SMARTBRIDGE IOT

ASSIGNMENT 2

Name: V A Monica

Registration number: 20BEC1189

Challenging task:

In wokwi, connect the push button and upload 0 and 1 to ibm cloud platform

Code:

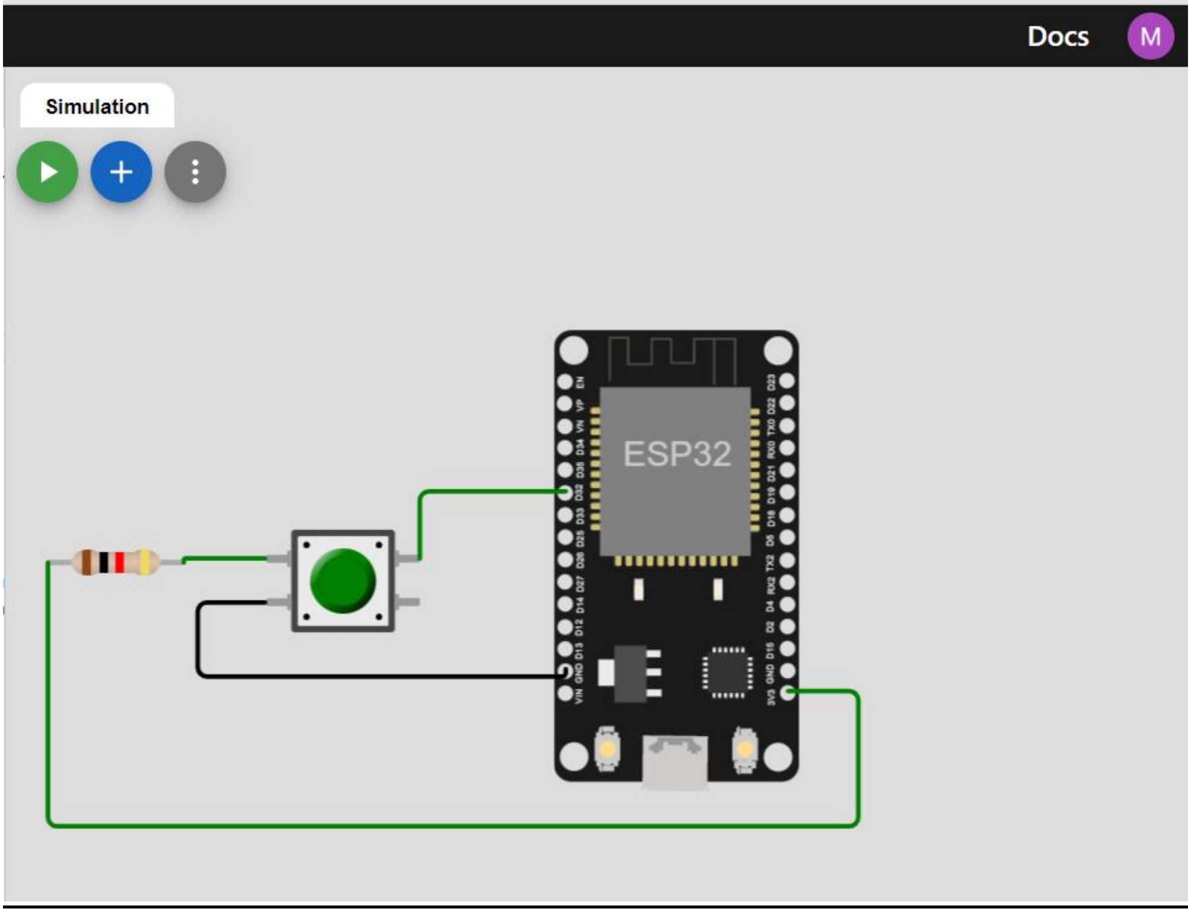
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

//-------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,portand
wificredential

void setup() {
  pinMode(32,INPUT);
  Serial.begin(115200);
  wificonnect();
  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 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);
}
```

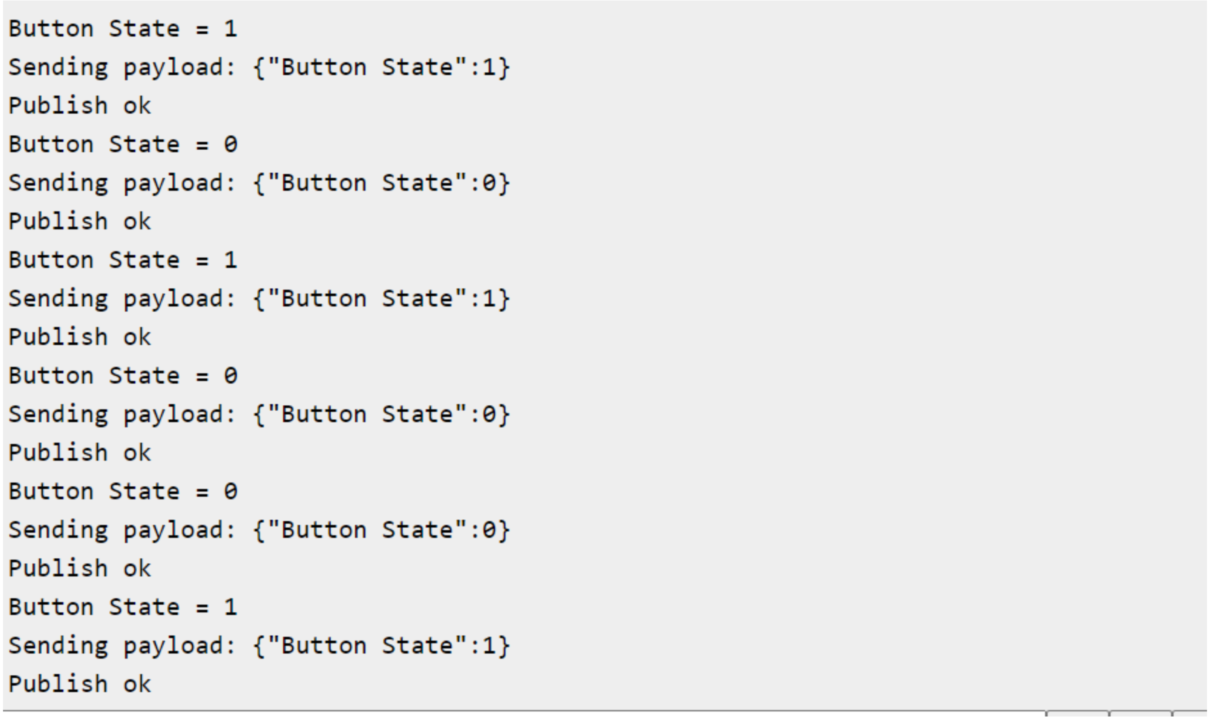
Wokwi link:

https://wokwi.com/projects/365852642909382657

Circuit diagram:

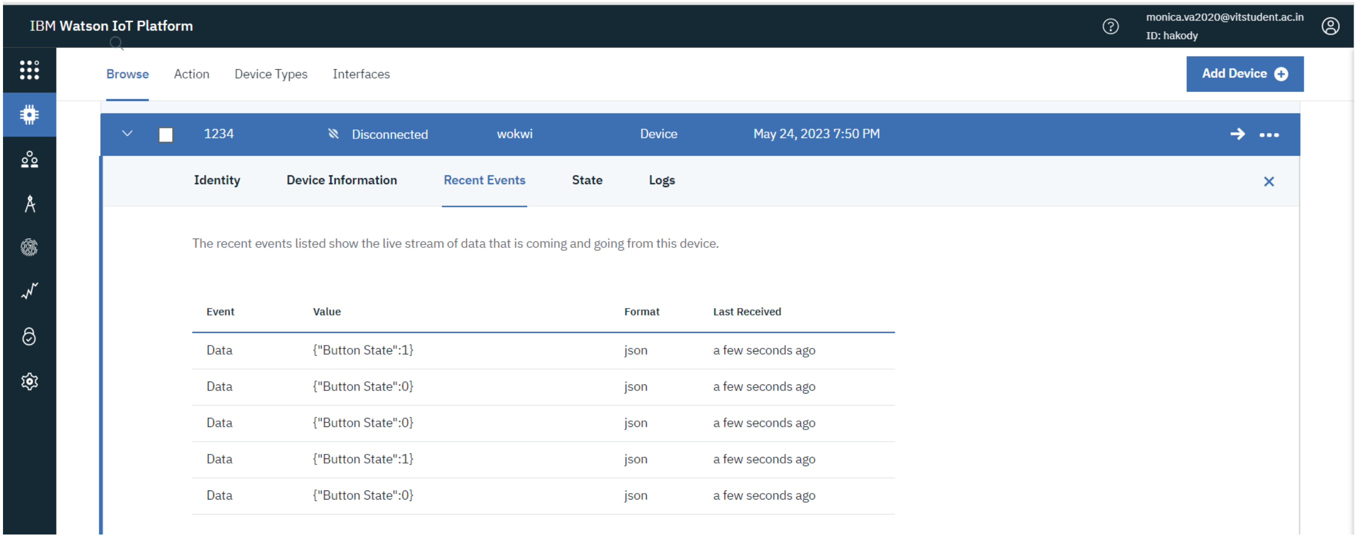


Output:



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

IBM cloud:



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":0}	json	a few seconds ago
Data	{"Button State":0}	json	a few seconds ago
Data	{"Button State":1}	json	a few seconds ago
Data	{"Button State":0}	json	a few seconds ago

Result:

Thus, the given task is performed and the outputs are observed in wokwi and IBM cloud respectively