

Assignment -4

Assignment Date	24 Oct 2022
Team ID	PNT2022TMID28579
Student Name	SREEDHAR M (Team Member)
Project Name	SmartFarmer-IoT Enabled Smart Farming Application

Question:

Write a 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

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "spmi5n"
#define DEVICE_TYPE "arduino"
#define DEVICE_ID "sreedhar_assignment"
#define TOKEN "asdfghjkl"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/sreedhar/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;
```

```

void setup()
{
    Serial.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect();
    mqttConnect();
}

void loop() {
    bool isNearby = dist < 100;
    digitalWrite(led, isNearby);

    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(client.subscribe(topic));
    }
}

```

```

        Serial.println("IBM 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");
        }
    }

    if(dist>100){
        String payload = "{\"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");
        }
    }
}
}

```

Output:

1. When distance greater than 100 cm

The screenshot shows the Wokwi IDE interface. On the left, the sketch.ino file is open, displaying the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "spmi5n"
6 #define DEVICE_TYPE "arduino"
7 #define DEVICE_ID "sreedhar_assignment"
8 #define TOKEN "asdfghjkl"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/sreedhar/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;
```

On the right, the simulation window shows an ESP32 board connected to an Ultrasonic Distance Sensor. The sensor's distance is displayed as 221cm. Below the simulation, the MQTT client's activity is shown:

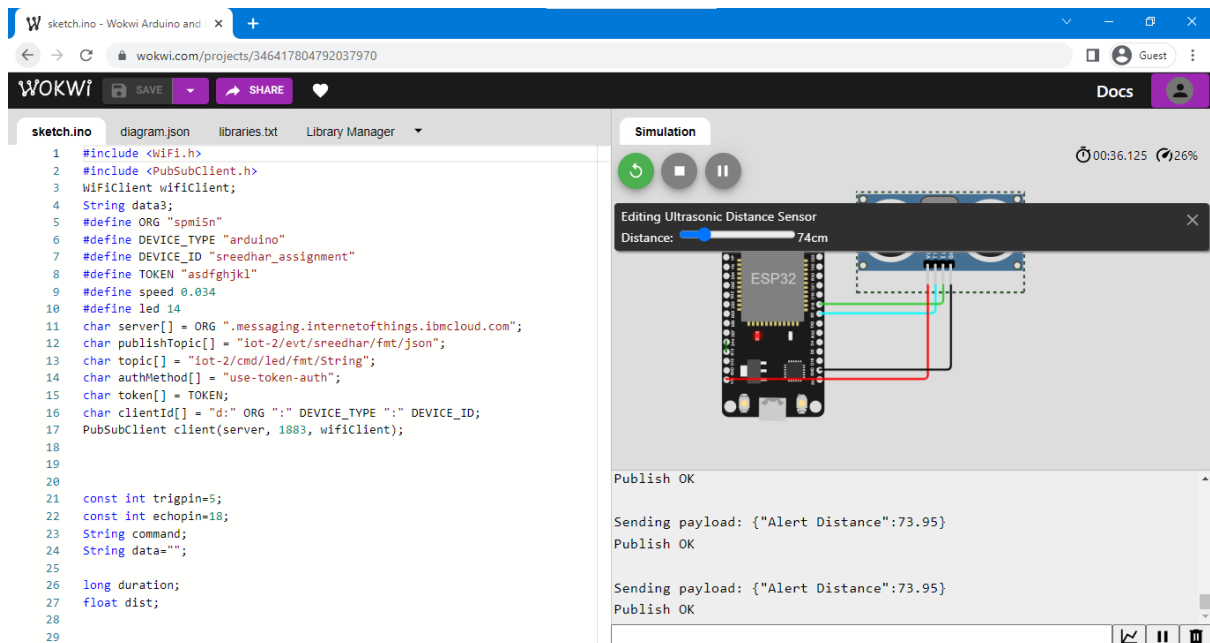
```
Publish OK
Sending payload: {"Distance":220.98}
Publish OK
Sending payload: {"Distance":220.97}
Publish OK
```

IBM RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform interface. The 'Recent Events' tab is selected, displaying a table of events for the device 'sreedhar'.

Event	Value	Format	Last Received
sreedhar	{"Distance":220.98}	json	a few seconds ago
sreedhar	{"Distance":220.98}	json	a few seconds ago
sreedhar	{"Distance":220.98}	json	a few seconds ago
sreedhar	{"Distance":220.97}	json	a few seconds ago
sreedhar	{"Distance":220.98}	json	a few seconds ago

2. When distance less than 100 cm



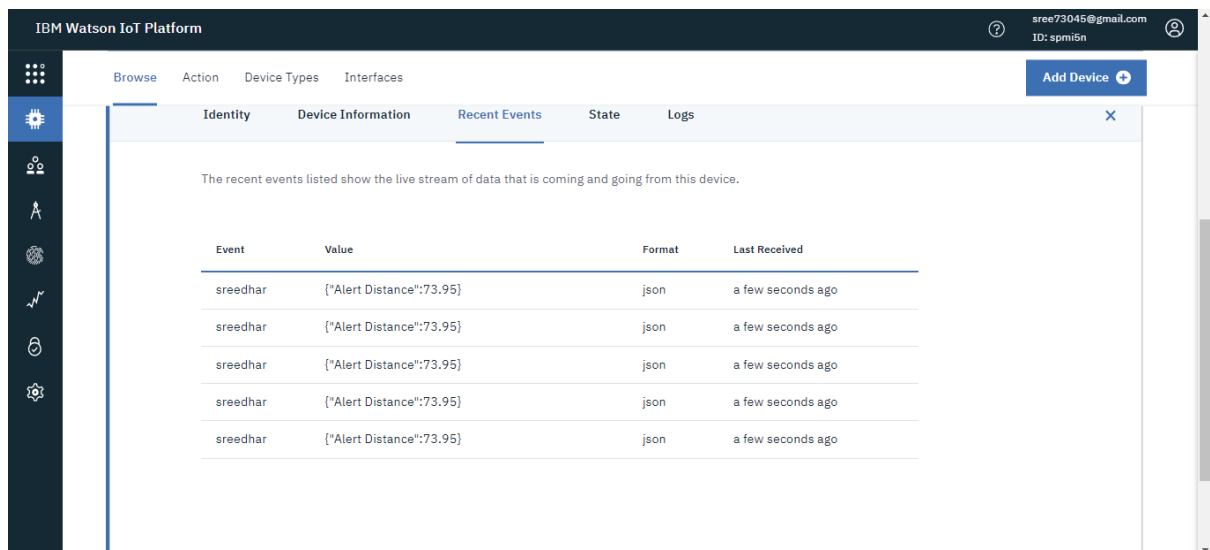
The screenshot shows the Wokwi IDE interface. On the left, the sketch editor displays an Arduino code snippet for an ESP32 connected to an ultrasonic sensor and publishing data to IBM Watson IoT. The code includes headers for WiFi and PubSubClient, defines constants for the device type, ID, token, and speed, and sets up the server, topic, and authentication. It also defines pins for a trigger and echo, and sets up a long duration and a float for distance.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "spm15n"
6 #define DEVICE_TYPE "arduino"
7 #define DEVICE_ID "sreedhar_assignment"
8 #define TOKEN "asdfghjkl"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/sreedhar/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;
```

On the right, the simulation window shows a 3D model of the ESP32 board with an ultrasonic sensor connected. A dialog box titled "Editing Ultrasonic Distance Sensor" shows a distance of 74cm. Below the simulation, the console output shows the following messages:

```
Publish OK
Sending payload: {"Alert Distance":73.95}
Publish OK
Sending payload: {"Alert Distance":73.95}
Publish OK
```

IBM RECENT EVENTS:



The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes "Browse", "Action", "Device Types", and "Interfaces". The left sidebar contains icons for various functions. The main content area is titled "Recent Events" and displays a table of events for a device named "sreedhar".

Event	Value	Format	Last Received
sreedhar	{"Alert Distance":73.95}	json	a few seconds ago
sreedhar	{"Alert Distance":73.95}	json	a few seconds ago
sreedhar	{"Alert Distance":73.95}	json	a few seconds ago
sreedhar	{"Alert Distance":73.95}	json	a few seconds ago
sreedhar	{"Alert Distance":73.95}	json	a few seconds ago

WOWKI LINK :- <https://wokwi.com/projects/346417804792037970>