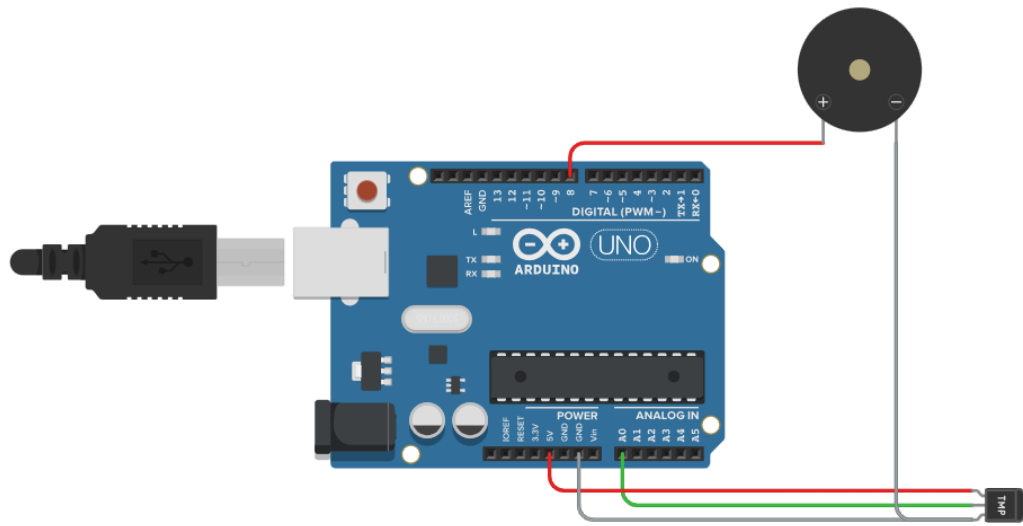


P (IIoT)

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

```
#define TEMP_PIN A0 // Pin where the TMP36 sensor is connected
#define BUZZER_PIN 8 // Buzzer pin
const float TEMPERATURE_THRESHOLD = 23.0; // Temperature threshold in Celsius
void setup() {
  // Initialize the buzzer pin as an output
  pinMode(BUZZER_PIN, OUTPUT);
  // Start the Serial Monitor for debugging
  Serial.begin(9600);
}
void loop() {
  // Read the temperature from the TMP36 sensor
  int tempReading = analogRead(TEMP_PIN);
  float voltage = tempReading * (5.0 / 1023.0);
  float temperatureC = (voltage - 0.5) * 100.0;
  // 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 > TEMPERATURE_THRESHOLD) {
    // Turn on the buzzer
    digitalWrite(BUZZER_PIN, HIGH);
    // Print an alert message to the Serial Monitor
    Serial.println("ALERT: Temperature is too high!");
  } else {
    digitalWrite(BUZZER_PIN, LOW);
  }
  // Wait for a short period before the next loop
  delay(1000);
}
```



Output :

```
ALERT: Temperature is too high!  
Temperature: 24.78 C  
ALERT: Temperature is too high!  
Temperature: 24.78 C  
ALERT: Temperature is too high!  
Temperature: 24.78 C  
ALERT: Temperature is too high!  
Temperature: 55.08 C  
ALERT: Temperature is too high!  
Temperature: 61.93 C  
ALERT: Temperature is too high!  
Temperature: 61.93 C  
ALERT: Temperature is too high!  
Temperature: 75.12 C  
ALERT: Temperature is too high!  
Temperature: 101.03 C  
ALERT: Temperature is too high!  
Temperature: 101.03 C
```

Pr (IIoT)

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

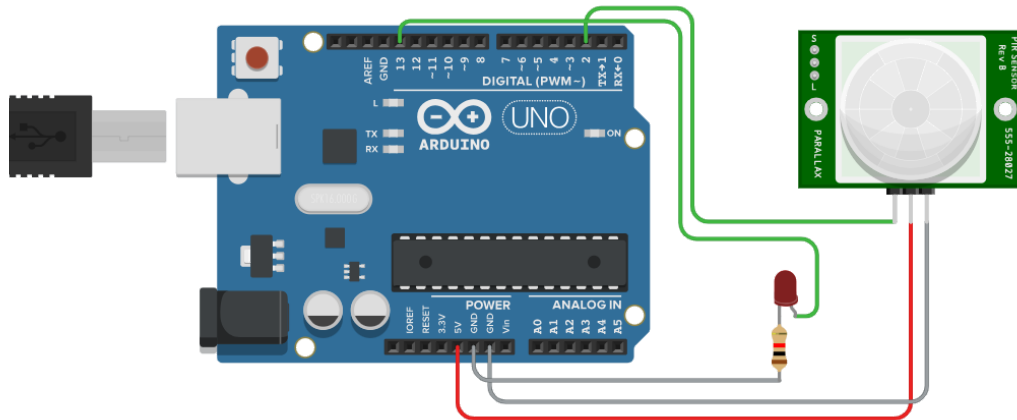
```
// Define pin numbers

const int pirPin = 2; // PIR sensor input pin

const int ledPin = 13; // LED output pin (built-in on many Arduino boards)

void setup() {
  pinMode(pirPin, INPUT); // Set PIR pin as input
  pinMode(ledPin, OUTPUT); // Set LED pin as output
  Serial.begin(9600); // Initialize serial communication for debugging
}

void loop() {
  int pirState = digitalRead(pirPin); // Read PIR sensor state
  if (pirState == HIGH) { // If motion is detected
    digitalWrite(ledPin, HIGH); // Turn on LED
    Serial.println("Motion detected!");
  } else {
    digitalWrite(ledPin, LOW); // Turn off LED
    Serial.println("No motion");
  }
  delay(500); // Wait for a second before rechecking
}
```



Output :

```
No motion
No motion
No motion
No motion
No motion
No motion
No motion
No motion
Motion detected!
Motion detected!
No motion
```

Pr (IIoT)

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

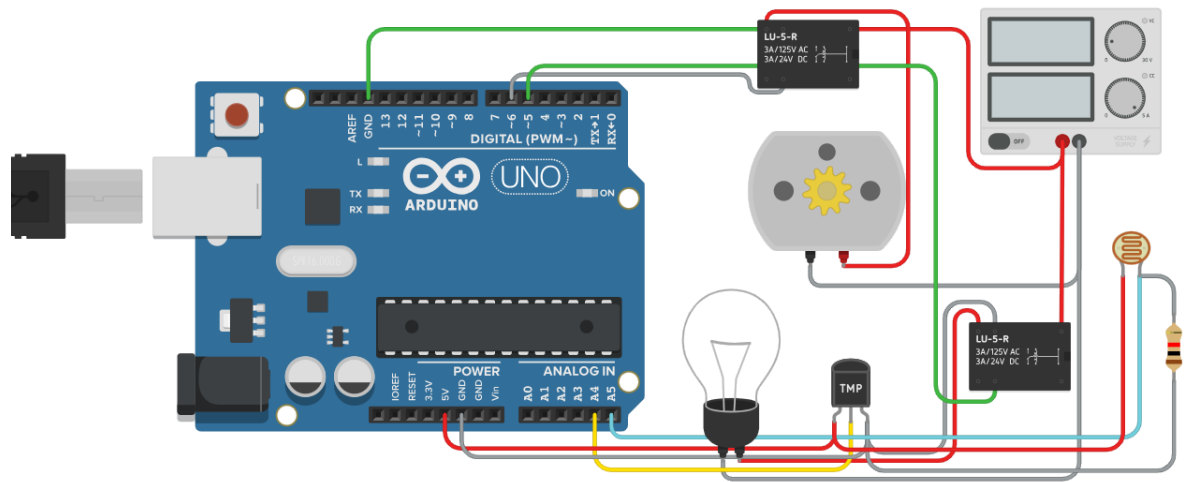
```
float x,y,z,temp;

void setup()
{
  // pinMode(8, INPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(A5, INPUT);
  Serial.begin(9600);
}

void loop()
{
  // x= digitalRead(8);
  y= analogRead(A5);
  z= analogRead(A4);
  Serial.println(x);
  Serial.println(y);
  Serial.println(z);
  temp = (double)z / 1024;
  temp = temp * 5;
  temp = temp - 0.5;
  temp = temp * 100;
  //if ( (x>0) )
  //{
  if ((y<550)&&(temp>30))

  {
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
  }
  else if((y<550)&&(temp<30))
```

```
{  
digitalWrite(5, HIGH);  
digitalWrite(6, LOW);  
}  
else if((y>550)&&(temp>30))  
{  
digitalWrite(5, LOW);  
digitalWrite(6, HIGH);  
}  
else if((y>550)&&(temp<30))  
{  
digitalWrite(5, LOW);  
digitalWrite(6, LOW);  
}  
/*else  
{  
digitalWrite(5, LOW);  
digitalWrite(6, LOW);  
}*/  
}
```



Output :

```

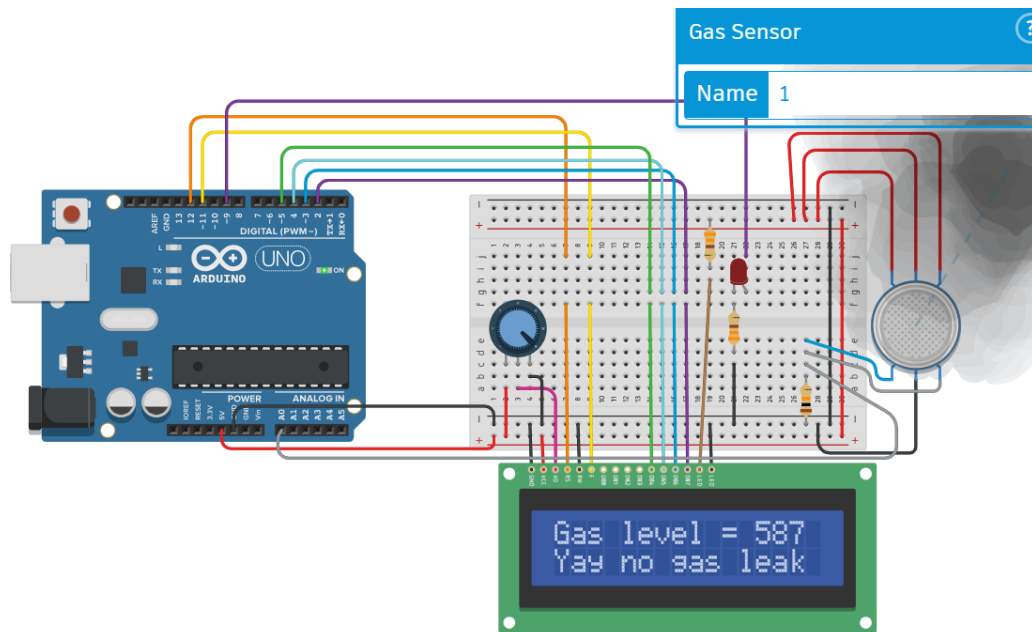
21.00
0.00
411.00
121.00
0.00
362.00
121.00
0.00
323.00
121.00
0.00
323.00
121.00
0.00
323.00
121.00

```

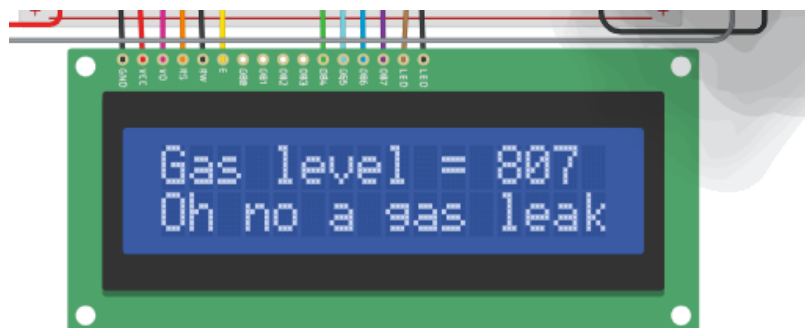
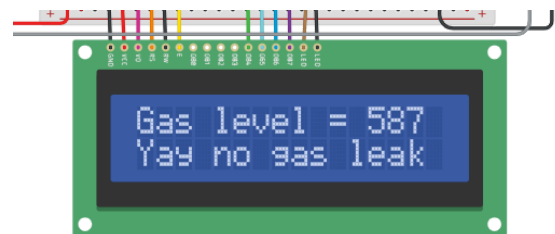
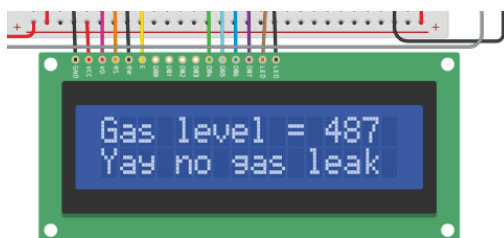
Pr (IIoT)

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

```
#include <LiquidCrystal.h>
int gas;
int wait = 100;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  lcd.begin(16, 2);
  pinMode(9,OUTPUT);
  pinMode(A0,INPUT);
}
void loop() {
  gas = analogRead(A0);
  if(gas>680){
    digitalWrite(9,HIGH);
    lcd.setCursor(0,0);
    lcd.print("Gas level = ");
    lcd.print(gas);
    lcd.setCursor(0,1);
    lcd.print("Oh no a gas leak");
    delay(wait);
  }
  else {
    digitalWrite(9,LOW);
    lcd.setCursor(0,0);
    lcd.print("Gas level = ");
    lcd.print(gas);
    lcd.setCursor(0,1);
    lcd.print("Yay no gas leak ");
    delay(wait);
  }
}
```

Output :



(IIoT)

Write a program for sending sensor data to the cloud and storing it in a database

```
//076260305129-8 62/2
String ssid = "Simulator Wifi"; // SSID to connect to
String password = ""; // Our virtual wifi has no password
String host = "api.thingspeak.com"; // Open Weather Map API
const int httpPort = 80;
String uri = "/update?api_key=HD64O90IN8CFLR1Q&field1=0&field2=0";

int setupESP8266(void) {
    // Start our ESP8266 Serial Communication
    Serial.begin(115200); // Serial connection over USB to computer
    Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266
    delay(10); // Wait a little for the ESP to respond
    if (!Serial.find("OK")) return 1;

    // Connect to 123D Circuits Simulator Wifi
    Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
    delay(10); // Wait a little for the ESP to respond
    if (!Serial.find("OK")) return 2;

    // Open TCP connection to the host:
    Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\", " + httpPort);
    delay(50); // Wait a little for the ESP to respond
    if (!Serial.find("OK")) return 3;

    return 0;
}

void anydata(void) {

    int temp = map(analogRead(A0),20,358,-40,125);

    // Construct our HTTP call
    String httpPacket = "GET " + uri + String(temp) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
    int length = httpPacket.length();

    // Send our message length
    Serial.print("AT+CIPSEND=");
    Serial.println(length);
    delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

    // Send our http request
```

}



S



Host: `api.thingspeak.com`

```
AT+CIPSEND=95
GET /update?api_key=HD64090IN8CF1R1Q&field1=0&field2=024 HTTP/1.1
Host: api.thingspeak.com
```

```
AT+CIPSEND=95
GET /update?api_key=HD64090IN8CFLR1Q&field1=0&field2=024 HTTP/1.1
Host: api.thingspeak.com
```

```
AT+CIPSEND=95
GET /update?api_key=HD64090IN8CFLR1Q&field1=0&field2=024 HTTP/1.1
```

Pr (IIoT)

Write a program for building a small-scale IIoT network using wireless communication protocols

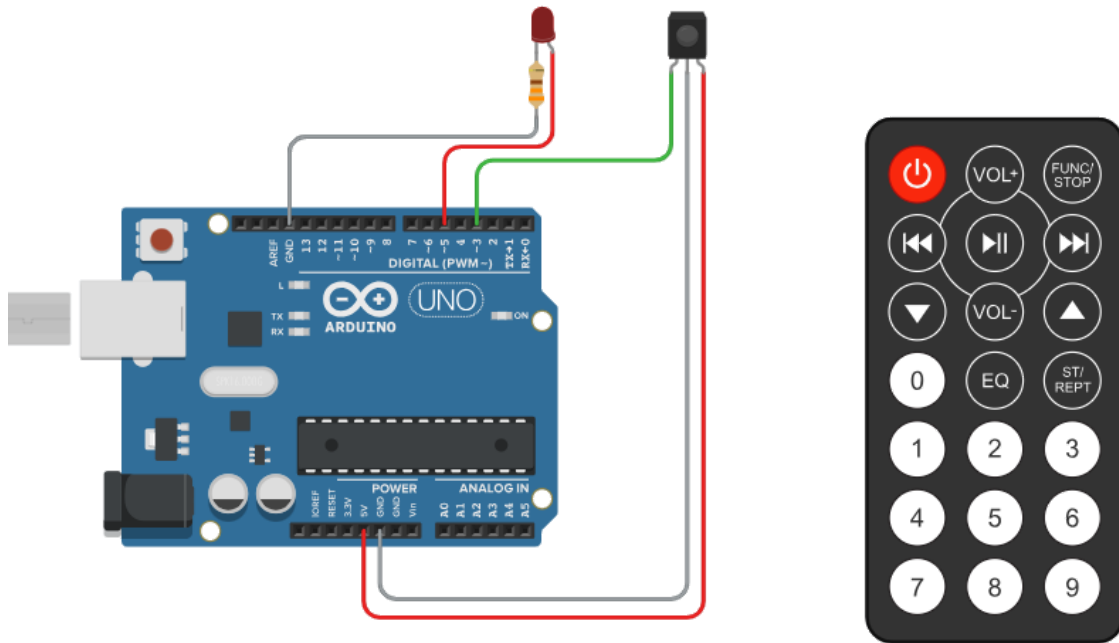
```
#include <IRremote.hpp>

const int rcvPin=3;
IRrecv irrecv(rcvPin);
decode_results results;

void setup()
{
    Serial.begin(9600);
    irrecv.enableIRIn(); // Start the receiver
    pinMode(5, OUTPUT);
}

void loop() {
    if(IrReceiver.decode()) {
        auto value= IrReceiver.decodedIRData.decodedRawData;
        //switch(results.value)
        switch(value)
        {
            case 4010852096:
                Serial.println("1"); // Button 1
                digitalWrite(5,HIGH);
                break;
            case 3994140416: // Template
                Serial.println("2"); // Button
                digitalWrite(5,LOW);
                break;
            default: Serial.println(value);
        }
        IrReceiver.resume(); // Receive the next value
    }
}
```

Circuit



Output :

