

ASSIGNMENT 4
MAHENDRA INSTITUTE OF TECHNOLOGY

Date	28 October2022
Team ID	PNT2022TMID17220
Project Name	Signs with smart connectivity for better road safety
Maximum Marks	2 Marks

Question:

Write code and connections in wowki for ultrasonic sensor. Whenever distance is less than 100cms send “alert” to IBM cloud and display in device recent events.

Program Code:

```
#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 "n6r19n"//IBM Organization ID
#define DEVICE_TYPE "ElangoIoT"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "06112002"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "98765432" //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()
{
    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);
    }
    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 defination 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());
}

```

```

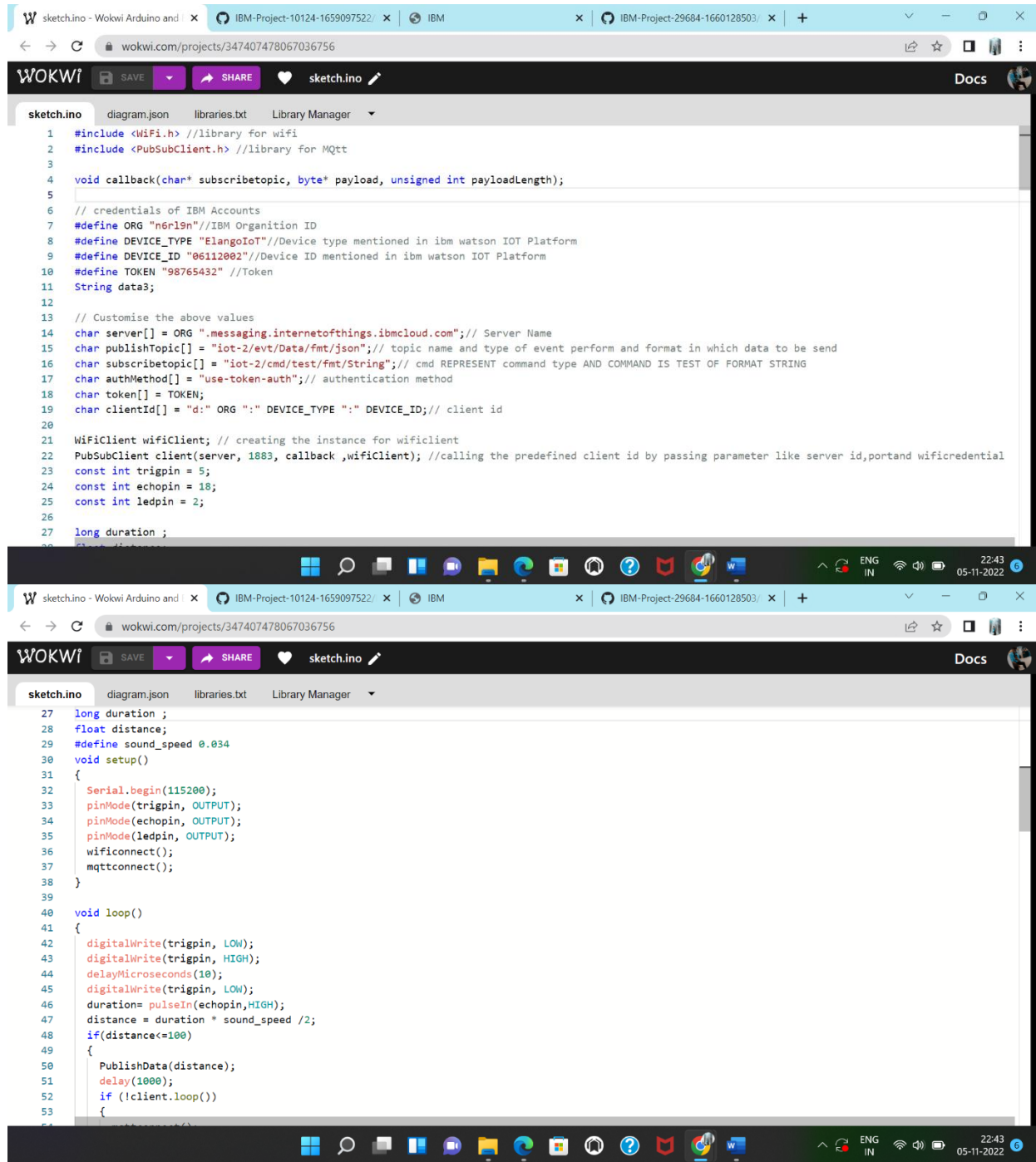
}

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="";
}

```

OUTPUT:



```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
5
6 // credentials of IBM Accounts
7 #define ORG "n6r19n" //IBM Organization ID
8 #define DEVICE_TYPE "ElangoIoT" //Device type mentioned in ibm watson IOT Platform
9 #define DEVICE_ID "06112002" //Device ID mentioned in ibm watson IOT Platform
10 #define TOKEN "98765432" //Token
11 String data3;
12
13 // Customise the above values
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and format in which data to be send
16 char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
17 char authMethod[] = "use-token-auth"; // authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; // client id
20
21 WiFiClient wifiClient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential
23 const int trigpin = 5;
24 const int echopin = 18;
25 const int ledpin = 2;
26
27 long duration ;
28 float distance;
29 #define sound_speed 0.034
30 void setup()
31 {
32   Serial.begin(115200);
33   pinMode(trigpin, OUTPUT);
34   pinMode(echopin, OUTPUT);
35   pinMode(ledpin, OUTPUT);
36   wifiConnect();
37   mqttConnect();
38 }
39
40 void loop()
41 {
42   digitalWrite(trigpin, LOW);
43   digitalWrite(trigpin, HIGH);
44   delayMicroseconds(10);
45   digitalWrite(trigpin, LOW);
46   duration = pulseIn(echopin, HIGH);
47   distance = duration * sound_speed / 2;
48   if (distance <= 100)
49   {
50     PublishData(distance);
51     delay(1000);
52     if (!client.loop())
53     {
54       // Reconnect to MQTT broker
55     }
56   }
57 }
```

W sketch.ino - Wokwi Arduino and x IBM-Project-10124-1659097522/ x IBM x IBM-Project-29684-1660128503/ x +

wokwi.com/projects/347407478067036756

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```
53 {
54   mqttconnect();
55 }
56 digitalWrite(ledpin, HIGH);
57 Serial.println("Alert !!");
58 Serial.println(distance);
59 }
60 else
61 {
62   digitalWrite(ledpin, LOW);
63 }
64 delay(10); // this speeds up the simulation
65 }
66
67 // Retrieving to Cloud
68
69 void PublishData(float distance)
70 {
71   mqttconnect();// Function call for connecting to ibm
72   // creating the String in form JSON to update the data to ibm cloud
73   String payload = "{\"Alert !! \": ";
74   payload += distance;
75   payload += "}";
76   Serial.print("Sending payload : ");
77   Serial.println(payload);
78   if (client.publish(publishTopic, (char*) payload.c_str()))
79   {
```

Windows taskbar: 22:43 05-11-2022

W sketch.ino - Wokwi Arduino and x IBM-Project-10124-1659097522/ x IBM x IBM-Project-29684-1660128503/ x +

wokwi.com/projects/347407478067036756

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```
79 {
80   Serial.println("Publish ok");// If it successfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publi
81 }
82 else
83 {
84   Serial.println("Publish failed");
85 }
86 }
87
88 void mqttconnect()
89 {
90   if (!client.connected())
91   {
92     Serial.print("Reconnecting client to ");
93     Serial.println(server);
94     while (!client.connect(clientId, authMethod, token))
95     {
96       Serial.print(".");
97       delay(500);
98     }
99     initManagedDevice();
100     Serial.println();
101   }
102 }
103
104 void wificonnect() // Function definition for wificonnect
105 {
```

Windows taskbar: 22:43 05-11-2022

W sketch.ino - Wokwi Arduino and x IBM-Project-10124-1659097522/ x IBM x IBM-Project-29684-1660128503/ x +

wokwi.com/projects/347407478067036756

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```
105 {
106   Serial.println();
107   Serial.print("Connecting to ");
108   WiFi.begin("Wokwi-GUEST", "", 6); // Passing the wifi credentials to establish the connection
109   while (WiFi.status() != WL_CONNECTED)
110   {
111     delay(500);
112     Serial.print(".");
113   }
114   Serial.println("");
115   Serial.println("WiFi connected");
116   Serial.println("IP address: ");
117   Serial.println(WiFi.localIP());
118 }
119
120 void initManagedDevice()
121 {
122   if (client.subscribe(subscribetopic))
123   {
124     Serial.println((subscribetopic));
125     Serial.println("subscribe to cmd OK");
126   }
127   else
128   {
129     Serial.println("subscribe to cmd FAILED");
130   }
131 }
```

Windows taskbar: 22:43 05-11-2022

W sketch.ino - Wokwi Arduino and x IBM-Project-10124-1659097522/ x IBM x IBM-Project-29684-1660128503/ x +

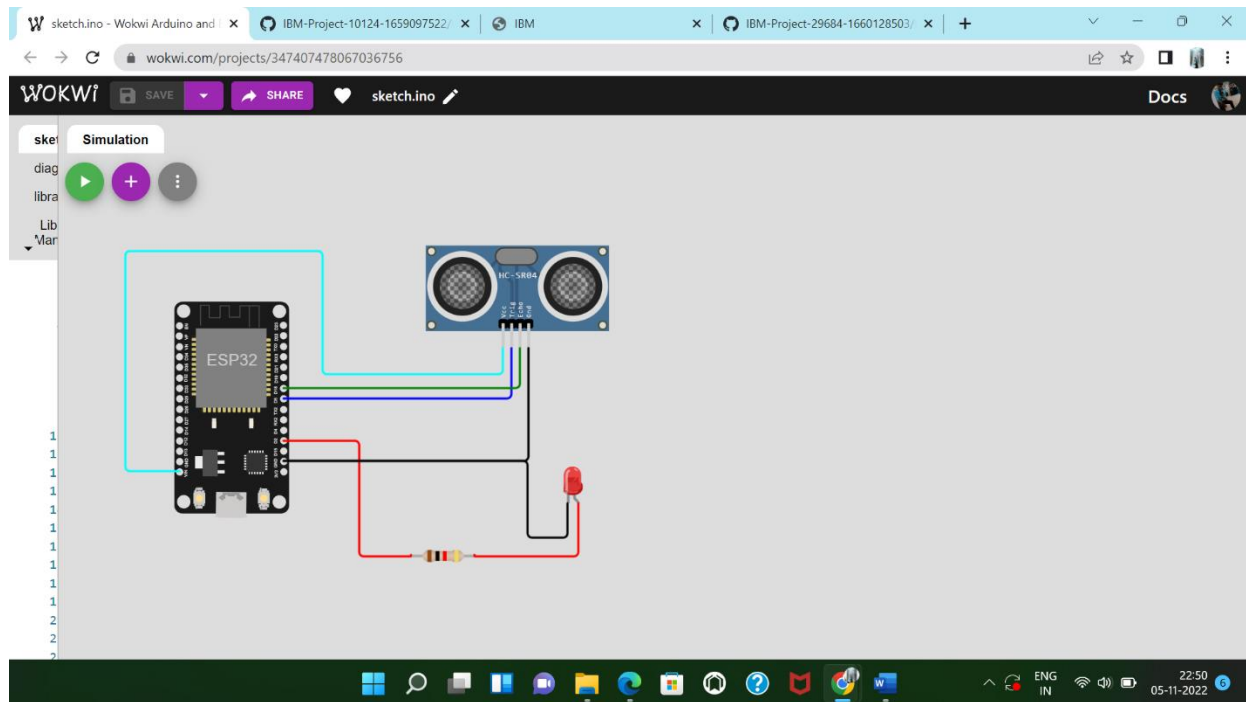
wokwi.com/projects/347407478067036756

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```
127 else
128 {
129   Serial.println("subscribe to cmd FAILED");
130 }
131 }
132
133 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
134 {
135   Serial.print("callback invoked for topic: ");
136   Serial.println(subscribetopic);
137   for (int i = 0; i < payloadLength; i++)
138   {
139     //Serial.print((char)payload[i]);
140     data3 += (char)payload[i];
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 }
```

Windows taskbar: 22:43 05-11-2022



Referral Link: <https://wokwi.com/projects/347407478067036756>