Part A

Class B Tech CSE 4th Year **Sub: Internet of Things Lab**

Aim: Study and Demonstration of various Sensors, Relays, and Servo motor

Prerequisite: Basics of programming, microcontrollers and basic electronics

Outcome:

1. Learning of various sensor modules and their connections with microcontroller boards.
2. Controlling Servo motors using various Sensors and controlling devices using relays.

Theory:

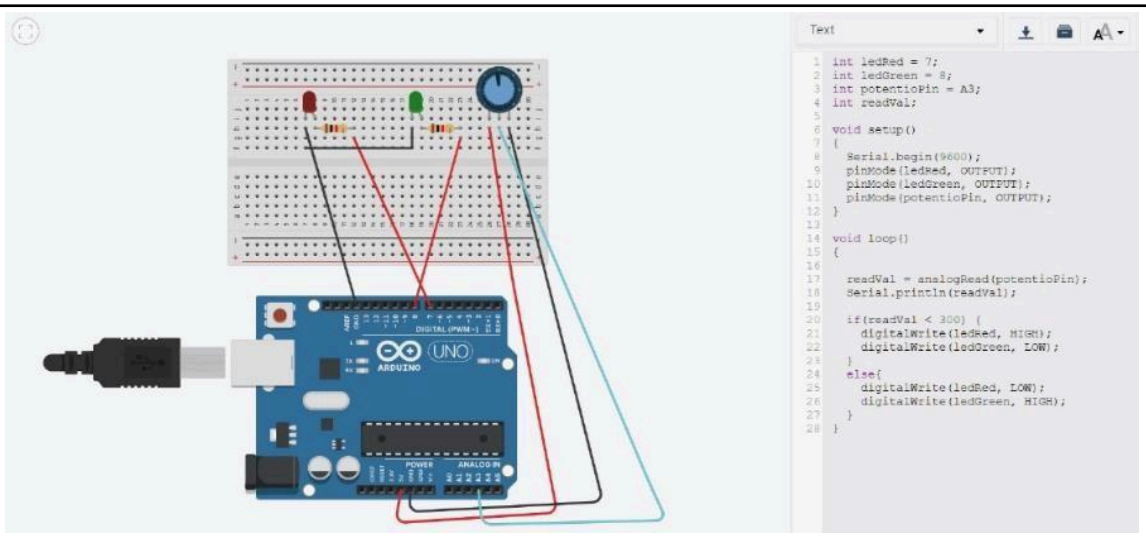
1. **Study various sensors, such as photosensitive, potentiometer, temperature, humidity, soil, etc.**
2. **Understanding Servo motors and controlling them using a photosensitive resistor and Potentiometer.**
3. **LED on/ off using a relay switch.**
4. **Create a circuit diagram for the Police Siren sound along with Red and Blue blinking lights.**

Part B (Write for an individual)

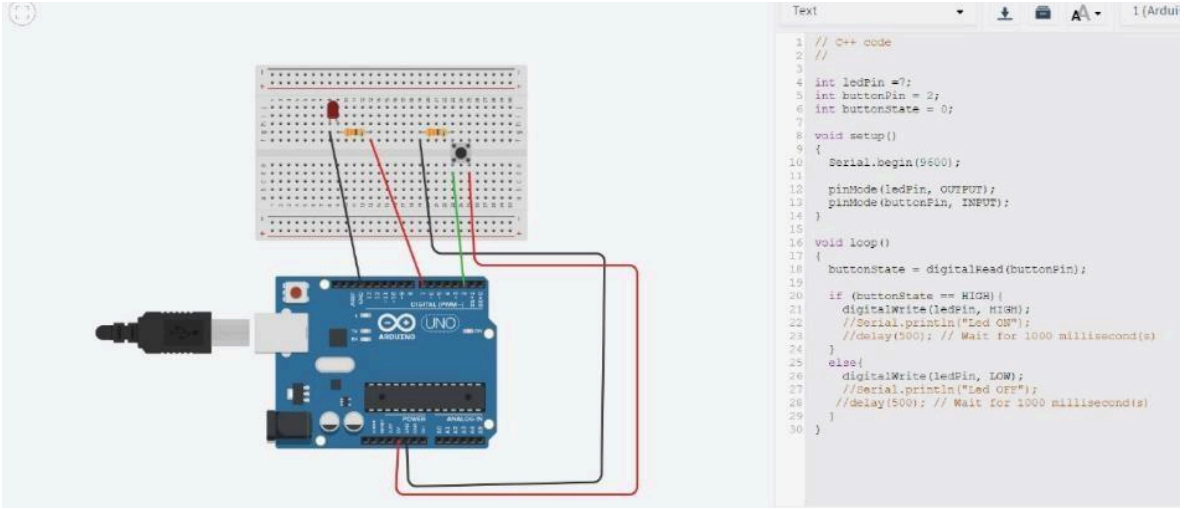
Steps:

1. Connect the photosensitive sensor to the Arduino/ESP board by linking its VCC and GND pins to the board's 5V and GND pins, respectively, and connect its signal output to an analog input pin on the board.
2. Connect the potentiometer to the Arduino/ESP board by attaching its VCC and GND pins to the board's 5V and GND pins, and connect its signal output to an analog input pin.
3. Connect the temperature sensor (e.g., LM35) to the Arduino/ESP board by wiring its VCC and GND pins to the board's 5V and GND pins, and connect its output pin to an analog input pin.
4. Connect the humidity sensor (e.g., DHT11) to the Arduino/ESP board by connecting its VCC and GND pins to the board's 5V and GND pins, and connect its data pin to a digital input pin.
5. Connect the soil moisture sensor to the Arduino/ESP board by linking its VCC and GND pins to the board's 5V and GND pins, and connect its signal output to an analog input pin.
6. Wire the relay module to the Arduino/ESP board by connecting the relay's VCC and GND pins to the board's 5V and GND pins, and connect the relay's input pin to a digital output pin.
7. Connect the servo motor to the Arduino/ESP board by attaching the servo's VCC and GND pins to the board's 5V and GND pins, and connect the servo's control pin to a PWM-capable digital output pin.
8. Develop and upload code to the Arduino/ESP board to read data from the sensors, control the relay, and adjust the servo motor position based on sensor inputs.
9. Test the setup by observing sensor readings, relay operation, and servo movement in response to changes in sensor values.

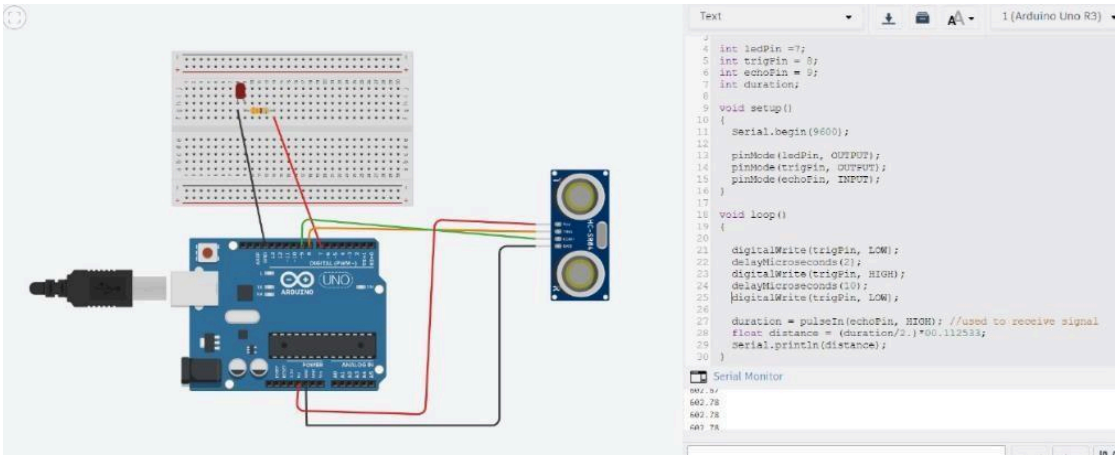
Output:



```
int ledRed = 7;
int ledGreen = 8;
int potentioPin = A3;
int readVal;

void setup()
{
  Serial.begin(9600);
  pinMode(ledRed, OUTPUT);
  pinMode(ledGreen, OUTPUT);
  pinMode(potentioPin, OUTPUT);
}

void loop()
{

  readVal = analogRead(potentioPin);
  Serial.println(readVal);

  if(readVal < 300) {
    digitalWrite(ledRed, HIGH);
    digitalWrite(ledGreen, LOW);
  }
  else{
    digitalWrite(ledRed, LOW);
    digitalWrite(ledGreen, HIGH);
  }
}
```



```
// C++ code
//

int ledPin =7;
int buttonPin = 2;
int buttonState = 0;

void setup()
{
  Serial.begin(9600);

  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH){
    digitalWrite(ledPin, HIGH);
    //Serial.println("Led ON");
    //delay(500); // Wait for 1000 millisecond(s)
  }
  else{
    digitalWrite(ledPin, LOW);
    //Serial.println("Led OFF");
   //delay(500); // Wait for 1000 millisecond(s)
  }
}
```



```
int ledPin =7;
int trigPin = 8;
int echoPin = 9;
int duration;

void setup()
{
  Serial.begin(9600);

  pinMode(ledPin, OUTPUT);
  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); //used to receive signal
  float distance = (duration/2.)*00.112533;
  Serial.println(distance);
}
```

Observation & Learning:

The sensors successfully provided accurate readings, with the photosensitive sensor responding to light changes, the potentiometer adjusting according to its position, the temperature sensor reflecting temperature variations, the humidity sensor showing changes in humidity levels, and the soil moisture sensor indicating soil moisture content. The relay module functioned correctly, turning on and off as commanded, and the servo motor responded accurately to control signals, adjusting its position as expected.

Through this experiment, you learned how to interface various sensors and actuators with an Arduino/ESP board. You gained practical experience in reading data from different types of sensors, controlling relays for switching applications, and manipulating servo motors for precise movements. This enhanced your understanding of integrating and managing multiple components in a circuit and their applications in real-world scenarios.

Conclusion:

In conclusion, the experiment effectively demonstrated the integration and functionality of various sensors, relays, and a servo motor with an Arduino/ESP board. The sensors provided accurate data, the relay module operated correctly for switching tasks, and the servo motor adjusted its position precisely. This experiment reinforced your ability to interface multiple components, manage their interactions, and apply them in practical applications, deepening your understanding of sensor integration and control systems.