

Assignment 4

Name	Dharani R
Team ID	PNT2022TMID38327
Project Name	IOT Based smart Crop production system for Agriculture

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

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "3i42s3"
#define DEVICE_TYPE "Dharani"
#define DEVICE_ID "Assignment-4"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/dharani/fmt/json";
char topic[] = "iot-2/cmd/status/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=19;
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 less than 100 cm

The screenshot displays the Wokwi IoT simulation interface. On the left, the code for the ESP32 is shown, which includes the necessary libraries and defines the device type, ID, and token. It also sets up the HC-SR04 sensor and configures the MQTT client to publish distance data to a specific topic on IBM Cloud IoT. The simulation output on the right shows the sensor detecting a distance of 86.96 cm, which is then published as a JSON payload to the MQTT topic.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "3i42s3"
6 #define DEVICE_TYPE "Dharani"
7 #define DEVICE_ID "Assignment_4"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12
13 char publishTopic[] = "iot-2/evt/dharani/fmt/json";
14 char topic[] = "iot-2/cmd/status/fmt/String";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 PubSubClient client(server, 1883, wificlient);
19
20 const int trigpin=19;
21 const int echopin=18;
22 String command;
23 String data="";
24 long duration;
25 float dist;
26
27 void setup()
28 {
29   Serial.begin(115200);
30   pinMode(led, OUTPUT);
```

Simulation Output:

```
Publish OK
Sending payload: {"Alert Distance":86.96}
Publish OK
Sending payload: {"Alert Distance":86.96}
Publish OK
```

IBM RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays details for a device named 'Assignment_4', which is 'Connected' and has a status of 'Dharani'. The 'Recent Events' tab is selected, showing a table of events.

Event	Value	Format	Last Received
dharani	{"Alert Distance":86.96}	json	a few seconds ago
dharani	{"Alert Distance":86.99}	json	a few seconds ago
dharani	{"Alert Distance":87.01}	json	a few seconds ago
dharani	{"Alert Distance":86.96}	json	a few seconds ago
dharani	{"Alert Distance":86.96}	json	a few seconds ago

2. When distance greater than 100 cm

The screenshot shows the Wokwi IDE interface. On the left, the 'esp32-dht22.ino' file is open, displaying C++ code for an ESP32 microcontroller. The code includes libraries for WiFi and MQTT, defines device information, and sets up an Ultrasonic Distance Sensor. The right side of the interface shows a 3D simulation of the ESP32 board with the sensor attached. A 'Simulation' window is open, showing the sensor's distance reading as 400cm. Below the simulation, a console window displays the MQTT publish logs.

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "3i42s3"
6 #define DEVICE_TYPE "Dharani"
7 #define DEVICE_ID "Assignment_4"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12
13 char publishTopic[] = "iot-2/evt/dharani/fmt/json";
14 char topic[] = "iot-2/cmd/status/fmt/String";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 PubSubClient client(server, 1883, wifiClient);
19
20 const int trigpin=19;
21 const int echopin=18;
22 String command;
23 String data="";
24 long duration;
25 float dist;
26
27 void setup()
28 {
29   Serial.begin(115200);
30   pinMode(led, OUTPUT);
  
```

Simulation window: Editing Ultrasonic Distance Sensor, Distance: 400cm

Console output:

```

Publish OK
Sending payload: {"Distance":399.98}
Publish OK
Sending payload: {"Distance":399.96}
Publish OK
  
```

IBM RECENT EVENTS:

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various IoT functions. The main content area shows details for a device named 'Assignment_4', which is 'Connected'. The 'Recent Events' tab is selected, displaying a table of live data streams.

Device: Assignment_4, Status: Connected, Name: Dharani, Last Seen: Nov 23, 2022 11:40 AM

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
dharani	{"Distance":399.96}	json	a few seconds ago
dharani	{"Distance":399.96}	json	a few seconds ago
dharani	{"Distance":399.94}	json	a few seconds ago
dharani	{"Distance":399.96}	json	a few seconds ago
dharani	{"Distance":399.98}	json	a few seconds ago

The bottom of the image shows a Windows taskbar with the search bar, task icons, and system tray information including the time 11:44 and date 23-11-2022.