ASSIGNMENT 4

Team ID: **PNT2022TMID52302**
Project Name: **Gas Leakage monitoring & Alerting system for Industries**

Submitted by **PADMANABHAN P (963519106025)**

QUESTION:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.
Upload document with wokwi share link and images of IBM cloud

Solution Code:

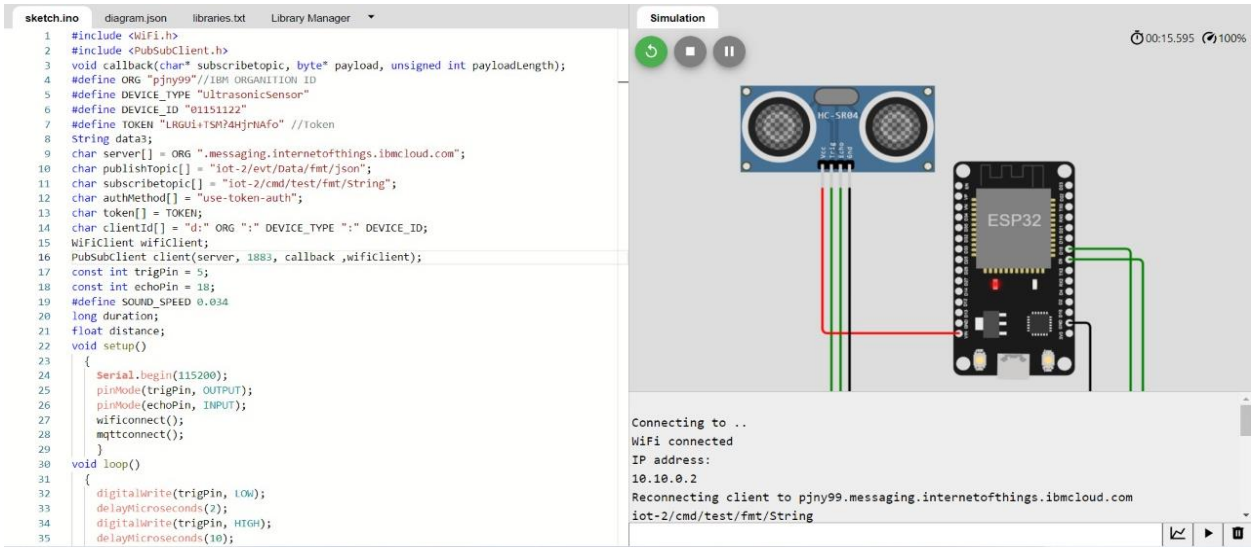
```
#include<WiFi.h>
#include<PubSubClient.h>
voidcallback(char* subscribetopic, byte* payload, unsignedintpayloadLength);
#define ORG "pjny99"//IBM ORGANITION ID
#define DEVICE_TYPE "UltrasonicSensor"
#define DEVICE_ID "01151122"
#define TOKEN "LRGUi+TSM?4HjrNAfo"//Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
charpublishTopic[] = "iot-2/evt/Data/fmt/json";
charsubscribetopic[] = "iot-2/cmd/test/fmt/String";
charauthMethod[] = "use-token-auth";
char token[] = TOKEN;
charclientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClientwifiClient;
PubSubClientclient(server, 1883, callback ,wifiClient);
constinttrigPin = 5;
constintechoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
voidsetup()
  {
    Serial.begin(115200);
    pinMode(trigPin, OUTPUT);
```

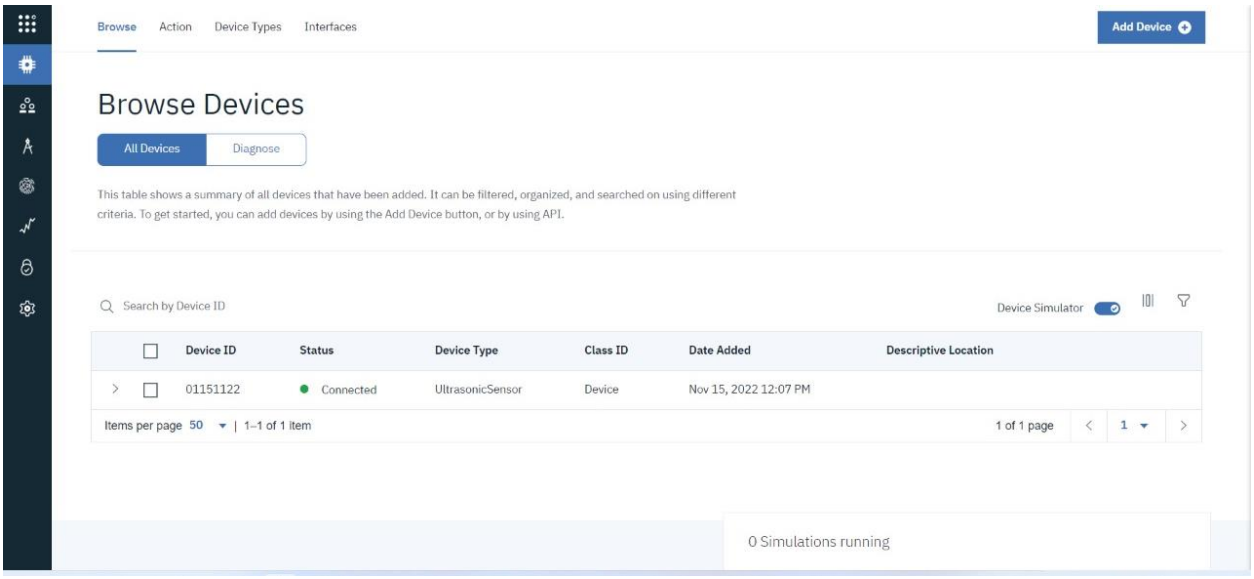
```
    pinMode(echoPin, INPUT);
    wificonnect();
    mqttconnect();
    }
voidloop()
  {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * SOUND_SPEED/2;
    Serial.print("Distance (cm): ");
    Serial.println(distance);
    if(distance<100) {
      Serial.println("ALERT!!");
      delay(1000);
      PublishData(distance);
      delay(1000);
      if (!client.loop()) {
        mqttconnect();
        }
      }
    delay(1000);
    }
voidPublishData(floatdist)
{
  mqttconnect();
  String payload = "{\"Distance\":";
  payload += dist;
  payload += ",\"ALERT!!\":""\"Distance less than 100cms\"";
  payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
    } else {
    Serial.println("Publish failed");
    }
  }
voidmqttconnect()
{
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
```

```
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print("."); delay(500);
      }
    initManagedDevice();
    Serial.println();
    }
  }
voidwificonnect()
{
  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());
  }
voidinitManagedDevice()
{
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
    } else {
    Serial.println("subscribe to cmd FAILED");
    }
  }
voidcallback(char* subscribetopic, byte* payload, unsignedintpayloadLength)
{
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (inti = 0; i<payloadLength; i++) {
    Serial.print((char)payload[i]);
    data3 += (char)payload[i];
    }
  Serial.println("data: "+ data3);
  data3="";
}
```

wokwi link : https://wokwi.com/projects/347736057197888083

OUTPUT :


sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
#define ORG "pjny99"//IBM ORGANITION ID
#define DEVICE_TYPE "UltrasonicSensor"
#define DEVICE_ID "01151122"
#define TOKEN "LRGUi+TSM?4HjrNAfo" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/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);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup()
{
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
Simulation
00:15.595
100%
HC-SR04
ESP32
Connecting to ..
WiFi connected
IP address:
10.10.0.2
Reconnecting client to pjny99.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String


Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
01151122
Connected
UltrasonicSensor
Device
Nov 15, 2022 12:07 PM
Items per page 50
1–1 of 1 item
1 of 1 page
1
0 Simulations running