

ASSIGNMENT 4

Date	6 nov 2022
Team ID	PNT2022TMID26079
Project Name	Gas Leakage Monitoring and Alerting System
Name	D.SANJAY DARSHAN

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

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "cyu5zw"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_Controller"
#define DEVICE_ID "ultrasonic_sensor"
#define TOKEN "LT(R4?kOsKh@bMY&aC"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/event_1/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);
  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);
}
```

```

}
void PublishData(float dist) {
  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");
  }
}

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="";
}

```

Wokwi link:

<https://wokwi.com/projects/347930226528879186>

output:
WOKWI

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  WiFiClient wifiClient;
4  String data3;
5  #define ORG "cyu5zw"//IBM ORGANITION ID
6  #define DEVICE_TYPE "ESP32_Controller"//Device type mentioned in ibm
7  #define DEVICE_ID "ultrasonic_sensor"//Device ID mentioned in ibm wat
8  #define TOKEN "LT(R4?kOsKh@bMY&aC"
9  #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/event_1/fmt/json";
13 char topic[] = "iot-2/cmd/led/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
```

↺

■

⏸

00:59.698 95%

Connecting to Wifi..WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to cyu5zw.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

Sending payload: {"Distance":399.94}
Publish OK

Sending payload: {"Distance":399.92}
Publish OK

Sending payload: {"Distance":399.92}
Publish OK

IBM CLOUD:



Browse

Action

Device Types

Interfaces

Add Device +



Identity

Device Information

Recent Events

State

Logs

✕

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"d":{"Name":"ultrasonic sensor","distance":7,"al...	json	a few seconds ago
event_1	{"d":{"Name":"ultrasonic sensor","distance":7,"al...	json	a few seconds ago
event_1	{"d":{"Name":"ultrasonic sensor","distance":7,"al...	json	a few seconds ago
event_1	{"d":{"Name":"ultrasonic sensor","distance":7,"al...	json	a few seconds ago
event_1	{"d":{"Name":"ultrasonic sensor","distance":7,"al...	json	a few seconds ago

Items per page 50 ▼ | 1–2 of 2 Items

1 of 1 page



1 ▼



1 Simulation running