

Assignment -4

Wowki & IBM Cloud

Assignment Date	27 October 2022
Student Name	Angleena Reji
Student Roll Number	310819106006
Maximum Marks	2 Marks

Question:

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

Code:

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

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

```
#include <ArduinoJson.h>
```

```
WiFiClient wifiClient;
```

```
#define ORG "zy1xr9"
```

```
#define DEVICE_TYPE "ChildTestDeviceType"
```

```
#define DEVICE_ID "12345"
```

```
#define TOKEN "vk)OK)nv0wO&S0XJ_1"
```

```
#define speed 0.034
```

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/abcd_1/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="";

String lat="14.167589";

String lon="80.248510";

String name="point2";

String icon="";

long duration;

int 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(1000);

        }

        initManagedDevice();

        Serial.println();

    }
}

void initManagedDevice() {
    if (client.subscribe(topic)) {

        Serial.println(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){  
        dist=100-dist;  
        icon="fa-trash";  
    }else{  
        dist=0;  
        icon="fa-trash-o";  
    }  
  
    DynamicJsonDocument doc(1024);  
    String payload;  
    doc["Name"]=name;  
    doc["Latitude"]=lat;  
    doc["Longitude"]=lon;
```

```

doc["Icon"]=icon;

doc["FillPercent"]=dist;

serializeJson(doc, payload);

delay(3000);

Serial.print("\n");

Serial.print("Sending payload: ");

Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {

    Serial.println("Publish OK");

} else {

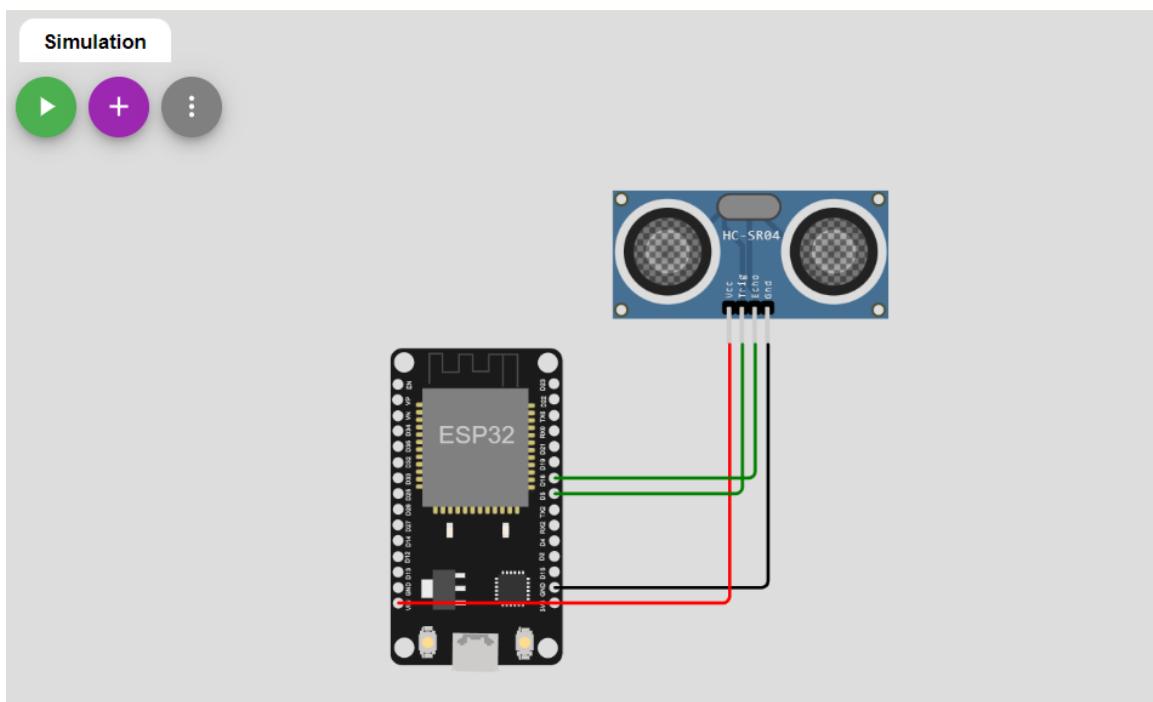
    Serial.println("Publish FAILED");

}

}

```

CONNECTIONS:



OUTPUT:

Wowki output:

The screenshot shows the Wokwi IDE interface. On the left, the Arduino sketch for an ESP32 is displayed, featuring code for connecting to an IBM Cloud IoT instance and publishing data. On the right, a simulation window shows an ESP32 board connected to an ultrasonic sensor. The sensor's distance is set to 82cm. Below the simulation, the console output shows the device publishing data to the IBM Cloud IoT platform.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 WiFiClient wificlient;
6
7 #define ORG "zy1xr9"
8 #define DEVICE_TYPE "ChildTestDeviceType"
9 #define DEVICE_ID "12345"
10 #define TOKEN "(vk)OK)nv0w0&50X)_1"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/abcd_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
22 const int trigpin=5;
23 const int echopin=18;
24 String command;
25 String data="";
26 String lat="14.167589";
27 String lon="80.248510";
28 String name="point2";
29 String icon="";
30
31 long duration;
32 int dist;
33
```

Simulation window: Editing Ultrasonic Distance Sensor. Distance: 82cm.

Console output:

```
trash", "FillPercent": 28}
Publish OK

Sending payload:
{"Name": "point2", "Latitude": "14.167589", "Longitude": "80.248510", "Icon": "fa-trash", "FillPercent": 19}
Publish OK
```

IBM Cloud Output:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes links to Browse, Action, Device Types, and Interfaces. The main content area displays a list of devices, with the selected device (ID: 12345) shown in detail. The device is connected and has a status of 'ChildTestDeviceType'. The 'Recent Events' tab is active, showing a list of events with columns for Event, Value, Format, and Last Received. The events are JSON payloads containing location data.

Event	Value	Format	Last Received
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago

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

1 of 1 page

1 Simulation running

LINK: <https://wokwi.com/projects/346870679866442322>