

NAME: R. GAUTHAM

REG NO: 20BEC1336

COURSE: Internet of Things (IOT)

WEEK 2 ASSIGNMENT

AIM:

Connect push button values 0 and 1 and upload to IBM Cloud using Wokwi Simulator

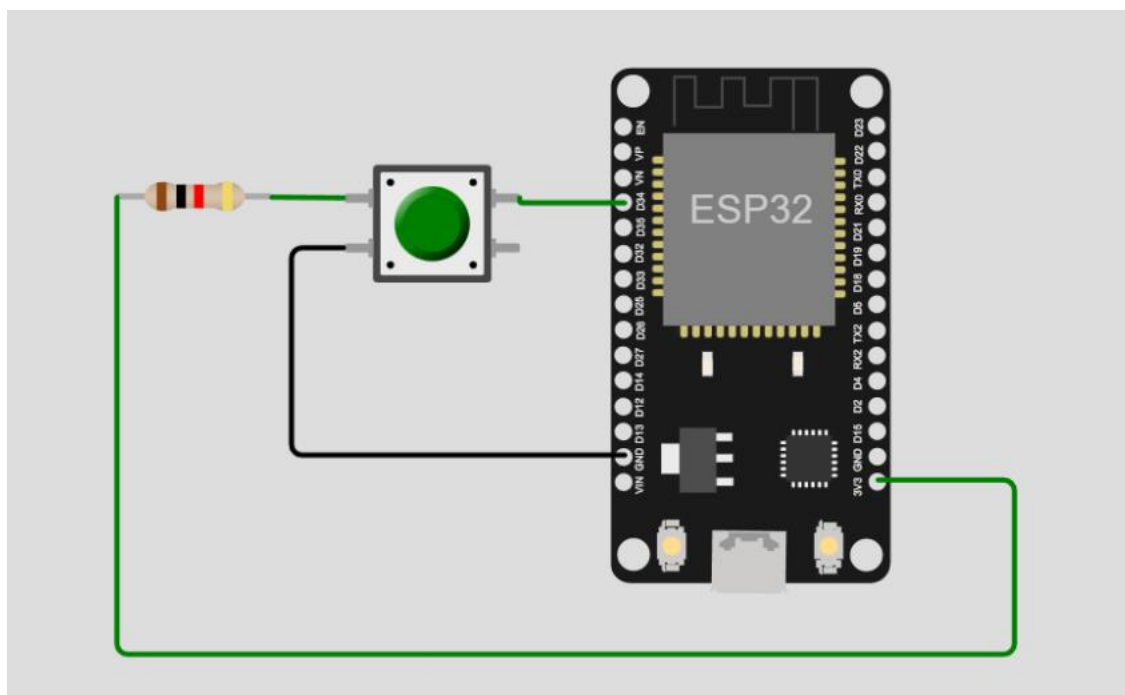
SOFTWARE USED:

WOKWI Simulator

WOKWI LINK:

<https://wokwi.com/projects/365976950496980993>

CIRCUIT DIAGRAM:



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 "t39hi2"//IBM ORGANITION ID
#define DEVICE_TYPE "Learning"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "31929095"//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,portand
wificredential

void setup() {
  pinMode(34,INPUT);
  Serial.begin(115200);
  wificonnect();
  mqttconnect();
}

void loop() {
  int buttonstate = digitalRead(34);
  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 defination 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)) {

```

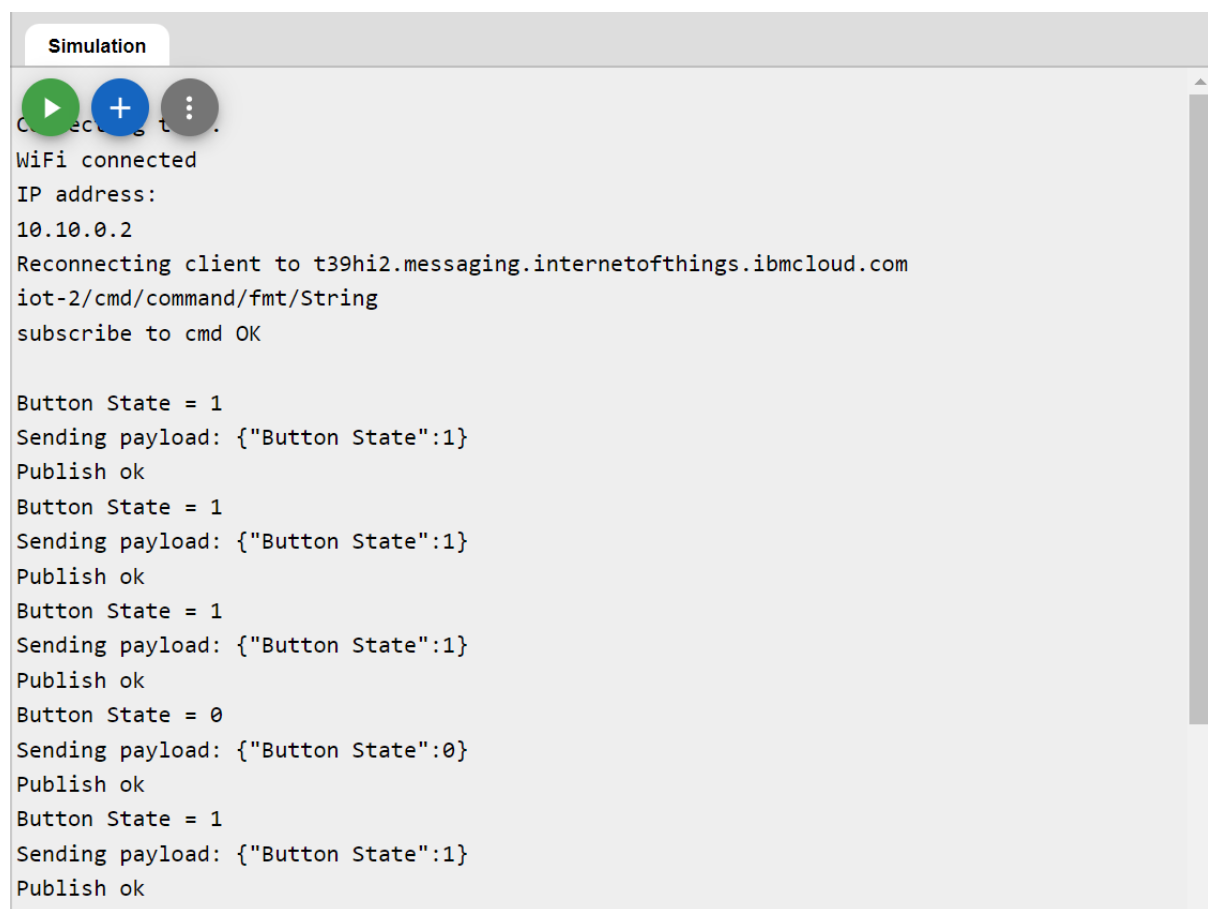
```

        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);
}

```

OUTPUTS:



The screenshot shows a simulation window with a title bar 'Simulation'. Below the title bar are three circular icons: a green play button, a blue plus button, and a grey three-dot menu button. The main area of the window displays a log of messages in a monospaced font. The messages include connection status, IP address, MQTT client reconnection details, a command subscription confirmation, and a series of button state updates (1 and 0) with corresponding JSON payloads and 'Publish ok' confirmations.

```

Connecting to...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to t39hi2.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

Button State = 1
Sending payload: {"Button State":1}
Publish ok
Button State = 1
Sending payload: {"Button State":1}
Publish ok
Button State = 1
Sending payload: {"Button State":1}
Publish ok
Button State = 0
Sending payload: {"Button State":0}
Publish ok
Button State = 1
Sending payload: {"Button State":1}
Publish ok

```

Browse	Action	Device Types	Interfaces	Add Device +	
Identity	Device Information	Recent Events	State	Logs	X
The recent events listed show the live stream of data that is coming and going from this device.					
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":1}	json	a few seconds ago		
Data	{"Button State":0}	json	a few seconds ago		
Data	{"Button State":0}	json	a few seconds ago		

RESULTS:

Given task was carried out successfully.