

Assignment - 4

Assignment Date	17 October 2022
Student Name	M KARTHIKA
Student Roll Number	49621911074
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

CODE 1 :

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "46ym7"//IBM ORGANITION ID
#define DEVICE_TYPE "IOT_GAS_LEAKAGE"//Device type mentioned in ibm
watson IOT Platform#define DEVICE_ID "22112001"//Device ID mentioned in ibm
watson IOT Platform #define TOKEN "1911074abcdefgh" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  wificonnect();
  mqttconnect();
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * SOUND_SPEED/2;
```

```

Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(float dist) {
mqttconnect();
String payload = "{\"Distance\": ";
payload += dist;
payload += ", \"ALERT!!\": \"\"Distance less than 100cms\"\"";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
} else {
Serial.println("Publish failed");
}
}
void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
}
}

```

```
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd 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++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}
```

Wokwi Link :

<https://wokwi.com/projects/347021585567187540>

Output and Simulation :

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

```
1 //-----credentials of IBM Accounts-----
2 #define ORG "46ym7"//IBM ORGANITION ID
3 #define DEVICE_TYPE "IOT_GAS_LEAKAGE"//Device type mentioned in ibm watson IOT Platform
4 #define DEVICE_ID "22112001"//Device ID mentioned in ibm watson IOT Platform
5 #define TOKEN "1911074abcdegh" //Token
6 String data3;
7 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
8 char publishTopic[] = "iot-2/evt/Data/fmt/json";
9 char subscribetopic[] = "iot-2/cmd/test/fmt/String";
10 char authMethod[] = "use-token-auth";
11 char token[] = TOKEN;
12 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
13 WiFiClient wifiClient;
14 PubSubClient client(server, 1883, callback, wifiClient);
15 const int trigPin = 5;
16 const int echoPin = 18;
17 #define SOUND_SPEED 0.034
18 long duration;
19 float distance;
20 void setup() {
21   Serial.begin(115200);
22   pinMode(trigPin, OUTPUT);
23   pinMode(echoPin, INPUT);
24   wifiConnect();
25   mqttConnect();
26 }
27 void loop()
28 {
29   digitalWrite(trigPin, LOW);
30   delayMicroseconds(2);
31   // ... (rest of the code is partially obscured)
```

The simulation window on the right shows an Ultrasonic Distance Sensor connected to an ESP32. The distance is 87cm. The console output shows the following messages:

```
Publish ok
Distance (cm): 86.96
ALERT!!
Sending payload: {"Distance":86.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 86.96
ALERT!!
```

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

The screenshot displays the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, showing a table of events. The table has the following columns: Event, Value, Format, and Last Received.

Event	Value	Format	Last Received
Data	{"Distance":86.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":86.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":86.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":86.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":86.96,"ALERT!!":"Distance less than ...	json	a few seconds ago

The dashboard also shows the user profile '1911074@nec.edu.in' and the device ID '46ym7'.