

```

#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "vico"; //NOMBRE DE NUESTRA RED WIFI
const char* password = "xxx"; //CONTRASEÑA DE NUESTRA RED WIFI
WiFiClient wifiClient;

const char* mqttBroker = "broker.hivemq.com";
const char* mqttClientName = "Victoria"; //CAMBIAR EL USUARIO
const char* mqttUser = "";
const char* mqttPass = "";
const char* name_topic = "efdi/led/vicoPlaca"; //CAMBIAR EL TOPICO
const char* send_topic = "efdi/led/vicoWokwi"; //CAMBIAR EL TOPICO
PubSubClient mqttClient(wifiClient);

bool state = false;
int LED = 2;
int BUT = 4;

void setup() {

  Serial.begin(115200);

  pinMode(LED, OUTPUT);
  pinMode(BUT, INPUT);

  // Connect to wifi
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  // MQTT setup
  mqttClient.setServer(mqttBroker, 1883);
  mqttClient.setCallback(callback);
}

void loop() {
  // Check if we are still connected to the MQTT broker
  if (!mqttClient.connected()) {

```

```

    mqttConnect();
}

// Let PubSubClient library do his magic
mqttClient.loop();

// Add your publish code here -----
if (!digitalRead(BUT)) {
    publishMessage(send_topic, "on", true);
}

else{
    publishMessage(send_topic, "off", true);
}
delay(1000);
}

void mqttConnect() {

    while (!mqttClient.connected()) {

        Serial.print("Attempting MQTT connection...");

        if (mqttClient.connect(mqttClientName, mqttUser, mqttPass)) {

            Serial.println("connected");
            mqttClient.publish("hello", mqttClientName);

            // Topic(s) subscription
            //mqttClient.subscribe(send_topic);
            mqttClient.subscribe(name_topic);

        } else {

            Serial.print("failed, rc=");
            Serial.print(mqttClient.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);

        }
    }
}

void callback(char* topic, byte* payload, unsigned int length) {
    String incomingMessage = "";
    for (int i = 0; i < length; i++){
        incomingMessage += (char)payload[i];
    }
}

```

```

Serial.println("Message arrived[" + String(topic) + "]" + incomingMessage);
if(incomingMessage == "on"){
    digitalWrite(2, HIGH);
    Serial.println("LED ON");
}
else if(incomingMessage == "off"){
    digitalWrite(2, LOW);
    Serial.println("LED OFF");
}
}

//===== publishing as string
void publishMessage(const char* topic, String payload, boolean retained) {
    if (mqttClient.publish(topic, payload.c_str(), true))
        Serial.println("Message published [" + String(topic) + "]: " + payload);
}

```