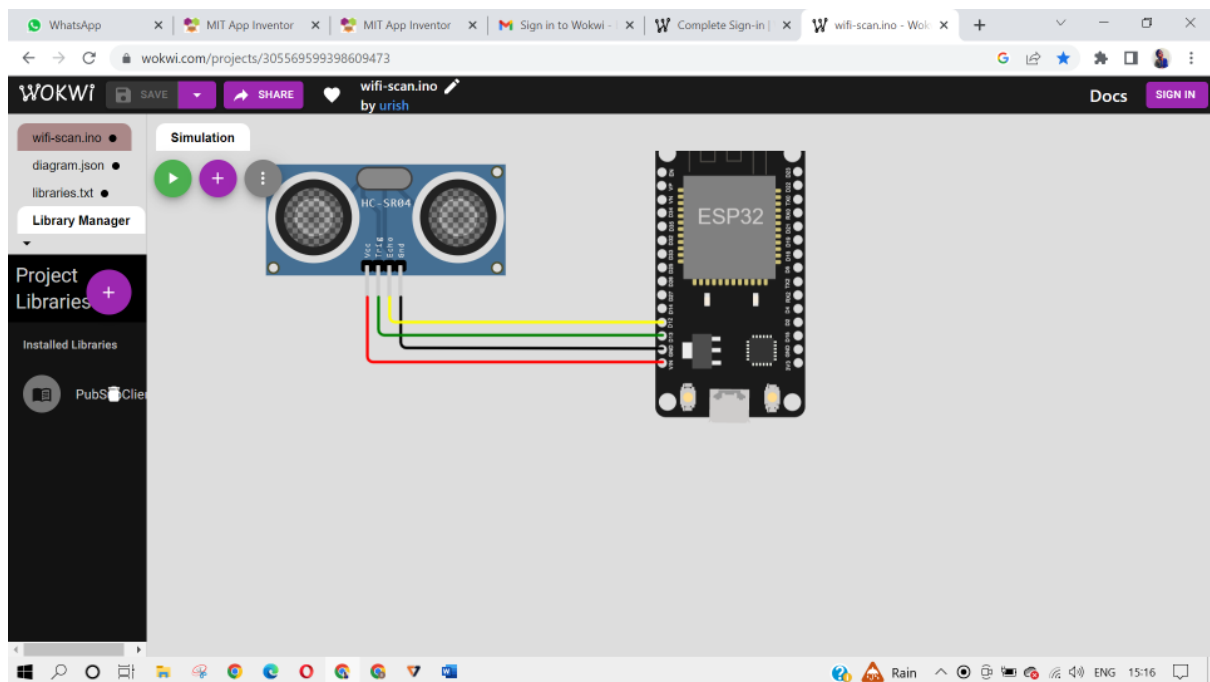


## ASSIGNMENT – 4

Write a code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cm and send an “alert” to the IBM cloud and display in the device recent events.

### CIRCUIT DIAGRAM:



### LINK:

<https://wokwi.com/projects/305569599398609473>

## OUTPUT SCREENSHOT AND CODE:

WOKWI

SAVE SHARE

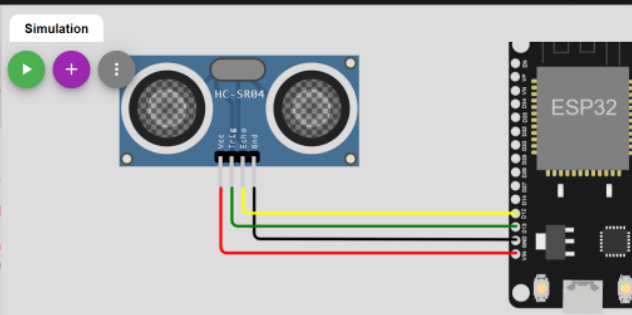
wifi-scan.ino by urish

Docs SIGN IN

wifi-scan.ino • diagram.json • libraries.txt • Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define TRIG_PIN 13
4 #define ECHO_PIN 12
5
6
7
8 void callback(char* subscribetopic, byte* payload, unsigned int payl
9
10 //-----credentials of IBM Accounts-----
11
12 #define ORG "Sgraia" //IBM ORGANIZATION ID
13 #define DEVICE_TYPE "sensor" //Device type mentioned in ibm watson IoT
14 #define DEVICE_ID "sensor_1" //Device ID mentioned in ibm watson IoT
15 #define TOKEN "12345678" //Token
16
17
18
19 //----- Customise the above values -----
20 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Se
21 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and ty
22 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRE
23 char authMethod[] = "use-token-auth"; // authentication method
24 char token[] = TOKEN;
25 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
26
27
28
```

Simulation



23°C Rain showers

WOKWI

SAVE SHARE

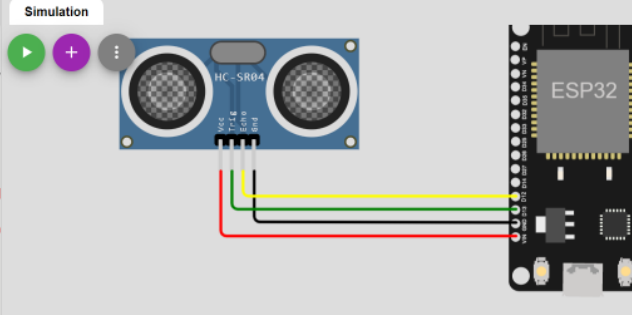
wifi-scan.ino by urish

Docs SIGN IN

wifi-scan.ino • diagram.json • libraries.txt • Library Manager

```
29 WiFiClient wificlient; // creating the instance for wificlient
30 PubSubClient client(server, 1883, callback, wificlient); //calling
31
32
33 void setup() // configureing the ESP32
34 {
35   Serial.begin(115200);
36   pinMode(TRIG_PIN, OUTPUT);
37   digitalWrite(TRIG_PIN, LOW);
38   pinMode(ECHO_PIN, INPUT);
39   delay(10);
40   Serial.println();
41   wificlient.connect();
42   mqttconnect();
43 }
44
45 void loop() // Recursive Function
46 {
47   digitalWrite(TRIG_PIN, HIGH);
48   delayMicroseconds(10);
49   digitalWrite(TRIG_PIN, LOW);
50   float duration_us = pulseIn(ECHO_PIN, HIGH);
51   float distance = 0.017 * duration_us;
52
53   if(distance < 100)
54   {
55     PublishData(distance, "ALERT");
56   }
57 }
```

Simulation



Rain

WOKWI

SAVE SHARE wifi-scan.ino by urish Docs SIGN UP

wifi-scan.ino • diagram.json • libraries.txt • Library Manager

```

57   PublishData(distance, "SAFE");
58 }
59
60 delay(1000);
61 if (!client.loop()) {
62   mqttconnect();
63 }
64 }
65
66 .....retrieving to Cloud.....
67
68 void PublishData(float d, char s[]) {
69   mqttconnect(); //function call for connecting to IBM
70   /*
71    * creating the String in form JSON to update the data to IBM
72    */
73   String payload = "{\"Distance\": ";
74   payload+=d;
75   payload+=",";
76   payload+="\"MESSAGE\": ";
77   payload+=s;
78   payload+="\"";
79   payload+="\"";
80   payload+="\"";
81   payload+="\"";
82   payload+="\"";
83 }
84

```

Simulation

ESP32

HC-SR04

Simulation controls: Play, Stop, Reset, and a menu icon.

System tray: Rain, network, volume, and system clock (15:13).

WOKWI

SAVE SHARE wifi-scan.ino by urish Docs SIGN UP

wifi-scan.ino • diagram.json • libraries.txt • Library Manager

```

84
85 Serial.print("Sending payload: ");
86 Serial.println(payload);
87
88
89 if (client.publish(publishTopic, (char*) payload.c_str())) {
90   Serial.println("Publish ok");// if it successfully upload data on
91 } else {
92   Serial.println("Publish failed");
93 }
94
95 }
96
97
98 void mqttconnect() {
99   if (!client.connected()) {
100     Serial.print("Reconnecting client to ");
101     Serial.println(server);
102     while (!client.connect(clientId, authMethod, token)) {
103       Serial.print(".");
104       delay(500);
105     }
106   }
107   initManagedDevice();
108   Serial.println();
109 }
110
111 void wificonnect() //function definition for wificonnect

```

Simulation

ESP32

HC-SR04

Simulation controls: Play, Stop, Reset, and a menu icon.

System tray: Rain, network, volume, and system clock (15:13).

WOKWI

SAVE SHARE wifiscan.ino by urish Docs SIGN IN

wifiscan.ino • diagram.json • libraries.txt • Library Manager

```

111 void wificonnect() //function definition for wificonnect
112 {
113     Serial.println();
114     Serial.print("Connecting to ");
115
116     WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to
117     while (WiFi.status() != WL_CONNECTED) {
118         delay(500);
119         Serial.print(".");
120     }
121     Serial.println("");
122     Serial.println("WiFi connected");
123     Serial.println("IP address: ");
124     Serial.println(WiFi.localIP());
125 }
126
127 void initManagedDevice() {
128     if (client.subscribe(subscribetopic)) {
129         Serial.println((subscribetopic));
130         Serial.println("subscribe to cmd OK");
131     } else {
132         Serial.println("subscribe to cmd FAILED");
133     }
134 }
135
136 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength) {
137 }
138

```

Simulation

ESP32

HC-SR04

WOKWI

SAVE SHARE wifiscan.ino by urish Docs SIGN IN

wifiscan.ino • diagram.json • libraries.txt • Library Manager

```

1 {
2   "version": 1,
3   "author": "19C047 LAKSHMI SREE",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -83.74, "left": 120.97, "width": 120, "height": 44, "color": "black", "attrs": {} },
7     { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -65.62, "left": -75.27, "width": 120, "height": 44, "color": "blue", "attrs": { "distance": "169" } },
8   ],
9   "connections": [
10    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
11    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
12    [ "esp:GND.2", "ultrasonic1:GND", "black", [ "v-0.44", "h-120.97" ] ],
13    [ "esp:VIN", "ultrasonic1:VCC", "red", [ "h0" ] ],
14    [ "esp:D12", "ultrasonic1:ECHO", "yellow", [ "h0" ] ],
15    [ "esp:D13", "ultrasonic1:TRIG", "green", [ "h0" ] ]
16  ]
17 }

```

Simulation

ESP32

HC-SR04

WOKWI

SAVE SHARE wifiscan.ino by urish Docs SIGN IN

wifiscan.ino • diagram.json • libraries.txt • Library Manager

```

1 {
2   "version": 1,
3   "author": "19C047 LAKSHMI SREE",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -83.74, "left": 120.97, "width": 120, "height": 44, "color": "black", "attrs": {} },
7     { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -65.62, "left": -75.27, "width": 120, "height": 44, "color": "blue", "attrs": { "distance": "169" } },
8   ],
9   "connections": [
10    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
11    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
12    [ "esp:GND.2", "ultrasonic1:GND", "black", [ "v-0.44", "h-120.97" ] ],
13    [ "esp:VIN", "ultrasonic1:VCC", "red", [ "h0" ] ],
14    [ "esp:D12", "ultrasonic1:ECHO", "yellow", [ "h0" ] ],
15    [ "esp:D13", "ultrasonic1:TRIG", "green", [ "h0" ] ]
16  ]
17 }

```

Simulation

ESP32

HC-SR04

