***SMARTBRIDGE EXTERNSHIP***

***(Internet Of Things)***

**ASSIGNMENT 2**

**Question:** *In wokwi connect push button and upload 0 and 1 to ibm cloud*

**Code:**

sk#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 "x44ini"//IBM ORGANITION 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;

//-------- 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 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": "Shubhankar",

"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" ] ],

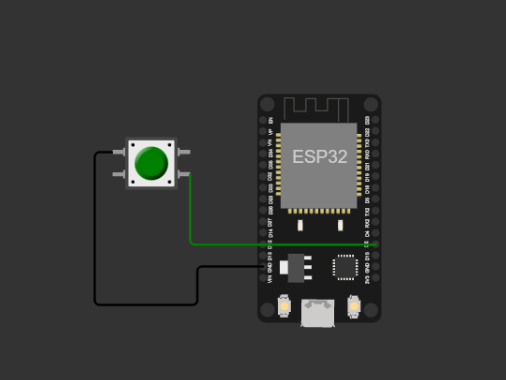
[ "btn1:1.l", "esp:GND.2", "black", [ "h-14.53", "v130", "h87.73", "v-32.73" ] ]

],

"dependencies": {}

}

**Diagram:**



**Outputs:**