

Assignment - 4

| | |
|---------------------|-----------------|
| Assignment Date | 17 October 2022 |
| Student Name | NINSHIYA MARY J |
| Student Roll Number | 49621911092 |
| 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 "12qa0h"//IBM ORGANITION ID
#define DEVICE_TYPE "IOT_GAS_LEAKAGE"//Device type mentioned in ibm
watson IOT Platform#define DEVICE_ID "16102001"//Device ID mentioned in ibm
watson IOT Platform #define TOKEN "1911092abcdefgh" //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 :



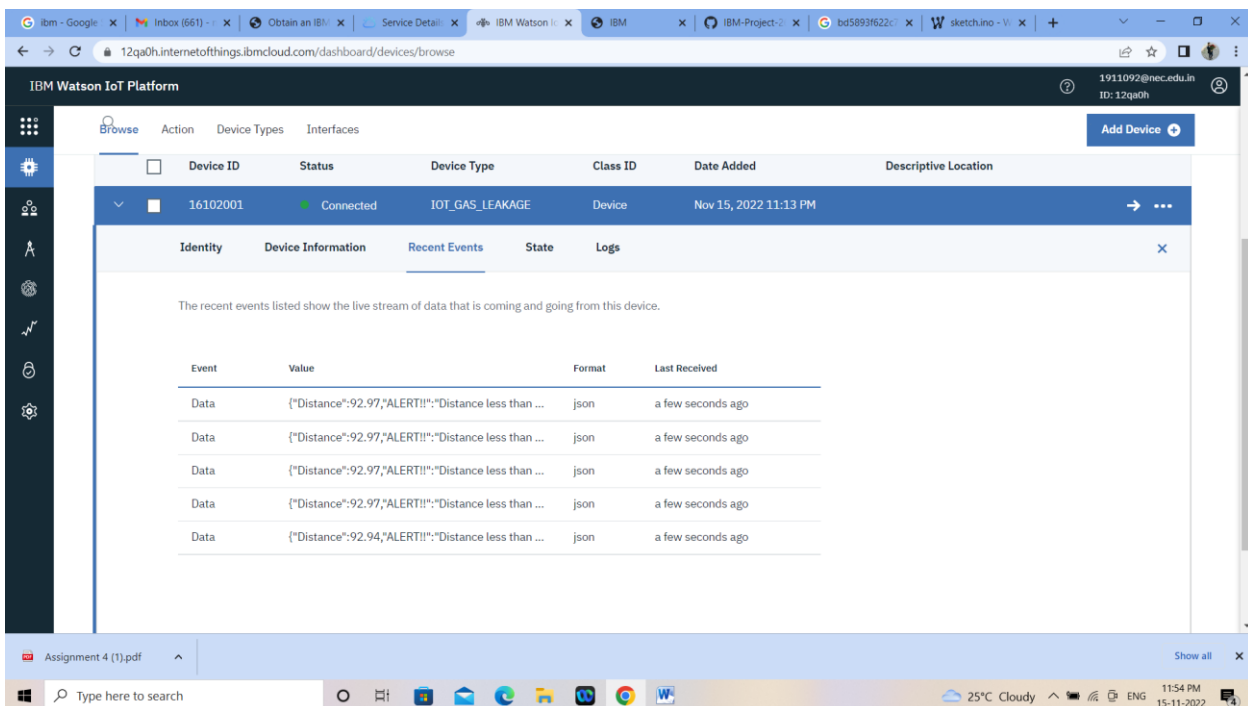
The screenshot shows the Wokwi web interface with a sketch.ino file loaded. The code is a C++ program for an Arduino Uno that connects to the IBM Watson IoT Platform via MQTT. It defines a device type 'IOT_GAS_LEAKAGE', a device ID '16102001', and a token '1911092abcdegh'. The program sets up a WiFi client and an MQTT client, then enters a loop that sends a payload containing distance and an alert message every 10 seconds.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int
4 payloadLength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "12qa0h"//IBM ORGANIZATION ID
7 #define DEVICE_TYPE "IOT_GAS_LEAKAGE"//Device type mentioned in ibm watson IOT Platform
8 #define DEVICE_ID "16102001"//Device ID mentioned in ibm watson IOT Platform
9 #define TOKEN "1911092abcdegh" //Token
10 String data;
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Data/fmt/json";
13 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 WiFiClient wificlient;
18 PubSubClient client(server, 1883, callback, wificlient);
19 const int trigPin = 5;
20 const int echoPin = 18;
21 #define SOUND_SPEED 0.034
22 long duration;
23 float distance;
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wificlient.connect();
29   mqtt.connect();
30 }
31 void loop()
32 {
33   digitalWrite(trigPin, LOW);
34   delayMicroseconds(2);
35   digitalWrite(trigPin, HIGH);
36   delayMicroseconds(10);
37   digitalWrite(trigPin, LOW);
38   duration = pulseIn(echoPin, HIGH);
39   distance = (duration/2) / SOUND_SPEED;
40   if (distance < 100) {
41     String payload = "{\"Distance\":\"" + String(distance) + "\",\"ALERT!!\":\"Distance less than 100cms\"}";
42     client.publish(publishTopic, payload);
43     Serial.println("Distance (cm): " + String(distance));
44     Serial.println("ALERT!!");
45     String payload2 = "{\"Distance\":\"" + String(distance) + "\",\"ALERT!!\":\"Distance less than 100cms\"}";
46     client.publish(subscribeTopic, payload2);
47     Serial.println("Publish ok");
48   }
49   delay(10000);
50 }
```

The simulation output shows the following sequence of events:

```
ALERT!!
Sending payload: {"Distance":92.97,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 92.97
ALERT!!
Sending payload: {"Distance":92.97,"ALERT!!":"Distance less than 100cms"}
Publish ok
```

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



The screenshot shows the IBM Watson IoT Platform dashboard. The device 16102001 is listed as 'Connected' with a status of 'IOT_GAS_LEAKAGE'. The 'Recent Events' tab is selected, showing a list of events with the following details:

| Event | Value | Format | Last Received |
|-------|---|--------|-------------------|
| Data | {"Distance":92.97,"ALERT!!":"Distance less than ... | json | a few seconds ago |
| Data | {"Distance":92.97,"ALERT!!":"Distance less than ... | json | a few seconds ago |
| Data | {"Distance":92.97,"ALERT!!":"Distance less than ... | json | a few seconds ago |
| Data | {"Distance":92.97,"ALERT!!":"Distance less than ... | json | a few seconds ago |
| Data | {"Distance":92.94,"ALERT!!":"Distance less than ... | json | a few seconds ago |