

Assignment-4

Assignment Date	25 October 2022
Student Name	Mathivaanan A
Student Roll Number	731719106008
Maximum Marks	2 Marks

Question-4:

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 "948e13"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "502645"
#define TOKEN "F1eqrGOoNLATFjk?Sz"
#define speed 0.034

#define led 14

char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/shreedharen/fmt/json"; char
topic[] = "iot-2/cmd/led/fmt/String"; char authMethod[] =
"use-token-auth"; char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

const int trigpin=5;
```

```
const int echopin=18;
```

```
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,LO
W);
  digitalWrite(trigpin,HIG
H);
  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>100){
    String payload = "{"Distance\":";
payload += dist; payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload:
");
Serial.println(payload);
        if(client.publish(publishT
opic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
    }else {
        Serial.println("Publish FAILED");
    }

}

}

```

Connection:

The screenshot displays the Wokwi web-based IDE for an ESP32. The left pane shows the source code for `esp32-dht22.ino`, which includes the `WiFi` and `PubSubClient` libraries. The code defines an MQTT client for the `3yngbh` organization, connecting to `ibmcloud.com`. It sets up a pin for an LED and includes a `setup()` function. The right pane shows a simulation of the ESP32 board connected to an `HC-SR04` ultrasonic sensor. The simulation log indicates a successful WiFi connection to `10.10.0.2` and the reconnection of the MQTT client to `3yngbh.messaging.internetofthings.ibmcloud.com`.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "3yngbh"
6 #define DEVICE_TYPE "Assignment"
7 #define DEVICE_ID "1234"
8 #define TOKEN "234567890"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/shreedharen/fmt/json";
13 char topic[] = "iot-2/cmd/led/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
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
```

Simulation Log:

```
Connecting to Wifi.....WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
3yngbh.messaging.internetofthings.ibmcloud.com
.....
```

Output:

Output: (IBM Cloud)

The screenshot displays the IBM Watson IoT Platform dashboard in a Mozilla Firefox browser. The dashboard shows a list of devices under the 'Browse' tab. Two devices are listed: one with ID 502645 (Disconnected) and another with ID NodeMCU_1 (Connected). The 'NodeMCU_1' device is selected, and its details are shown in a modal window. The modal window has tabs for Identity, Device Information, Recent Events, State, and Logs. The 'Recent Events' tab is active, showing a table of raw data. The table has columns for Property, Value, Type, Event, and Last Received. Three events are listed: randomNumber (88), temp (26), and hum (96), all of type 'Number' and event 'event_1', received 'a few seconds ago'. At the bottom of the modal, it says 'Showing Raw Data | No Interfaces Available'. The main dashboard also shows 'Items per page 50' and '1-2 of 2 items'. A status bar at the bottom indicates '1 Simulation running'.

Device ID	Status	Device Type	Class ID	Date Added
502645	Disconnected	NodeMCU	Device	Nov 8, 2022 4:12 PM
NodeMCU_1	Connected	NodeMCU	Device	Nov 10, 2022 9:54 AM

Property	Value	Type	Event	Last Received
randomNumber	88	Number	event_1	a few seconds ago
temp	26	Number	event_1	a few seconds ago
hum	96	Number	event_1	a few seconds ago

Link: <https://wokwi.com/projects/347932864457736788>