

Assignment -4

Assignment Date	21 OCTOBER 2022
Student Name	VINOTHKUMAR B
Student Roll Number	513419106045
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.

solution

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "x390n3"
#define DEVICE_TYPE "Vk"
#define DEVICE_ID "94424"
#define TOKEN "VkRlPIRUAVV-2kBa03"
#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=18;
const int echopin=19;
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="";
}

```

The screenshot shows 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 "x390n3"
6 #define DEVICE_TYPE "Vk"
7 #define DEVICE_ID "94424"
8 #define TOKEN "VkrIPIRUAVV-2k8a03"
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=18;
22 const int echopin=19;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);

```

On the right, the simulation window shows an HC-SR04 sensor connected to an ESP32. The console output shows the following messages:

```

Publish OK
Sending payload: {"Normal Distance":44.97}
Publish OK
Sending payload: {"Normal Distance":44.97}
Publish OK

```

The screenshot shows the IBM Watson IoT Platform dashboard. The device with ID 94424 is shown as 'Connected'. The 'Recent Events' tab displays the following data:

Event	Value	Format	Last Received
Data	{"Normal Distance":44.97}	json	a few seconds ago
Data	{"Normal Distance":44.97}	json	a few seconds ago
Data	{"Normal Distance":44.97}	json	a few seconds ago
Data	{"Normal Distance":45}	json	a few seconds ago
Data	{"Normal Distance":44.97}	json	a few seconds ago