

SMARTBRIDGE EXTERNSHIP

Internet Of Things

Date: 27th May 2023

Jyoti Prakash Behura
20BCE7355
VIT-AP

Assignment 2: In wokwi connect push button and upload 0 and 1 to ibm cloud

Code:

sketch.ino

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
#define button 4
#define LED 5
int buttonPin;

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

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

#define ORG "9f8w1x"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//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;

//----- 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() {
  pinMode(buttonPin, INPUT_PULLUP);
  Serial.begin(9600);
  wificonnect();
  mqttconnect();
}

void loop() {
  int buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    Serial.println("Button state: 1");
  } else {
    Serial.println("Button state: 0");
  }

  delay(100);
  if (!client.loop()) {
    mqttconnect();
  } // Adjust delay as needed
}
/*.....retrieving to
Cloud.....*/

// void PublishData() {
//   mqttconnect();
//   String payload = "{\"temp\":";
//   payload += temp;
//   payload += "," "\"Humid\":";
//   payload += humid;
//   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="";
}

```

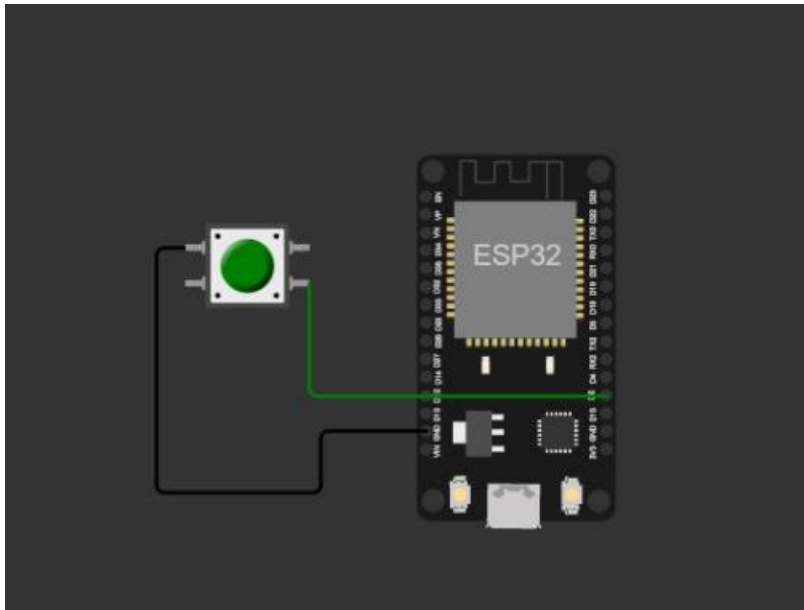
diagram.json

```

{
  "version": 1,
  "author": "JYOTI PRAKASH BEHURA 20BCE7355",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0,
    "attrs": {} },
    {
      "type": "wokwi-pushbutton",
      "id": "btn1",
      "top": 38.73,
      "left": -124.27,
      "attrs": { "color": "green" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "esp:D2", "btn1:2.r", "green", [ "h0" ] ],
    [ "btn1:1.1", "esp:GND.2", "black", [ "h-14.53", "v130", "h87.73", "v-
32.73" ] ]
  ],
  "dependencies": {}
}

```

Diagram:



Outputs:

sketch.ino

```
sketch.ino • diagram.json libraries.txt Library Manager
1  #include <WiFi.h> //library for wifi
2  #include <PubSubClient.h> //library for MQTT
3  #define button 4
4  #define LED 5
5  int buttonPin;
6
7  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
8
9  //-----credentials of IBM Accounts-----
10
11 #define ORG "9f8wlx" //IBM ORGANITION ID
12 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
13 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
14 #define TOKEN "12345678" //Token
15 String data3;
16
17
18 //----- Customise the above values -----
19 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
20 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform
21 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
22 char authMethod[] = "use-token-auth"; // authentication method
23 char token[] = TOKEN;
24 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
25
26
27 //-----
28 WiFiClient wifiClient; // creating the instance for wificlient
29 PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client
30 void setup() {
31     pinMode(buttonPin, INPUT_PULLUP);
32     Serial.begin(9600);
33     wifiConnect();
34     mqttConnect();
35 }
```

```
sketch.ino • diagram.json libraries.txt Library Manager
37 void loop() {
38   int buttonState = digitalRead(buttonPin);
39
40   if (buttonState == HIGH) {
41     Serial.println("Button state: 1");
42   } else {
43     Serial.println("Button state: 0");
44   }
45
46   delay(100);
47   if (!client.loop()) {
48     mqttconnect();
49   } // Adjust delay as needed
50 }
51
52 /*.....retrieving to Cloud.....
53
54 // void PublishData() {
55 //   mqttconnect();
56 //   String payload = "{\"temp\"";
57 //   payload += temp;
58 //   payload += "," + "\"Humid\"";
59 //   payload += humid;
60 //   payload += "\"}";
61 //   if (client.publish(publishTopic, (char*) payload.c_str())) {
62 //     Serial.println("Publish ok");// if it successfully upload data on the cloud then it
63 //   } else {
64 //     Serial.println("Publish failed");
65 //   }
66 // }
67 // }
68
69
```

```
sketch.ino • diagram.json libraries.txt Library Manager
70 void mqttconnect() {
71   if (!client.connected()) {
72     Serial.print("Reconnecting client to ");
73     Serial.println(server);
74     while (!client.connect(clientId, authMethod, token)) {
75       Serial.print(".");
76       delay(500);
77     }
78
79     initManagedDevice();
80     Serial.println();
81   }
82 }
83 void wificonnect() //function definition for wificonnect
84 {
85   Serial.println();
86   Serial.print("Connecting to ");
87
88   WiFi.begin("wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
89   while (WiFi.status() != WL_CONNECTED) {
90     delay(500);
91     Serial.print(".");
92   }
93   Serial.println("");
94   Serial.println("WiFi connected");
95   Serial.println("IP address: ");
96   Serial.println(WiFi.localIP());
97 }
98
```

```

98
99 void initManagedDevice() {
100     if (client.subscribe(subscribetopic)) {
101         Serial.println((subscribetopic));
102         Serial.println("subscribe to cmd OK");
103     } else {
104         Serial.println("subscribe to cmd FAILED");
105     }
106 }
107
108 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
109 {
110
111     Serial.print("callback invoked for topic: ");
112     Serial.println(subscribetopic);
113     for (int i = 0; i < payloadLength; i++) {
114         //Serial.print((char)payload[i]);
115         data3 += (char)payload[i];
116     }
117     Serial.println("data: "+ data3);
118     if(data3=="lighton")
119     {
120         Serial.println(data3);
121         digitalWrite(LED,HIGH);
122     }
123     else
124     {
125         Serial.println(data3);
126         digitalWrite(LED,LOW);
127     }
128     data3="";
129 }
130

```

Diagram.json

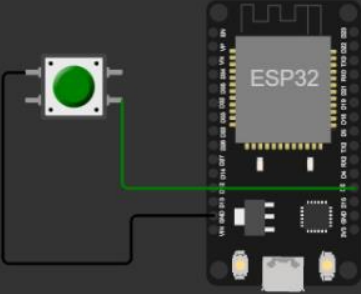
```

sketch.ino • diagram.json  libraries.txt  Library Manager
1  {
2      "version": 1,
3      "author": "JYOTI PRAKASH BEHURA 20BCE7355",
4      "editor": "wokwi",
5      "parts": [
6          { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0, "attrs": {} },
7          {
8              "type": "wokwi-pushbutton",
9              "id": "btn1",
10             "top": 38.73,
11             "left": -124.27,
12             "attrs": { "color": "green" }
13         }
14     ],
15     "connections": [
16         [ "esp:TX0", "$serialMonitor:RX", "", [] ],
17         [ "esp:RX0", "$serialMonitor:TX", "", [] ],
18         [ "esp:D2", "btn1:2.r", "green", [ "h0" ] ],
19         [ "btn1:1.l", "esp:GND.2", "black", [ "h-14.53", "v130", "h87.73", "v-32.73" ] ]
20     ],
21     "dependencies": {}
22 }

```

Output:

Simulation



Button state: 1
Button state: 1
Button state: 1

IBM Watson IoT Platform

Browse

Action

Device Types

Interfaces

Add Device

Delete

1 item selected

Cancel

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	abcd	Device	May 28, 2023 11:28 AM	

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	{\"randomNumber\":35}	json	a few seconds ago
event_1	{\"randomNumber\":86}	json	a minute ago
event_1	{\"randomNumber\":37}	json	3 minutes ago
event_1	{\"randomNumber\":10}	json	4 minutes ago

1 Simulation running

