

## ASSIGNMENT-4

### QUESTION

Write code and connections in wowki for the ultrasonic sensor. Whenever the distance is less than 100 cms send a **alert to the IBM cloud** and display in the device recent events.

Upload document with wokwi share link and images of IBM cloud.

### CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
#define ORG "92zbfc"
#define DEVICE_TYPE "esp32"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  wificonnect();
  mqttconnect();
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * SOUND_SPEED/2;
  Serial.print("Distance (cm): ");
```

```

Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(float dist) {
mqttconnect();
String payload = "{\"Distance\":\"";
payload += dist;
payload += "\",\"ALERT!!\":\"\"Distance less than 100cms\"";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
} else {
Serial.println("Publish failed");
}
}
void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
}

```

```

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println(subscribetopic);
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (int i = 0; i < payloadLength; i++)
{
data3 += (char)payload[i];
}
Serial.println("data: " + data3);
data3="";
}

```

Wokwi Link: <https://wokwi.com/projects/348283018223288915>

## OUTPUT & SIMULATION:

The screenshot displays the Wokwi online IDE interface. On the left, the 'sketch.ino' file contains the following code:

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
4 #define ORG "92zbfc"
5 #define DEVICE_TYPE "esp32"
6 #define DEVICE_ID "12345"
7 #define TOKEN "12345678"
8 String data3;
9 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
10 char publishTopic[] = "iot-2/evt/Data/fmt/json";
11 char subscribetopic[] = "iot-2/cmd/test/fmt/String";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15 WiFiClient wificlient;
16 PubSubClient client(server, 1883, callback, wificlient);
17 const int trigPin = 5;
18 const int echoPin = 18;
19 #define SOUND_SPEED 0.034
20 long duration;
21 float distance;
22 void setup() {
23   Serial.begin(115200);
24   pinMode(trigPin, OUTPUT);
25   pinMode(echoPin, INPUT);
26   wificlient.connect();
27   mqttconnect();
28 }
29 void loop()

```

On the right, the 'Simulation' window shows a visual representation of the ESP32 board and the HC-SR04 ultrasonic sensor. A dialog box titled 'Editing Ultrasonic Distance Sensor' indicates a distance of 94cm. The console output shows the following sequence of events:

```

ALERT!!
Sending payload: {"Distance":93.94,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 93.94
ALERT!!
Sending payload: {"Distance":93.94,"ALERT!!":"Distance less than 100cms"}
Publish ok

```

## Alert to IBM cloud:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various IoT functions. The main content area is titled 'pybd31.internetofthings.ibmcloud.com/dashboard/devices/browse'. It features a tabbed interface with 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of live data streams. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. It lists five events, all labeled 'event\_1', with values in JSON format: {"ultrasonic":3}, {"ultrasonic":91}, {"ultrasonic":93}, {"ultrasonic":74}, and {"ultrasonic":16}. Each event is received in 'json' format and is noted as 'a few seconds ago'. A status box at the bottom right indicates '2 Simulations running'. The bottom of the image shows a Windows taskbar with the search bar, application icons, and system tray information including temperature (26°C), weather (Cloudy), and date/time (17:56, 14-11-2022).

IBM Watson IoT Platform

pybd31.internetofthings.ibmcloud.com/dashboard/devices/browse

Identity Device Information **Recent Events** State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"ultrasonic":3}	json	a few seconds ago
event_1	{"ultrasonic":91}	json	a few seconds ago
event_1	{"ultrasonic":93}	json	a few seconds ago
event_1	{"ultrasonic":74}	json	a few seconds ago
event_1	{"ultrasonic":16}	json	a few seconds ago

2 Simulations running

Type here to search

26°C Cloudy 17:56 14-11-2022