

Assignment -4

Assignment Date	10 Oct 2022
Student Name	A. GOWTHAM
Student Register Number	422719106004
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>
WiFiClient wifiClient;
String data3;
#define ORG "ncj2k2"
#define DEVICE_TYPE "Ultrasonic_sensor"
#define DEVICE_ID "987654321"
#define TOKEN "bIqHU?Ocx!pYiRSYh+"
#define speed 0.034
#define led 12

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=4;
const int echopin=2;
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>101 && dist<400){
        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("Warning crosses 110cm -- it automaticaly 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:

Device Event Information

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area displays the 'Recent Events' tab for a device with ID 987654321, which is an 'Ultrasonic_sensor' and is currently 'Disconnected'. The device was last seen on Oct 31, 2022 at 2:05 PM. Below this, a table lists recent events:

Event	Value	Format	Last Received
Data	{"Alert Distance":64.96}	json	a few seconds ago
Data	{"Alert Distance":64.96}	json	a few seconds ago
Data	{"Alert Distance":74.99}	json	a few seconds ago
Data	{"Alert Distance":43.94}	json	a few seconds ago
Data	{"Alert Distance":33.97}	json	a few seconds ago

Below the event table, a list of devices is shown, including 'demo_123_1', 'demo_123_2', and 'demo_123_3'. A status bar at the bottom indicates '4 Simulations running'.

Wokwi project info

The screenshot shows the Wokwi project interface. The left sidebar contains a 'sketch.ino' file. The main area displays the code for the sketch, which is a C++ program for an Arduino Uno. The code uses the 'Ultrasonic' library to measure distance and publish the data to a topic. The right sidebar shows a 'Simulation' window with a visual representation of the circuit, including an Arduino Uno and an Ultrasonic sensor. The simulation is running, and the distance is shown as 61cm. The bottom status bar indicates '09:42 AM 11/1/2022'.

```
84  
85 {  
86   digitalWrite(trigpin,LOW);  
87   digitalWrite(trigpin,HIGH);  
88   delayMicroseconds(10);  
89   digitalWrite(trigpin,LOW);  
90   duration=pulseIn(echopin,HIGH);  
91   dist=duration*speed/2;  
92   if(dist<100){  
93     String payload = "{\"Alert Distance\"";  
94     payload += dist;  
95     payload += "}";  
96     Serial.print("\n");  
97     Serial.print("Sending payload: ");  
98     Serial.println(payload);  
99     if (client.publish(publishTopic, (char*) payload.c_str()))  
100   {  
101     Serial.println("Publish OK");  
102   }  
103 }  
104 }  
105 }  
106 if(dist>101 && dist<400){  
107   String payload = "{\"normal distance\"";  
108   payload += dist;  
109   payload += "}";  
110   Serial.print("\n");  
111   Serial.print("Sending payload: ");  
112   Serial.println(payload);  
113   if (client.publish(publishTopic, (char*) payload.c_str()))  
114   {  
115     Serial.println("Publish OK");  
116   }  
117 }
```