

Assignment-4

Assignment Date	26 October 2022
Student Name	Mr. G. Suriyakrishna
Student Roll Number	911019104026
Maximum Marks	2 Marks

QUESTION-1:

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

PROGRAM:

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

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

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

```
WiFiClient wifiClient;
```

```
#define ORG "kr9fjo"
```

```
#define DEVICE_TYPE "TestDeviceType"
```

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

```
#define TOKEN "VJsSC148dk1dCN3UqS"
```

```
#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");  
  
}  
  
}
```

OUTPUT:

The screenshot displays the Wokwi IDE interface. On the left, the 'sketch.ino' file is open, showing an Arduino sketch for an ESP32. The sketch includes headers for WiFi, PubSubClient, and ArduinoJson. It defines an MQTT client with a specific server, topic, and token. The main logic involves connecting to the MQTT server and publishing data. On the right, the 'Simulation' tab shows a visual representation of the ESP32 board connected to an HCSR04 ultrasonic sensor. The sensor's VCC pin is connected to the ESP32's 5V pin, and its GND pin is connected to the ESP32's GND pin. The simulation controls (play, stop, and reset buttons) are visible above the sensor.

```
1 #include <WiFi.h>  
2 #include <PubSubClient.h>  
3 #include <ArduinoJson.h>  
4  
5 WiFiClient wifiClient;  
6  
7 #define ORG "kr9fjo"  
8 #define DEVICE_TYPE "TestDeviceType"  
9 #define DEVICE_ID "12345"  
10 #define TOKEN "VjSSC148dk1dCNBUqS"  
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
```

IBM Watson IoT Platform

venkateshg01@gmail.com
ID: cllvfe

Browse Action Device Types Interfaces

Search by Device ID

Device Simulator

Add Device

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
12345	Disconnected	Raspberrypi	Device	Oct 31, 2022 3:19 PM		venkateshg01@gmail.com

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
Distance	{"distance":74}	json	a few seconds ago
Distance	{"distance":89}	json	a few seconds ago
Distance	{"distance":12}	json	a few seconds ago
Distance	{"distance":52}	json	a few seconds ago
Distance	{"distance":45}	json	a few seconds ago

1 Simulation running

Activate Windows
Go to Settings to activate Windows.

Simulation

Publish ok

Measured distance: 18.94

Sending payload: {"ALERT_MESSAGE":1,"DISTANCE":18.94}

Publish ok

Measured distance: 18.94

Sending payload: {"ALERT_MESSAGE":1,"DISTANCE":18.94}

Publish ok

Graph icon | Pause icon | Stop icon