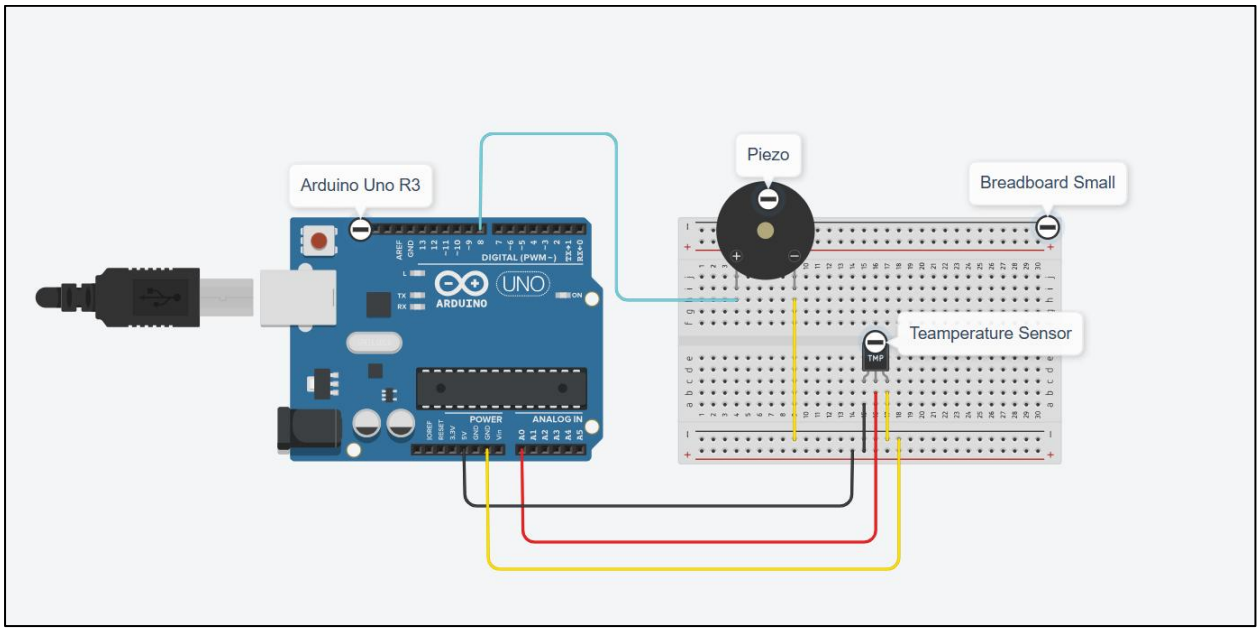
Experiment No. 01

Title - Write a program for sending alert messages to the user for controlling and interacting with your environment.

Program

```
// Environment Alert System using LM35 Temperature Sensor and Buzzer
// Components: Arduino Uno, LM35, Piezo Buzzer
const int tempPin = A0;   // LM35 sensor connected to Analog pin A0
const int buzzerPin = 9;  // Buzzer connected to Digital pin 9
float temperature;  // Variable to store temperature reading
void setup() {
  Serial.begin(9600);         // Start Serial Communication
  pinMode(buzzerPin, OUTPUT); // Set buzzer pin as output
  Serial.println("Environment Monitoring System Started...");
}
void loop() {
  int sensorValue = analogRead(tempPin);  // Read analog signal from LM35
  temperature = (sensorValue * 5.0 * 100.0) / 1024.0;  // Convert to °C
  Serial.print("Current Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
  // High temperature alert
  if (temperature > 35.0) {
    Serial.println("ALERT: High Temperature Detected!");
    tone(buzzerPin, 1000);  // Turn on buzzer sound
    delay(1000);
    noTone(buzzerPin);      // Turn off buzzer
  }
  // Low temperature alert
  else if (temperature < 15.0) {
    Serial.println("WARNING: Low Temperature Detected!");
    tone(buzzerPin, 500);
    delay(500);
    noTone(buzzerPin);
  }
  // Normal temperature condition
  else {
    Serial.println("Temperature Normal.");
    noTone(buzzerPin);
  }
  delay(2000);  // Wait 2 seconds before next reading
}
```

System Architecture



Output

Serial Monitor

```
Environment Monitoring System Started...
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Environment Monitoring System Started...
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
Current Temperature: 74.71 Â°C
ALERT: High Temperature Detected!
```

Expriment No.2

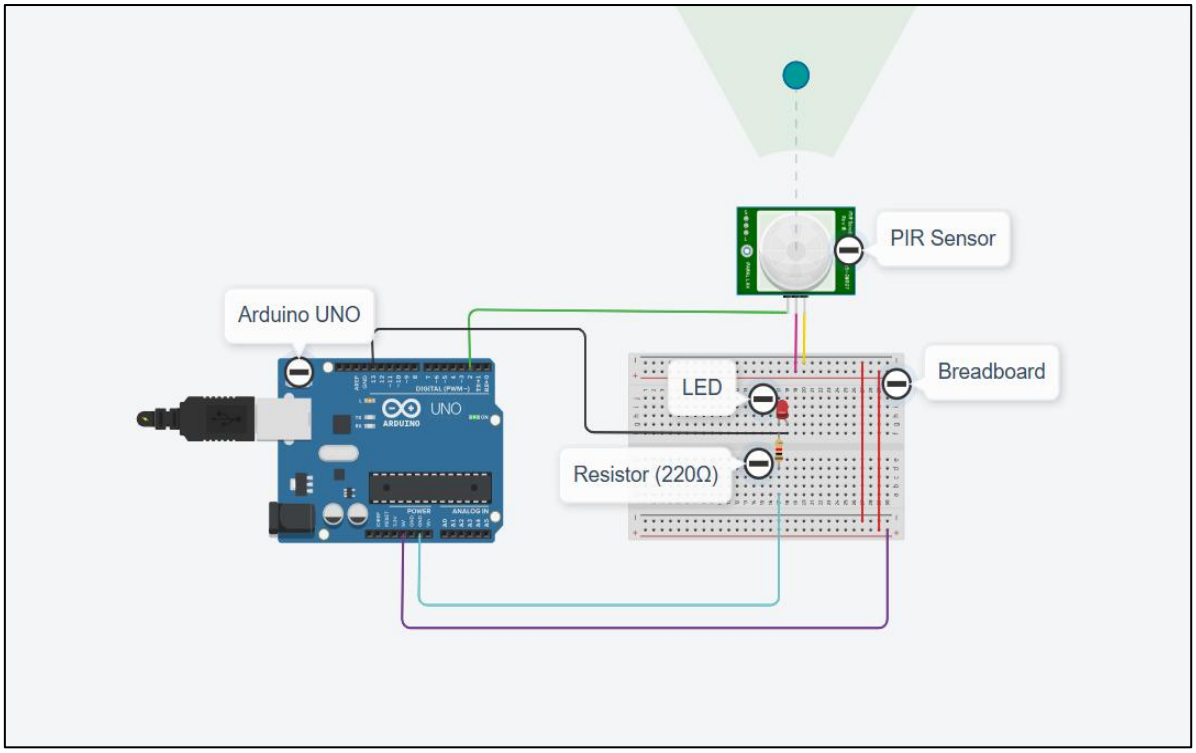
Title - Write an Arduino/ Raspberry pi program for interfacing with PIR sensor Experiment.

Program

```
// PIR Motion Detection System using Arduino UNO
// Author: Shrikrishna Sutar
// Platform: Tinkercad Simulation

const int pirPin = 2;     // PIR sensor output pin
const int ledPin = 13;    // LED pin
int motionState = LOW;    // To store current motion status
int val = 0;              // Sensor reading
void setup() {
  pinMode(pirPin, INPUT);   // PIR sensor as input
  pinMode(ledPin, OUTPUT);  // LED as output
  Serial.begin(9600);       // Start serial communication
  Serial.println("PIR Motion Detection System Initialized...");
  delay(2000);              // Sensor stabilization time
}
void loop() {
  val = digitalRead(pirPin);  // Read PIR sensor value
  if (val == HIGH) {          // Motion detected
    digitalWrite(ledPin, HIGH);
    if (motionState == LOW) {
      Serial.println("⚠️ Motion Detected!");
      motionState = HIGH;
    }
  }
  else {                      // No motion
    digitalWrite(ledPin, LOW);
    if (motionState == HIGH) {
      Serial.println("No Motion.");
      motionState = LOW;
    }
  }
  delay(500);  // Short delay for stability
}
```

System Architecture



Output

Serial Monitor

```
PIR Motion Detection System Initialized...
â Ï¸ Motion Detected!
PIR Motion Detection System Initialized...
â Ï¸ Motion Detected!
```

Experiment No. 03

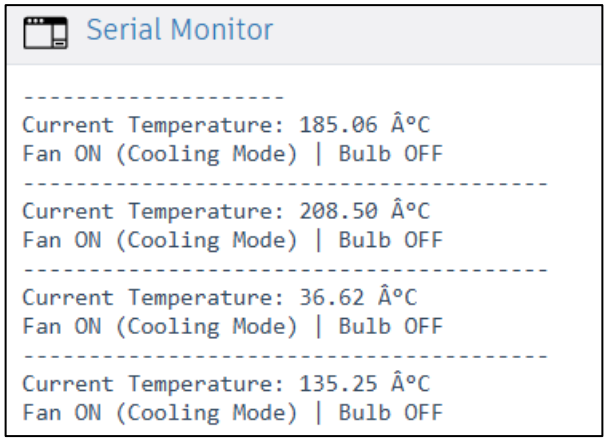
Title - Write a program for developing an IIoT application for energy monitoring and optimization.

Program

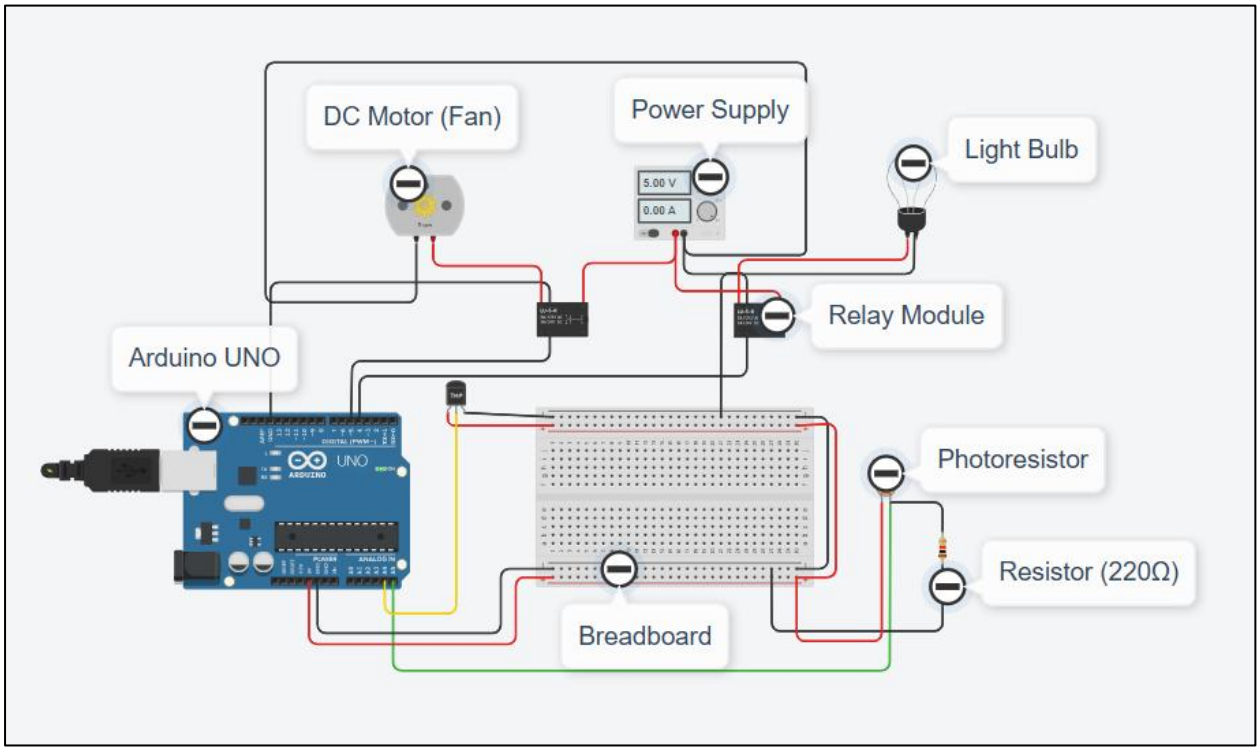
```
// IIoT-Based Smart Home Energy Monitoring and Optimization
// Platform: Tinkercad Simulation
const int tempPin = A0;     // TMP36/LM35 temperature sensor pin
const int bulbRelay = 7;    // Relay for bulb
const int motorRelay = 8;   // Relay for fan (DC motor)
float temperature;          // To store temperature readings
void setup() {
  Serial.begin(9600);
  pinMode(tempPin, INPUT);
  pinMode(bulbRelay, OUTPUT);
  pinMode(motorRelay, OUTPUT);
  Serial.println("Smart Home Energy Monitoring System Initialized...");
  delay(2000);
}
void loop() {
  int sensorValue = analogRead(tempPin);
  // Convert analog reading to temperature (for TMP36 or LM35)
  temperature = (sensorValue * 5.0 * 100.0) / 1024.0;
  Serial.print("Current Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
  // Energy optimization logic
```

```
 if (temperature > 30) {
  digitalWrite(motorRelay, HIGH);  // Turn ON fan
  digitalWrite(bulbRelay, LOW);    // Turn OFF bulb
  Serial.println("Fan ON (Cooling Mode) | Bulb OFF");
 }
 else if (temperature < 20) {
  digitalWrite(bulbRelay, HIGH);   // Turn ON bulb
  digitalWrite(motorRelay, LOW);   // Turn OFF fan
  Serial.println("Bulb ON (Heating/Lighting Mode) | Fan OFF");
 }
 else {
  digitalWrite(bulbRelay, LOW);
  digitalWrite(motorRelay, LOW);
  Serial.println("Temperature Normal - Energy Saving Mode");
 Serial.println("----------------------------------------");
 delay(2000);
}
```

Output



System Architecture



Experiment No. 04

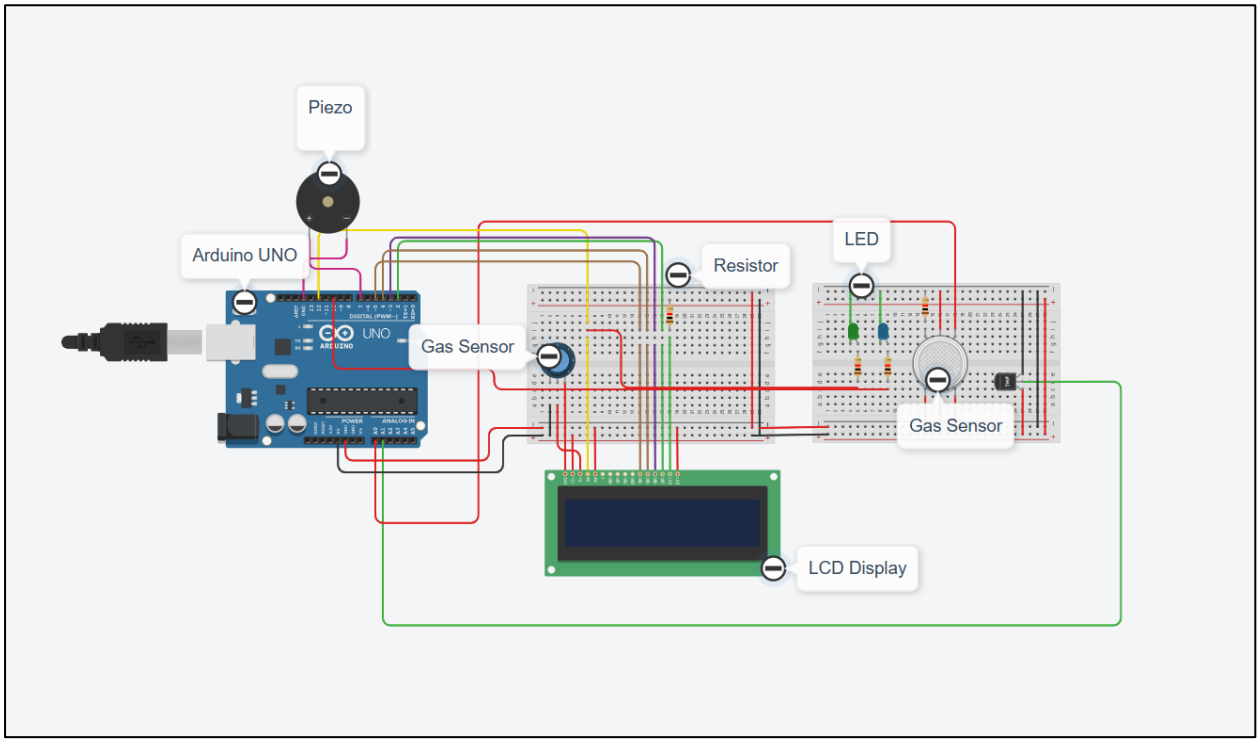
Title - Write a program for implementing security measures in an IIoT system.

Program

```
#include <LiquidCrystal.h>
// LCD Pins: RS, EN, D4, D5, D6, D7
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
const int gasSensorPin = A0;   // Gas sensor analog input
const int tempSensorPin = A1;  // TMP sensor analog input
const int potPin = A2;         // Potentiometer for simulated access control
const int buzzerPin = 8;       // Buzzer pin
const int redLed = 9;          // Red LED - Alert
const int greenLed = 10;       // Green LED - Safe
int gasValue = 0;
int tempValue = 0;
int accessValue = 0;
bool authenticated = false;
void setup() {
  pinMode(buzzerPin, OUTPUT);
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(gasSensorPin, INPUT);
  pinMode(tempSensorPin, INPUT);
  pinMode(potPin, INPUT);
  lcd.begin(16, 2);
  Serial.begin(9600);
  lcd.print("IIoT Security Sys");
  lcd.setCursor(0, 1);
  lcd.print("Initializing...");
  delay(2000);
  lcd.clear();
}
void loop() {
  // --- Authentication Simulation ---
  accessValue = analogRead(potPin);
  if (accessValue > 600) {  // If potentiometer turned to high
    authenticated = true;
  } else {
    authenticated = false;
  }
  gasValue = analogRead(gasSensorPin);
  tempValue = analogRead(tempSensorPin);
  float temperature = (tempValue * 5.0 / 1024.0) * 100; // TMP sensor formula
  // --- Display Sensor Data ---
```

```
  lcd.setCursor(0, 0);
  lcd.print("Gas:");
  lcd.print(gasValue);
  lcd.print(" Tmp:");
  lcd.print((int)temperature);
  lcd.setCursor(0, 1);
  if (authenticated) {
    lcd.print("Access: OK     ");
  } else {
    lcd.print("Access: DENIED ");
  }
  // --- Security Logic ---
  if (!authenticated) {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzerPin, 1000);
    Serial.println("Unauthorized Access Detected!");
  } else {
    // Check for sensor abnormalities
    if (gasValue > 400 || temperature > 40) {
      digitalWrite(redLed, HIGH);
      digitalWrite(greenLed, LOW);
      tone(buzzerPin, 1500);
      lcd.clear();
      lcd.print("⚠ Alert Triggered!");
      delay(2000);
      lcd.clear();
      Serial.println("Alert: High Gas or Temp!");
    } else {
      digitalWrite(redLed, LOW);
      digitalWrite(greenLed, HIGH);
      noTone(buzzerPin);
    }
  }
  delay(1000);
}
```

System Architecture



Experiment No. 05

Title - Write A Program For Sending Sensor Data To The Cloud And Storing It In A Database.

Program

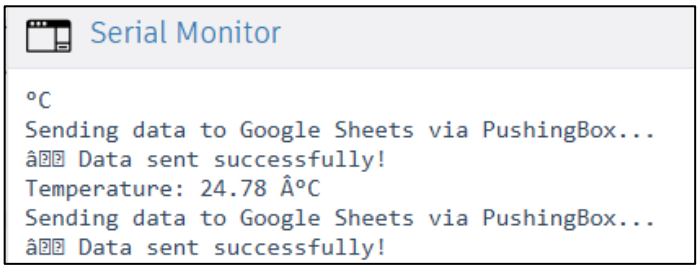
```
int sensorPin = A0; // TMP36 connected to A0
float temperatureC;
void setup() {
  Serial.begin(9600);
  Serial.println("IIoT Energy Monitoring System Started...");
}
void loop() {
  int sensorValue = analogRead(sensorPin);
  float voltage = sensorValue * (5.0 / 1023.0);
  temperatureC = (voltage - 0.5) * 100.0;

  // Simulate data upload
  Serial.print("Temperature: ");
  Serial.print(temperatureC);
  Serial.println(" °C");

  Serial.println("Sending data to Google Sheets via PushingBox...");
  Serial.println("☑ Data sent successfully!");

  delay(10000); // every 10 seconds
}
```

Output



System Architecture