

Assignment -4

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT

#define ECHO_PIN 5
#define TRIG_PIN 4

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

//-----credentials of IBM Accounts-----

#define ORG "a4aah7"//IBM ORGANITION ID
#define DEVICE_TYPE "Esp32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "861895"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of
event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT
command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, NULL ,wifiClient); //calling the predefined
client id by passing parameter like server id,portand wificredential

void setup() // configureing the ESP32
{
    Serial.begin(115200);
    pinMode(LED_BUILTIN, 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 = pulseIn(ECHO_PIN, HIGH);
    return duration * 0.034 / 2;
}

void loop() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    int duration = pulseIn(ECHO_PIN, HIGH);
    float distance = duration * 0.034 / 2;

    Serial.print("Distance: ");
    Serial.println(distance);

    if(distance <100)
        Serial.println("Alert");
    else if(distance >100)
        Serial.println("Distance is maintained");
    PublishData(distance);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

/*.....retrieving to
Cloud. .... */

void PublishData(float distance) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"Distance\":";
    payload += distance;
    payload += "," " \"Status\":";
    payload += "\"Alert\"";
    payload += "}";

    Serial.print("Sending payload: ");
    Serial.println(payload);
}

```

```

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it sucessfully upload data on the cloud
        then it will print publish ok in Serial monitor or else it will print publish
        failed
    } 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() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
    the connection
    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");
    }
}
}

```

Device Recent Events Screenshot

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area shows the details for a device with ID '861895', which is 'Disconnected' and of type 'raspberrypi'. The 'Recent Events' tab is selected, showing a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are listed as 'event_1' with values like '{"DISTANCE":17.86,"STATUS":"alert"}' in 'json' format, received 'a few seconds ago'. A status message at the bottom indicates '1 Simulation running'.

IBM Watson IoT Platform

ac19uec143@smartintents.com
ID: e4eah7

861895 Disconnected raspberrypi Device 3 Oct 2022 19:10

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	{"DISTANCE":17.86,"STATUS":"alert"}	json	a few seconds ago
event_1	{"DISTANCE":70.59,"STATUS":"alert"}	json	a few seconds ago
event_1	{"DISTANCE":58.62,"STATUS":"alert"}	json	a few seconds ago
event_1	{"DISTANCE":89.08,"STATUS":"alert"}	json	a few seconds ago
event_1	{"DISTANCE":32.59,"STATUS":"alert"}	json	a few seconds ago

1 Simulation running