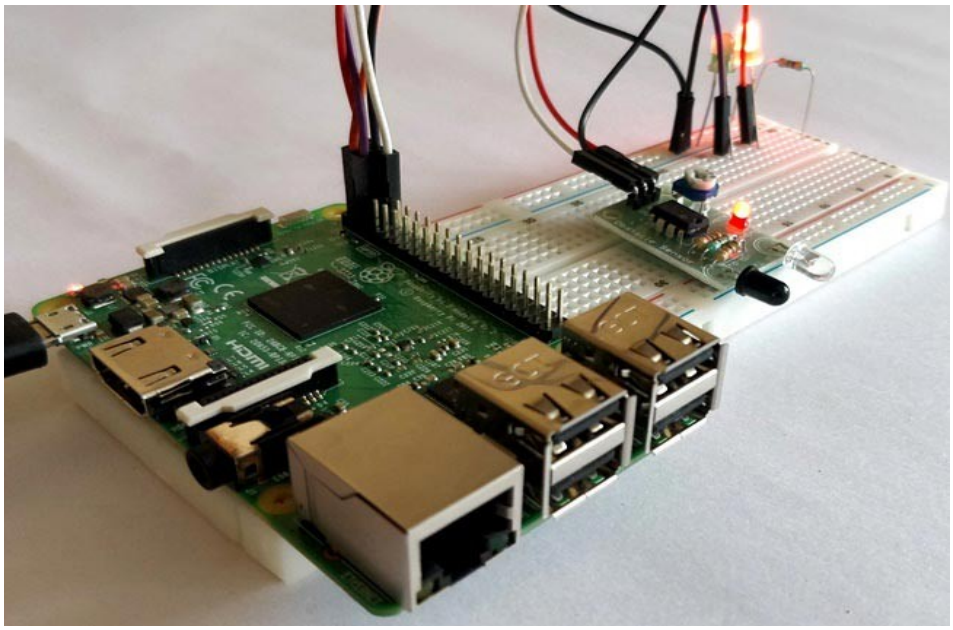
OUTPUT:



PROGRAM:

```
import RPi.GPIO as GPIO
import time
PIR_PIN = 17
BUZZER_PIN = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(PIR_PIN, GPIO.IN)
GPIO.setup(BUZZER_PIN, GPIO.OUT)
try:
    while True:
        motion = GPIO.input(PIR_PIN)
        if motion == GPIO.HIGH:
            print("Motion detected!")
            GPIO.output(BUZZER_PIN, GPIO.HIGH)
            time.sleep(1)
            GPIO.output(BUZZER_PIN, GPIO.LOW)
         time.sleep(0.1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

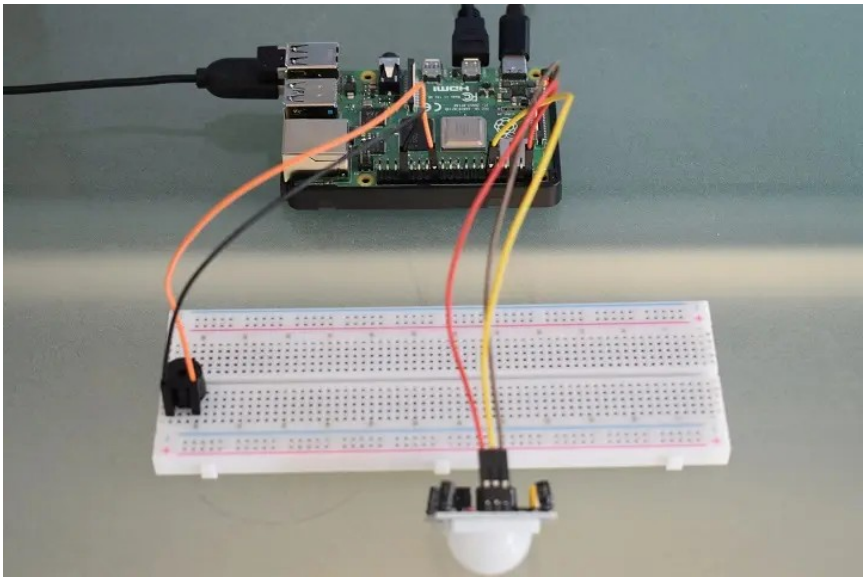
OUTPUT:



PROGRAM:

```
import RPi.GPIO as GPIO
import time
PIR_PIN = 17
LED_PINR = 18
LED_PING = 15
GPIO.setmode(GPIO.BCM)
GPIO.setup(PIR_PIN, GPIO.IN)
GPIO.setup(LED_PINR, GPIO.OUT)
GPIO.setup(LED_PING, GPIO.OUT)
try:
    while True:
        motion = GPIO.input(PIR_PIN)
        if motion == GPIO.HIGH:
            print("Motion detected!")
            GPIO.output(LED_PINR, GPIO.HIGH)
         GPIO.output(LED_PING, GPIO.LOW)
        else:
            GPIO.output(LED_PINR, GPIO.LOW)
         GPIO.output(LED_PING, GPIO.HIGH)
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

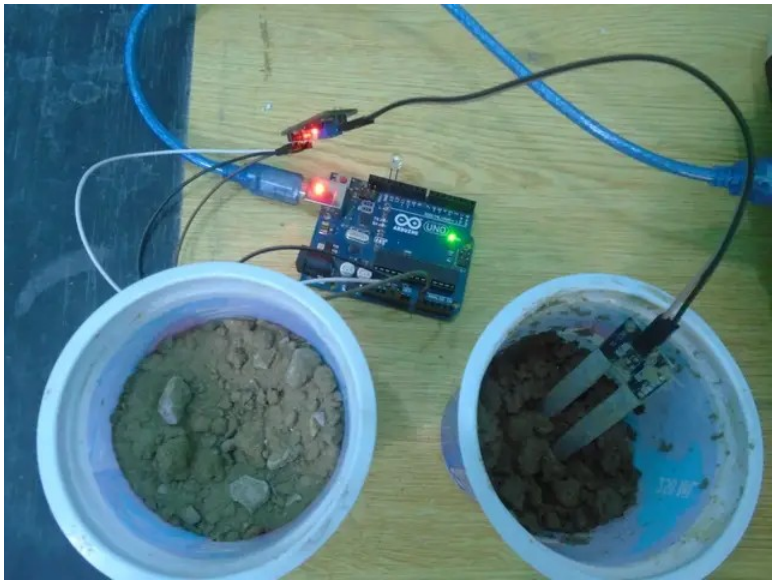
OUTPUT:



PROGRAM:

```
int moistureSensorPin = A0;
int ledPin = 8;
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int moistureLevel = analogRead(moistureSensorPin);
  if (moistureLevel < 500){
    digitalWrite(ledPin, HIGH);
    Serial.println("Soil is too dry!");
  } else{
    digitalWrite(ledPin, LOW);
    Serial.println("Soil moisture is okay.");
  }
  delay(1000);
}
```

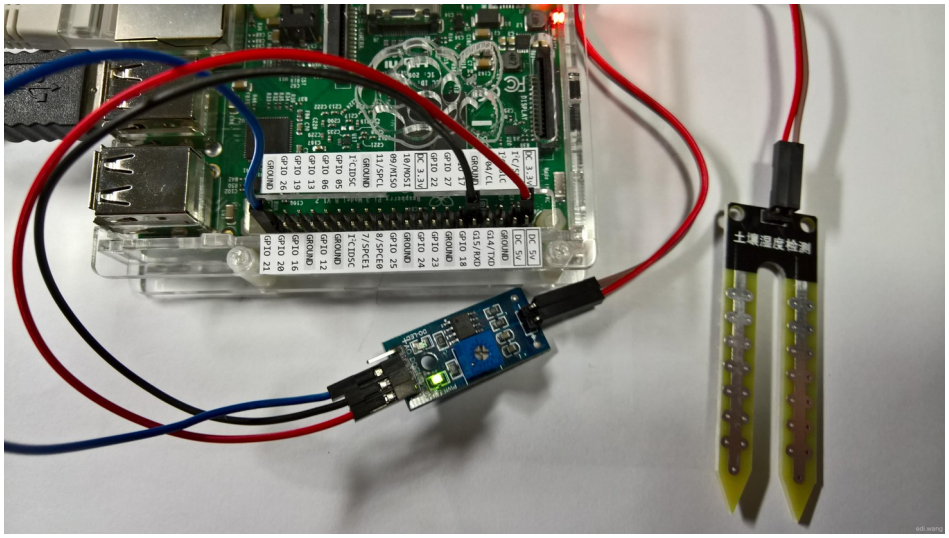
OUTPUT:



PROGRAM:

```
import RPi.GPIO as GPIO
import time
MOISTURE_SENSOR_PIN = 17
LED_PIN = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(MOISTURE_SENSOR_PIN, GPIO.IN)
GPIO.setup(LED_PIN, GPIO.OUT)
try:
    while True:
        moisture_level = GPIO.input(MOISTURE_SENSOR_PIN)
        if moisture_level == GPIO.LOW:
            print("Soil is too dry!")
            GPIO.output(LED_PIN, GPIO.HIGH)
        else:
            GPIO.output(LED_PIN, GPIO.LOW)
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

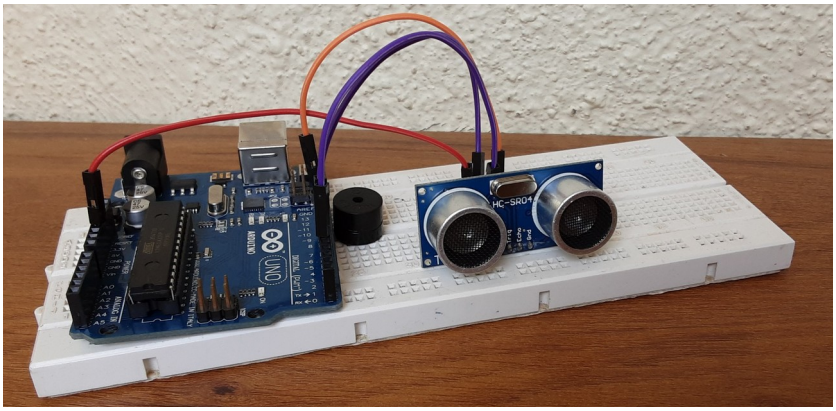
OUTPUT:



PROGRAM:

```
const int trigPin = 7;
const int echoPin = 6;
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);}
void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);  }
```

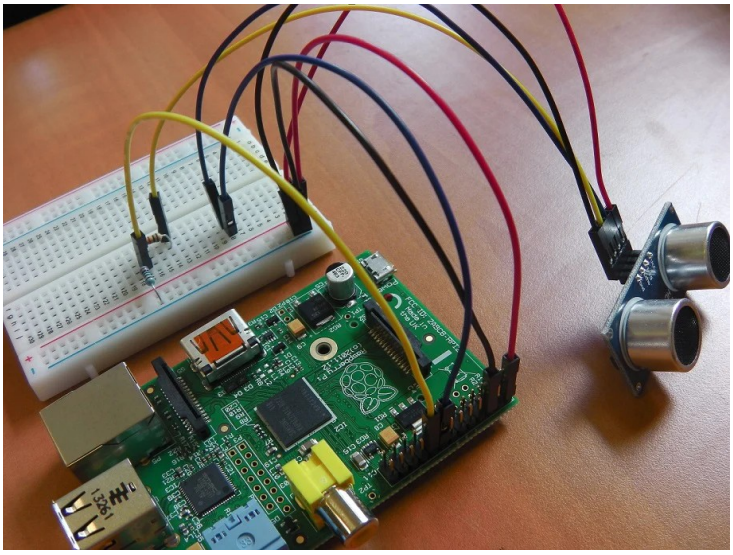
OUTPUT:



PROGRAM:

```
import RPi.GPIO as GPIO
import time
TRIG_PIN = 17
ECHO_PIN = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG_PIN, GPIO.OUT)
GPIO.setup(ECHO_PIN, GPIO.IN)
def measure_distance():
    GPIO.output(TRIG_PIN, True)
    time.sleep(0.00001)
    GPIO.output(TRIG_PIN, False)
    while GPIO.input(ECHO_PIN) == 0:
        pulse_start = time.time()
    while GPIO.input(ECHO_PIN) == 1:
        pulse_end = time.time()
    pulse_duration = pulse_end - pulse_start
    distance = (pulse_duration * 34300) / 2
    return distance
try:
    while True:
        distance = measure_distance()
        print(f"Distance: {distance:.2f} cm")
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

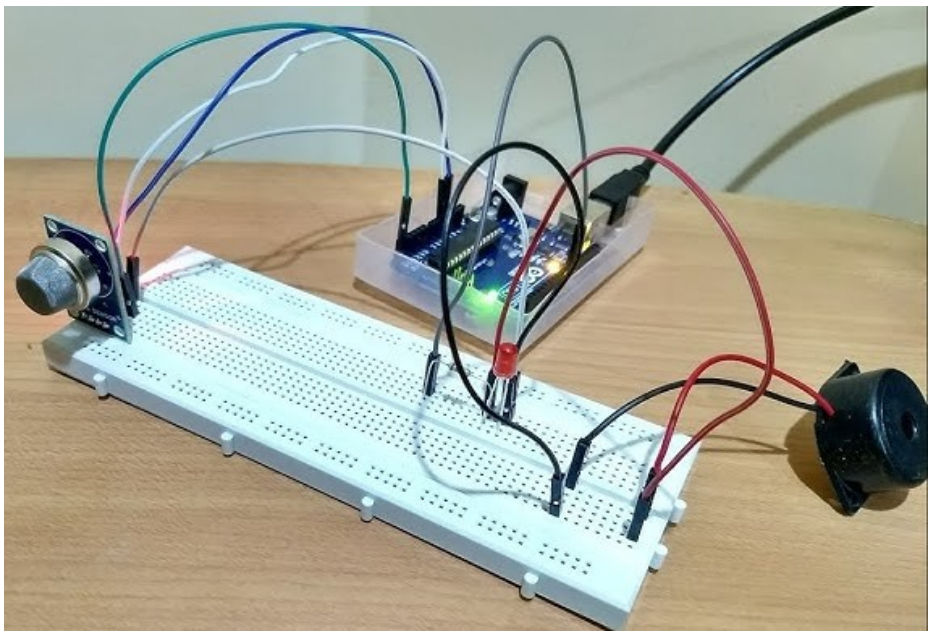
OUTPUT:



PROGRAM:

```
int smokeSensorPin = A0;
int buzzerPin = 8;
void setup() {
  pinMode(smokeSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int sensorValue = analogRead(smokeSensorPin);
  int threshold =130;
  if (sensorValue > threshold) {
    Serial.println("Smoke detected!");
    digitalWrite(buzzerPin, HIGH);
  } else {
    Serial.println("No smoke detected.");
    digitalWrite(buzzerPin, LOW);
  }
  delay(1000);
}
```

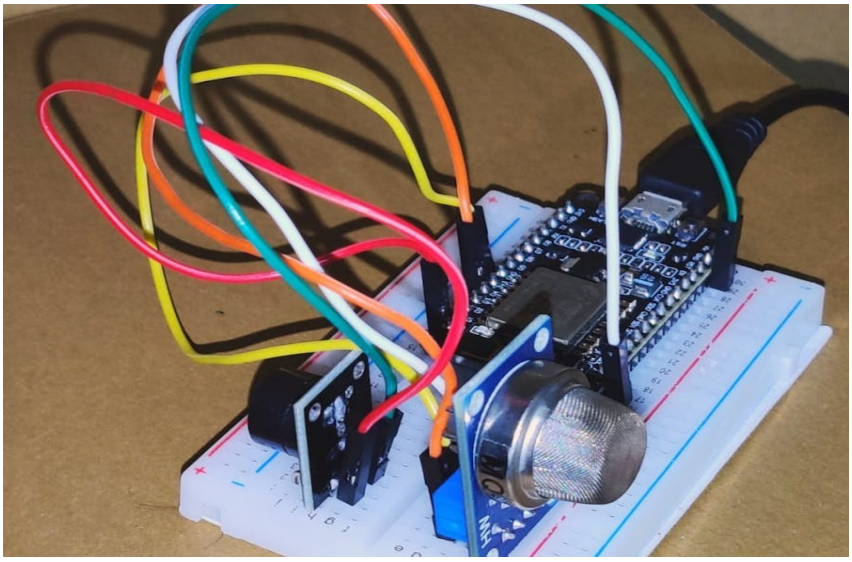
OUTPUT:



PROGRAM:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
sensor_pin = 18
buzzer_pin = 17
GPIO.setup(sensor_pin, GPIO.IN)
GPIO.setup(buzzer_pin, GPIO.OUT)
try:
    while True:
        if GPIO.input(sensor_pin):
            print("Gas/smoke detected!")
            GPIO.output(buzzer_pin, GPIO.HIGH)
        else:
            print("No gas/smoke detected.")
            GPIO.output(buzzer_pin, GPIO.LOW)
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

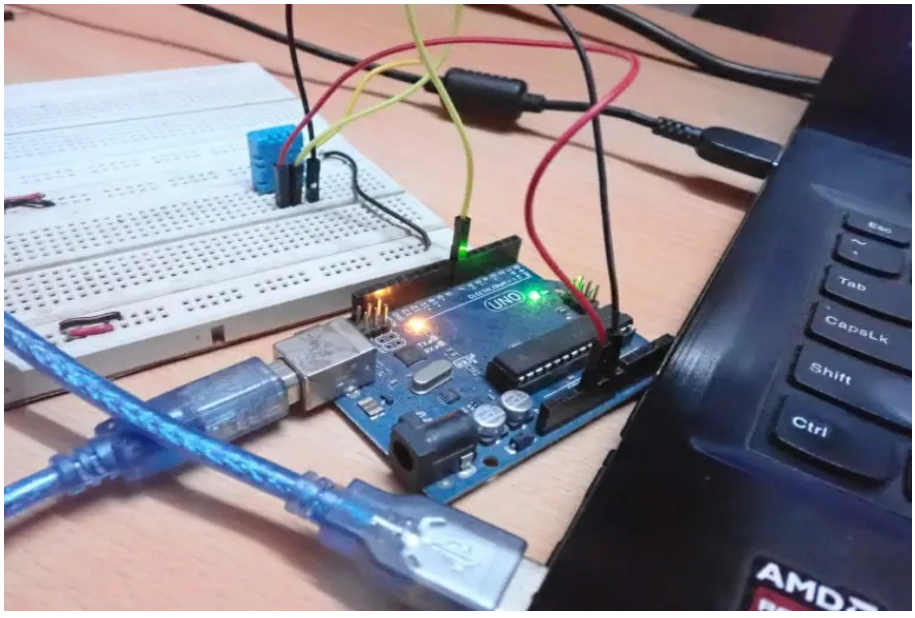
OUTPUT:



PROGRAM:

```
#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");} else {
    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.print("%\t");
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.println("°C");}}
```

OUTPUT:



PROGRAM:

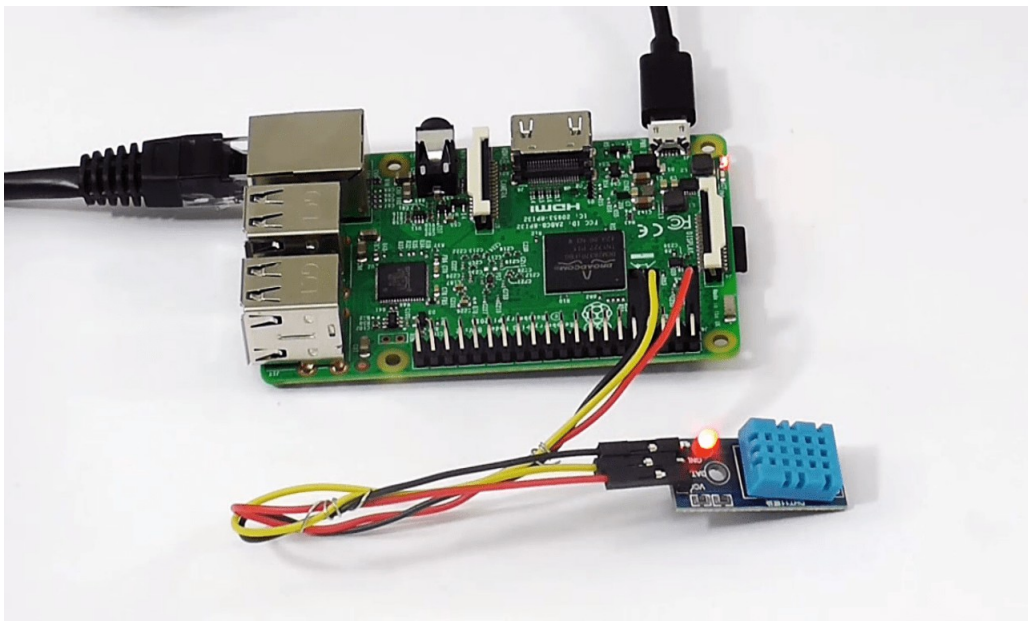
```
import Adafruit_DHT

# Set up the sensor type and GPIO pin
sensor = Adafruit_DHT.DHT11
pin = 4

while True:
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

    if humidity is not None and temperature is not None:
        print(f"Temperature: {temperature}°C, Humidity: {humidity}%")
    else:
        print("Failed to read from the DHT sensor.")
```

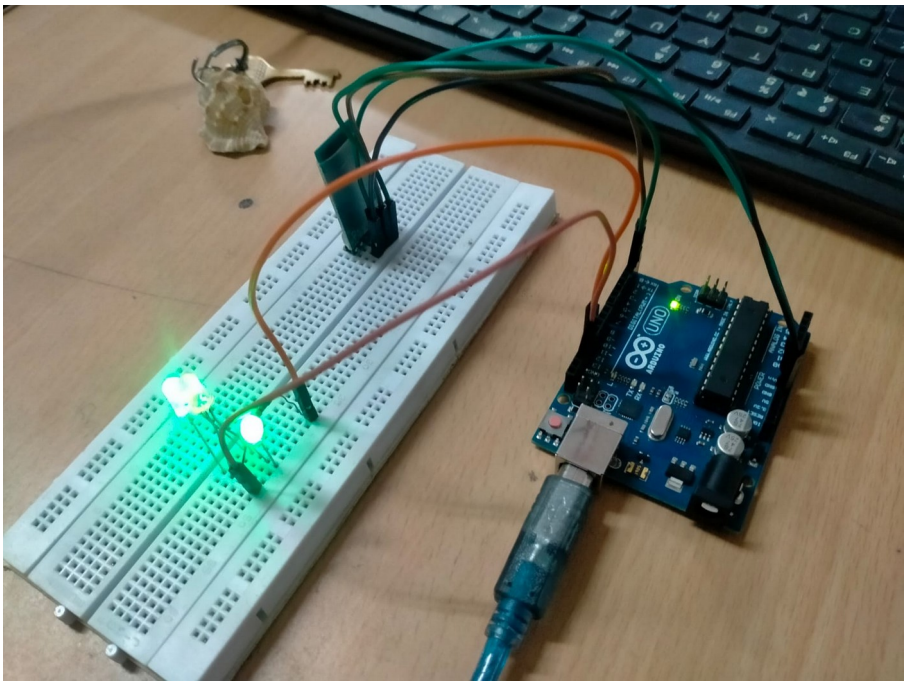
OUTPUT:



PROGRAM:

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(0, 1); // RX, TX
int ledPin = 9;
char command;
void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  bluetooth.begin(9600);}
void loop() {
  if (bluetooth.available() > 0) {
    command = bluetooth.read();
    if (command == '1') {
      digitalWrite(ledPin, HIGH);}
    if (command == '0') {
      digitalWrite(ledPin, LOW);}}}
```

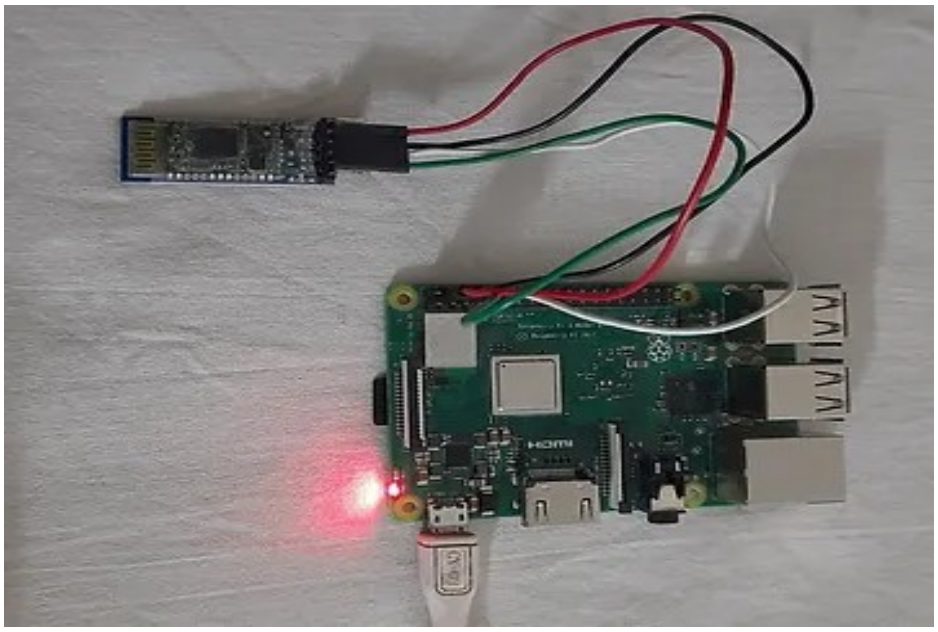
OUTPUT:



PROGRAM:

```
import bluetooth
from gpiozero import LED
import time
led = LED(17)  # Change to the GPIO pin you are using
server_sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
port = 1
server_sock.bind(("", port))
server_sock.listen(1)
print("Waiting for a Bluetooth connection...")
client_sock, address = server_sock.accept()
print("Accepted connection from", address)
try:
    while True:
        data = client_sock.recv(1024).decode('utf-8')
        if not data:
            break
        if data == "on":
            led.on()
        elif data == "off":
            led.off()
except KeyboardInterrupt:
    pass
client_sock.close()
server_sock.close()
```

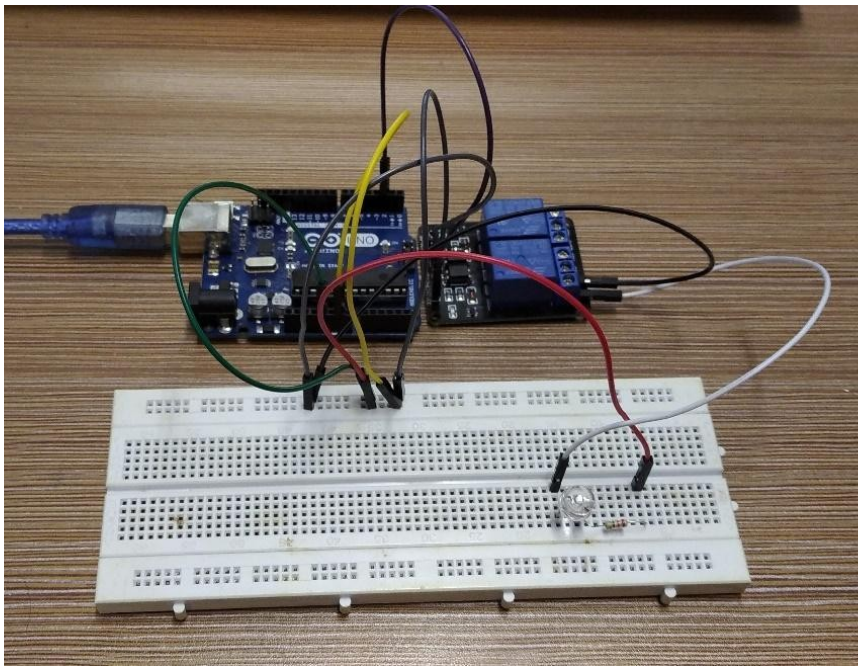
OUTPUT:



PROGRAM:

```
int relayPin = 7;
void setup() {
  pinMode(relayPin, OUTPUT);
}
void loop() {
  digitalWrite(relayPin, HIGH);
  delay(1000);
  digitalWrite(relayPin, LOW);
  delay(1000);
}
```

OUTPUT:



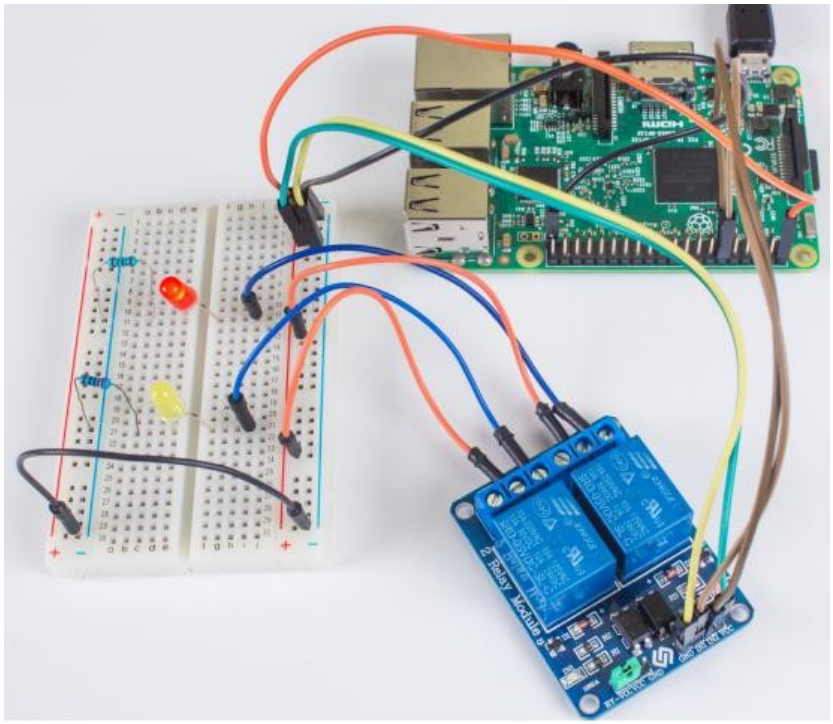
PROGRAM:

```
from gpiozero import LED, OutputDevice
from time import sleep

relay = OutputDevice(17, active_high=False)
led = LED(18)

while True:
    relay.on()
    led.on()
    sleep(1)
    relay.off()
    led.off()
    sleep(1)
```

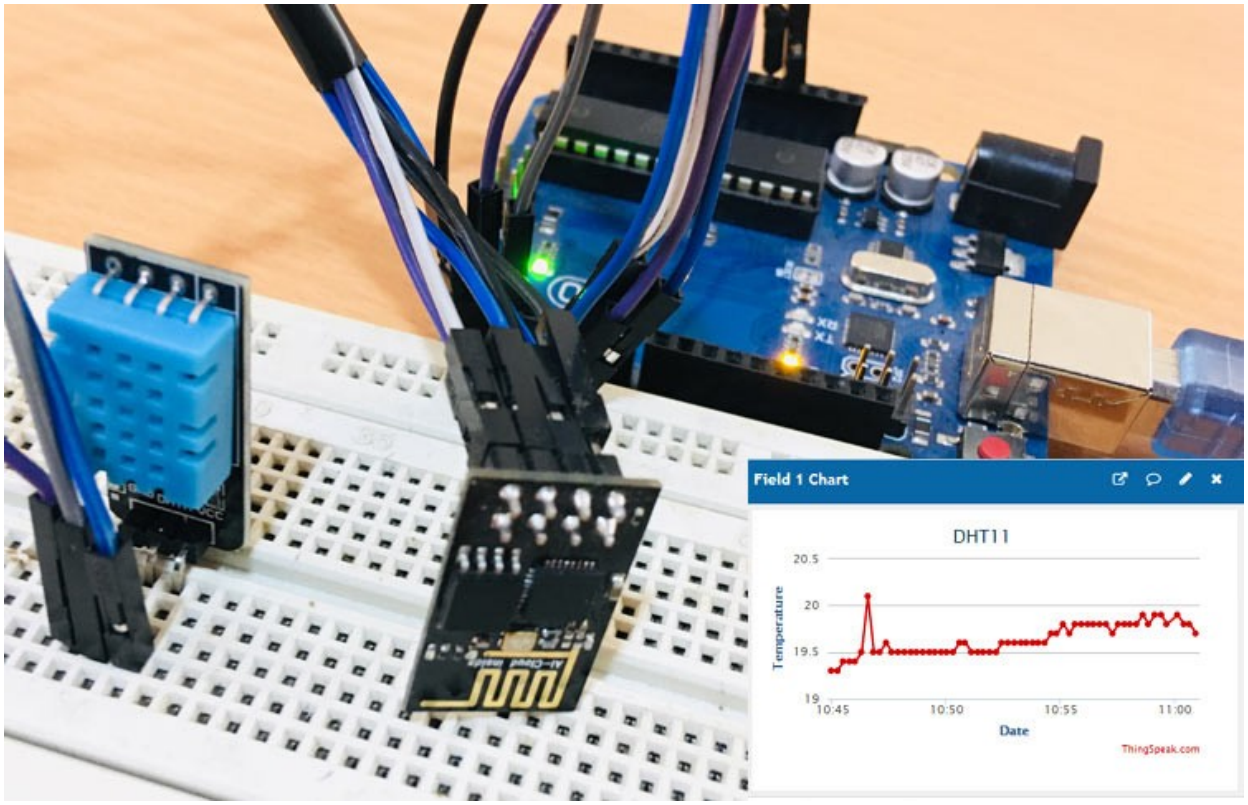
OUTPUT:



PROGRAM:

```
#include <ESP8266WiFi.h>
#include <DHT.h>
const char* ssid = "<Wi-Fi Network Name>";
const char* password = "<Wi-Fi Network Password>";
const char* thingSpeakAPI = "<ThingSpeak Write API Key>";
const int DHTPin = 2;  // GPIO pin to which the DHT sensor is connected
const int UpdateInterval = 30000;  // Update data every 30 seconds (in milliseconds)
DHT dht(DHTPin, DHT22);
WiFiClient client;
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  dht.begin();}
void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    float temperature = dht.readTemperature();
    if (!isnan(temperature)) {
      sendTemperatureData(temperature);
    }}
  delay(UpdateInterval);
}
void sendTemperatureData(float temp) {
  if (client.connect("api.thingspeak.com", 80)) {
    String data = "field1=" + String(temp);
    data += "&status=MQTTPUBLISH";  // Optional status field
    client.println("POST /update HTTP/1.1");
    client.println("Host: api.thingspeak.com");
    client.println("Connection: close");
    client.println("X-THINGSPEAKAPIKEY: " + thingSpeakAPI);
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.print("Content-Length: ");
    client.println(data.length());
    client.println();
    client.print(data);
  } else {
    Serial.println("Failed to connect to ThingSpeak.");
  }
  client.stop();
}
```

OUTPUT:


Field 1 Chart
DHT11
20.5
20
19.5
19
Temperature
10:45
10:50
10:55
11:00
Date
ThingSpeak.com