

ASSIGNMENT – 4

WOKWI SIMULATION

Assignment Date	4 th November 2022
Student Name	Abhirami J.S
Student Roll Number	960219106003
Maximum Marks	2 Marks

Question-1:

Write a code and make a connection in WOKWI for ultrasonic sensor. Whenever distance is less than 100 , send “alert” to IBM cloud and display in device recent events

PROGRAM

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "093eak"
#define DEVICE_TYPE "Abhi2001"
#define DEVICE_ID "Abhirami2001"
#define TOKEN "CB0?Hb&T?84QL+rOSv"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Abhirami/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,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>100){
String payload = "{\"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");
}else {
Serial.println("Publish FAILED");
}
}
}
}
}

```

OUTPUT:

WOKWI SIMULATION

The simulation shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The code in the editor is as follows:

```

76 duration=pulseIn(echopin,HIGH);
77 dist=duration*speed/2;
78 if(dist<100){
79   String payload = "{\"Alert Distance\":\"";
80   payload += dist;
81   payload += "\"}";
82   Serial.print("\n");
83   Serial.print("Sending payload: ");
84   Serial.println(payload);
85   if (client.publish(publishTopic, (char*) payload.c_str())) {
86     Serial.println("Publish OK");
87   }
88 }
89 if(dist>100){
90   String payload = "{\"Distance\":\"";
91   payload += dist;
92   payload += "\"}";
93   Serial.print("\n");
94   Serial.print("Sending payload: ");
95   Serial.println(payload);
96   if(client.publish(publishTopic, (char*) payload.c_str())) {
97     Serial.println("Publish OK");
98   }else {
99     Serial.println("Publish FAILED");
100   }
101 }
102 }
103

```

The terminal output shows the following sequence of events:

```

Publish OK
Sending payload: {"Distance":343.94}
Publish OK
Sending payload: {"Distance":343.94}
Publish OK

```

When Distance < 100 :

The screenshot shows the WOKWI simulation interface. On the left, the sketch.ino file is open, displaying the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "093eak"
6 #define DEVICE_TYPE "Abhi2001"
7 #define DEVICE_ID "Abhirami2001"
8 #define TOKEN "CB0?Hb&T?84QL+r0Sv"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"
12 char publishTopic[] = "iot-2/evt/Abhirami/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 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(trigpin, OUTPUT);
```

The simulation window on the right shows an ESP32 microcontroller connected to an Ultrasonic Distance Sensor. The sensor's distance is 62cm. The console output shows the following messages:

```
Publish OK
Sending payload: {\"Alert Distance\":61.97}
Publish OK
Sending payload: {\"Alert Distance\":61.97}
Publish OK
```

When Distance < 100 :

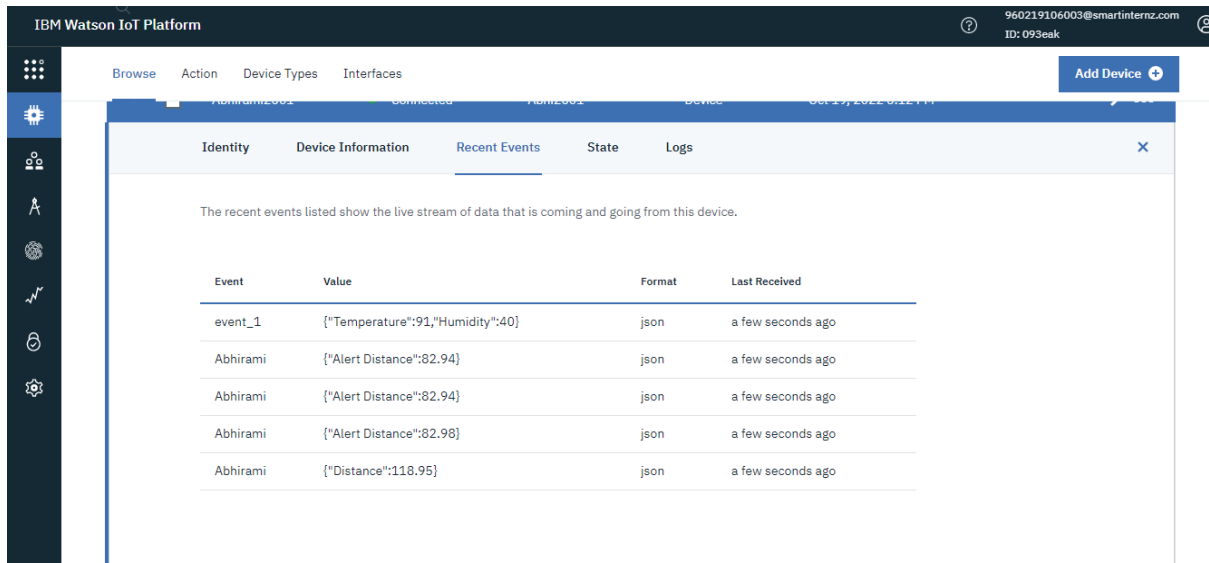
The screenshot shows the WOKWI simulation interface. On the left, the sketch.ino file is open, displaying the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "093eak"
6 #define DEVICE_TYPE "Abhi2001"
7 #define DEVICE_ID "Abhirami2001"
8 #define TOKEN "CB0?Hb&T?84QL+r0Sv"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"
12 char publishTopic[] = "iot-2/evt/Abhirami/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 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(trigpin, OUTPUT);
```

The simulation window on the right shows an ESP32 microcontroller connected to an Ultrasonic Distance Sensor. The sensor's distance is 344cm. The console output shows the following messages:

```
Sending payload: {\"Distance\":343.96}
Publish OK
Sending payload: {\"Distance\":343.96}
Publish OK
Sending payload: {\"Distance\":343.94}
```

IBM CLOUD OUTPUT :



The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows the 'Recent Events' tab for a device named 'Abhirami2002'. The tab is titled 'Identity' and 'Device Information'. Below the tabs, a message states: 'The recent events listed show the live stream of data that is coming and going from this device.' A table lists the recent events with columns for 'Event', 'Value', 'Format', and 'Last Received'.

Event	Value	Format	Last Received
event_1	{"Temperature":91,"Humidity":40}	json	a few seconds ago
Abhirami	{"Alert Distance":82.94}	json	a few seconds ago
Abhirami	{"Alert Distance":82.94}	json	a few seconds ago
Abhirami	{"Alert Distance":82.98}	json	a few seconds ago
Abhirami	{"Distance":118.95}	json	a few seconds ago

WOKWI LINK :

<https://wokwi.com/projects/347468236342690386>