Project Development Phase
SPRINT 1

Domain	Internet of Things
Team ID	PNT2022TMID29662
Project Name	Gas Leakage monitoring and Alerting system

Connect sensors and Arduino with python.

Program:

```
      #include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt
#include <stdlib.h>
#include <time.h>
#include <Stepper.h>

const int stepsPerRevolution = 200;  // change this to fit the number of steps
per revolution
// for your motor

// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 15, 2, 4, 5);


int led=22;
int buzzer=21;
String data3;

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
#define ORG "q1cni1"//IBM ORGANITION ID
#define DEVICE_TYPE "gas1"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1361"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token

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 COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential


void setup() {
  Serial.begin(9600);
  Serial.println("Hello, ESP32!");
  pinMode(22, OUTPUT);
  pinMode(21, OUTPUT);
  delay(10);
  myStepper.setSpeed(60);
  wificonnect();
  mqttconnect();
}

void loop() {
  srand((int)time(0));
  int gas = rand()%1000;
  delay(2000);
  Serial.println(gas);
  if(gas>400){
    digitalWrite(led,1);
    tone(buzzer,262,1000);
    myStepper.step(stepsPerRevolution);
    PublishData(gas);
    delay(5000);
  }
  else{
    digitalWrite(led,0);
    PublishData1(gas);
  }
}

//String alert="ALERT";
void PublishData(float distance) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"GAS_Concentration\":";
  payload += distance;
  payload += "," "\"Message\":\"ALERT\"";
  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");
  }

}
void PublishData1(float distance) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"GAS_Concentration\":";
  payload += distance;
  payload += "," "\"Message\":\"SAFE\"";
  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");
  }

}

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)
{

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

Circuit Diagram: