

Assignment -4

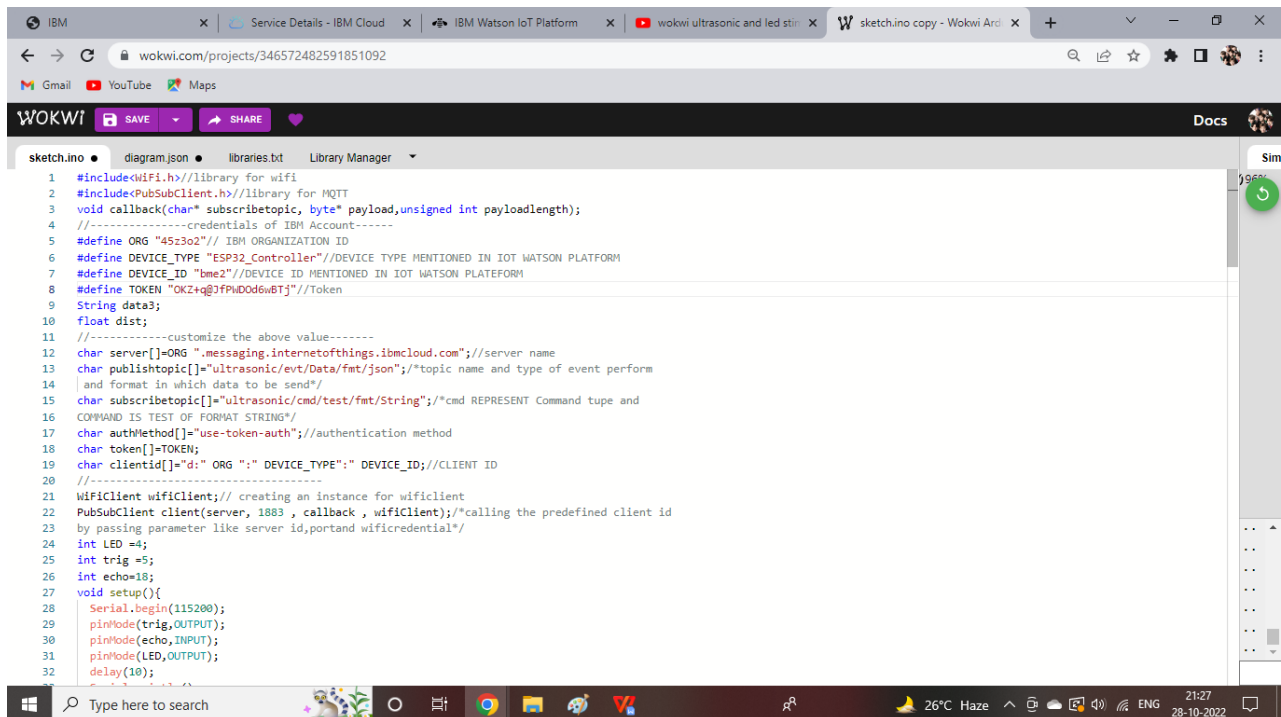
Wokwi Programming

Assignment Date	29 October 2022
Student Name	V.Hema Dharshini
Student Roll Number	912419104009
Maximum Marks	2 Marks

Question-1:

Write code and connections in wokwi for ultrasonic sensor.
Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.
Upload document with wokwi share link and images of ibm cloud.

Solution:
Coding:-



```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 void callback(char* topic, byte* payload, unsigned int payloadlength);
4 //-----credentials of IBM Account-----
5 #define ORG "452302" // IBM ORGANIZATION ID
6 #define DEVICE_TYPE "ESP32_Controller" //DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
7 #define DEVICE_ID "bme2" //DEVICE ID MENTIONED IN IOT WATSON PLATFORM
8 #define TOKEN "OKZ+q@3fPhD0d6wBTj" //Token
9 String data3;
10 float dist;
11 //-----customize the above value-----
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //server name
13 char publishTopic[] = "ultrasonic/evt/Data/fmt/json"; //topic name and type of event perform
14 //and format in which data to be send*/
15 char subscribeTopic[] = "ultrasonic/cmd/test/fmt/String"; //cmd REPRESENT Command tupe and
16 //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 an instance for wifi client
22 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id
23 //by passing parameter like server id, port and wifi credential*/
24 int LED = 4;
25 int trig = 5;
26 int echo = 18;
27 void setup() {
28   Serial.begin(115200);
29   pinMode(trig, OUTPUT);
30   pinMode(echo, INPUT);
31   pinMode(LED, OUTPUT);
32   delay(10);
33 }
```

```
31 pinMode(LED,OUTPUT);
32 delay(10);
33 Serial.println();
34 wificonnect();
35 mqttconnect();
36
37 void loop() {
38   digitalWrite(trig,LOW);
39   digitalWrite(trig,HIGH);
40   delayMicroseconds(10);
41   digitalWrite(trig,LOW);
42   float dur=pulseIn(echo,HIGH);
43   float dist=(dur * 0.0343)/2;
44   Serial.print("distance in cm");
45   Serial.println(dist);
46   PublishData(dist);
47   delay(1000);
48   if (!client.loop()){
49     mqttconnect();
50   }
51 }
52 /*.....retriving to cloud.....*/
53 void PublishData(float dist){
54   mqttconnect();//function call for connecting to ibm
55   /*creating the string in form of JSON to update the data to ibm cloud*/
56   String object;
57   if(dist<100)
58   {
59     digitalWrite(LED,HIGH);
60     Serial.println("no object is near");
61     object="Near";
62   }
```

```
66 Serial.println( "no object found ");
67 object="No";
68 }
69 String payload="{\"distance\":";
70 payload +=dist;
71 payload +=\",\" \"object\":\":";
72 payload += object;
73 payload += "\":";
74
75 Serial.print("Sending payload: ");
76 Serial.println(payload);
77 if(client.publish(topic,(char*) payload.c_str())){
78   Serial.println("Publish ok");// if its successfully upload data on the cloud then it will print
79   publish ok in serial monitor or else it will print publish failed*/
80 } else{
81   Serial.println("Publish failed");
82 }
83
84 void mqttconnect(){
85   if(!client.connected()){
86     Serial.print("Reconnecting client to ");
87     Serial.println(server);
88     while(!client.connect(clientid,authMethod, token)){
89       Serial.print(".");
90       delay(500);
91     }
92     initManagedDevice();
93     Serial.println();
94   }
95 }
96 void wificonnect()//function definition for wificonnect
97 {
98   Serial.println();
```

```
98 Serial.println();
99 Serial.print("Connecting to ");
100 WiFi.begin("vivo 1816", "taetae95");//PASSING THE WIFI CREDENTIALS TO ESTABLISH CONNECTION
101 while (WiFi.status() !=WL_CONNECTED){
102   delay(500);
103   Serial.print(".");
104 }
105 Serial.println("");
106 Serial.println("WiFi connected");
107 Serial.println("IP address:");
108 Serial.println(WiFi.localIP());
109 }
110 void initManagedDevice(){
111   if(client.subscribe(subscribetopic)){
112     Serial.println(subscribetopic);
113     Serial.println("subscribe to cmd OK");
114   }else{
115     Serial.println("subscribe to cmd failed");
116   }
117 }
118 void callback(char* subscribetopic,byte*payload,unsigned int payloadLength)
119 {
120   Serial.print("callback invoked for topic: ");
121   Serial.println(subscribetopic);
122   for(int i=0; i< payloadLength; i++){
123     //Serial.print((char)payload[i]);
124     data3 +=(char)payload[i];
125   }
126   //Serial.println("dta: "+ data3);
127   //if(data3=="Near")
128   //{
129   //Serial.println(data3);
```


DATA SENT TO IBM CLOUD ON OBJECT BEING DETECTED

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A table lists devices, with 'bme2' selected. Below the table, the 'Recent Events' tab is active, displaying a list of events with columns for 'Event', 'Value', 'Format', and 'Last Received'. The events show distance data being sent to the cloud.

Event	Value	Format	Last Received
Data	{\"distance\":97.82,\"object\":\"Near\"}	json	a few seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	json	a few seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	json	a few seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	json	a few seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	json	a few seconds ago

The screenshot shows the Wokwi IDE interface. On the left, the 'sketch.ino' file is open, displaying the Arduino code for connecting to IBM Watson IoT. On the right, a 'Simulation' window shows a 3D model of an ESP32 board connected to an ultrasonic sensor. The simulation output on the right shows the device sending data to the cloud.

```
1 #include<WiFi.h> //library for wifi
2 #include<PubSubClient.h> //library for MQTT
3 void callback(char* topic, byte* payload, unsigned int length) {
4 //-----credentials of IBM Account-----
5 #define ORG "45z3o2" // IBM ORGANIZATION ID
6 #define DEVICE_TYPE "ESP32_Controller" //DEVICE TYPE MENTIONED IN IOT WATSON
7 #define DEVICE_ID "bme2" //DEVICE ID MENTIONED IN IOT WATSON
8 #define TOKEN "OKZ+q@JfPWdOd6wBTj" //Token
9 String data3;
10 float dist;
11 //-----customize the above value-----
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
13 char publishtopic[] = "ultrasonic/evt/Data/fmt/json"; //topic name
14 //and format in which data to be send*/
15 char subscribetopic[] = "ultrasonic/cmd/test/fmt/string"; //command
16 //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 an instance for wificlient
22 PubSubClient client(server, 1883, callback, wificlient);
```

OUTPUT:

<https://wokwi.com/projects/346780783662334547>