

Date	20 October 2022
Student Name	ARSHA VARSHINEE.V
Student Register Number/Roll Number	211419106034/2019PECEC115
Maximum Marks	2 Marks

ASSIGNMENT-4

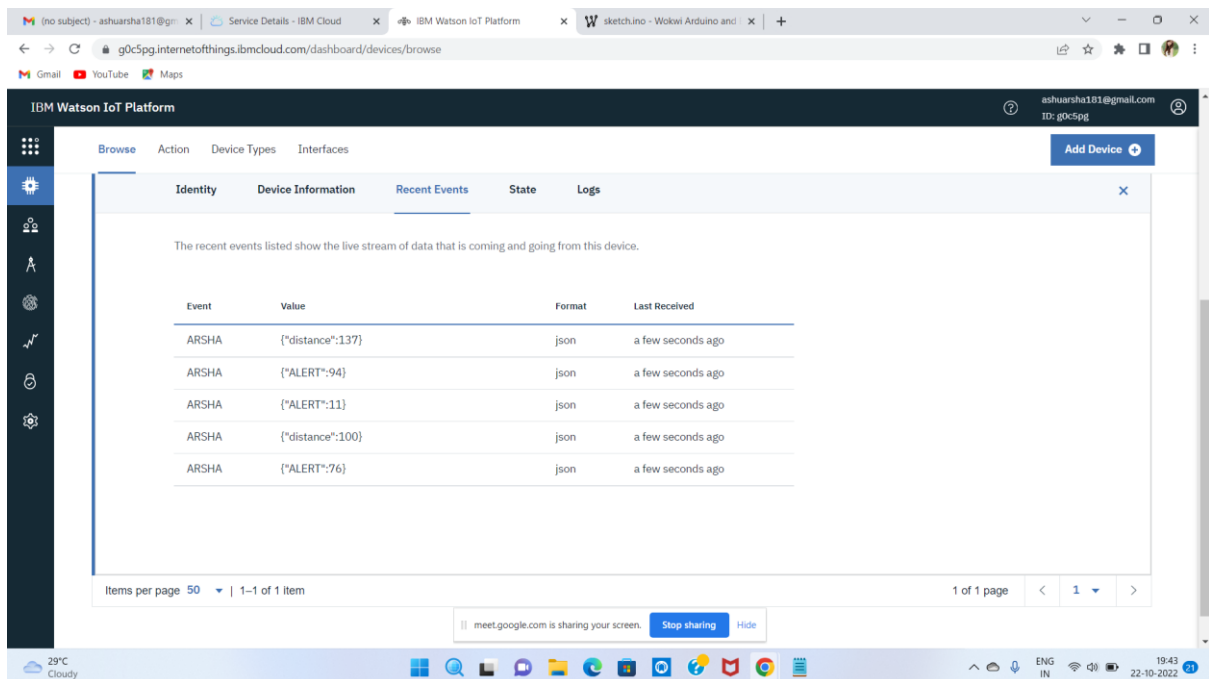
QUESTION:

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 documents with wokwi share link and images of ibm cloud.

WOKWI LINK:

[https://wokwi.com/projects/346227912198849107\](https://wokwi.com/projects/346227912198849107)

IBM CLOUD(DEVICE RECENT EVENTS):



The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a table of events for the device 'ARSHA'. The table has columns for Event, Value, Format, and Last Received. The events are as follows:

Event	Value	Format	Last Received
ARSHA	{"distance":137}	json	a few seconds ago
ARSHA	{"ALERT":94}	json	a few seconds ago
ARSHA	{"ALERT":111}	json	a few seconds ago
ARSHA	{"distance":100}	json	a few seconds ago
ARSHA	{"ALERT":76}	json	a few seconds ago

The dashboard also shows a sidebar with navigation options and a top bar with user information. A meeting sharing bar is visible at the bottom of the screen.

WOKWI SIMULATION:

The screenshot shows the Wokwi web interface. On the left is the 'sketch.ino' editor with the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "g0c5pg"
5 #define DEVICE_TYPE "temp"
6 #define DEVICE_ID "temp123"
7 #define TOKEN "xV5zpvJ?n411)w2@k8"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/ARSHA/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:ORG:"DEVICE_TYPE:"DEVICE_ID";
16
17 WiFiClient wifiClient;
18 PubSubClient client(server, 1883, callback, wifiClient);
19
20 #define ECHO_PIN 12
21 #define TRIG_PIN 13
22 #define led 14
23
24 void setup() {
25   // put your setup code here, to run once:
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(TRIG_PIN, OUTPUT);
29   pinMode(ECHO_PIN, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33 float readDistanceCM() {
```

On the right is the 'Simulation' window showing a circuit diagram of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor and an LED. The simulation log shows the following output:

```
publish ok
Measured distance: 106.00
Sending payload:{"distance":106.00}
publish ok
Measured distance: 60.00
Sending payload:{"ALERT":60.00}
publish ok
```

The screenshot shows the Wokwi web interface. On the left is the 'sketch.ino' editor with the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "g0c5pg"
5 #define DEVICE_TYPE "temp"
6 #define DEVICE_ID "temp123"
7 #define TOKEN "xV5zpvJ?n411)w2@k8"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/ARSHA/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:ORG:"DEVICE_TYPE:"DEVICE_ID";
16
17 WiFiClient wifiClient;
18 PubSubClient client(server, 1883, callback, wifiClient);
19
20 #define ECHO_PIN 12
21 #define TRIG_PIN 13
22 #define led 14
23
24 void setup() {
25   // put your setup code here, to run once:
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(TRIG_PIN, OUTPUT);
29   pinMode(ECHO_PIN, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33 float readDistanceCM() {
```

On the right is the 'Simulation' window showing a circuit diagram of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor and an LED. The simulation log shows the following output:

```
publish ok
Measured distance: 65.00
Sending payload:{"ALERT":65.00}
publish ok
Measured distance: 119.00
Sending payload:{"distance":119.00}
publish ok
```