

**Assignment: 2**

**Company: Smart-Internz**

**Domain: Internet of Things**

**Date: 27-5-2023**

**Name: H. Shyam**

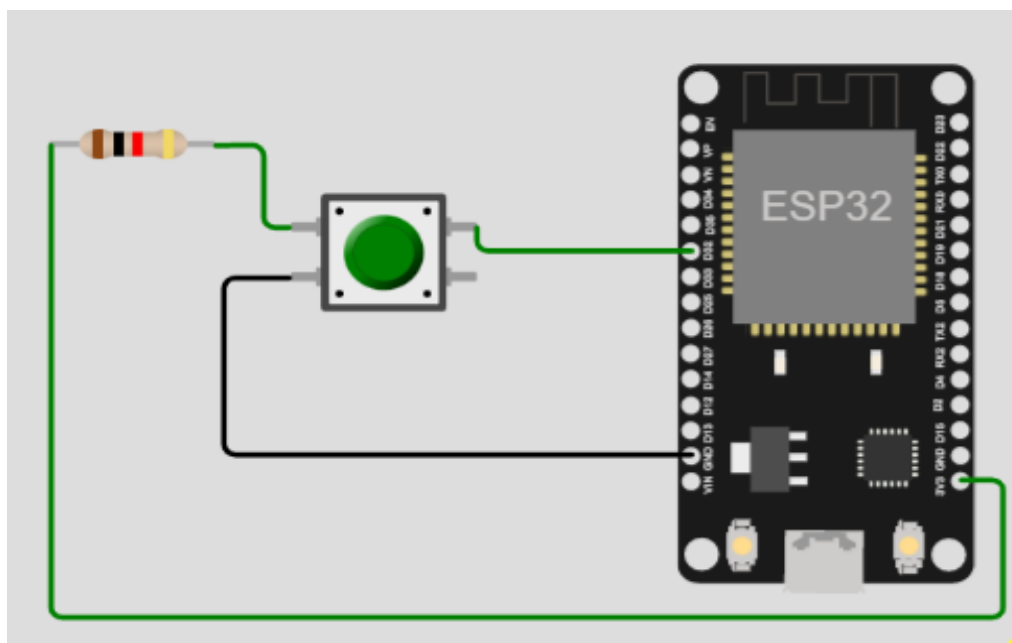
**Reg. no.: 20BEC1055**

**Task:** In Wokwi simulator, connect push button & upload 0 and 1 to IBM cloud.

**Wokwi Link:**

<https://wokwi.com/projects/365969714855667713>

**Circuit:**



Add the PubSubClient library to the Library manager pane & run the code below



## CODE:

```
1  #include <WiFi.h> //library for wifi
2  #include <PubSubClient.h> //library for MQTT
3  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
4
5  //-----credentials of IBM Accounts-----
6
7  #define ORG "tiwmk9" //IBM ORGANITION ID
8  #define DEVICE_TYPE "SHWokwi" //Device type mentioned in ibm watson IOT Platform
9  #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
10 #define TOKEN "12345678" //Token
11 String data3;
12 float h, t;
13
14 //----- Customise the above values -----
15 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
16 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and format in which data to be send
17 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
18 char authMethod[] = "use-token-auth"; // authentication method
19 char token[] = TOKEN;
20 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
21
22 //-----
23 WiFiClient wificlient; // creating the instance for wificlient
24 PubSubClient client(server, 1883, callback, wificlient);
25 //calling the predefined client id by passing parameter like server id, port and wificredential
26
27 void setup() {
28     pinMode(32, INPUT);
29     Serial.begin(115200);
30     wificlient.connect();
31     mqttconnect();
32 }
33
34 void loop() {
35     int buttonstate = digitalRead(32);
36     Serial.print("Button State = ");
37     Serial.println(buttonstate);
38     PublishData(buttonstate);
39     delay(1000);
40     if (!client.loop()) {
41         mqttconnect();
42     }
43 }
```

```

45  /*.....retrieving to Cloud.....*/
46
47  void PublishData(bool buttonstate) {
48      mqttconnect();//function call for connecting to ibm
49      String payload = "{\"Button State\":\"";
50      payload += buttonstate;
51      payload += "\"}";
52
53      Serial.print("Sending payload: ");
54      Serial.println(payload);
55
56      if (client.publish(publishTopic, (char*) payload.c_str())) {
57          Serial.println("Publish ok");
58      } else {
59          Serial.println("Publish failed");
60      }
61  }
62
63
64  void mqttconnect() {
65      if (!client.connected()) {
66          Serial.print("Reconnecting client to ");
67          Serial.println(server);
68          while (!client.connect(clientId, authMethod, token)) {
69              Serial.print(".");
70              delay(500);
71          }
72
73          initManagedDevice();
74          Serial.println();
75      }
76
77  void wificonnect() //function defination for wificonnect
78  {
79      Serial.println();
80      Serial.print("Connecting to ");
81      WiFi.begin("Wokwi-GUEST", "", 6);//passing wifi credentials to establish connection
82      while (WiFi.status() != WL_CONNECTED) {
83          delay(500);
84          Serial.print(".");
85      }
86      Serial.println("");
87      Serial.println("WiFi connected");
88      Serial.println("IP address: ");
89      Serial.println(WiFi.localIP());
90  }
91

```

```

92 void initManagedDevice() {
93     if (client.subscribe(subscribetopic)) {
94         Serial.println((subscribetopic));
95         Serial.println("subscribe to cmd OK");
96     } else {
97         Serial.println("subscribe to cmd FAILED");
98     }
99 }
100
101 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
102 {
103     Serial.print("callback invoked for topic: ");
104     Serial.println(subscribetopic);
105 }

```

## **OUTPUT:**

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a search bar and a table of devices. One device, '1234', is highlighted with a status of 'Connected' and a device type of 'SHWokwi'. Below the device list, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events with columns for 'Event', 'Value', 'Format', and 'Last Received'.

Event	Value	Format	Last Received
Data	{"Button State":1}	json	a few seconds ago
Data	{"Button State":1}	json	a few seconds ago
Data	{"Button State":0}	json	a few seconds ago
Data	{"Button State":0}	json	a few seconds ago

**RESULT:** Given task was carried out successfully.