2a . Basic Programming using Arduino - LED and Switch Interface

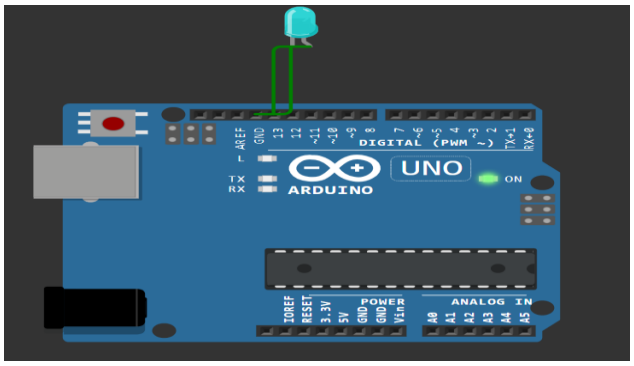
A.Blink the onboard LED in Uno

Program :

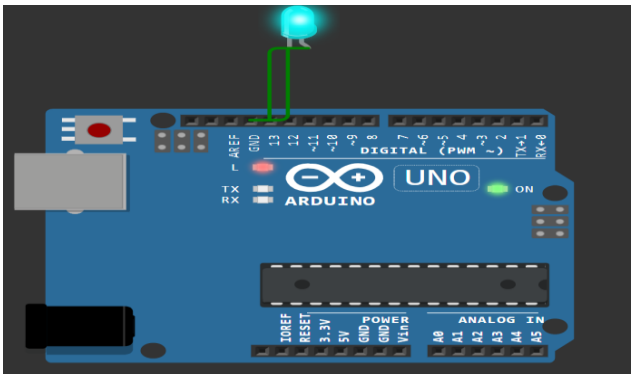
```
const int ledPin = 13;

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

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

Output:

LED OFF:



LED ON:



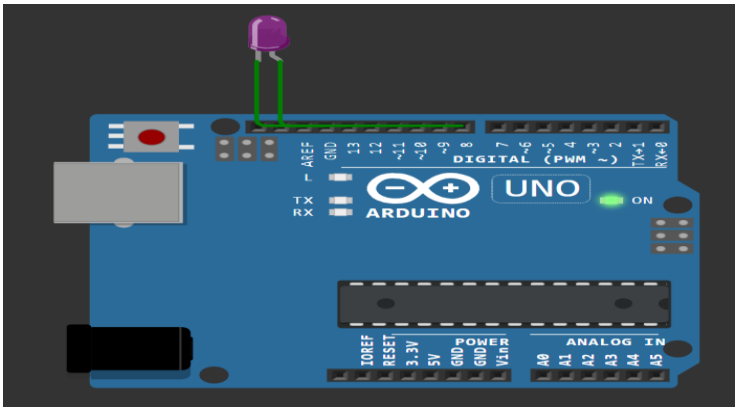
B . Blink the user LED in Uno using the port D8.

Program:

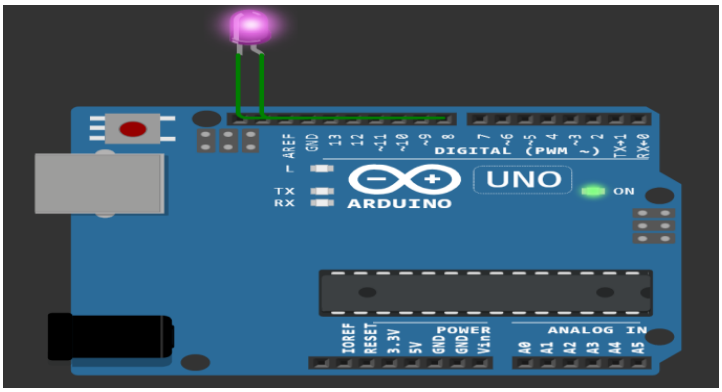
```
void setup() {
        pinMode(8, OUTPUT);
}
void loop() {
        digitalWrite(8,HIGH); delay(1000);
        digitalWrite(8,LOW); delay(1000);
}
```

Output:

USER LED OFF:



USER LED ON:



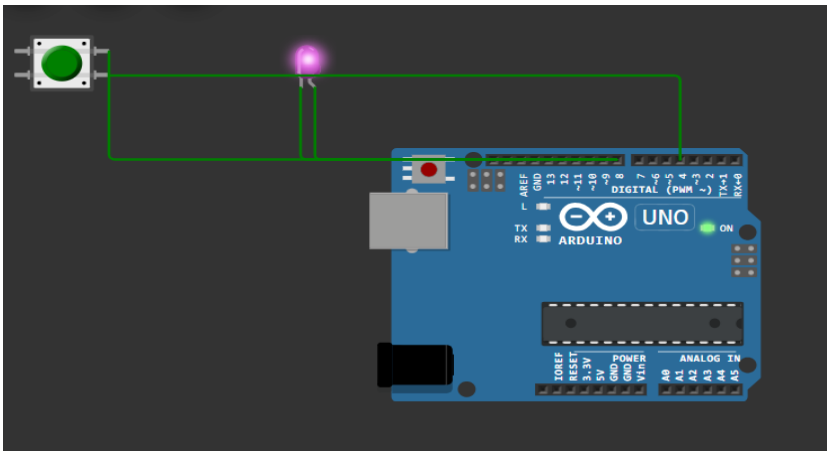
C. Switch ON/OFF a LED using a digital Switch-button using External Pullup variable.

Program:

```
int button = 0;
void setup()
{
 pinMode(4, INPUT);
 pinMode(8, OUTPUT);
}
void loop()
{
 button = digitalRead(4);
 if (button == LOW)
   digitalWrite(8, HIGH);
 else
   digitalWrite(8, LOW);
}
```

Output:

Button state: LOW



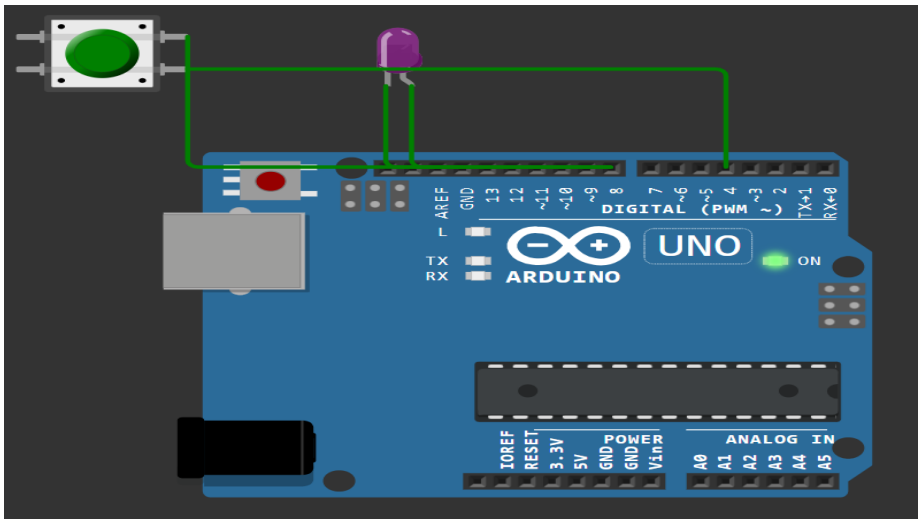
D. Switch ON/OFF a LED using a digital Switch-button using External Pulldown variable.

Program:

```
int button = 0;
void setup()
{
 pinMode(4, INPUT);
 pinMode(8, OUTPUT);
}
void loop()
{
 button = digitalRead(4);
 if (button == HIGH)
  digitalWrite(8, HIGH);
 else
  digitalWrite(8, LOW);
}
```

Output:

Button state: HIGH



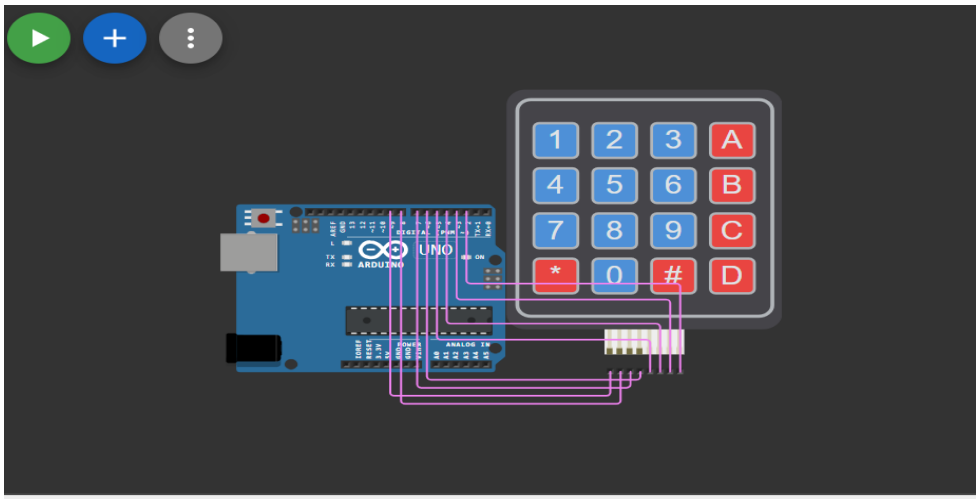
E. Switch/Keypad Interface:

Program:

```
const int ROWS = 4;
const int COLS = 4;
int rowPins[ROWS] = {9, 8, 7, 6};
int colPins[COLS] = {5, 4, 3, 2};
char keys[ROWS][COLS] = {
 {'0', '1', '2', '3'},
 {'4', '5', '6', '7'},
 {'8', '9', 'A', 'B'},
 {'C', 'D', 'E', 'F'}
};
void setup() {
 Serial.begin (9600);
 for (int i = 0; i < ROWS; i++)
 {
  pinMode(rowPins[i], INPUT_PULLUP);
 }
 for (int i = 0; i < COLS; i++)
 {
  pinMode(colPins[i], OUTPUT);
  digitalWrite(colPins[i], HIGH);
 }
}
void loop()
{
 for (int col = 0; col < COLS; col++)
 {
  digitalWrite(colPins[col], LOW);
  for (int row = 0; row < ROWS; row++)
  {
   if (digitalRead(rowPins[row]) == LOW)
   {
    Serial.println(keys[row][col]);
    delay(300);
   }
  }
```

```
    digitalWrite(colPins[col], HIGH);
  }
}
```

Output:




```
4
*
5
8
7
```

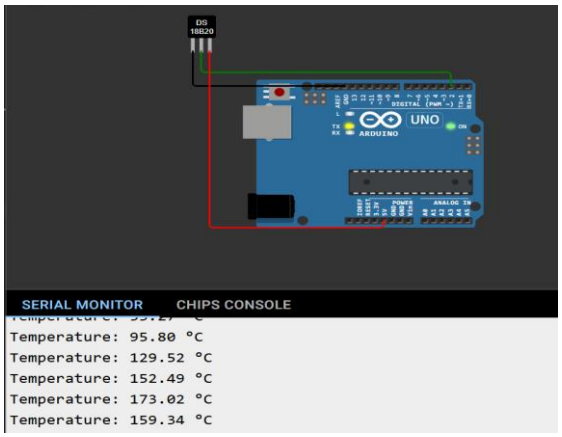
2b. Basic Programming using Arduino - Analog & Digital Sensor Interface

A. Temperature Sensor Analog Interface:

Program:

```
const int sensorPin = A0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(sensorPin);
  float voltage = sensorValue * (5.0 / 1023.0);
  float temperatureC = voltage * 100;
  Serial.print("Temperature: ");
  Serial.print(temperatureC);
  Serial.println(" °C");
  delay(1000);
}
```

Output:

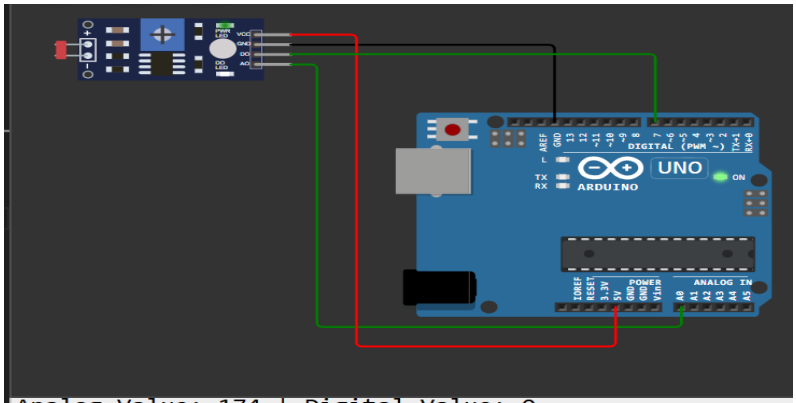


B. LDR Sensor Digital Interface:

Program:

```
const int ldrPin = A0;
void setup() {
 Serial.begin(9600);
}
void loop() {
 int analogValue = analogRead(ldrPin);
 int digitalValue = analogValue > 512 ? 1 : 0;
 Serial.print("Analog Value: ");
 Serial.print(analogValue);
 Serial.print(" | Digital Value: ");
 Serial.println(digitalValue);
 delay(500);
}
```

Output:




```
Analog Value: 174 | Digital Value: 0
Analog Value: 174 | Digital Value: 0
Analog Value: 834 | Digital Value: 1
Analog Value: 834 | Digital Value: 1
Analog Value: 834 | Digital Value: 1
Analog Value: 834 | Digital Value: 1
Analog Value: 834 | Digital Value: 1
```

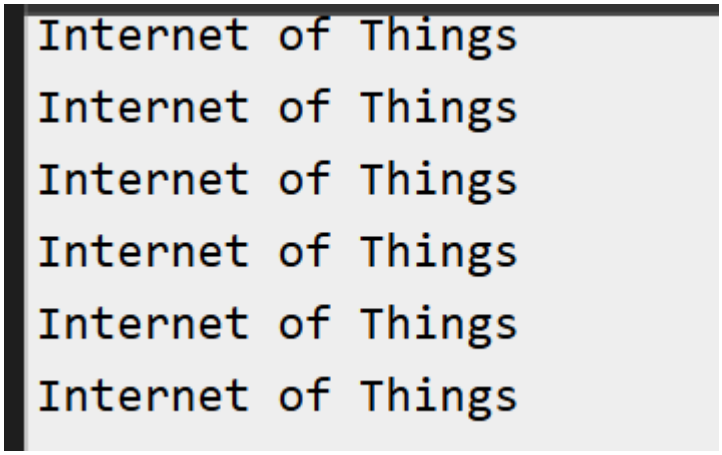
2c. Basic Programming using Arduino - Serial Communication

Task 1 : Displaying a message on the Serial Monitor

Program:

```
void setup()
{
  Serial.begin(9600);
  delay(1000);
}
void loop()
{
  Serial.println("Internet of Things");
}
```

Output:



Task 2 : Displaying a stored variable value on the Serial Monitor

Program:

```
void setup()
{
  Serial.begin(9600);
  delay(1000);
}
void loop()
{
  int x = 10;
  int y = 23;
  Serial.println(x);
  Serial.println(y);
}
```

Output:

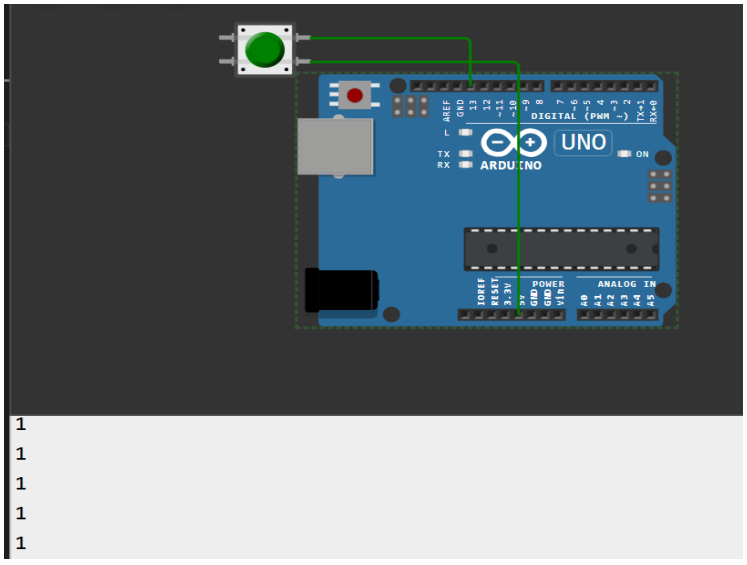
```
10
23
10
23
10
23
```

Task 3 : Displaying state of a digital input pin 9

Program:

```
void setup()
{
 Serial.begin(9600);
 delay(1000);
}
void loop()
{
 int value = digitalRead(13);
 Serial.println(value);
}
```

Output:

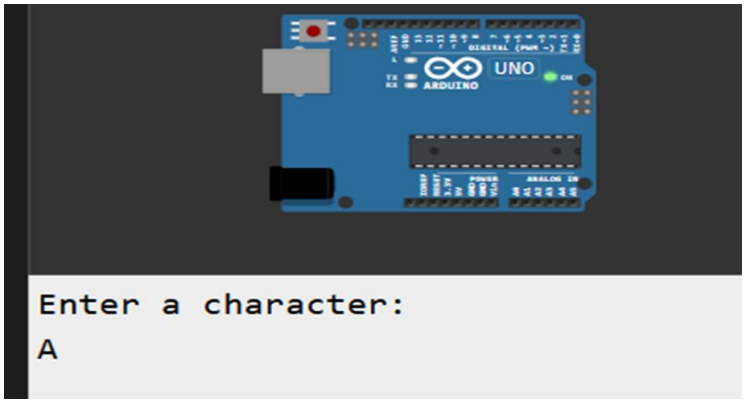


Task 4 : Interact with Arduino by saying your name through Serial monitor

Program:

```
char input;
void setup()
{
 Serial.begin(9600);
}
void loop()
{
 if (Serial.available() > 0)
 {
  input = Serial.read();
  Serial.println("Entered Character:");
  Serial.print(input);
 }
}
```

Output:

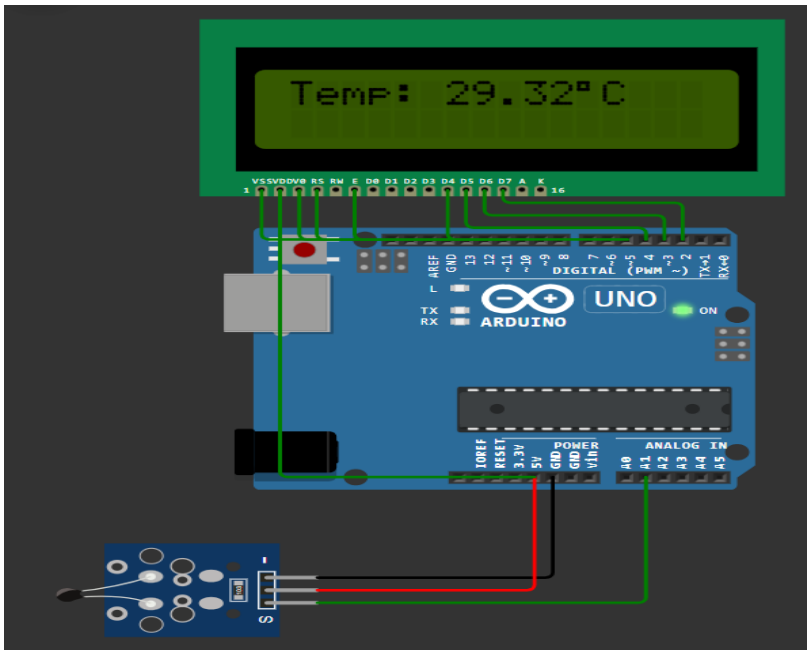


2d. Basic Programming using Arduino - Local display of sensor data using LCD

Program:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int tempPin = A1;
float temperature;

void setup() {
  lcd.begin(16, 2);
  pinMode(tempPin, INPUT);
}

void loop() {
  int analogValue = analogRead(tempPin);
  float voltage = analogValue * (5.0 / 1023.0);
  temperature = (voltage * 1.0) + (0.05 * 500);
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print((char)223);
  lcd.print("C");
  delay(1000);
  lcd.clear();
}
```

Output:



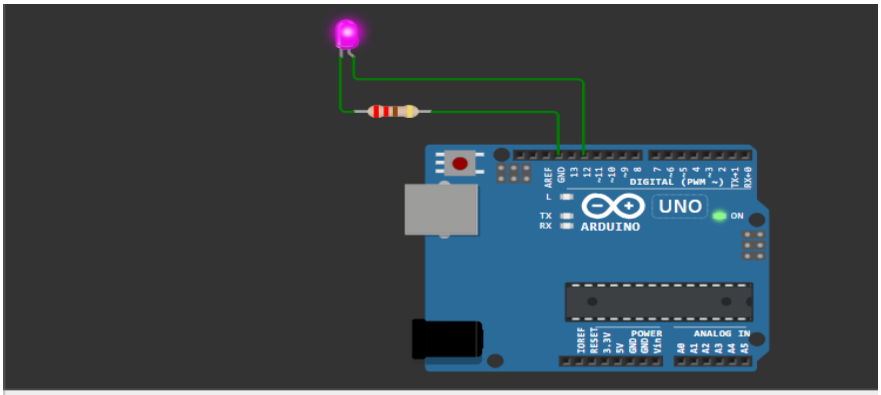
2e. Basic Programming using Arduino - Display of Sensor values in Mobile handset using Bluetooth

Program:

```
void setup() {
 Serial.begin(9600);         // Serial monitor (acts like Bluetooth in Wokwi)
 pinMode(12, OUTPUT);        // LED/Buzzer connected to pin 12
 Serial.println("Bluetooth Simulation Ready!");
 Serial.println("Type '1' to turn ON LED, '2' to turn OFF LED");
}
void loop() {
 char data; // declare here so it can be used inside this loop
 if (Serial.available() > 0) { // Check if data received
  data = Serial.read();      // Read the command
  // Ignore newline or carriage return
  if (data == '\n' || data == '\r') {
   return;
  }

  switch (data) {
   case '1': // Turn ON LED
    digitalWrite(12, HIGH);
    Serial.println("LED ON");
    break;

   case '2': // Turn OFF LED
    digitalWrite(12, LOW);
```

```
      Serial.println("LED OFF");
      break;

    default: // Any other key
      Serial.println("Invalid Command");
      break;
  }
 }
 delay(50); // Prevent flooding
}
```

Output:




```
Bluetooth Simulation Ready!
Type '1' to turn ON LED, '2' to turn OFF LED
LED ON
LED OFF
LED ON
```

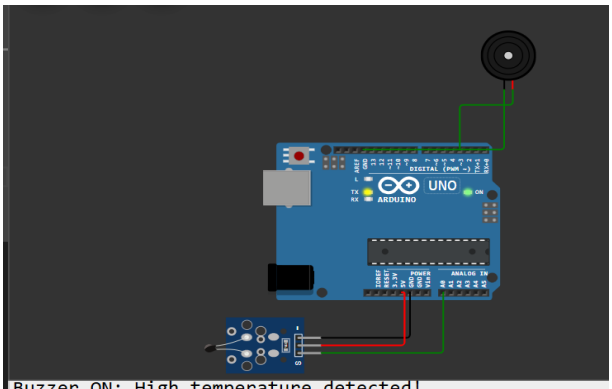
6. Design and development of a Temperature Detection System using LM35 Temperature sensor

Program:

```
// Define the analog pin for LM35 and digital pin for the buzzer
const int sensorPin = A0;
const int buzzerPin = 3;
// Temperature threshold in Celsius
const float tempThreshold = 35.0;
void setup() {
  // Initialize serial communication
  Serial.begin(9600);
  // Set the buzzer pin as output
  pinMode(buzzerPin, OUTPUT);
  // Turn the buzzer off initially
  digitalWrite(buzzerPin, LOW);
}
void loop() {
  // Read the analog value from the LM35 sensor
  int sensorValue = analogRead(sensorPin);
  // Convert the analog value to voltage
  float voltage = sensorValue * (5.0 / 1023.0);
  // Convert the voltage to temperature in Celsius
  float temperatureC = voltage * 100;
  // Print the temperature to the Serial Monitor
  Serial.print("Temperature: ");
  Serial.print(temperatureC);
```

```
  Serial.println(" °C");
  // Check if the temperature exceeds the threshold
  if (temperatureC > tempThreshold) {
    // Turn on the buzzer
    digitalWrite(buzzerPin, HIGH);
    Serial.println("Buzzer ON: High temperature detected!");
  } else {
    // Turn off the buzzer
    digitalWrite(buzzerPin, LOW);
    Serial.println("Buzzer OFF: Temperature is normal.");
  }
  // Delay for a short interval before the next reading
  delay(1000);
}
```

Output:



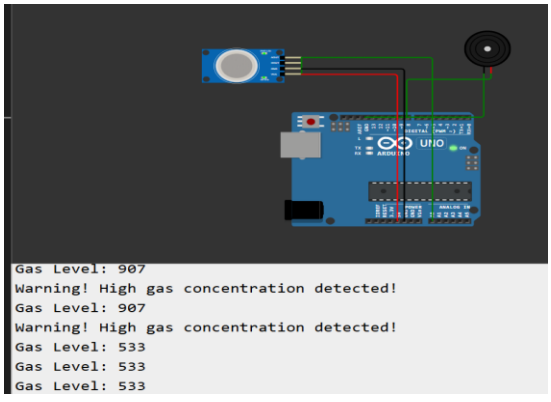

```
Buzzer ON: High temperature detected!
Temperature: 255.62 °C
Buzzer ON: High temperature detected!
Temperature: 255.62 °C
Buzzer ON: High temperature detected!
Temperature: 255.62 °C
Buzzer ON: High temperature detected!
```

7. Design and development of a Gas Detection System using MQ5 sensor

Program:

```
#define MQ5_PIN A0
#define BUZZER_PIN 8
void setup() {
 pinMode(BUZZER_PIN, OUTPUT);
 Serial.begin(9600);
}
void loop() {
 int gasValue = analogRead(MQ5_PIN);
 Serial.print("Gas Level: ");
 Serial.println(gasValue);
 if (gasValue > 800) {
  Serial.println("Warning! High gas concentration detected!");
 } else {
  digitalWrite(BUZZER_PIN, LOW);
 }
 delay(1000);
}
```

Output:



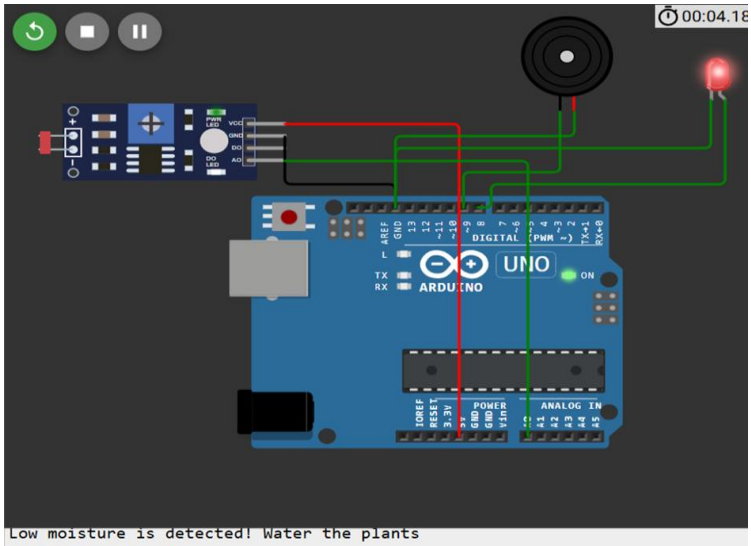
8. Design and development of a Moisture Detection System using Soil Moisture Sensor

Program:

```
#define sensorPin A0
#define sensorPower 7
#define LED_PIN 8
void setup () {
 pinMode(sensorPower, OUTPUT);
 Serial.begin(9600);
}
void loop () {
 // put your main code here, to run repeatedly:
 Serial.print("Analog output: ");
 int val = readSensor();
 Serial.println(val);
 if (val > 1000) {
  Serial.println("Low moisture is detected! Water the plants");
  digitalWrite(LED_PIN, HIGH);
 } else {
  digitalWrite(LED_PIN, LOW);
 }
 Delay (1000);
}
int readSensor() {
 digitalWrite(sensorPower, HIGH);
 delay (10);
 int val = analogRead(sensorPin);
```

```
  digitalWrite(sensorPower, LOW);
  return val;
}
```

Output:



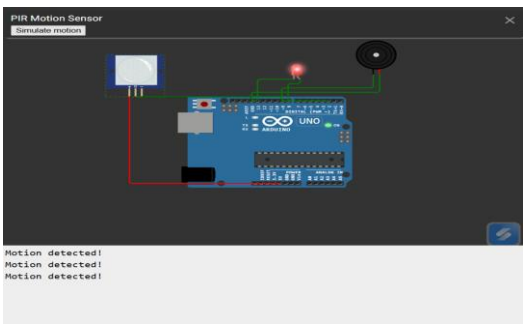

```
Low moisture is detected! Water the plants
Analog output: 1011
Low moisture is detected! Water the plants
Analog output: 1011
Low moisture is detected! Water the plants
Analog output: 1011
Low moisture is detected! Water the plants
```

9. Design and development of an Intrusion Detection System using PIR sensor

Program:

```
#define PIR_PIN 2 // Digital pin connected to PIR sensor OUT
#define LED_PIN 8 // LED connected to pin 8
void setup() {
 pinMode(PIR_PIN, INPUT); // Set PIR sensor as input
 pinMode(LED_PIN, OUTPUT); // Set LED pin as output
 Serial.begin(9600); // Start serial communication
}
void loop() {
 int motionDetected = digitalRead(PIR_PIN); // Read PIR sensor state
 if (motionDetected == HIGH) { // If motion is detected
  Serial.println("Motion detected!");
  digitalWrite(LED_PIN, HIGH); // Turn ON LED
  delay(5000); // Keep LED on for 5 seconds
 } else {
  digitalWrite(LED_PIN, LOW); // Turn OFF LED
 }
 delay(1000); // Wait 1 second before checking again
}
```

Output:



10. Design and development of a Heartbeat Monitoring system using Heart beat sensor.

Program:

```
#define SENSOR_PIN A0 // Heartbeat Sensor Pin

int bpm;

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

void loop() {
  int sensorValue = analogRead(SENSOR_PIN);
  bpm = map(sensorValue, 0, 1023, 60, 120); // Basic BPM mapping
  Serial.print("Heart Rate: ");
  Serial.print(bpm);
  Serial.println(" BPM");
  delay(1000);
}
```

Output: