Date	19 November 2022
Team ID	PNT2022TMID53557
Project Name	**Project –**Gas Leakage Monitoring and Alerting System

SOURCE CODE:

```
#include<Servo.h>

#include<LiquidCrystal_I2C.h>

#include <WiFi.h>

#include <PubSubClient.h>

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

int GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial sgps(13, 15); //Rx ,  Tx gps

SoftwareSerial sgsm(3, 1); // Rx ,  Tx gsm

LiquidCrystal_I2C lcd(32, 16, 2);

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

#define ORG "oqhi1j"//IBM ORGANITION ID

#define DEVICE_TYPE "NODEMCU"//Device type mentioned in ibm watson
IOT Platform

#define DEVICE_ID "BHAVAN0108"//Device ID mentioned in ibm watson
IOT Platform

#define TOKEN "bharathi0503" //Token

String data3;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/Gas/fmt/json";

char publishTopic2[] = "iot-2/evt/Data/fmt/json";
```

```
char subscribetopic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);

#define KNOB 3
Servo myservo;
int green = 2;
int yellow = 3;
int red = 4;
int siren = 5;
int gas = A3;
int sensorValue = 0;
int c = 0;
int m = 0;
String latitude = "0.000000";
String longitude = "0.000000";
int t=0,h=0,p=0;

void setup()
{
  Serial.begin(9600);
  myservo.attach(KNOB);
  myservo.write(90);
  sgsm.begin(9600);
```

```
  sgps.begin(9600);
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("GAS LEAKAGE");
  lcd.setCursor(4,1);
  lcd.print("DETECTION");
  delay(3000);
  lcd.clear();
  pinMode(green, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(siren, OUTPUT);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
  wificonnect();
  mqttconnect();
}

void loop()
{
  sensorValue = random(500,1000);
  t=random(36,38);
  h=random(62,68);
  p=random(20,80);
```

```
Serial.print("Temperature: ");
Serial.println(t);
Serial.print("Humidity: ");
Serial.println(h);
Serial.print("Gas Level: ");
Serial.println(sensorValue);
Serial.print("Pressure: ");
Serial.println(p);
if(g >= 750)
{
  latitude = "13.147327";
  longitude = "80.226269";
}
else
{
  latitude = "0.000000";
  longitude = "0.000000";
}
if(sensorValue > 500 && c==0)
{
  lcd.clear();
  Serial.println("GAS DETECTED");
  Serial.println("SMS: GAS IS DETECTED!!");
  myservo.write(90);
  Serial.println("SMS: THE KNOB IS CLOSED");
  sendSMS("GAS IS DETECTED!!");
  sendSMS("THE KNOB IS CLOSED");
```

```
  lcd.setCursor(0,0);
  lcd.print("GAS DETECTED");
  lcd.setCursor(0,1);
  lcd.print("KNOB IS CLOSED");
  delay(1000);
  c=1;
  m=1;
 }
if(sensorValue < 499)
 {
  c=0;
  myservo.write(0);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Gas Value: ");
  lcd.setCursor(11,0);
  lcd.print(sensorValue);
  if(m == 1)
  {
    Serial.println("LEAKAGE STOPPED");
    Serial.println("THE KNOB IS OPENED");
    lcd.setCursor(0,0);
    lcd.print("LEAKAGE STOPPED");
    lcd.setCursor(0,1);
    lcd.print("THE KNOB IS OPENED");
    m=0;
    sendSMS("LEAKAGE HAS BEEN STOPPED");
```

```
  sendSMS("THE KNOB IS OPENED");
}
if(sensorValue > 500)
{
  lcd.setCursor(0,1);
  lcd.print("GAS DETECTED");
  digitalWrite(red, HIGH);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
  tone(siren, 200);
}
else if(sensorValue > 281 && sensorValue < 500)
{
  lcd.setCursor(0,1);
  lcd.print("            ");
  digitalWrite(yellow, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(green, LOW);
  noTone(siren);
}
else
{
  lcd.setCursor(0,1);
  lcd.print("            ");
  digitalWrite(green, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
```

```
      noTone(siren);
    }
    delay(1000);
  }
}

void sendSMS(char*message)
{
  while (sgps.available() > 0)
    if (gps.encode(sgps.read()))
    {
      if (gps.location.isValid())
      {
        sgsm.listen();
        sgsm.print("\r");
        delay(1000);
        sgsm.print("AT+CMGF=1\r"); // AT COMMAND TO SEND SMS
        delay(1000);
        sgsm.print("AT+CMGS=\"+919025681637\"\r"); // REGISTERED
NUMBER TO SEND SMS
        delay(1000);
        //The text of the message to be sent.
        sgsm.print(message);
        sgsm.print("https://www.google.com/maps/?q="); // MAPS
        sgsm.print(gps.location.lat(), 6); // LAT
        sgsm.print(",");
        sgsm.print(gps.location.lng(), 6); // LONG    delay(1000);
        sgsm.write(0x1A);
```

```
        delay(1000);
      }
    }
}


void PublishData(int temp, int hum, int gas, int pres, String lat, String lng)
{
  mqttconnect();
  String payload2 = "{\"d\":{\"temperature\":";
  payload2 += t;
  payload2 += ","\"humidity\":";
  payload2 +=h;
  payload2 += ","\"gasLevel\":";
  payload2 += gas;
  payload2 += ","\"pressure\":";
  payload2 += pres;
  payload2 += ","\"latitude\":";
  payload2 += lat;
  payload2 += ","\"longitude\":";
  payload2 += lng;
  payload2 +="}}";
  Serial.print("Sending Payload: ");
  Serial.println(payload2);
  if (client.publish(publishTopic2, (char*) payload2.c_str()))
  {
    Serial.println("Published");
```

```
  }
  else
  {
    Serial.println("Not Published");
  }
}


void mqttconnect()
{
  if (!client.connected())
  {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token))
    {
      Serial.print(".");
      delay(500);
    }
  initManagedDevice();
  Serial.println();
  }
}



void wificonnect()
```

```
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}



void initManagedDevice()
{
  if (client.subscribe(subscribetopic))
  {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  }
  else
  {
    Serial.println("subscribe to cmd FAILED");
```

```
  }
}



void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++)
  {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  data3="";
}
```