

ASSIGNMENT 4

Date	October 2022
Team ID	PNT2022TMID38325
Project Name	Smart Solution for Railways

QUESTION :

Write code and connection in wowki 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> //library for wifi
#include <PubSubClient.h> //library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "2icro4"
#define DEVICE_TYPE "Subathra"
#define DEVICE_ID "Subathra_Assignment-4"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/Subathra/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");
}
}

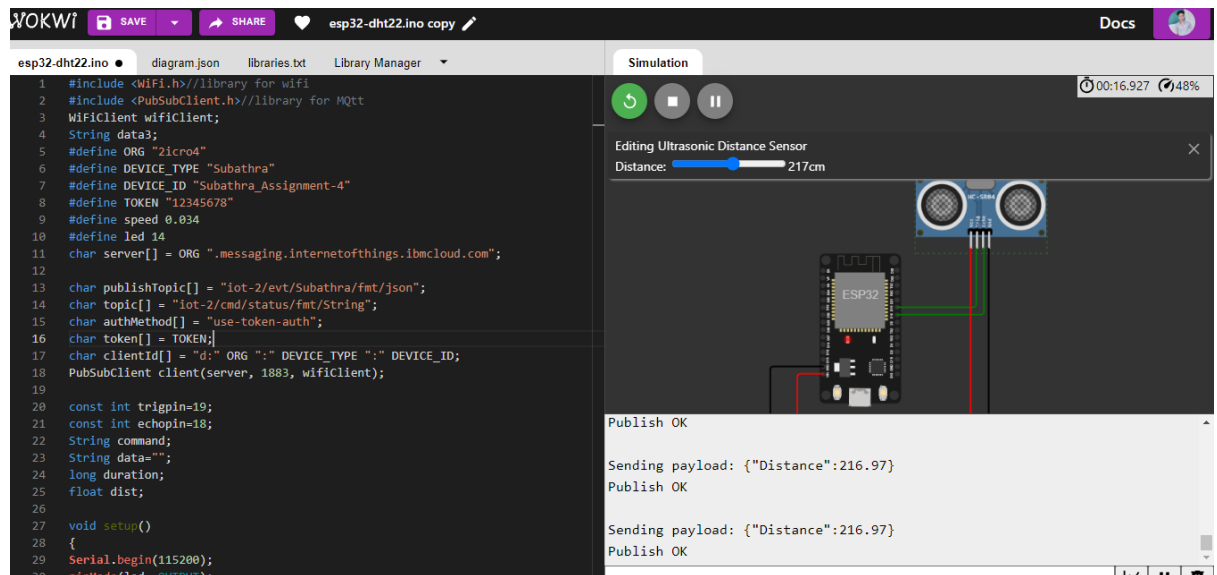
void publishData()
{
digitalWrite(trigpin, LOW);
digitalWrite(trigpin, HIGH);
delayMicroseconds(10);
digitalWrite(trigpin, LOW);
duration=pulseIn(echopin, HIGH);
dist=duration*speed/2;
if(dist<100)
{
String payload = "{\"Alert Distance\":\"";
payload += dist;

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 greater than 100 cm



The screenshot shows the Wokwi IDE interface. On the left, the Arduino code for an ESP32 is displayed. The code includes libraries for WiFi and MQTT, defines device information, and sets up an MQTT client. It also includes logic for an ultrasonic sensor and an LED. On the right, the simulation window shows a 3D model of the ESP32 and an ultrasonic sensor. The sensor's distance is set to 217cm. Below the simulation, the output console shows the MQTT client sending a payload: {"Distance":216.97}.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data;
5 #define ORG "2icro4"
6 #define DEVICE_TYPE "Subathra"
7 #define DEVICE_ID "Subathra_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/Subathra/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: Editing Ultrasonic Distance Sensor, Distance: 217cm

Publish OK

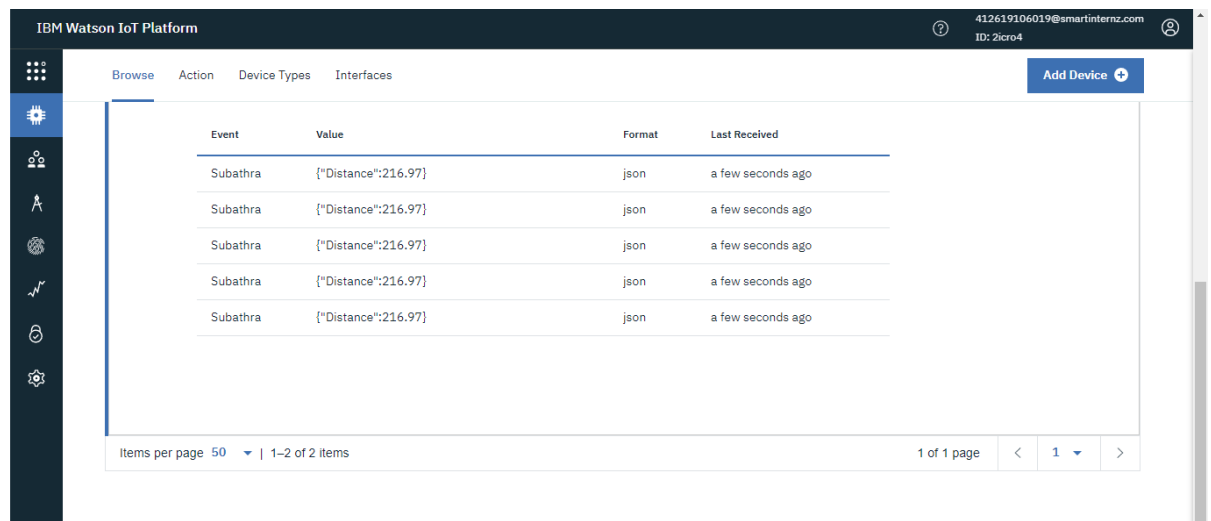
Sending payload: {"Distance":216.97}

Publish OK

Sending payload: {"Distance":216.97}

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 main content area displays a table of recent events. The table has four columns: Event, Value, Format, and Last Received. The events are all from the device 'Subathra' and have a value of '{"Distance":216.97}' in json format, received 'a few seconds ago'.

Event	Value	Format	Last Received
Subathra	{"Distance":216.97}	json	a few seconds ago
Subathra	{"Distance":216.97}	json	a few seconds ago
Subathra	{"Distance":216.97}	json	a few seconds ago
Subathra	{"Distance":216.97}	json	a few seconds ago
Subathra	{"Distance":216.97}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

1 of 1 page

2) When distance less than 100

The WOKWI IDE interface displays the following code for `esp32-dht22.ino`:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "2icro4"
6 #define DEVICE_TYPE "Subathra"
7 #define DEVICE_ID "Subathra_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/Subathra/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=10;
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);
```

The simulation console shows the following output:

```
Publish OK
Sending payload: {"Alert Distance":93.96}
Publish OK
Sending payload: {"Alert Distance":93.96}
Publish OK
```

IBM RECENT EVENTS

The IBM Watson IoT Platform interface shows the following table of recent events:

Event	Value	Format	Last Received
Subathra	{"Alert Distance":93.96}	json	a few seconds ago
Subathra	{"Alert Distance":93.96}	json	a few seconds ago
Subathra	{"Alert Distance":93.96}	json	a few seconds ago
Subathra	{"Alert Distance":93.96}	json	a few seconds ago
Subathra	{"Alert Distance":93.96}	json	a few seconds ago

Items per page: 50 | 1-2 of 2 items | 1 of 1 page