FINAL CODE

Date	18 NOVEMBER 2022
Team ID	PNT2022TMID35909
Project Name	Gas Leakage Monitoring & Alerting System for Industries

```
#include <LiquidCrystal.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define LED 23
#define BUZZER_PIN 19  // define type of sensor DHT 11
String command;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

#define ORG "ckdbr5"//IBM ORGANITION ID
#define DEVICE_TYPE "123"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "252725"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "27252527"     //Token
String data3;
int trigger;
float h, t;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and
format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command
type AND
char authMethod[] = "use-token-auth";// authentication method

char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

const int DHT_PIN = 25;
DHTesp dhtSensor;
LiquidCrystal lcd(4,15,5,18,21,22);
int ThreshHold = 60;

WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
void setup() {
  Serial.begin(9600);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  lcd.begin(16,2);
  pinMode(LED,OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  wificonnect();
  mqttconnect();
}


void loop() {

  delay(5000);
  digitalWrite(LED, HIGH);
  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  Serial.println("Temperature: " + String(data.temperature, 2) + "°C");
  Serial.println("Humidity: " + String(data.humidity, 1) + "%");

float gassensor=random(0,100);

  Serial.print(F("Gas Concentration: "));
  Serial.println(gassensor);

  if (gassensor>ThreshHold)
  {
    trigger=1;
    Serial.println(F("GAS LEAKED ALERT!"));
    Serial.println();
    lcd.clear();
    lcd.print ("GAS LEAKAGE :(");
    tone(BUZZER_PIN,31);
    delay (1000);
    lcd.clear();
    lcd.print ("ALERT!!!");
    delay(1000);
    noTone(BUZZER_PIN);

  }

  else
  {
    trigger=0;
    Serial.println(F("SAFE!"));
    Serial.println();
    lcd.clear();

lcd.print ("ALL GOOD :)");
    delay(1000);
    lcd.clear();
    lcd.print ("SAFE!");
    delay(1000);
  }

  PublishData(data.temperature,data.humidity,gassensor,trigger);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

/*....................................retrieving to Cloud...............................*/

void PublishData(float temp, float humid , float sensorvalue ,int trigger) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"Temperature\":";
  payload += temp;
  payload += "," "\"Humidity\":";
  payload += humid;
  payload += "," "\"GasConcentration\":";
  payload += sensorvalue;
  payload += "," "\"Status\":";
  payload += trigger;
  payload += "}";

  Serial.print("Sending payload: ");
  Serial.println(payload);

  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print
publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
    delay(2000);
  }
}
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() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  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)
{
  delay(3000);
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {

    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="shutoff")
  {
    Serial.println(data3);
    digitalWrite(LED,LOW);
  }
  data3="";
}
```