

Assignment -4

Assignment Date	20 OCTOBER 2022
Student Name	HARIHARAN A
Student Roll Number	737819ECR050
Maximum Marks	2 Marks

Question-1:

Write code and connections in wowki for ultrasonic sensor.

Whenever distance is less than 100 cms send “alert” to IBM cloud and display in device recent events.

Solution:

WOWKI LINK:

<https://wokwi.com/projects/347396071297647187>

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#define TRIGGER 2
```

```
#define ECHO 15
```

```
#define sound 0.034
```

```
int distance;
```

```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

```
//-----credentials of IBM Accounts-----
```

```
#define ORG "2jk8e5"
```

```
#define DEVICE_TYPE "Gas_leakage_Detection_Device"
```

```
#define DEVICE_ID "Gas_Detector_1"
```

```
#define TOKEN "123456789"
```

```
String data3;
```

```
//----- Customise the above values -----
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
```

```
char publishTopic[] = "iot-2/evt/Data/fmt/json";
```

```
char subscribetopic[] = "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);
void setup()
{
  Serial.begin(115200);
  pinMode(TRIGGER, OUTPUT);
  pinMode(ECHO, INPUT);
  delay(10);
  Serial.println();
  Serial.println("Reconnecting client to wp72r7.messaging.internetofthings.ibmcloud.com\niot-
2/cmd/test/fmt/String\nsubscribe to cmd OK\n\n399 cms.\n399 cms.\n243 cms.\n151 cms.\n35 cms.\nSending
payload: {\"message\": \"alert\"}\nPublish ok\n41 cms.\nSending payload: {\"message\": \"alert\"}\nPublish
ok\n41 cms.\nSending payload: {\"message\": \"alert\"}\nPublish ok\n41 cms.\nSending payload:
{\"message\": \"alert\"}\nPublish ok");
  wificonnect();
  mqttconnect();
}
void loop()
{
  digitalWrite(TRIGGER, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER, LOW);
  int time=pulseIn(ECHO,HIGH);
  distance=(time*sound)/2;
  Serial.print("Distance:");
  Serial.print(distance);
  Serial.println("cms");
  if(distance<100){
    //PublishData(distance);
  }
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
/*.....retrieving to Cloud ..... */
void PublishData(int d) {
  //mqttconnect();
  String payload = \"{\"message\": \"alert\"}\";
  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++) {
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  data3="";
}

```

CIRCUIT DIAGRAM:

Sketch Code:

```

1  #include <WiFi.h>
2  #include <PubSubClient.h>
3
4  #define TRIGGER 2
5  #define ECHO 15
6  #define sound 0.034
7  int distance;
8
9  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
10
11  //-----credentials of IBM Accounts-----
12
13  #define ORG "2jk8e5"
14  #define DEVICE_TYPE "Gas_Leakage_Detection_Device"
15  #define DEVICE_ID "Gas_Detector_1"
16  #define TOKEN "123456789"
17  String data3;
18
19  //----- Customise the above values -----
20
21  char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
22  char publishTopic[] = "iot-2/evt/Data/fmt/json";
23  char subscribetopic[] = "iot-2/cmd/test/fmt/String";
24  char authMethod[] = "use-token-auth";
25  char token[] = TOKEN;
26  char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
27
28  //-----
29
30  WiFiClient wificlient;
31  PubSubClient client(server, 1883, callback ,wificlient);
32  void setup()

```

Simulation Output:

```

Publish ok
Distance:47cms
Sending payload: {"message":"alert"}
Publish ok
Distance:47cms
Sending payload: {"message":"alert"}
Publish ok

```

IBM CLOUD RECENT EVENTS:

IBM Watson IoT Platform

Device: Gas_Detector_1

Device Type: Gas_Leakage_Detection_Device

Device Status: Disconnected

Device Added: Oct 8, 2022 10:10 AM

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago