

## Assignment - 4

Assignment Date	17 October 2022
Student Name	Dharaneesh KT
Student Roll Number	737819CSR033
Maximum Marks	2 Marks

### Question-1:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud

### CODE 1 :

```
#include <WiFi.h> #include
<PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "cbseji"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
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)
{

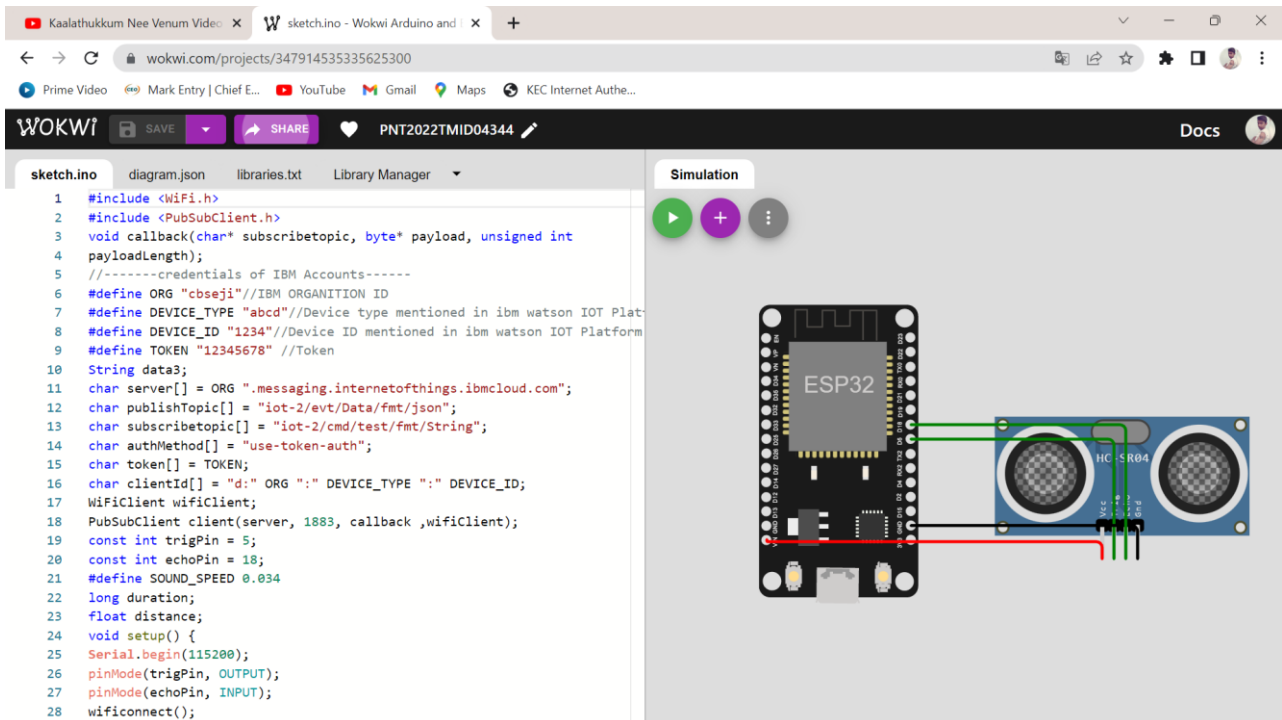
```

```
Serial.print("callback invoked for topic: ");  
Serial.println(subscribetopic); for (int i = 0;  
i    <    payloadLength;    i++)    {  
//Serial.print((char)payload[i]);    data3    +=  
(char)payload[i];  
}  
Serial.println("data: "+ data3); data3="";  
}
```

## Wokwi Link :

<https://wokwi.com/projects/347914535335625300>

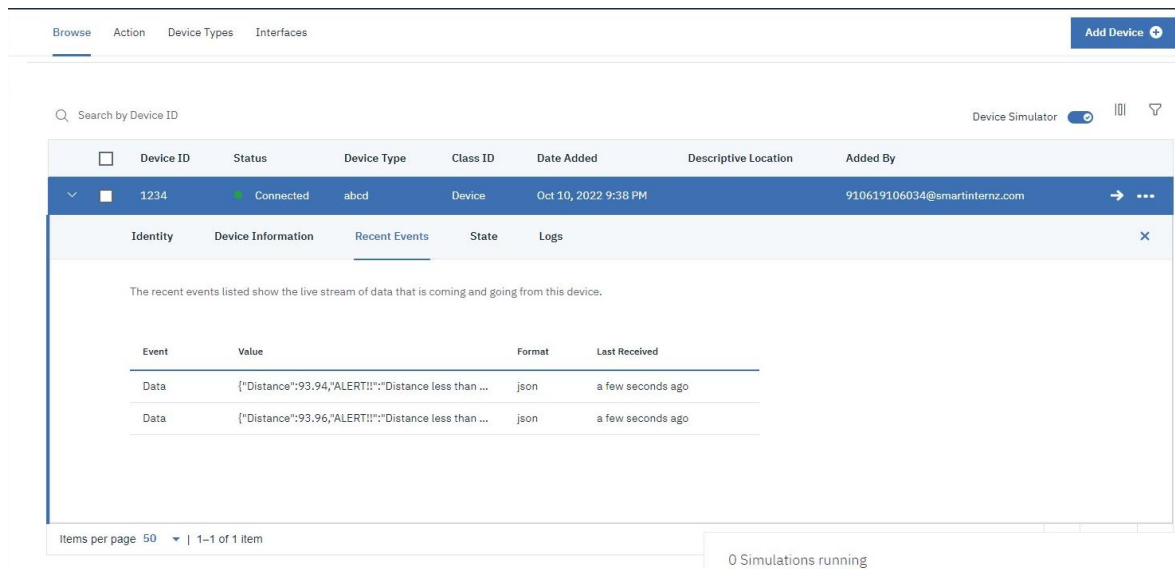
## Output and Simulation :



The screenshot displays the Wokwi web interface. On the left, the 'sketch.ino' file is open, showing an Arduino sketch that includes the `WiFi.h` and `PubSubClient.h` libraries. The sketch defines an ESP32 device type 'abcd' with a device ID '1234' and a token '12345678'. It sets up a WiFi connection and a PubSubClient for an IBM Watson IoT Platform. The sketch also defines a trig pin (5) and an echo pin (18) for an ultrasonic sensor. The `setup()` function initializes the serial port and pin modes. The `loop()` function (partially visible) would handle the sensor readings and cloud communication.

On the right, the 'Simulation' tab is active, showing a 3D model of the ESP32 board connected to an HC-SR04 ultrasonic sensor module. The sensor is connected to the ESP32's trig pin (5) and echo pin (18).

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



The screenshot shows the Wokwi device simulator interface. At the top, there are tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. Below the search bar, a table lists the devices. The first device is '1234', which is 'Connected' and has a device type of 'abcd'. The table also shows the date added (Oct 10, 2022 9:38 PM) and the user who added it (910619106034@smartintemz.com).

Below the table, there is a section for 'Recent Events'. It contains a table with the following data:

Event	Value	Format	Last Received
Data	{\"Distance\":93.94,\"ALERT!!!\":\"Distance less than ...	json	a few seconds ago
Data	{\"Distance\":93.96,\"ALERT!!!\":\"Distance less than ...	json	a few seconds ago

At the bottom of the interface, it says 'Items per page 50' and '1-1 of 1 item'. On the right side, it says '0 Simulations running'.