

ASSIGNMENT 04

NAME : JAYAPREETHI U

ROLL NUMBER : CITC1904081

TEAM ID : PNT2022TMID52822

WOKWI CODE:

Link : <https://wokwi.com/projects/348109588842676818>

```
#include <WiFi.h>//library for wifi
#include <WiFiClient.h>
#include <PubSubClient.h>//library for MQTT
// creating the instance by passing pin and typr of dht connected
float distance;
#define sound_speed 0.034
int trigpin=18;
int echopin=19;
int led=5;
int LED=9;
long duration;
String message;// creating the instance by passing pin and typr of dht
connected

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

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

#define ORG "9w42t3"
#define DEVICE_TYPE "Wokwi_Bavi"
#define DEVICE_ID "123456789"
#define TOKEN "+59ElkeP8c2d82)X?U" //Token
String data3;
float h, t;

//----- 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, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential
void setup()// configureing the ESP32
{
    Serial.begin(115200);
    pinMode(trigpin,OUTPUT);
    pinMode(echopin,INPUT);
    pinMode(led,OUTPUT);
    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop()// Recursive Function
{

digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delay(1000);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
distance=duration*sound_speed/2;
Serial.println("distance"+String(distance)+"cm");
    if(distance<100)
    {
        message="Alert";
        digitalWrite(led,HIGH);
    } else
{
    message="No problem";
    digitalWrite(led,LOW);
}
    delay(1000);
    PublishData(distance,message);
    // if (!client.loop()) {
    //     mqttconnect();
    // }
}

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

void PublishData(float d, String a) {
    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 += d; payload += "}";
payload += "," "{\"message\":";
payload += a;
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");
    }
}

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

```

IBM WATSON IoT PLATFORM:

IBM Watson IoT Platform

1904072ece@cit.edu.in
ID: 9w42t3

Browse

Action

Device Types

Interfaces

Search by Device ID

Device Simulator

Device ID

Status

Device Type

Class ID

Date Added

>

12345

Disconnected

NodeMCU

Device

Nov 11, 2022 11:44 PM

▼

123456789

Disconnected

Wokwi_Bavi

Device

Nov 12, 2022 10:43 AM

→ ...

Identity

Device Information

Recent Events

State

Logs

Device ID

123456789

Device Type

Wokwi_Bavi

Date Added

Nov 12, 2022 10:43 AM

Added By

1904072ece@cit.edu.in

Connection Status

Disconnected

Items per page 50

1–2 of 2 items

1 of 1 page

<

1

>

IBM Watson IoT Platform

1904072ece@cit.edu.in
ID: 9w42t3

Browse

Action

Device Types

Interfaces

Search by Device ID

Device Simulator

Device ID

Status

Device Type

Class ID

Date Added

>

12345

Disconnected

NodeMCU

Device

Nov 11, 2022 11:44 PM

>

123456789

Disconnected

Wokwi_Bavi

Device

Nov 12, 2022 10:43 AM

>

1904072

Connected

citibavi

Device

Nov 12, 2022 1:10 PM

Items per page 50

1–3 of 3 items

1 of 1 page

<

1

>

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

0 Simulations running

OUTPUT SNIPS:

WOKWI SAVE SHARE ASSIGNMENT 04 ibm Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <WiFiClient.h>
3 #include <PubSubClient.h> //library for MQTT
4 // creating the instance by passing pin and type of dht connected
5 float distance;
6 #define sound_speed 0.034
7 int trigpin=18;
8 int echopin=19;
9 int led=5;
10 int LED=9;
11 long duration;
12 String message; // creating the instance by passing pin and type of dht connected
13
14 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
15
16 //-----credentials of IBM Accounts-----
17
18 #define ORG "9w42t3" //IBM ORGANIZATION ID
19 #define DEVICE_TYPE "citibavi" //Device type mentioned in ibm watson IOT Platform
20 #define DEVICE_ID "1984072" //Device ID mentioned in ibm watson IOT Platform
21 #define TOKEN "g0Z?m?FsdX9XHCW!L" //Token
22 String data3;
23 float h, t;
24
25 //----- Customise the above values -----
26
27 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
28 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
29 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
```

Simulation

00:50.837 5%

Publish ok
distance399.94cm
Sending payload: {"distance":399.94}, {"message":No problem}
Publish ok
distance399.96cm
Sending payload: {"distance":399.96}, {"message":No problem}
Publish ok

WOKWI SAVE SHARE ASSIGNMENT 04 ibm Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <WiFiClient.h>
3 #include <PubSubClient.h> //library for MQTT
4 // creating the instance by passing pin and type of dht connected
5 float distance;
6 #define sound_speed 0.034
7 int trigpin=18;
8 int echopin=19;
9 int led=5;
10 int LED=9;
11 long duration;
12 String message; // creating the instance by passing pin and type of dht connected
13
14 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
15
16 //-----credentials of IBM Accounts-----
17
18 #define ORG "9w42t3" //IBM ORGANIZATION ID
19 #define DEVICE_TYPE "citibavi" //Device type mentioned in ibm watson IOT Platform
20 #define DEVICE_ID "1984072" //Device ID mentioned in ibm watson IOT Platform
21 #define TOKEN "g0Z?m?FsdX9XHCW!L" //Token
22 String data3;
23 float h, t;
24
25 //----- Customise the above values -----
26
27 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
28 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
29 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
```

Simulation

01:23.546 86%

Publish ok
distance244.95cm
Sending payload: {"distance":244.95}, {"message":No problem}
Publish ok
distance244.95cm
Sending payload: {"distance":244.95}, {"message":No problem}
Publish ok

