

Experiment 4

Program:

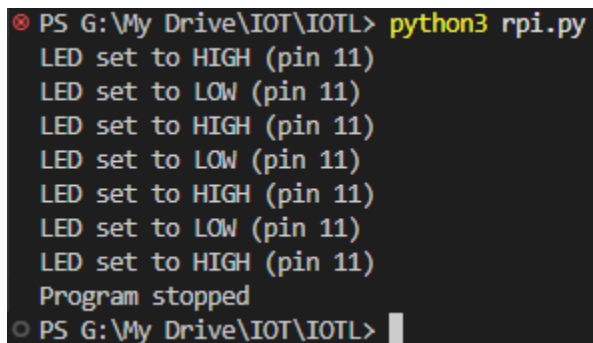
```
import RPi.GPIO as GPIO
import time

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 not visible. The prompt shows the command 'python3 rpi.py' being executed. The output consists of alternating lines of 'LED set to HIGH (pin 11)' and 'LED set to LOW (pin 11)', followed by 'Program stopped'. The prompt then returns to 'PS G:\My Drive\IOT\IOTL>'.

```
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)
Program stopped
PS G:\My Drive\IOT\IOTL>
```


Experiment 5

Write a program using Arduino to control LED (One or more ON/OFF). Or Blinking

Program:

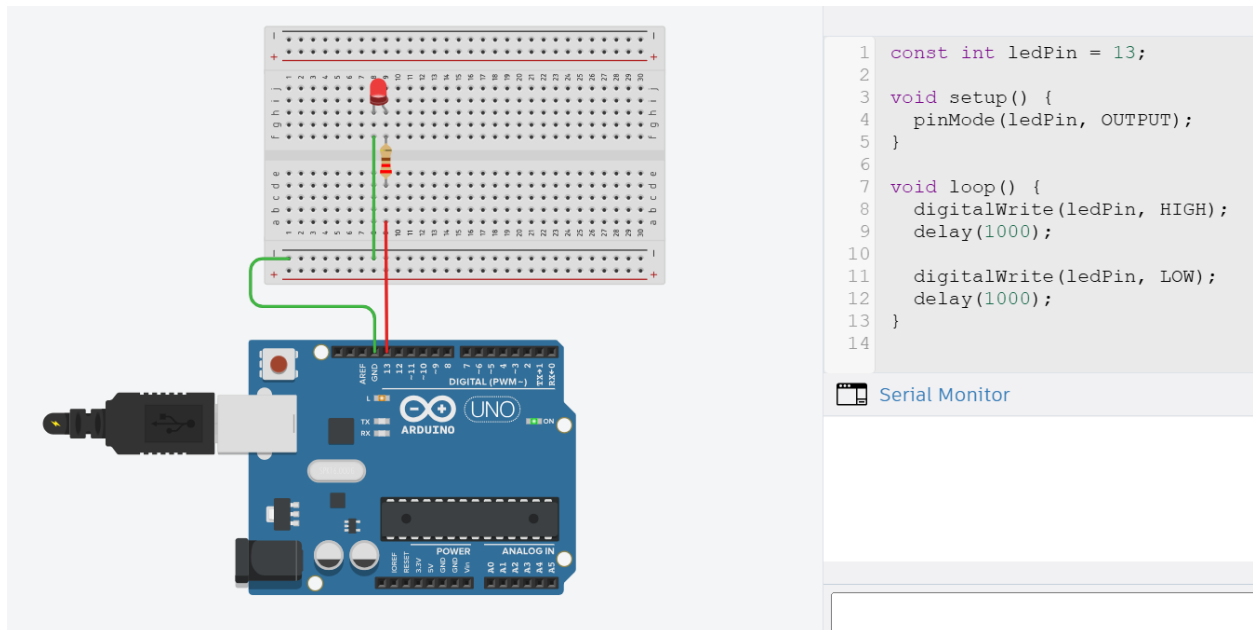
```
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);

  digitalWrite(ledPin, LOW);
  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);
  }
}
```

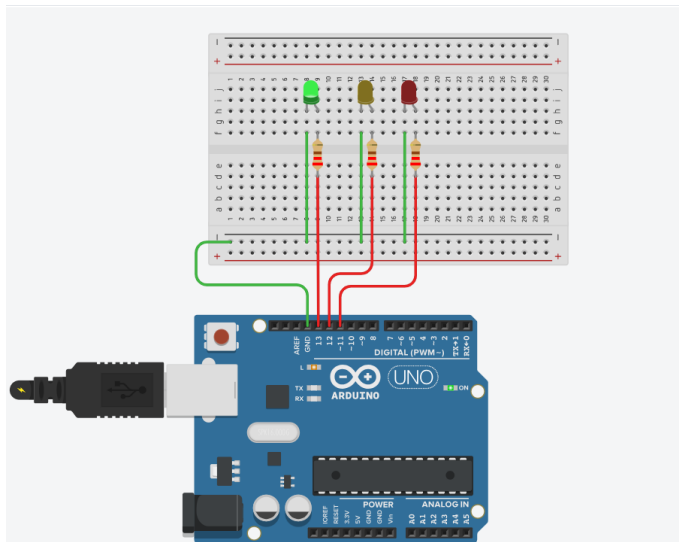
```

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

delay(100);
}

```

Output:



```

1  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

Experiment 7

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);
    }
  }
}
```

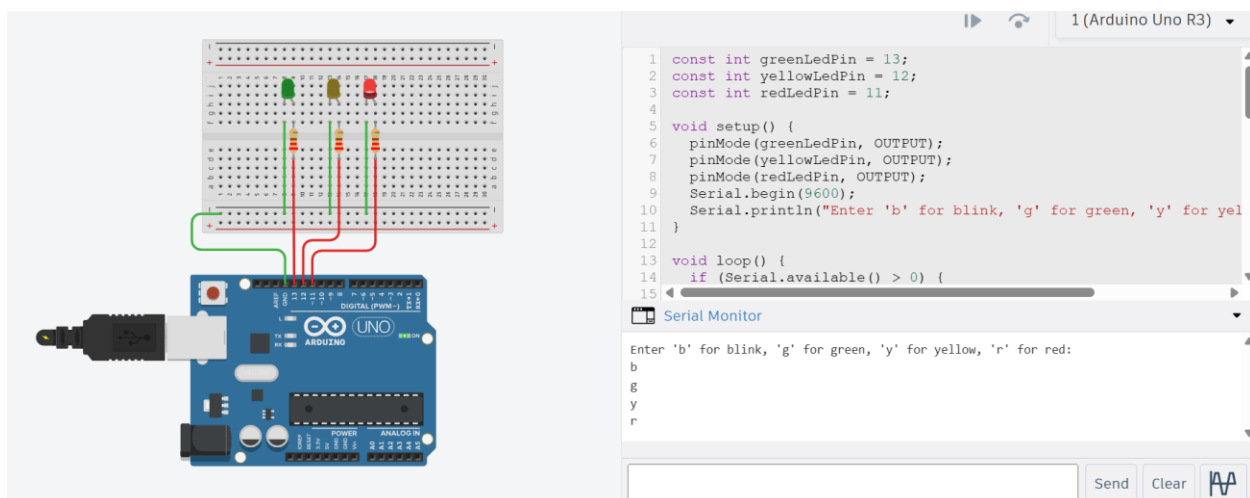
```

        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:



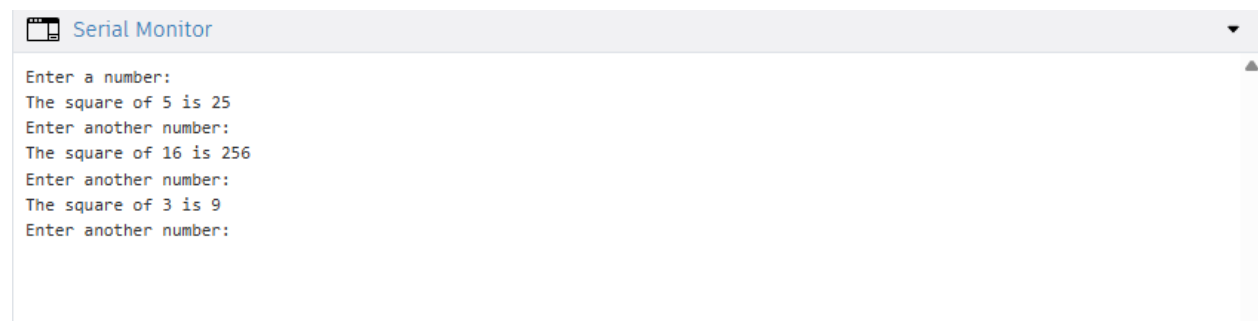
Experiment 8

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: ");  
}  
void loop() {  
    if (Serial.available() > 0) {  
        int num = Serial.parseInt();  
        int squared = num * num;  
        Serial.print("The square of ");  
        Serial.print(num);  
        Serial.print(" is ");  
        Serial.println(squared);  
        Serial.println("Enter another number: ");  
    }  
}
```

Output:



Experiment 9

Write a program to control the color of the LED by turning 3 different potentiometers. One will be read for the value of Red, one for the value of Green, and one for the value of Blue

Program:

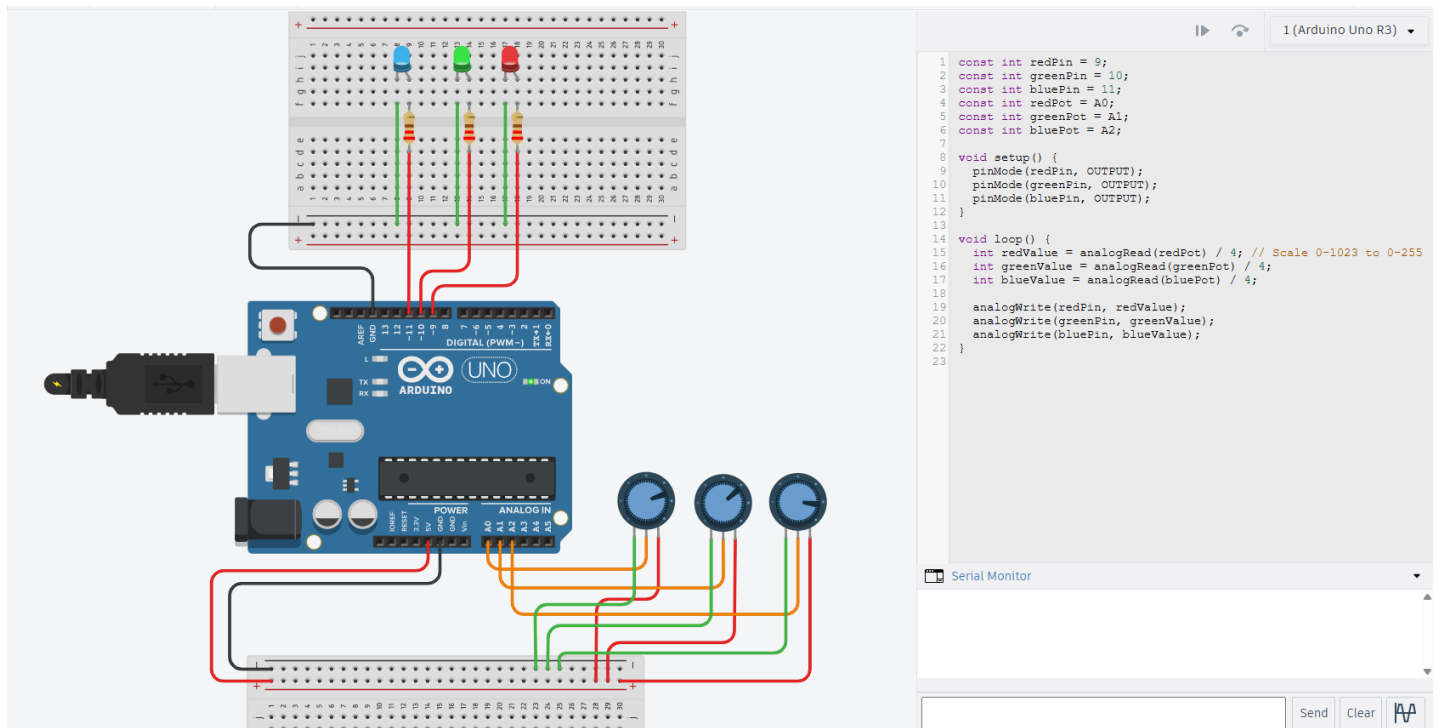
```
const int redPin = 9;
const int greenPin = 10;
const int bluePin = 11;
const int redPot = A0;
const int greenPot = A1;
const int bluePot = A2;

void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  int redValue = analogRead(redPot) / 4; // Scale 0-1023 to 0-255
  int greenValue = analogRead(greenPot) / 4;
  int blueValue = analogRead(bluePot) / 4;

  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}
```

Output:



Experiment 10

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

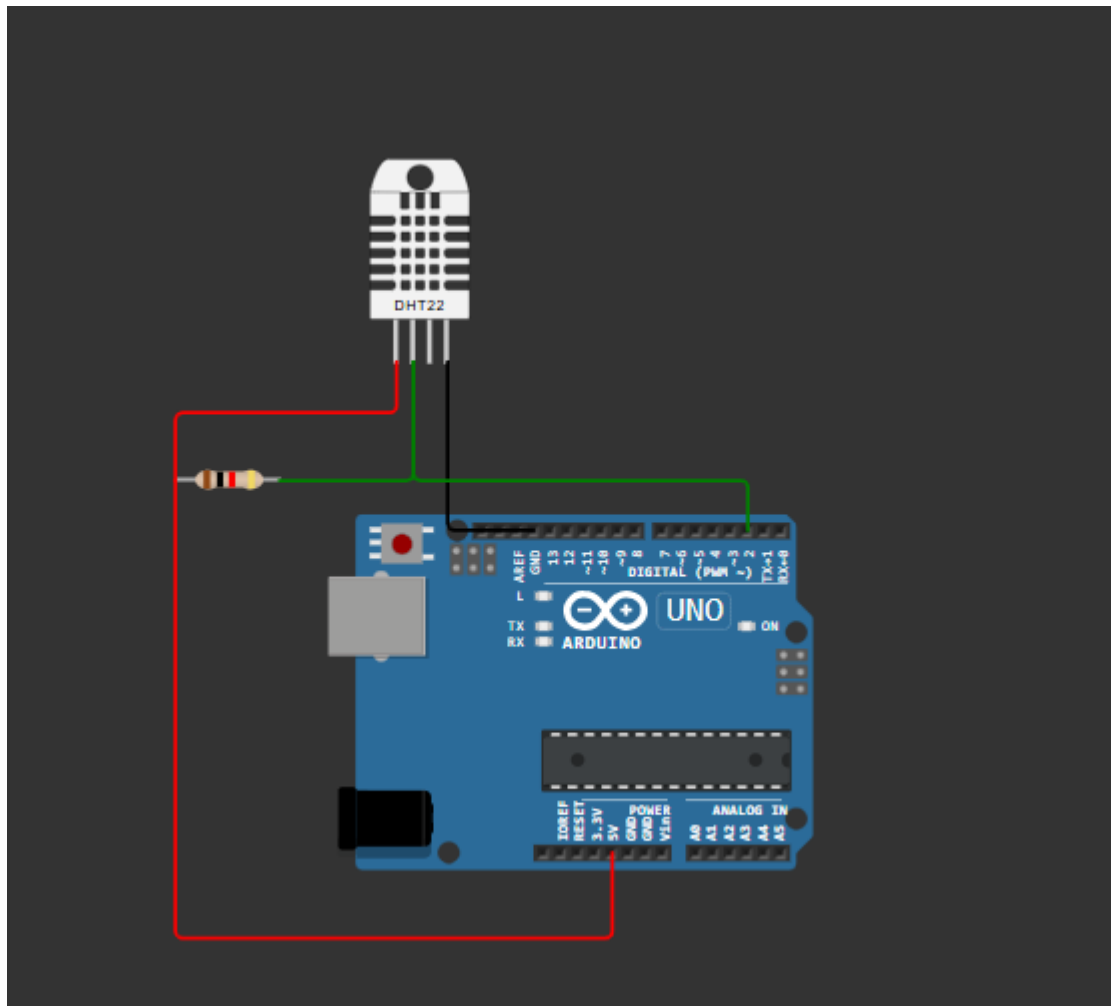
void setup() {
  Serial.begin(9600);
  dht.begin();
  Serial.println("DHT11 Temperature Monitor");
  Serial.println("-----");
  delay(1000);
}

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");  
}
```



Experiment 11

Write a program, read the temperature sensor and send the values to the serial monitor on the computer

```
#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 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;
  }
  Serial.print("Current Temperature: ");
  Serial.print(tempC, 1);
  Serial.println(" C");
}
```

Experiment 12

Write a program to show the temperature and shows a graph of the recent measurements

```
#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 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;
  }
  Serial.println(tempC, 1);
}
```


Practical 13

Write a program using piezo element and use it to play a tune after someone knocks

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

const int tuneNotes[] = {100, 200, 300, 400, 500};
const int noteCount = 5;
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);
}
```

