

## Assignment -4 Wowki & IBM Cloud

Assignment Date	04 October 2022
Student Name	Nandhakumar S
Student Roll Number	310819106054
Maximum Marks	2 Marks

### Question-1:

Write code and connections in wowki for the ultrasonic sensor.

Whenever the distance is less than 100cms sent "alert" to IBM cloud and display in device recent events.

### Code:

```
#include <WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

WiFiClient wifiClient; String

data3;

#define ORG "q26y5w"

#define DEVICE_TYPE "TestDeviceType"

#define DEVICE_ID "2022"

#define TOKEN "uu(rZQw9292EfSFGDX"

#define speed 0.034 #define led 14 char server[] = ORG

".messaging.internetofthings.ibmcloud.com"; char publishTopic[]

= "iot-2/evt/shreedharen/fmt/json";

char topic[] = "iot-2/cmd/led/fmt/String"; char authMethod[] =
```

```
"use-token-auth"; char token[] = TOKEN; char clientId[] = "d:"
```

```
ORG ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
PubSubClient client(server, 1883, wifiClient);
```

```
const int trigpin=5; const int echopin=18;
```

```
String command; String data=""; long duration;
```

```
float dist; void setup()
```

```
{
```

```
Serial.begin(115200);
```

```
pinMode(led, OUTPUT);
```

```
pinMode(trigpin, OUTPUT);
```

```
pinMode(echopin, INPUT);
```

```
wifiConnect();
```

```
mqttConnect(); } void loop()
```

```
{ bool isNearby = dist < 100;
```

```
digitalWrite(led, isNearby);
```

```
publishData(); delay(500); if
```

```
(!client.loop())
```

```
{ mqttConnect();
```

```
} }
```

```
void
```

```
wifiConnect() {  
Serial.print("Connecting to "); Serial.print("Wifi");
```

```
WiFi.begin("Wokwi-GUEST", "", 6); while
```

```
(WiFi.status() != WL_CONNECTED) { delay(500);
```

```
Serial.print(".");
```

```
}
```

```
Serial.print("WiFi connected, IP address: ");
```

```

Serial.println(WiFi.localIP());

} void mqttConnect() { if
(!client.connected()) {

Serial.print("Reconnecting MQTT client to ");

Serial.println(server); while (!client.connect(clientId, authMethod,
token)) { Serial.print("."); delay(500); } initManagedDevice();

Serial.println(); } } void
initManagedDevice() { if
(client.subscribe(topic)) {

// Serial.println(client.subscribe(topic)); Serial.println("IBM subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

} } void publishData() {

digitalWrite(trigpin,LOW);

digitalWrite(trigpin,HIGH);

delayMicroseconds(10);

digitalWrite(trigpin,LOW);

duration=pulseIn(echopin,HIGH);

dist=duration*speed/2; if(dist<100){

String payload = "{\"Alert Distance\":\""; payload
+= dist; payload += "\"}"; Serial.print("\n");

Serial.print("Sending payload: ");

Serial.println(payload); if (client.publish(publishTopic,
(char*) payload.c_str())) {

Serial.println("Publish OK");

} } if(dist>100){

```

```
String payload = "{\"Distance\":";
payload += dist; payload += "}";

Serial.print("\n");

Serial.print("Sending payload: ");

Serial.println(payload); if(client.publish(publishTopic,
(char*) payload.c_str())) {

Serial.println("Publish OK");

}else

{

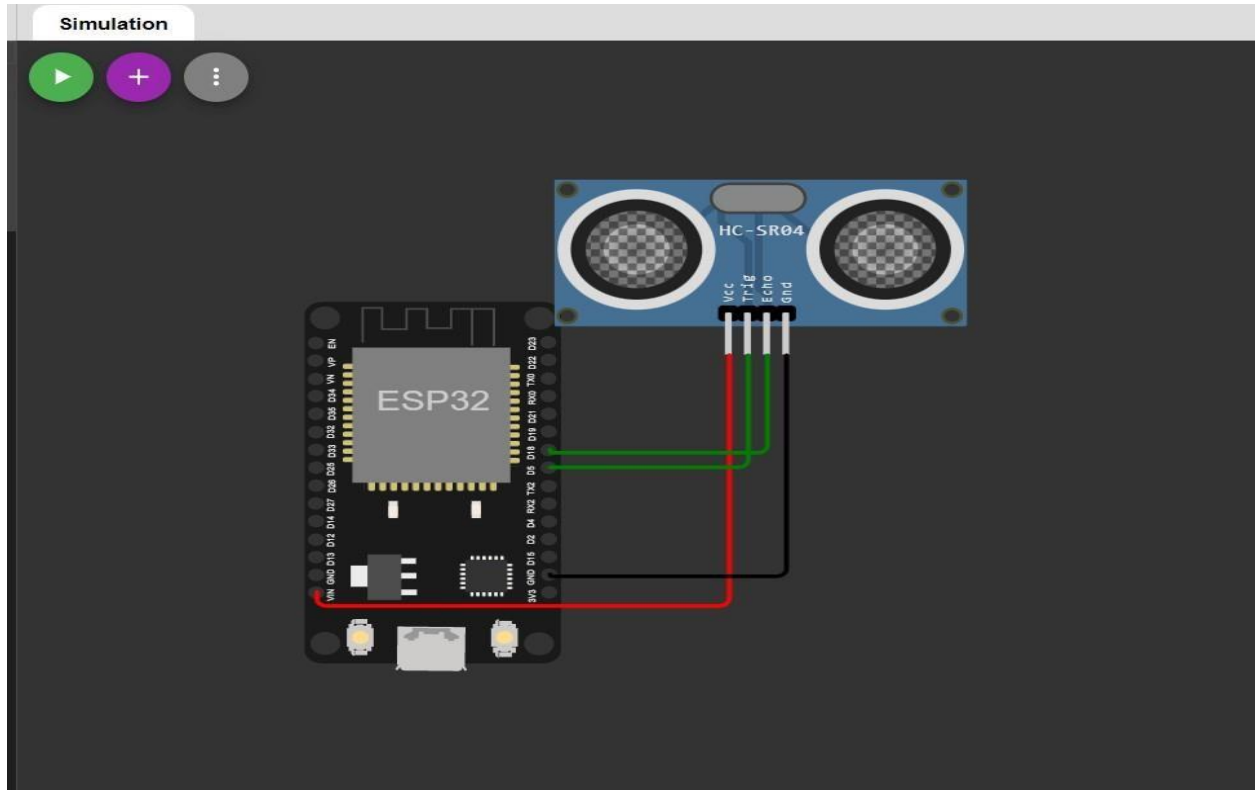
Serial.println("Publish FAILED");

}

}

}
```

## Connections:



## Output:

IBM IBM-EPBL/IBM-Project-17 Resource list - IBM Cloud Ultra sonic sensor copy - \ x

wokwi.com/projects/347425070237352532

WOKWI SAVE SHARE Ultra sonic sensor copy Docs SIGN IN

sketch.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 WiFiClient wificlient;
6
7 #define ORG "q26y5w"
8 #define DEVICE_TYPE "TestDeviceType"
9 #define DEVICE_ID "2022"
10 #define TOKEN "uu(rZQw9292EfSFGDX"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.co
14 char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wificlient);
20 void publishData();
21
22 const int trigpin=5;

```

Simulation

Sending payload:  
{"Name": "point2", "Latitude": "14.167589", "Longitude": "80.248510", "Icon": "fa-trash-o", "FillPercent": 0}  
Publish OK

Sending payload:

26°C 12:44 AM 05-11-2022

## Output:(IBM Cloud)

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes the IBM logo and several open tabs: 'IBM-EPBL/IBM-Pro', 'Service Details - IB', 'IBM Watson IoT Pl', and 'Ultra sonic sensor'. The browser address bar shows the URL 'q26y5w.internetofthings.ibmcloud.com/dashboard/devices/browse'. The dashboard header identifies the user as '310819106054@smartinternz.com' with ID 'q26y5w'. The main content area is titled 'Browse' and features a sidebar with icons for various IoT functions. The central panel displays a message: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this message is a table with four columns: 'Event', 'Value', 'Format', and 'Last Received'. The table contains five rows of data, all with the event name 'event\_test' and a JSON value representing temperature and humidity. The format for all entries is 'json', and the last received time for all is 'a few seconds ago'. At the bottom of the dashboard, a status box indicates '1 Simulation running'.

Event	Value	Format	Last Received
event_test	{"temp":96,"hum":8}	json	a few seconds ago
event_test	{"temp":71,"hum":21}	json	a few seconds ago
event_test	{"temp":83,"hum":10}	json	a few seconds ago
event_test	{"temp":56,"hum":52}	json	a few seconds ago
event_test	{"temp":48,"hum":1}	json	a few seconds ago

1 Simulation running

**Link : <https://wokwi.com/projects/347425070237352532>**