

Assignment-4

Date	11 November 2022
Name	Santhiya J
Roll Number	620119106081
Team ID	PNT2022TMID30897
Project Name	Gas Leakage Monitoring and Alerting System for Industries

Question :

Write code and connections in wokwi for ultrasonic sensors. That 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.

Wokwi:

<https://wokwi.com/projects/348220963754607187>

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>

WiFiClient wifiClient;

#define ORG "m30vo1"
#define DEVICE_TYPE "san"
#define DEVICE_ID "san_1"
#define TOKEN "iiYP@fboag(we0r@A&"
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json"; char
topic[] = "iot-2/cmd/home/fmt/String"; char authMethod[] =
"use-token-auth"; char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient); void
publishData(); const int trigpin=5; const int echopin=18;
String command;
String data=""; long
duration; float
dist;
void
setup()
{
    Serial.begin(115200);
    pinMode(trigpin, OUTPUT);
```

```

        pinMode(echopin, INPUT);
        wifiConnect(); mqttConnect();
    } void loop() {
        publishData(); delay(500);
        if (!client.loop()) {
            mqttConnect();
        }
    }
    void wifiConnect() {
        Serial.print("Connecting to "); Serial.print("Wifi");
        WiFi.begin("Wokwi-GUEST", "", 6); while
        (WiFi.status() != WL_CONNECTED) { delay(500);
            Serial.print(".");
        }
        Serial.print("WiFi connected, IP address: ");
        Serial.println(WiFi.localIP());
    } void mqttConnect() { if
    (!client.connected()) {
        Serial.print("Reconnecting MQTT client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token))
            { Serial.print("."); delay(500);
        }
        initManagedDevice();
        Serial.println();
    } }
    void initManagedDevice() { if
        (client.subscribe(topic)) {
            Serial.println("subscribe to cmd OK");
        } else
        {
            Serial.println("subscribe to cmd FAILED");
        }
    } }
    void publishData()
    { digitalWrite(trigpin, LOW);
        digitalWrite(trigpin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigpin, LOW);
        duration=pulseIn(echopin, HIGH);
        dist=duration*speed/2;
        if(dist<100){
            String payload = "{\"Alert distance\":\"";
            payload += dist; payload += "\"}";
            Serial.print("\n");
            Serial.print("Sending payload: ");
            Serial.println(payload);
            if (client.publish(publishTopic, (char*) payload.c_str())){
                Serial.println("Publish OK");
            } else
            {
                Serial.println("Publish FAILED");
            }
        }
    }

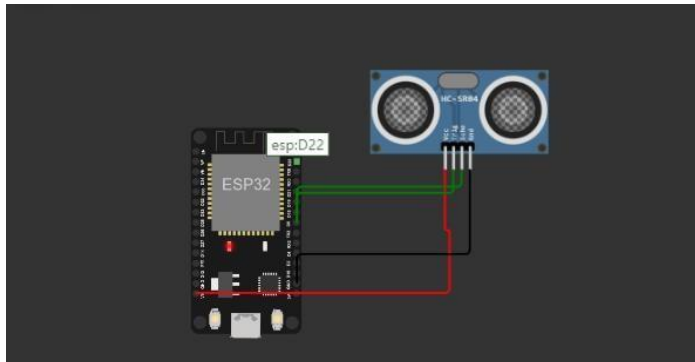
```

```

    }
  }
}

```

Diagram:



Wokwi Output:

Wokwi simulation interface showing the sketch and output.

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 WiFiClient wifiClient;
5
6 #define ORG "m30vol"
7 #define DEVICE_TYPE "san"
8 #define DEVICE_ID "san_1"
9 #define TOKEN "i1YP@fboag(we0n@A&"
10 #define speed 0.034
11
12
13 char server[] = ORG".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/event_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wifiClient);
20 void publishData();
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25 long duration;
26 float dist;
27 void setup()
28 {
29   Serial.begin(115200);
30   pinMode(trigpin, OUTPUT);
31   pinMode(echopin, INPUT);

```

Simulation output:

```

Sending payload: {"Alert distance":99.98}
Publish OK

Sending payload: {"Alert distance":99.98}
Publish OK

Sending payload: {"Alert distance":99.98}
Publish OK

Sending payload: {"Alert distance":99.98}
Publish OK

Sending payload: {"Alert distance":99.98}
Publish OK

```

Wokwi interface includes tabs for sketch.ino, diagram.json, libraries.txt, and Library Manager. The simulation window shows the ESP32 and HC-SR04 sensor connected. The output window shows the sensor readings being published to the MQTT broker.

IBM cloud output:

The screenshot shows the IBM Watson IoT Platform interface with a device simulation window open. The simulation window displays the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 WiFiClient wificlient;
5
6 #define ORG "m30vo1"
7 #define DEVICE_TYPE "san"
8 #define DEVICE_ID "san_1"
9 #define TOKEN "iiYP@fboag(we0r@A&"
10 #define speed 0.034
11
12
13 char server[] = ORG".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/event_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wificlient);
20 void publishData();
```

The simulation window also shows the following output:

```
Publish OK
Sending payload: {"Alert distance":99.98}
Publish OK
Sending payload: {"Alert distance":99.98}
Publish OK
```

The IBM Watson IoT Platform interface shows the "Browse Devices" page with a table of devices. The table has columns for "Device ID" and "Name". The device "san_1" is listed.

The screenshot shows the IBM Watson IoT Platform interface with the "Recent Events" tab selected. The table displays the following data:

Event	Value	Format	Last Received
event_1	{"Alert distance":99.98}	json	a few seconds ago
event_1	{"Alert distance":99.98}	json	a few seconds ago
event_1	{"Alert distance":99.98}	json	a few seconds ago
event_1	{"Alert distance":99.98}	json	a few seconds ago
event_1	{"Alert distance":99.99}	json	a few seconds ago

The IBM Watson IoT Platform interface also shows the "Add Device" button and the "Recent Events" tab selected.

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes the IBM Watson IoT Platform logo, a search bar, and a user profile section with the email 'soundarra2112001@gmail.com' and ID 'ironpk'. The main content area is titled 'ironpk.internetofthings.ibmcloud.com/dashboard/devices/browse'. The 'Browse' tab is selected, and the 'Recent Events' sub-tab is active. A table displays the following data:

Event	Value	Format	Last Received
event_1	{"randomNumber":4}	json	a few seconds ago
event_1	{"randomNumber":96}	json	a few seconds ago
event_1	{"randomNumber":11}	json	a few seconds ago
event_1	{"randomNumber":40}	json	a few seconds ago
event_1	{"randomNumber":52}	json	a few seconds ago

At the bottom of the dashboard, there is a status bar indicating '5 Simulation running' and a pagination control showing 'Items per page 50 | 1-1 of 1 item'.