

Assignment -4

Assignment Date	12 october 2022
Student Name	Srinivasan D
Student Roll Number	513419106040
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

Solution:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "mqy1m"
#define DEVICE_TYPE "esp"
#define DEVICE_ID "esp-32"
#define TOKEN "_wobQnsPnCX!120Zv6"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/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=4;
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 = "{\"Normal 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>101 && dist<111){
        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("Warning crosses 110cm -- it automatically of the loop");
            digitalWrite(led, HIGH);
        }else {
            Serial.println("Publish FAILED");
        }
    }
}

}

void callback(char* subscribeTopic, byte* payload, unsigned int
payloadLength){
    Serial.print("callback invoked for topic:");
    Serial.println(subscribeTopic);
    for(int i=0; i<payloadLength; i++){

```

```

    dist += (char)payload[i];
}
Serial.println("data:" + data3);
if(data3=="lighton"){
    Serial.println(data3);
    digitalWrite(led,HIGH);
}
data3="";
}

```

Output:

The screenshot displays the Wokwi IDE interface. On the left, the 'sketch.ino' file contains the following code:

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "mqyu1m"
6 #define DEVICE_TYPE "esp"
7 #define DEVICE_ID "esp-32"
8 #define TOKEN "...wobQnsPncXl120zv6"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Data/fmt/json";
13 char topic[] = "iot-2/cmd/home/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 void publishData();
19
20
21 const int trigpin=5;
22 const int echopin=4;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30

```

The right side of the IDE shows a 'Simulation' window. It features a visual representation of an LDR sensor module and an ESP8266 microcontroller. Below the visual, a log window displays the following messages:

```

Reconnecting MQTT client to mqyu1m.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

Sending payload: {"Normal Distance":37.98}
Publish OK

Sending payload: {"Normal Distance":37.98}
Publish OK

```

Service Details - IBM Watson IoT Platform | IBM Watson IoT Platform | wokwi - Yahoo India | sketch.ino - Wokwi | Fwd: ibm us code - |

mquylm.internetofthings.ibmcloud.com/dashboard/devices/browse

Gmail | YouTube | Maps | Inbox (52) - abirami... | 100 JavaScript Proj... | (52) Todo List using... | New Tab

IBM Watson IoT Platform

srinivasandehibabu@gmail.com
ID: mquylm

Browse | Action | Device Types | Interfaces

Add Device

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Normal Distance":37.98}	json	a few seconds ago
Data	{"Normal Distance":37.89}	json	a few seconds ago
Data	{"Normal Distance":37.98}	json	a few seconds ago
Data	{"Normal Distance":37.98}	json	a few seconds ago
Data	{"Normal Distance":37.98}	json	a few seconds ago

Items per page 50 | 1-3 of 3 items

1 of 1 page < 1 >

Wokwi link:

<https://wokwi.com/projects/347742426566230610>