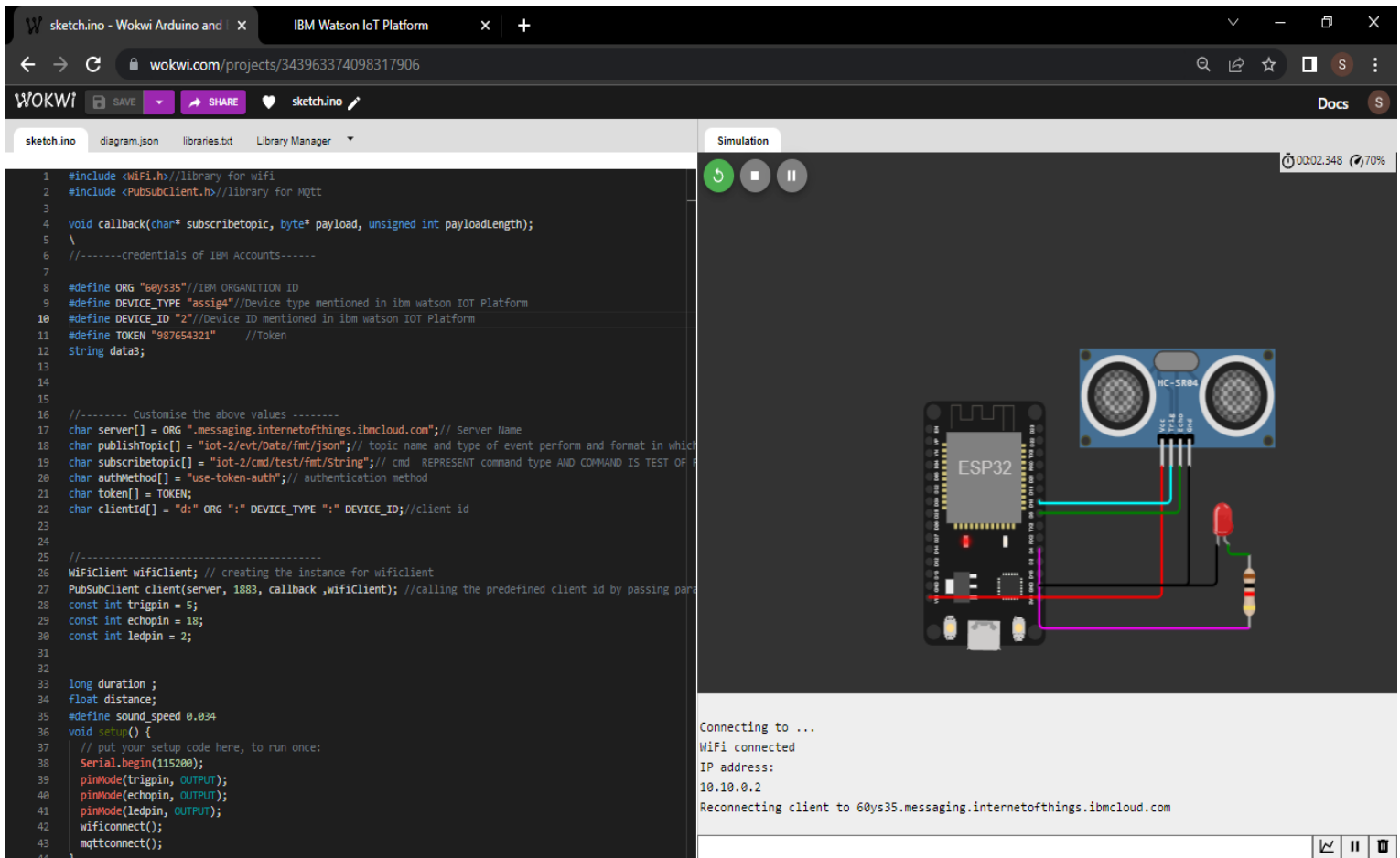


ASSIGNMENT – 04

- i. Write code and connection in Wowki for ultrasonic sensor.
- ii. Whenever distance is less than 100 cm send “**Alert**” to IBM cloud and display in device recent events.

Step 1 . Completed to build Circuit and run program.



The screenshot displays the Wokwi IDE interface with the following components:

- Code Editor:** Contains the following code:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 void callback(char* topic, byte* payload, unsigned int payloadLength);
5 \
6 //-----credentials of IBM Accounts-----
7
8 #define ORG "60ys35" //IBM ORGANIZATION ID
9 #define DEVICE_TYPE "assign4" //Device type mentioned in ibm watson IOT Platform
10 #define DEVICE_ID "2" //Device ID mentioned in ibm watson IOT Platform
11 #define TOKEN "987654321" //Token
12 String data;
13
14
15
16 //----- Customise the above values -----
17 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
18 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and format in which
19 char subscribeTopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command type AND COMMAND IS TEST OF #
20 char authMethod[] = "use-token-auth"; // authentication method
21 char token[] = TOKEN;
22 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
23
24
25 //-----
26 WiFiClient wifiClient; // creating the instance for wifiClient
27 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by passing param
28 const int trigpin = 5;
29 const int echopin = 18;
30 const int ledpin = 2;
31
32
33 long duration ;
34 float distance;
35 #define sound_speed 0.034
36 void setup() {
37   // put your setup code here, to run once:
38   Serial.begin(115200);
39   pinMode(trigpin, OUTPUT);
40   pinMode(echopin, OUTPUT);
41   pinMode(ledpin, OUTPUT);
42   wifiClient.connect();
43   mqttClient.connect();
44 }
```
- Simulation:** Shows a visual representation of the ESP32 board connected to an HC-SR04 ultrasonic sensor and an LED. The sensor is connected to pins 5 (Trig) and 18 (Echo) of the ESP32. The LED is connected to pin 2 (ledpin) and ground.
- Console Output:** Displays the following messages:

```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 60ys35.messaging.internetofthings.ibmcloud.com
```

W sketch.ino - Wokwi Arduino and x IBM Watson IoT Platform x +

wokwi.com/projects/343963374098317906

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

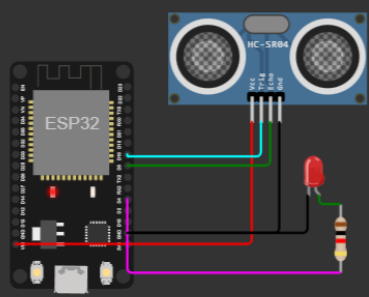
```

111
112 WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
113 while (WiFi.status() != WL_CONNECTED) {
114     delay(500);
115     Serial.print(".");
116 }
117 Serial.println("");
118 Serial.println("WiFi connected");
119 Serial.println("IP address: ");
120 Serial.println(WiFi.localIP());
121 }
122
123 void initManagedDevice() {
124     if (client.subscribe(subscribetopic)) {
125         Serial.println(subscribetopic);
126         Serial.println("subscribe to cmd OK");
127     } else {
128         Serial.println("subscribe to cmd FAILED");
129     }
130 }
131
132 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
133 {
134
135     Serial.print("callback invoked for topic: ");
136     Serial.println(subscribetopic);
137     for (int i = 0; i < payloadLength; i++) {
138         //Serial.print((char)payload[i]);
139         data3 += (char)payload[i];
140     }
141
142     Serial.println("data: " + data3);
143     if(data3=="lighton")
144     {
145         Serial.println(data3);
146     }
147     else
148     {
149         Serial.println(data3);
150     }
151     data3="";
152 }
153

```

Simulation

00:25.246 90%



Connecting to ...
 WiFi connected
 IP address:
 10.10.0.2
 Reconnecting client to 60ys35.messaging.internetofthings.ibmcloud.com

Step 2. Output in WOWKI
 (When Distance is below 100 cm send alert to user)

W sketch.ino - Wokwi Arduino and x IBM Watson IoT Platform x +

wokwi.com/projects/343963374098317906

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

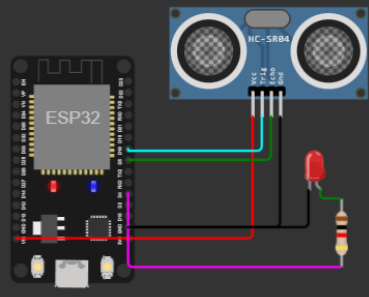
```

40 pinMode(echopin, OUTPUT);
41 pinMode(ledpin, OUTPUT);
42 wificonnect();
43 mqttconnect();
44 }
45
46 void loop() {
47     digitalWrite(trigpin, LOW);
48     digitalWrite(trigpin, HIGH);
49     delayMicroseconds(10);
50     digitalWrite(trigpin, LOW);
51
52     duration= pulseIn(echopin,HIGH);
53     distance = duration * sound_speed /2;
54     if(distance<=100){
55         PublishData(distance);
56         delay(1000);
57         if (!client.isConnected()) {
58             mqttconnect();
59         }
60         digitalWrite(ledpin, HIGH);
61         Serial.println("ALERT.....!!");
62         Serial.println(distance);
63     }
64     else
65     {
66         digitalWrite(ledpin, LOW);
67     }
68     // put your main code here, to run repeatedly:
69     delay(10); // this speeds up the simulation
70 }
71
72 /*.....Retrieving to Cloud.....*/
73
74 void PublishData(Float distance) {
75     mqttconnect();//function call for connecting to ibm
76
77     // creating the String in in-form JSON to update the data to ibm cloud
78     String payload = "{\"ALERT...!! \": ";
79     payload += distance;
80     payload += "}";
81
82     Serial.print("Sending payload: ");

```

Simulation

00:28.174 96%



0.00
 Sending payload: {"ALERT...!! ": 0.00}
 Publish ok
 ALERT.....!!
 0.00
 Sending payload: {"ALERT...!! ": 0.00}
 Publish ok

sketch.ino - Wokwi Arduino and x IBM Watson IoT Platform x +

7kb3es.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

410119106009@smartintemz.com
ID: 7kb3es

Browse Action Device Types Interfaces

Add Device +

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
28201816	Disconnected	wowiki	Device	Oct 23, 2022 1:05 PM
4321	Connected	assign4	Device	Nov 18, 2022 3:37 PM

Identity Device Information Recent Events State Logs

Device ID: 4321
Device Type: assign4
Date Added: Nov 18, 2022 3:37 PM
Added By: 410119106009@smartintemz.com
Connection Status: Connected
Connection Time: Nov 18, 2022 3:40 PM
Client Address: 50.31.197.64 Insecure

Step 3.Output in IBM CLOUD (Watson Platform)

sketch.ino - Wokwi Arduino and x IBM Watson IoT Platform x +

7kb3es.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

410119106009@smartintemz.com
ID: 7kb3es

Browse Action Device Types Interfaces

Add Device +

Device ID	Status	Device Type	Class ID	Date Added
28201816	Disconnected	wowiki	Device	Oct 23, 2022 1:05 PM
4321	Disconnected	assign4	Device	Nov 18, 2022 3:37 PM
56789	Disconnected	deepi_reshepeery	Device	Oct 22, 2022 5:58 PM
636903	Connected	suba	Device	Nov 5, 2022 2:24 PM

Identity Device Information Recent Events State Logs

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

Event	Value	Format	Last Received
Data	{"ALERT...!! ":"33.98"}	json	a few seconds ago
Data	{"ALERT...!! ":"33.98"}	json	a few seconds ago
Data	{"ALERT...!! ":"33.98"}	json	a few seconds ago
Data	{"ALERT...!! ":"33.98"}	json	a few seconds ago
Data	{"ALERT...!! ":"33.98"}	json	a few seconds ago

6789 Disconnected Arduino Device Oct 20, 2022 8:52 PM

PROGRAM:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
\
//-----credentials of IBM Accounts-----

#define ORG "60ys35" //IBM ORGANITION ID
#define DEVICE_TYPE "assign4" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "4321" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "0987654321" //Token
String data3;
//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined
client id by passing parameter like server id,portand wificredential
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;

long duration ;
float distance;
#define sound_speed 0.034
void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, OUTPUT);
    pinMode(ledpin, OUTPUT);
    wificonnect();
    mqttconnect();
}

void loop() {
    digitalWrite(trigpin, LOW);
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin, LOW);
```

```

duration= pulseIn(echopin,HIGH);
distance = duration * sound_speed /2;
if(distance<=100){
    PublishData(distance);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
    digitalWrite(ledpin, HIGH);
    Serial.println("ALERT...!!!");
    Serial.println(distance);
}
else
{
    digitalWrite(ledpin, LOW);
}
// put your main code here, to run repeatedly:
delay(10); // this speeds up the simulation
}

/*.....retrieving to
Cloud.....*/

void PublishData(float distance) {
    mqttconnect();//function call for connecting to ibm

    // creating the String in in form JSON to update the data to ibm cloud
    String payload = "{\"ALERT...!! \": ";
    payload += distance;
    payload += "}";

    Serial.print("Sending payload: ");
    Serial.println(payload);

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it sucessfully upload data on the cloud
        then it will print publish ok in Serial monitor or else it will print publish
        failed
    } 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() //function definition for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish
the connection
    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);
    if(data3=="lighton")
    {
        Serial.println(data3);
    }
    else
    {
        Serial.println(data3);
    }
    data3="";
}

```