

## Week 2 assignment

ROSHAN KUMAR L

VIT CHENNAI

**Task** - In wokwi connect push button and upload 0 and 1 to ibm cloud

**Code –**

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

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

#define ORG "hakody" //IBM ORGANISATION ID
#define DEVICE_TYPE "wokwi" //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;
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, port and
wificredential

void setup() {
  pinMode(32, INPUT);
  Serial.begin(115200);
  wificlient();
  mqttconnect();
}

void loop() {
  int buttonstate = digitalRead(32);
```

```

    Serial.print("Button State = ");
    Serial.println(buttonstate);
    PublishData(buttonstate);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

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

void PublishData(bool buttonstate) {
    mqttconnect();//function call for connecting to ibm
    String payload = "{\"Button State\":\"";
    payload += buttonstate;
    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() //function definition for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

```

```

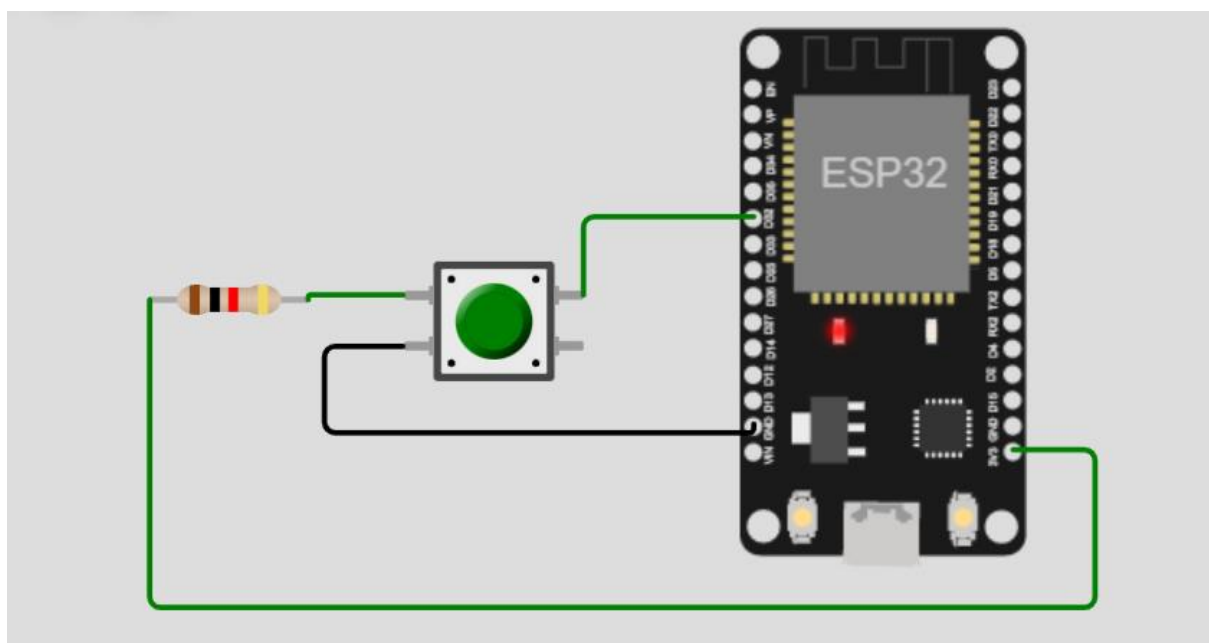
    WiFi.begin("Wokwi-GUEST", "", 6); //passing wifi credentials to establish
connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

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);
}

```

## SCHEMATIC



## OUTPUT

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

```
75 }
76
77 void wificonnect() //function definition for wificonnect
78 {
79   Serial.println();
80   Serial.print("Connecting to ");
81
82   WiFi.begin("Wokwi-GUEST", "", 6); //passing wifi credentials to establish connection
83   while (WiFi.status() != WL_CONNECTED) {
84     delay(500);
85     Serial.print(".");
86   }
87   Serial.println("");
88   Serial.println("WiFi connected");
89   Serial.println("IP address: ");
90   Serial.println(WiFi.localIP());
91 }
92
93 void initManagedDevice() {
94   if (client.subscribe(subscribetopic)) {
95     Serial.println((subscribetopic));
96     Serial.println("subscribe to cmd OK");
97   } else {
98     Serial.println("subscribe to cmd FAILED");
99   }
100 }
101
102 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength)
103 {
104   Serial.print("callback invoked for topic: ");
105   Serial.println(subscribetopic);
106 }
```

On the right, the 'Simulation' window shows a visual representation of the ESP32 microcontroller connected to a green LED. The simulation is running, as indicated by the play button icon and the timer showing 01:31.765. Below the simulation, the serial output window displays the following text:

```
Connecting to ....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to hakody.messaging.internetofthings.ibmcloud.com
.....
```