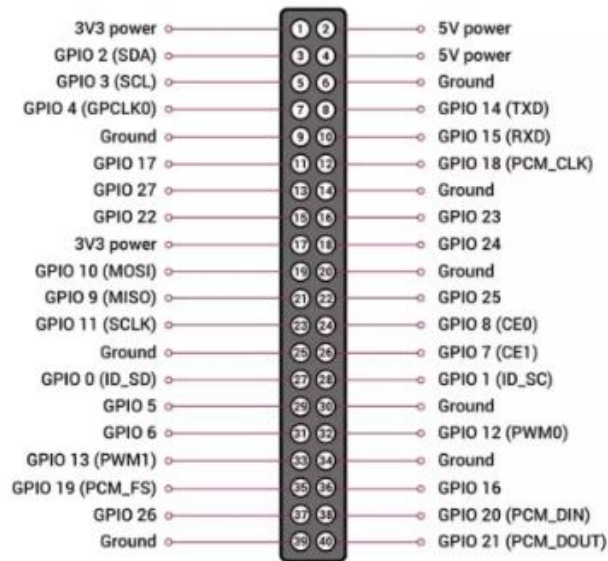


Study of LED Blinking using Raspberry Pi. PIN diagram of Raspberry Pi is given below.



Program:

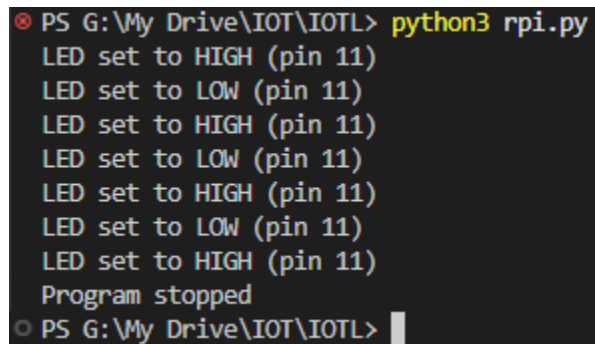
```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
led_pin = 11
GPIO.setup(led_pin, GPIO.OUT)

try:
    while True:
        GPIO.output(led_pin, GPIO.HIGH)
        print(f"LED set to HIGH (pin {led_pin})")
        time.sleep(0.5)
        GPIO.output(led_pin, GPIO.LOW)
        print(f"LED set to LOW (pin {led_pin})")
        time.sleep(0.5)

except KeyboardInterrupt:
    GPIO.cleanup()
    print("Program stopped")
```

Output:

A screenshot of a Windows command prompt window. The title bar is partially visible at the top. The command prompt shows the execution of a Python script named 'rpi.py'. The output of the script is displayed in the prompt, showing a sequence of LED state changes (HIGH and LOW) for pin 11, followed by a message indicating the program has stopped. The prompt is currently at the 'PS G:\My Drive\IOT\IOTL>' line.

```
PS G:\My Drive\IOT\IOTL> python3 rpi.py
LED set to HIGH (pin 11)
LED set to LOW (pin 11)
LED set to HIGH (pin 11)
LED set to LOW (pin 11)
LED set to HIGH (pin 11)
LED set to LOW (pin 11)
LED set to HIGH (pin 11)
LED set to LOW (pin 11)
LED set to HIGH (pin 11)
Program stopped
PS G:\My Drive\IOT\IOTL>
```

Write a program using Arduino to control 3 LEDs. - (ON/OFF) Blinking

Program:

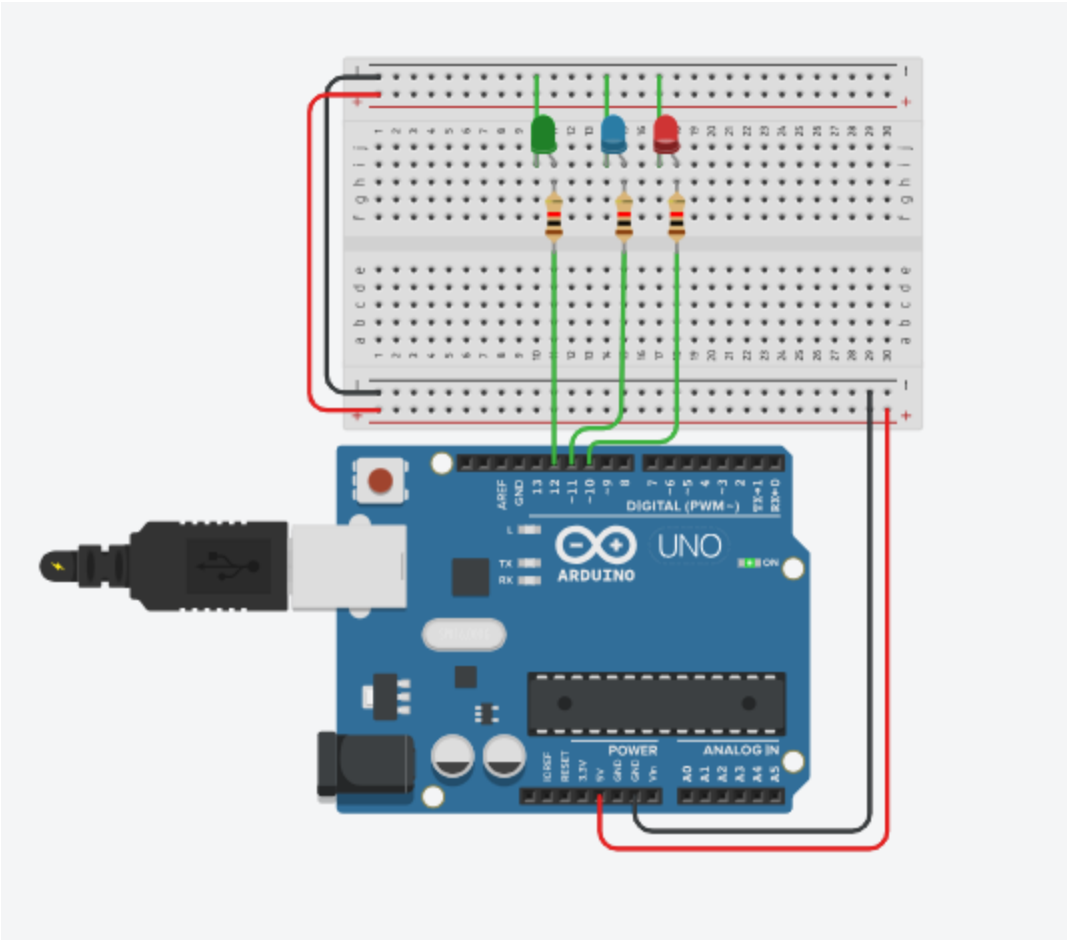
```
const int green = 12;
const int blue = 11;
const int red = 10;
void setup()
{
    pinMode(green,OUTPUT);
    pinMode(blue,OUTPUT);
    pinMode(red, OUTPUT);
}

void loop()
{
    digitalWrite(green,HIGH);
    digitalWrite(blue,LOW);
    digitalWrite(red,LOW);
    delay(1000);

    digitalWrite(green,LOW);
    digitalWrite(blue,HIGH);
    digitalWrite(red,LOW);
    delay(1000);

    digitalWrite(green,LOW);
    digitalWrite(blue,LOW);
    digitalWrite(red,HIGH);
    delay(1000);
}
```

Output:



Experiment 6

Create a program that illuminates the green LED if the counter is less than 100, illuminates the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is greater than 200

Program:

```
const int greenLedPin = 13; // Green LED pin
const int yellowLedPin = 12; // Yellow LED pin
const int redLedPin = 11;    // Red LED pin
int counter = 0;

void setup() {
  pinMode(greenLedPin, OUTPUT);
  pinMode(yellowLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
  Serial.begin(9600);
}

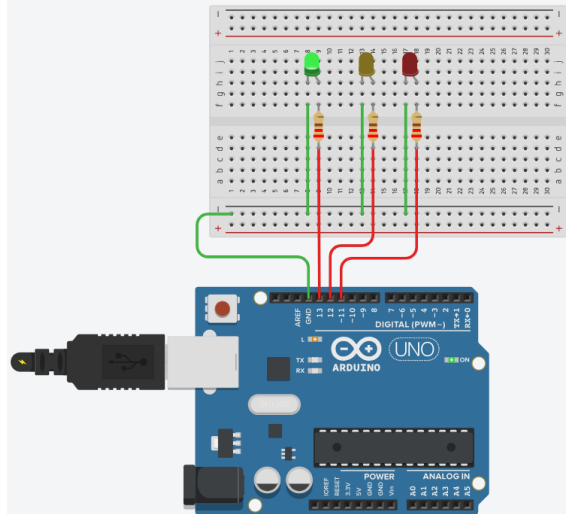
void loop() {
  counter++;
  Serial.print("Counter: ");
  Serial.println(counter);

  if (counter < 100) {
    digitalWrite(greenLedPin, HIGH);
    digitalWrite(yellowLedPin, LOW);
    digitalWrite(redLedPin, LOW);
  } else if (counter >= 100 && counter <= 200) {
    digitalWrite(greenLedPin, LOW);
    digitalWrite(yellowLedPin, HIGH);
    digitalWrite(redLedPin, LOW);
  } else if (counter > 200) {
    digitalWrite(greenLedPin, LOW);
    digitalWrite(yellowLedPin, LOW);
    digitalWrite(redLedPin, HIGH);
  }

  if (counter >= 300) {
    counter = 0;
  }

  delay(100);
}
```

Output:



1 (Arduino Uno R3) ▾

```
1  const int greenLedPin = 13; // Green LED pin
2  const int yellowLedPin = 12; // Yellow LED pin
3  const int redLedPin = 11; // Red LED pin
4  int counter = 0;
5
6  void setup() {
7      pinMode(greenLedPin, OUTPUT);
8      pinMode(yellowLedPin, OUTPUT);
9      pinMode(redLedPin, OUTPUT);
10     Serial.begin(9600);
11 }
12
13 void loop() {
14     counter++;
15     Serial.print("Counter: ");
```

Serial Monitor ▾

Counter: 49
Counter: 50
Counter: 51
Counter: 52
Counter: 53
Coun

Send Clear

Create a program so that when the user enters 'b' the green light blinks, 'g' the green light is illuminated
'y' the yellow light is illuminated and 'r' the red light is illuminated

Program:

```
const int greenLedPin = 13;
const int yellowLedPin = 12;
const int redLedPin = 11;

void setup() {
  pinMode(greenLedPin, OUTPUT);
  pinMode(yellowLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Enter 'b' for blink, 'g' for green, 'y' for yellow, 'r' for red:");
}

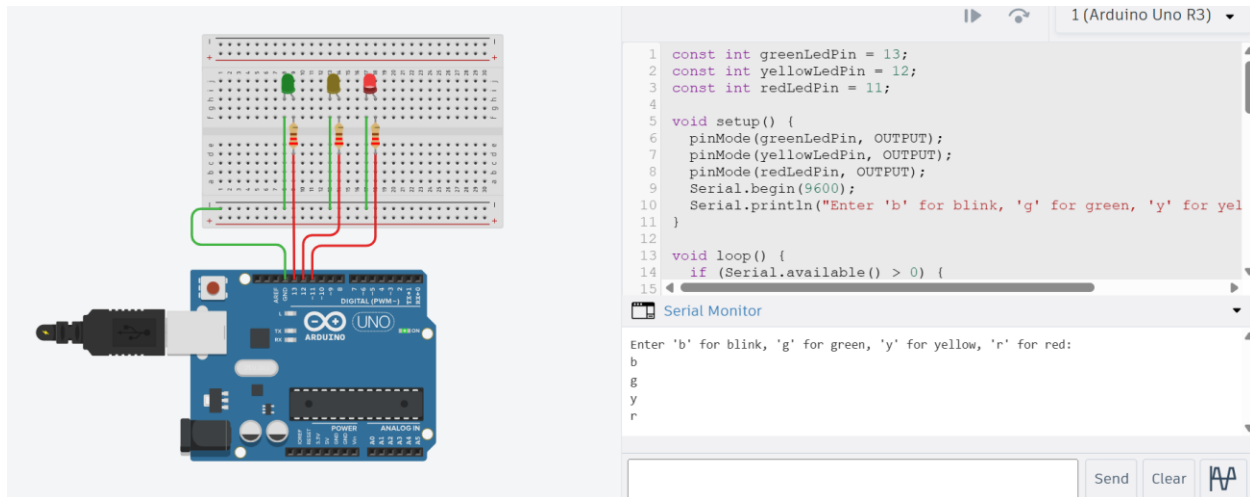
void loop() {
  if (Serial.available() > 0) {
    char input = Serial.read();
    Serial.println(input);

    switch (input) {
      case 'b':
        blinkGreen();
        break;
      case 'g':
        digitalWrite(greenLedPin, HIGH);
        digitalWrite(yellowLedPin, LOW);
        digitalWrite(redLedPin, LOW);
        break;
      case 'y':
        digitalWrite(greenLedPin, LOW);
        digitalWrite(yellowLedPin, HIGH);
        digitalWrite(redLedPin, LOW);
        break;
      case 'r':
        digitalWrite(greenLedPin, LOW);
        digitalWrite(yellowLedPin, LOW);
        digitalWrite(redLedPin, HIGH);
        break;
      default:
        Serial.println("Invalid input.");
    }
  }
}

void blinkGreen() {
  for (int i = 0; i < 5; i++) { // Blink 5 times
    digitalWrite(greenLedPin, HIGH);
    delay(250);
    digitalWrite(greenLedPin, LOW);
    delay(250);
  }
}
```

```
}  
}
```

Output:



Experiment 8

Write a program that asks the user for a number and outputs the number squared that is entered

Program: (Arduino Uno)

```
void setup()  
{  
  Serial.begin(9600);  
  Serial.println("Enter a number to calculate its square and cube");  
}  
  
void loop()  
{  
  if(Serial.available() > 0)  
  {  
    double num = Serial.parseInt();  
    double square = num * num;  
    double cube = num * num * num;  
    Serial.print("The square of: ");  
    Serial.print(num);  
    Serial.print(" is: ");  
    Serial.println(square);  
    Serial.print("The cube of: ");  
    Serial.print(num);  
    Serial.print(" is: ");
```

```
    Serial.println(cube);
    Serial.println("Enter another number to calc square and cube: ");
}
}
```

Experiment 9

Write a program to control the colour of 3 LEDs by turning 1 potentiometer, (LED one is red, LED two is green and LED three is blue)

Program:

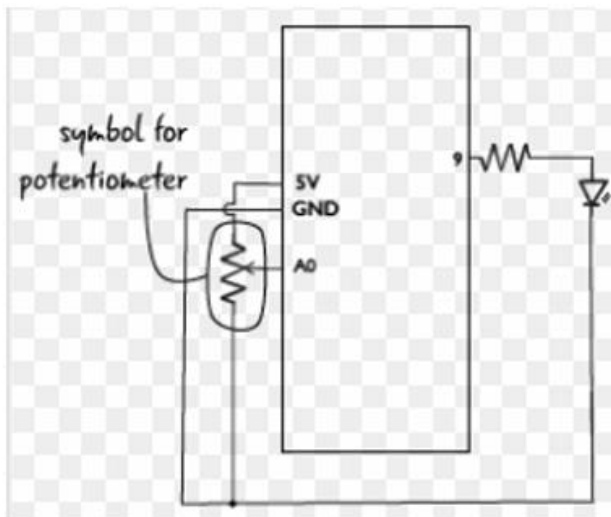
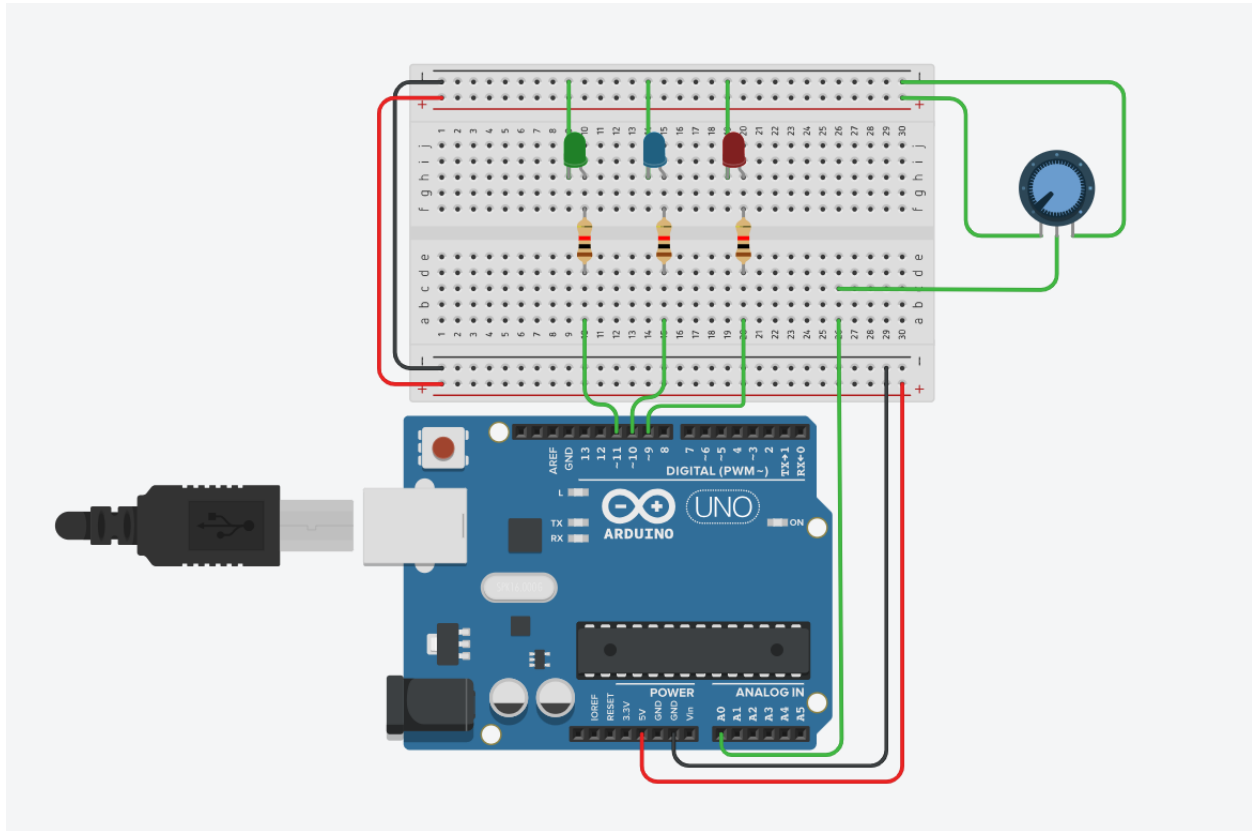
```
const int potPin = A0;
const int green = 11;
const int blue = 10;
const int red = 9;

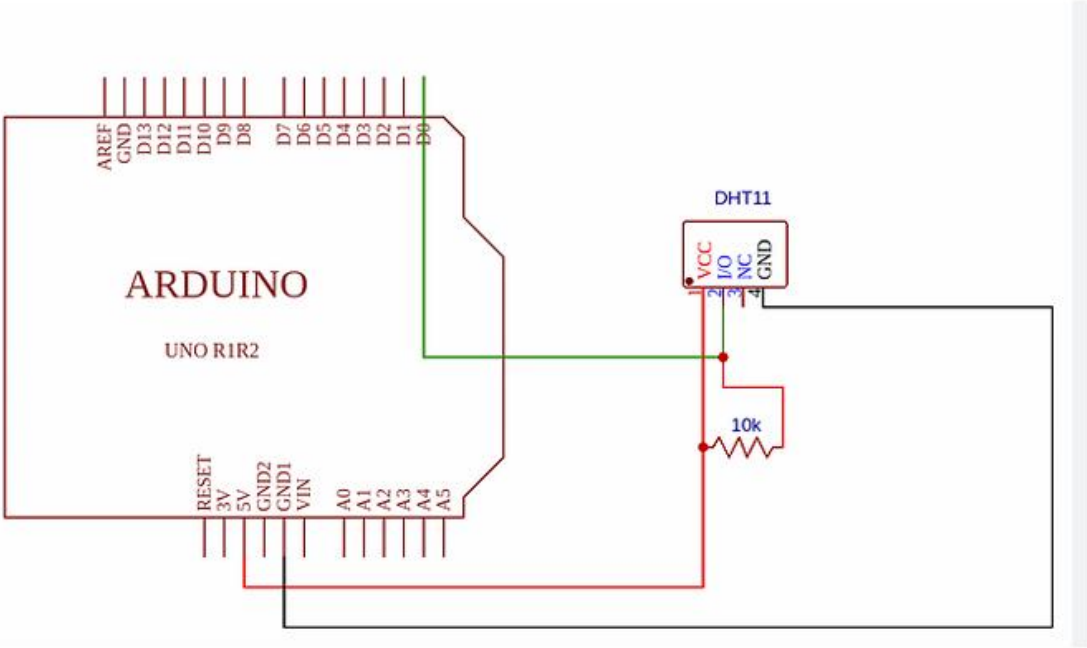
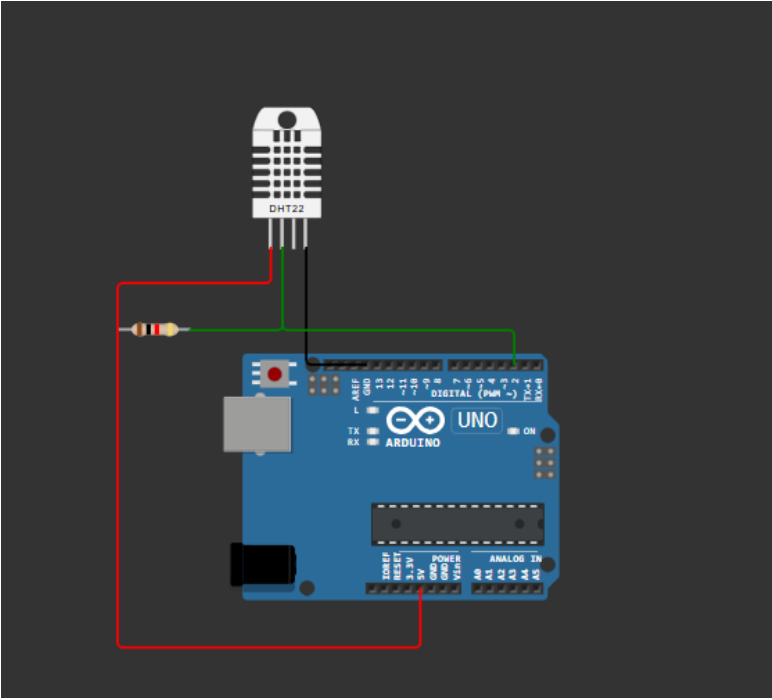
void setup()
{
    pinMode(potPin, INPUT);
    pinMode(green, OUTPUT);
    pinMode(blue, OUTPUT);
    pinMode(red, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    int potValue = analogRead(potPin) / 4;
    Serial.print("Current Brightness (0-255): ");
    Serial.println(potValue);

    analogWrite(green,potValue);
    analogWrite(blue,potValue);
    analogWrite(red, potValue);
}
```


Output:





Write a program read the temperature sensor and send the values of temperature and humidity to the serial monitor on the computer. Plot its graph.

```
#include <DHT.h>

const int dhtPin = 2;
const int dhtType = DHT11; // DHT sensor type

DHT dht(dhtPin, dhtType);

void setup() {
  Serial.begin(9600);
  dht.begin();
  Serial.println("DHT11 Temp/Humidity Monitor");
  Serial.println("-----");
  delay(1000);
}

void loop() {
  delay(2000);
  float tempC = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (isnan(tempC) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Serial.print(tempC, 1);
  Serial.print(', ');
  Serial.println(humidity, 1);
}
```

Write a program so it displays the temperature in Fahrenheit as well as the maximum and minimum temperatures it has seen.

```
#include <DHT.h>

const int dhtPin = 2;
const int dhtType = DHT11; // DHT sensor type

DHT dht(dhtPin, dhtType);

float currentTemp;
float maxTemp = -100.0; // Initialize to a very low value
float minTemp = 150.0; // Initialize to a very high value

void setup() {
    Serial.begin(9600);
    dht.begin();
    Serial.println("DHT11 Temperature Monitor");
    Serial.println("-----");
    delay(1000);
}

void loop() {
    delay(2000);
    float tempC = dht.readTemperature();

    if (isnan(tempC)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
    currentTemp = (tempC * 9.0 / 5.0) + 32.0;

    if (currentTemp > maxTemp) {
        maxTemp = currentTemp;
    }
    if (currentTemp < minTemp) {
        minTemp = currentTemp;
    }
    Serial.print("Current: ");
    Serial.print(currentTemp, 1);
    Serial.print(" F | Min: ");
    Serial.print(minTemp, 1);
    Serial.print(" F | Max: ");
    Serial.print(maxTemp, 1);
    Serial.println(" F");
}
```

```
}
```

Write a program using piezo element and use it to play a tune (8 notes) after someone knocks.

```
const int knockSensorPin = A0;
const int buzzerPin = 8;
const int threshold = 100;

const int tuneNotes[] = {100, 200, 300, 400, 500, 600, 700, 800};
const int noteCount = 8;
const int noteDuration = 150;

void setup() {
  pinMode(knockSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(knockSensorPin);
  Serial.println(sensorValue);

  if (sensorValue >= threshold) {

    for (int i = 0; i < noteCount; i++) {
      tone(buzzerPin, tuneNotes[i], noteDuration);
      delay(noteDuration + 30);
    }

    delay(500);
  }

  delay(10);
}
```

