

## IBM ASSIGNMENT-4

```
#include <WiFi.h>
#include <PubSubClient.h>

void callback(char* subscribetopic,byte* payload, unsigned int
payloadLength);

#define ORG "lizz0n"
#define DEVICE_TYPE "IOT"
#define DEVICE_ID "1"
#define TOKEN "12345678"

String data3;

char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/distance/fmt/json";
char subscribeTopic[]="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);

#define ECHO_PIN 2
#define TRIG_PIN 4
#define led 5

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(led, OUTPUT);
```

```

    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    wificonnect();
    mqttconnect();
}

float readDistanceCM() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    int duration=random(1,200);
    //Serial.println(duration);
    //duration = pulseIn(ECHO_PIN, HIGH);
    return duration ;
    //Serial.println(duration);

}

void loop() {
    float distance = readDistanceCM();
    //Serial.println(distance);

    bool isNearby = distance < 100;
    digitalWrite(led, isNearby);

    Serial.print("Measured distance: ");
    Serial.println(distance);
    if(distance<100){
        PublishData2(distance);
    }
}

```

```

    }else{
        PublishData1(distance);

    }
    //PublishData(distance);
    delay(1000);
    if(!client.loop()){
        mqttconnect();
    }

    //delay(2000);
}

void PublishData1(float dist){
    mqttconnect();
    String payload= "{\"distance\":";
    payload += dist;
    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 PublishData2(float dist){
    mqttconnect();

```

```

String payload= "{\\"ALERT\\":\"";
payload += dist;
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 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);
    if(data3=="lighton"){
        Serial.println(data3);
    }
}

```

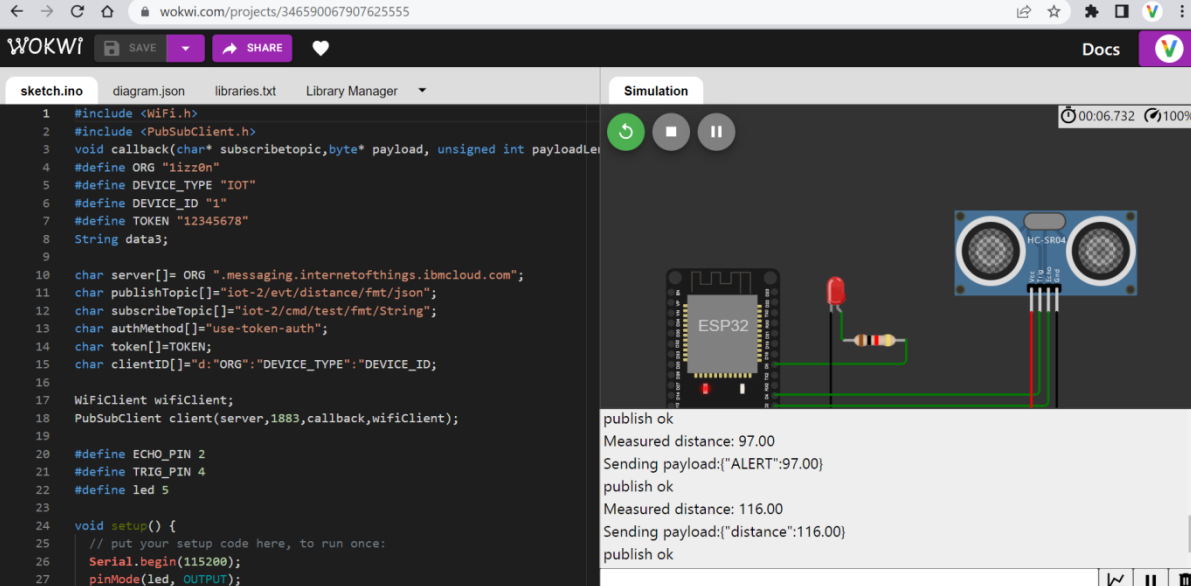
```

        digitalWrite(led,HIGH);
    }else{
        Serial.println(data3);
        digitalWrite(led,LOW);
    }
    data3="";
}

```

**WOKWI PROJECT LINK:** <https://wokwi.com/projects/346590067907625555>

Normal Case:



The screenshot displays the Wokwi web IDE interface. On the left, the 'sketch.ino' file is open, showing the following code:

```

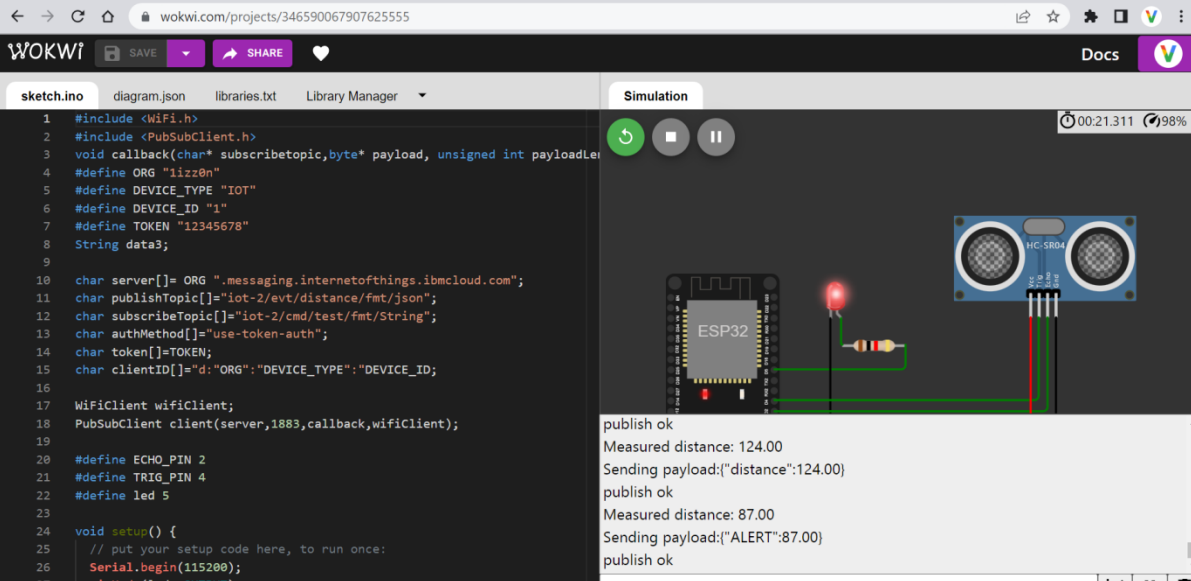
1 #include <Wifi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribtopic,byte* payload,unsigned int payloadLen)
4 #define ORG "lizz0n"
5 #define DEVICE_TYPE "IOT"
6 #define DEVICE_ID "1"
7 #define TOKEN "12345678"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/distance/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 2
21 #define TRIG_PIN 4
22 #define led 5
23
24 void setup() {
25 // put your setup code here, to run once:
26 Serial.begin(115200);
27 pinMode(led, OUTPUT);
28 }
29
30 void loop() {
31   // put your main code here, to run repeatedly:
32   // ...
33 }

```

On the right, the 'Simulation' window shows a visual representation of the ESP32 board, the HC-SR04 sensor, and an LED. The console output indicates the following sequence of events:

- publish ok
- Measured distance: 97.00
- Sending payload:["ALERT":97.00]
- publish ok
- Measured distance: 116.00
- Sending payload:["distance":116.00]
- publish ok

Alert Case:



The screenshot displays the Wokwi web IDE interface. On the left, the 'sketch.ino' file is open, showing the same code as in the normal case. On the right, the 'Simulation' window shows the same visual representation of the ESP32 board, the HC-SR04 sensor, and an LED. The console output indicates the following sequence of events:

- publish ok
- Measured distance: 124.00
- Sending payload:["distance":124.00]
- publish ok
- Measured distance: 87.00
- Sending payload:["ALERT":87.00]
- publish ok

Cloud storage:

IBM Watson IoT Platform

19110099@hiet.ac.in  
ID: 1izz0n

Browse

Action

Device Types

Interfaces

Add Device

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     |
|----------|------------------|--------|-------------------|
| distance | {"distance":114} | json   | a few seconds ago |
| distance | {"distance":140} | json   | a few seconds ago |
| distance | {"ALERT":35}     | json   | a few seconds ago |
| distance | {"distance":105} | json   | a few seconds ago |
| distance | {"ALERT":50}     | json   | a few seconds ago |