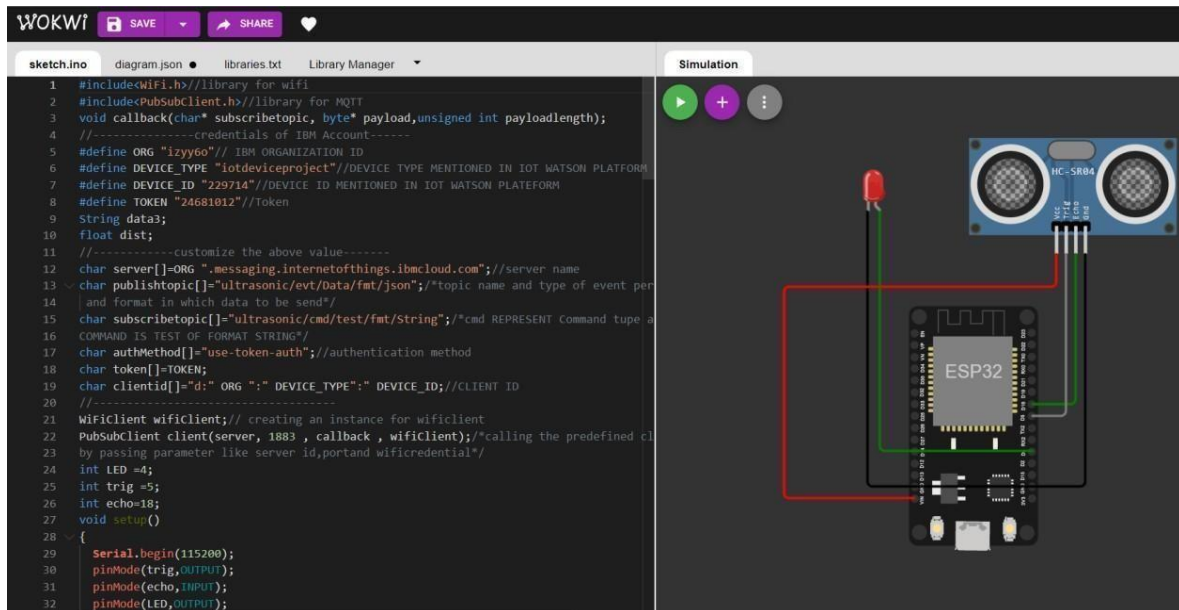


## Assignment -4

### Question-1:

Write code and connections in wokwi for ultrasonic sensors. Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.



WOKWI
SAVE
SHARE
Docs

sketch.ino
diagram.json
libraries.txt
Library Manager

```

32 pinMode(LED,OUTPUT);
33 delay(10);
34 wifiConnect();
35 mqttconnect();
36 }
37 void loop()//recursive function
38 {
39   digitalWrite(trig,LOW);
40   digitalWrite(trig,HIGH);
41   delayMicroseconds(10);
42   digitalWrite(trig,LOW);
43   float dur=pulseIn(echo,HIGH);
44   float dist=(dur * 0.0343)/2;
45   Serial.print("distance in cm");
46   Serial.println(dist);
47   PublishData(dist);
48   delay(1000);
49   if (!client.loop()){
50     mqttconnect();
51   }
52 }
53 /*.....Retriving to cloud.....*/
54 void PublishData(float dist){
55   mqttconnect();//function call for connecting to ibm
56   /*creating the string in form of JSON to update the data to ibm cloud*/
57   String object;
58   if(dist<100)
59   {
60     digitalWrite(LED,HIGH);
61     Serial.println("no object is near");
62     object="Near";
63   }

```

Simulation

WOKWI
SAVE
SHARE

sketch.ino
diagram.json
libraries.txt
Library Manager

```

63 }
64 else
65 {
66   digitalWrite(LED,LOW);
67   Serial.println("no object found");
68   object="No";
69 }
70 String payload="{\"distance\":";
71 payload +=dist;
72 payload +=",\" \"object\":\":";
73 payload += object;
74 payload += "\":";
75
76 Serial.print("Sending payload: ");
77 Serial.println(payload);
78 if(client.publish(publishtopic, (char*) payload.c_str())){
79   Serial.println("Publish ok");/* if its sucessfully upload data on the cloud then
80   publish ok in serial monitor or else it will print publish failed*/
81 } else{
82   Serial.println("Publish failed");
83 }
84 }
85 void mqttconnect(){
86   if(!client.connected()){
87     Serial.print("Reconnecting client to ");
88     Serial.println(server);
89     while(!client.connect(clientid,authMethod, token)){
90       Serial.print(".");
91       delay(500);
92     }
93   }
94   initManagedDevice();
95   Serial.println();

```

Simulation

WOKWI

SAVE

SHARE

Docs

sketch.ino

diagram.json

libraries.txt

Library Manager

```

123 initManagedDevice();
124 Serial.println();
125 }
126 }
127 void wificonnect()//function defenition for wificonnect
128 {
129     Serial.println();
130     Serial.print("connecting to ");
131     Wifi.begin("wokwi.GUEST", "",6);//PASSING THE WIFI CREDENTIALS TO ESTABLISH CONNE
132     while (Wifi.status() !=WL_CONNECTED){
133         delay(500);
134         Serial.print(".");
135     }
136     Serial.println("");
137     Serial.println("Wifi connected");
138     Serial.println("IP address");
139     Serial.println(Wifi.localIP());
140 }
141 void initManagedDevice(){
142     if(client.subscribe(subscribetopic)){
143         Serial.println(subscribetopic);
144         Serial.println("subscribe to cmd OK");
145     }else{
146         Serial.println("subscribe to cmd failed");
147     }
148 }
149 void callback(char* subscribetopic,byte*payload,unsigned int payloadLength)
150 {
151     Serial.print("callback invoked for topic: ");
152     Serial.println(subscribetopic);
153     for(int i=0; i< payloadLength; i++){
154         //Serial.print((char)payload[i]);
155         data3 +=(char)payload[i];
156     }
157 }

```

Simulation

WOKWI

SAVE

SHARE

Docs

sketch.ino

diagram.json

libraries.txt

Library Manager

```

111 void initManagedDevice(){
112     if(client.subscribe(subscribetopic)){
113         Serial.println(subscribetopic);
114         Serial.println("subscribe to cmd OK");
115     }else{
116         Serial.println("subscribe to cmd failed");
117     }
118 }
119 void callback(char* subscribetopic,byte*payload,unsigned int payloadLength)
120 {
121     Serial.print("callback invoked for topic: ");
122     Serial.println(subscribetopic);
123     for(int i=0; i< payloadLength; i++){
124         //Serial.print((char)payload[i]);
125         data3 +=(char)payload[i];
126     }
127     //Serial.println("dta: "+ data3);
128     //if(data3=="Wear")
129     //{
130     //Serial.println(data3);
131     //digitalWrite(LED,HIGH);
132     //}
133     //else
134     //{
135     //Serial.println(data3);
136     //digitalWrite(LED,LOW);
137     //}
138     data3="";
139 }

```

Simulation

## DATA SENT TO IBM CLOUD ON NO OBJECT DETECTED

The screenshot shows the IBM Cloud IoT Platform interface for a device named 'DISTANCEDETECT'. The device is in a 'Disconnected' state. The 'Recent Events' tab is selected, displaying a table of events. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. There are five rows of data, all with the same value: '{"distance":79.66,"object":"No"}'. The 'Format' column shows 'json' and the 'Last Received' column shows 'a few seconds ago'.

Event	Value	Format	Last Received
Data	{"distance":79.66,"object":"No"}	json	a few seconds ago
Data	{"distance":79.66,"object":"No"}	json	a few seconds ago
Data	{"distance":79.66,"object":"No"}	json	a few seconds ago
Data	{"distance":79.66,"object":"No"}	json	a few seconds ago
Data	{"distance":79.66,"object":"No"}	json	a few seconds ago

## WHEN OBJECT DETECTED BY ULTRASONIC DETECTOR SENSOR

The simulation shows an ESP32 microcontroller board connected to an HC-SR04 ultrasonic sensor. The sensor's VCC pin is connected to the ESP32's 5V pin, GND to GND, and Trig to D4. The Echo pin is connected to a red LED. The simulation output shows the following sequence of events:

```
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
Distancein cm141.21
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
```

