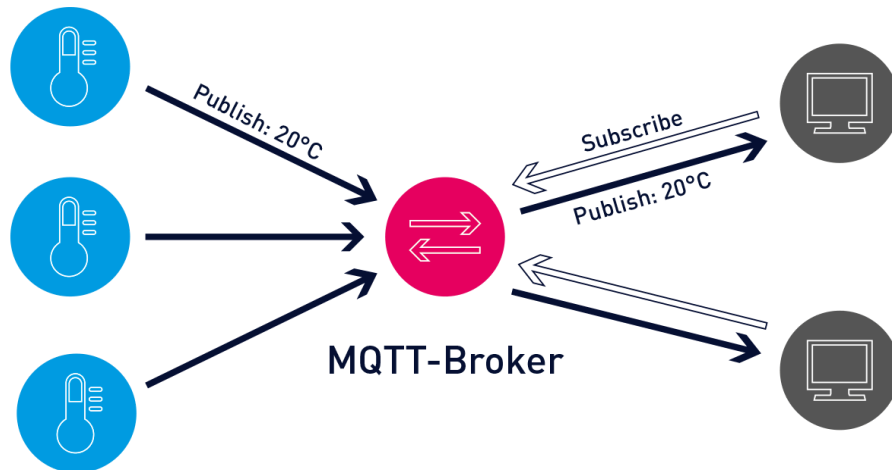


# Tuần 03 - Giao thức MQTT

## Giới thiệu

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông điệp (message) dựa trên mô hình Publish/Subscribe sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định. MQTT gồm 2 phần chính là Broker và Clients.



### 1. MQTT Broker

Hay còn ví như máy chủ, trung tâm hay trực chính giao tiếp. Nhiệm vụ chính của Broker là nhận thông điệp (message) từ client, xếp vào hàng đợi rồi chuyển đến một địa điểm cụ thể. Ngoài ra, nó còn đảm nhận thêm một vài tính năng khác như: bảo mật message, lưu trữ message, logs, ....

### 2. MQTT Client

MQTT Client là các thiết bị/ứng dụng Client kết nối đến Broker để thực hiện truyền nhận dữ liệu. Client thì được chia thành hai nhóm là Publisher và Subscriber. Một Client có thể có 1 trong 2 nhiệm vụ hoặc cả 2.

- Publisher là thiết bị gửi bản tin lên broker.
- Subscriber là người nhận bản tin mỗi khi có bản tin mới gửi lên Broker.

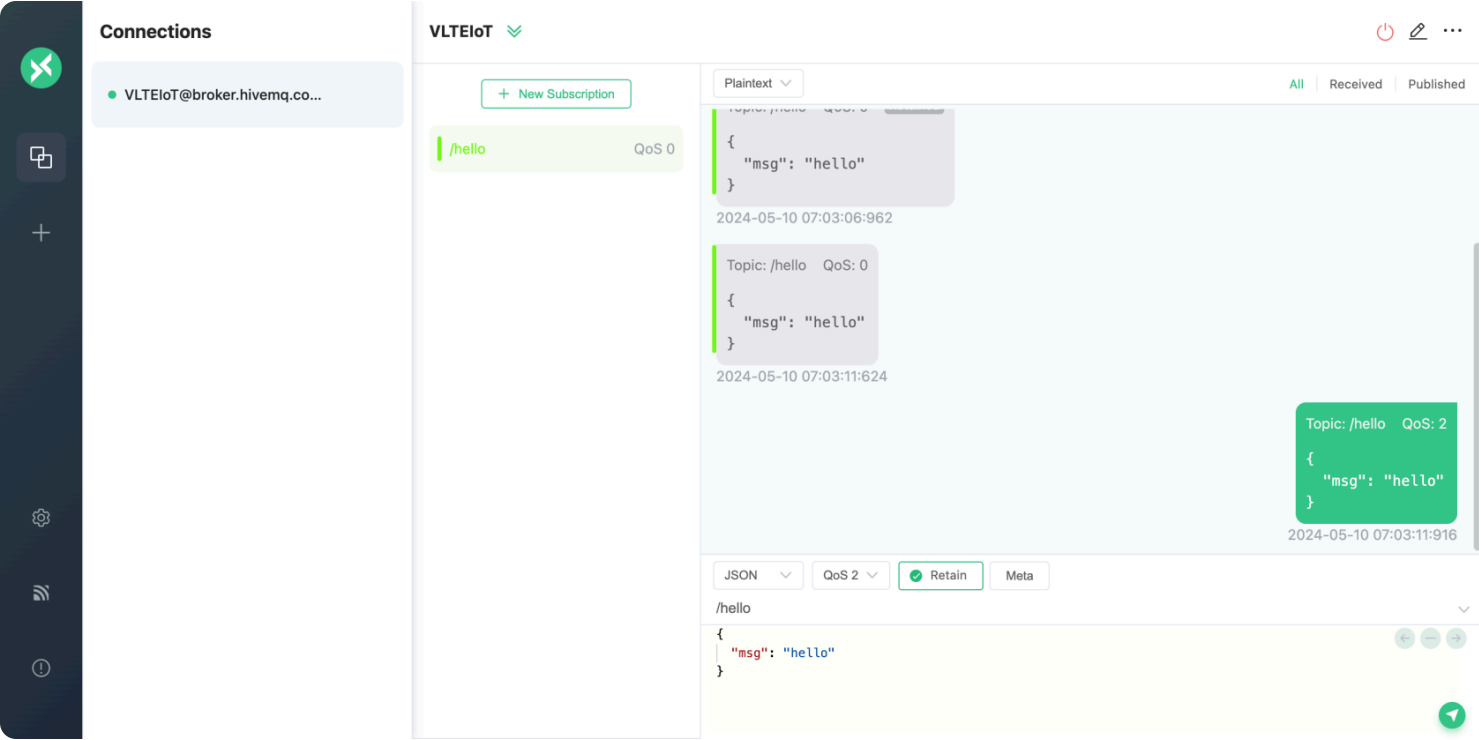
### 3. Thư viện kết nối MQTT trên ESP32/Arduino

Thư viện kết nối MQTT: <https://github.com/knolleary/pubsubclient>

Một số MQTT Borker public có thể sử dụng như:

Name	Broker Address	TCP Port	TLS Port	WebSocket Port
Eclipse	mqtt.eclipse.org	1883	-	80, 443
Mosquitto	test.mosquitto.org	1883	8883, 8884	80
<b>HiveMQ</b>	<b>broker.hivemq.com</b>	<b>1883</b>	<b>-</b>	<b>8000</b>
Flespi	mqtt.flespi.io	1883	8883	80, 443
Dioty	mqtt.dioty.co	1883	8883	8080, 8880
Fluux	mqtt.fluux.io	1883	8883	-
EMQX	broker.emqx.io	1883	8883	8083

3. Công cụ Test MQTT



Link: [http://www.emqx.io/online-mqtt-client#/recent\\_connections](http://www.emqx.io/online-mqtt-client#/recent_connections)

Lập trình với MQTT

1. Lập trình nhúng trên ESP32/Arduino

Sử dụng thư viện **WiFi.h** và **PubSubClient.h**.

- Thư viện **Wi-Fi** cho phép ESP32 thiết lập kết nối với mạng Wi-Fi.
- Thư viện **PubSubClient** cho phép ESP32 kết nối với MQTT broker.

Cấu hình các tham số sau: **Tên** và **Mật khẩu Wi-Fi**, **Địa chỉ** và **Cổng** của MQTT broker, **Topic**;

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
```

```
4 // WiFi
5 const char *ssid = "<SSID>";
6 const char *password = "<Password>";
7
8 // MQTT Broker
9 const char *mqtt_broker = "";
10 const char *topic = "";
11 const char *mqtt_username = "";
12 const char *mqtt_password = "";
13 const int mqtt_port = 1883;
```

Mở cổng Serial và thiết lập kết nối với Wi-Fi.

```
1 Serial.begin(115200);
2 WiFi.begin(ssid, password);
3 while (WiFi.status() != WL_CONNECTED) {
4     delay(500);
5     Serial.println("Connecting to WiFi..");
6 }
```

Sử dụng PubSubClient để thiết lập kết nối với MQTT broker.

#### a. Publish Messages & Subscribe

Sau khi kết nối với MQTT broker được thiết lập thành công, ESP32 sẽ xuất bản các tin nhắn đến topic và sau đó đăng ký topic.

```
1 // Publish một dữ liệu vào một topic
2 client.publish(topic, "Hello World");
3
4 // Subscribe một kênh nhận dữ liệu;
5 // Khi có dữ liệu gửi đến hàm "callback" sẽ được gọi
6 client.subscribe(topic);
```

#### b. Receive MQTT Messages

```
1 void callback(char *topic, byte *payload, unsigned int length) {
2     Serial.print("Message arrived in topic: ");
3     Serial.println(topic);
4     Serial.print("Message:");
5     for (int i = 0; i < length; i++) {
6         Serial.print((char) payload[i]);
7     }
8     Serial.println();
9     Serial.println("-----");
10 }
```

#### c. TLS/SSL Port

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) cung cấp kênh liên lạc an toàn giữa máy khách và máy chủ. Về cốt lõi, TLS và SSL là các giao thức mã hóa sử dụng cơ chế handshake để sử dụng các tham số khác nhau nhằm tạo kết nối an toàn giữa máy khách và máy chủ. Sau khi quá trình kết nối hoàn tất, giao tiếp được mã hóa giữa máy khách và máy chủ sẽ được thiết lập và không kẻ tấn công nào có thể xâm nhập bất kỳ phần nào của giao tiếp. Máy chủ cung cấp chứng chỉ X509 (thường do cơ quan đáng tin cậy cấp) mà khách hàng sử dụng để xác minh danh tính của máy chủ.

## 2. Lập trình ứng dụng

Giao thức MQTT Client được hỗ trợ trên nhiều ngôn ngữ khác như Javascript; Java; Python để gửi và nhận dữ liệu từ MQTT Borker. Một số **thư viện** hỗ trợ kết nối MQTT Client:

- Với Javascript có thể dùng thư viện Eclipse Paho JavaScript Client [Link tham khảo: https://www.eclipse.org/paho/index.php?page=clients/js/index.php](https://www.eclipse.org/paho/index.php?page=clients/js/index.php)
- Với Python có thể dùng thư viện paho-mqtt [Link tham khảo: https://github.com/eclipse/paho.mqtt.python](https://github.com/eclipse/paho.mqtt.python)
- Với php dùng thư viện <https://github.com/php-mqtt/client>

### a. JavaScript MQTT Client

Kết nối với máy chủ bằng WebSockets và Topic đã được đăng ký, sau khi đăng ký, nó sẽ publish thông báo đến Topic đó. Mọi tin nhắn liên quan đến Topic đã đăng ký sẽ được in ra bằng điều khiển Javascript. Yêu cầu sử dụng **cùng** một MQTT broker để hỗ trợ WebSockets hoặc sử dụng một cổng có thể chuyển tiếp giữa WebSockets và TCP.

```
1 <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript">
2 <script src="https://code.jquery.com/jquery-3.6.1.min.js" type="text/javascript"></script>
3
4 // Create a client instance
5 var client = new Paho.MQTT.Client("Broker Address", Number(WebSockets port), "clientId");
6
7 // Set callback handlers
8 client.onConnectionLost = onConnectionLost;
9 client.onMessageArrived = onMessageArrived;
10
11 // Connect the client
12 client.connect({onSuccess:onConnect});
13
14 // Called when the client connects
15 function onConnect() {
16     // Once a connection has been made, make a subscription and send a message.
17     console.log("onConnect");
18     client.subscribe("");
19     message = new Paho.MQTT.Message("");
20     message.destinationName = "";
21     client.send(message);
22 }
23
24 // Called when the client loses its connection
25 function onConnectionLost(responseObject) {
26     if (responseObject.errorCode !== 0) {
27         console.log("onConnectionLost:"+responseObject.errorMessage);
28     }
29 }
```

```
30 // Called when a message arrives
31 function onMessageArrived(message) {
32     console.log("onMessageArrived:"+message.payloadString);
33 }
```

## a. Python

Cài thư viện

```
1 pip install paho-mqtt
```

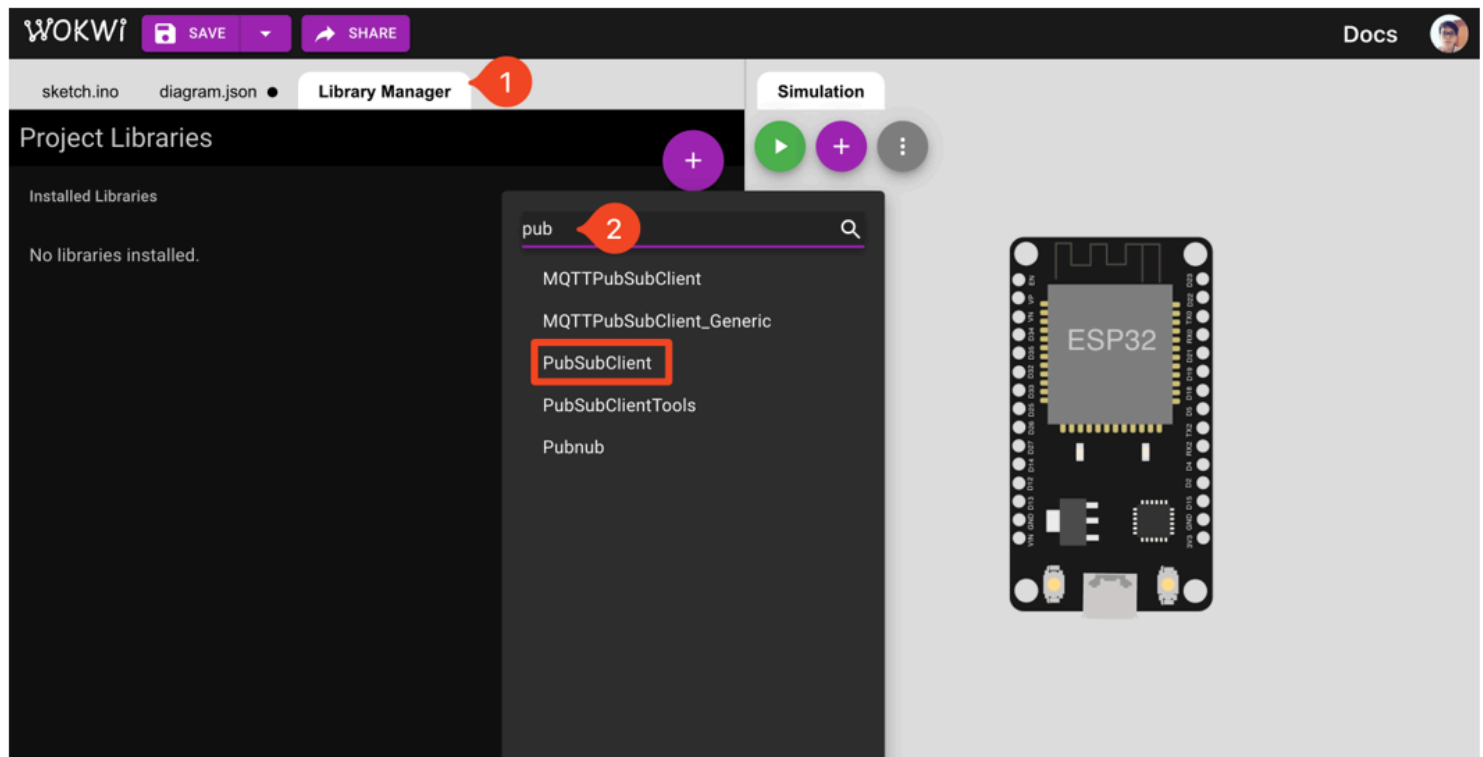
Đoạn chương trình ví dụ

```
1 import paho.mqtt.client as mqtt
2
3 # The callback for when the client receives a CONNACK response from the server.
4 def on_connect(client, userdata, flags, reason_code, properties):
5     print(f"Connected with result code {reason_code}")
6     client.subscribe("$SYS/#")
7
8 # The callback for when a PUBLISH message is received from the server.
9 def on_message(client, userdata, msg):
10     print(msg.topic+" "+str(msg.payload))
11
12 mqttc = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2)
13 mqttc.on_connect = on_connect
14 mqttc.on_message = on_message
15
16 mqttc.connect("mqtt.eclipseprojects.io", 1883, 60)
17 mqttc.loop_forever()
```

# Ví dụ

Sử dụng kết nối WIFI trên ESP32 kết nối đến MQTT Borker public và xây dựng ứng dụng điều khiển bật/tắt một LED từ xa ?

**Bước 1:** Cài đặt thư viện MQTT vào ESP32 như hình bên dưới:



**Bước 2:** Lập trình kết nối MQTT trên ESP32 điều khiển bật/tắt LED

```

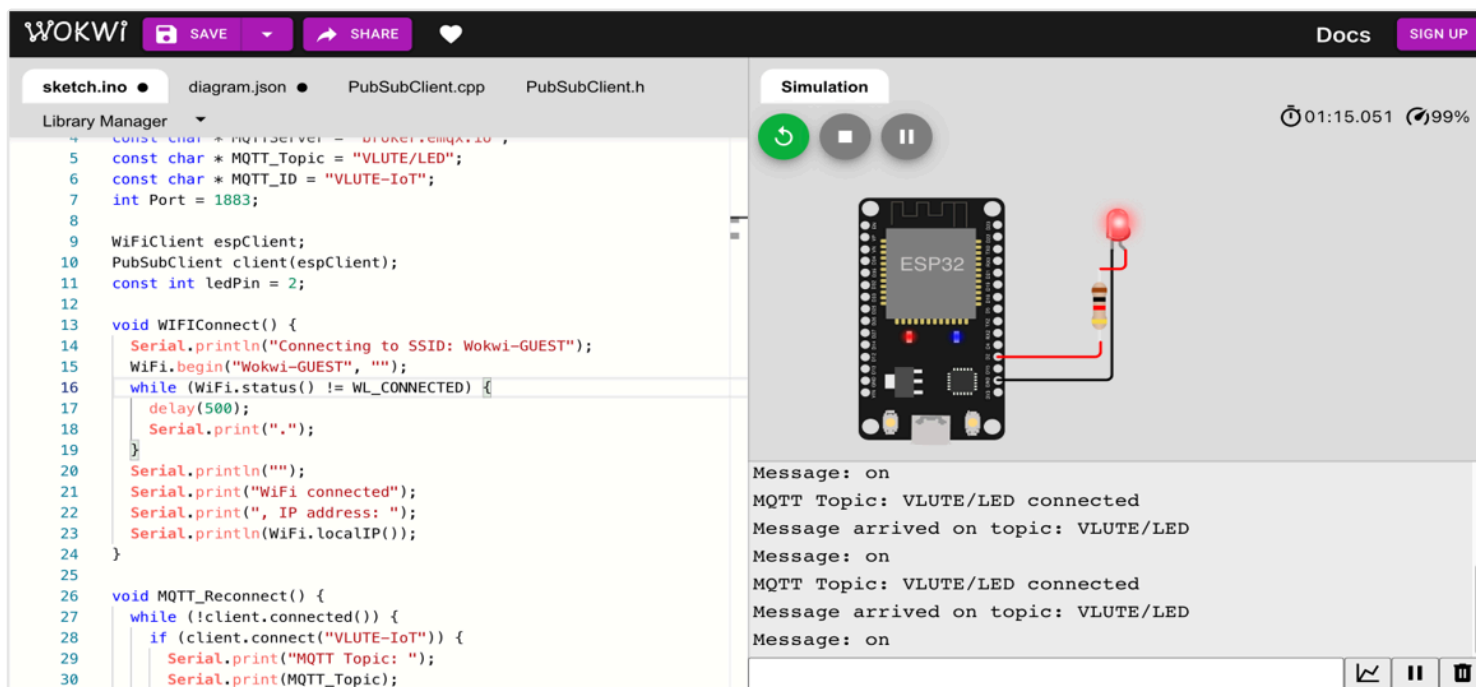
1 #include <WiFi.h>
2 #include "PubSubClient.h"
3
4 const char * MQTTServer = "";
5 const char * MQTT_Topic = "";
6
7 // Tạo ID ngẫu nhiên tại: https://www.guidgen.com/
8 const char * MQTT_ID = "";
9 int Port = 1883;
10
11 WiFiClient espClient;
12 PubSubClient client(espClient);
13 const int ledPin = 2;
14
15 void WIFIDConnect() {
16   Serial.println("Connecting to SSID: Wokwi-GUEST");
17   WiFi.begin("Wokwi-GUEST", "");
18   while (WiFi.status() != WL_CONNECTED) {
19     delay(500);
20     Serial.print(".");
21   }
22   Serial.println("");
23   Serial.print("WiFi connected");
24   Serial.print(", IP address: ");
25   Serial.println(WiFi.localIP());
26 }
27

```

```
28 void MQTT_Reconnect() {
29     while (!client.connected()) {
30         if (client.connect(MQTT_ID)) {
31             Serial.print("MQTT Topic: ");
32             Serial.print(MQTT_Topic);
33             Serial.print(" connected");
34             client.subscribe(MQTT_Topic);
35             Serial.println("");
36         } else {
37             Serial.print("failed, rc=");
38             Serial.print(client.state());
39             Serial.println(" try again in 5 seconds");
40             delay(5000);
41         }
42     }
43 }
44
45 void callback(char* topic, byte* message, unsigned int length) {
46     Serial.print("Message arrived on topic: ");
47     Serial.println(topic);
48     Serial.print("Message: ");
49     String stMessage;
50
51     for (int i = 0; i < length; i++) {
52         Serial.print((char)message[i]);
53         stMessage += (char)message[i];
54     }
55     Serial.println();
56     if (stMessage == "on") {
57         digitalWrite(ledPin, HIGH);
58     }
59     else if (stMessage == "off") {
60         digitalWrite(ledPin, LOW);
61     }
62 }
63
64 void setup() {
65     Serial.begin(115200);
66
67     WiFiConnect();
68     client.setServer(MQTTServer, Port);
69     client.setCallback(callback);
70     pinMode(ledPin, OUTPUT);
71 }
72
73 void loop() {
74     delay(10);
75     if (!client.connected()) {
76         MQTT_Reconnect();
77     }
78     client.loop();
```

79 }

Giao diện đầu nối và kết quả mô phỏng được thể hiện như hình bên dưới.



ESP32 sẽ nhận các giá trị này và so sánh để điều khiển trạng thái của LED tương ứng.

Đoạn chương trình ví dụ về dùng HTML JS để gửi dữ liệu điều khiển LED đến ESP32

```

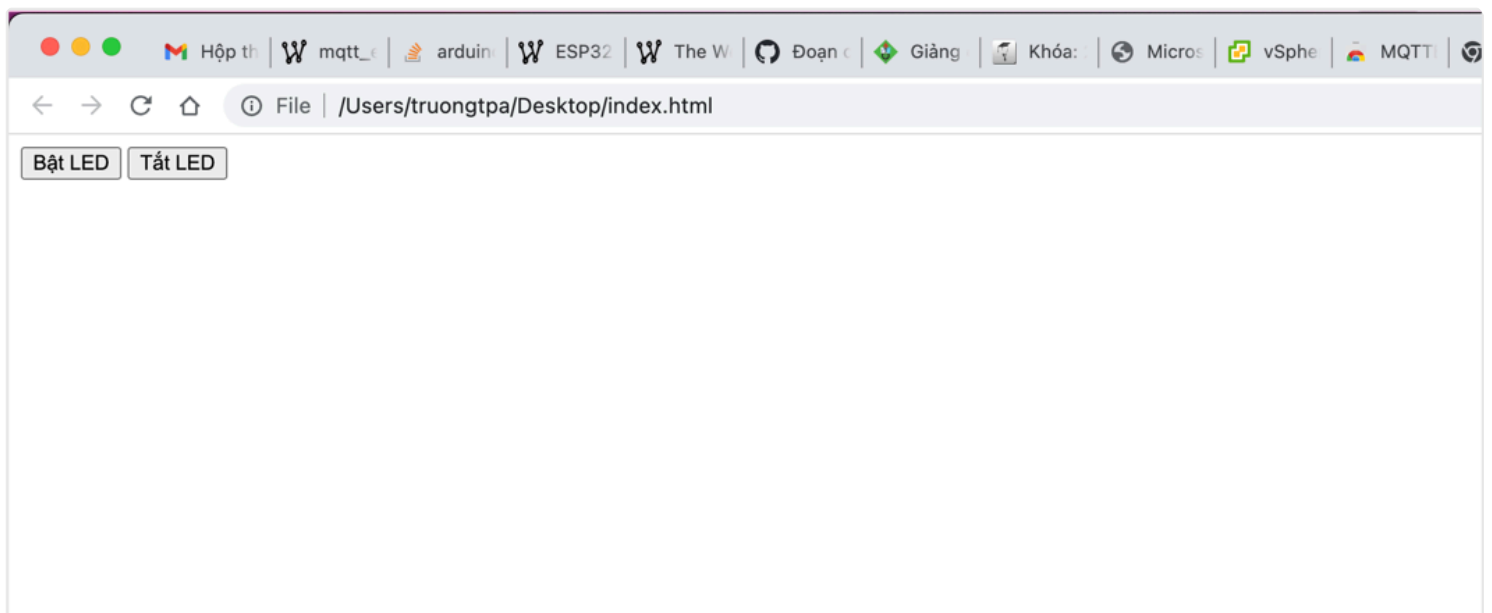
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <script src="https://cdn.jsdelivr.net/npm/paho-mqtt@1.0.1/mqttws31.min.js" type="text/javascript">
8    <script src="https://code.jquery.com/jquery-3.6.1.min.js" type="text/javascript"></script>
9  </head>
10 <body>
11   <button class="bat">Bật LED</button>
12   <button class="tat">Tắt LED</button>
13 </body>
14 </html>
15
16 <script>
17   $('.bat').click(function(){
18     message = new Paho.MQTT.Message("on");
19     message.destinationName = "VLUTE/LED";
20     client.send(message);
21   });
22
23   $('.tat').click(function(){

```



```
24     message = new Paho.MQTT.Message("off");
25     message.destinationName = "VLUTE/LED";
26     client.send(message);
27 });
28
29 client = new Paho.MQTT.Client("broker.emqx.io", Number(8083), "clientId");
30
31 client.onConnectionLost = onConnectionLost;
32 client.onMessageArrived = onMessageArrived;
33
34 client.connect({onSuccess:onConnect});
35
36 function onConnect() {
37     console.log("onConnect");
38     // client.subscribe("World");
39 }
40
41 function onConnectionLost(responseObject) {
42     if (responseObject.errorCode !== 0) {
43         console.log("onConnectionLost:"+responseObject.errorMessage);
44     }
45 }
46
47 function onMessageArrived(message) {
48     console.log("onMessageArrived:"+message.payloadString);
49 }
50 </script>
```

Sử dụng HTML JS điều khiển LED qua MQTT



## Bài tập

## Bài 1: Lập trình đấu nối 4 LED với bo mạch ESP32 thực hiện điều khiển thông qua MQTT trên ứng dụng MQTTX.

a. Bật/tắt tuần tự từng LED

b. Lần lượt chớp tắt ngẫu nhiên 1 đèn LED bất kỳ.

**Gợi ý:** Sử dụng hàm `random` để tạo một số ngẫu nhiên trong khoảng từ a đến b `int pinLED = random(min, max);`

```
1 void loop() {  
2   // print a random number from 0 to 299  
3   randomNumber = random(300);  
4  
5   // print a random number from 10 to 19  
6   randomNumber = random(10, 20);  
7 }
```

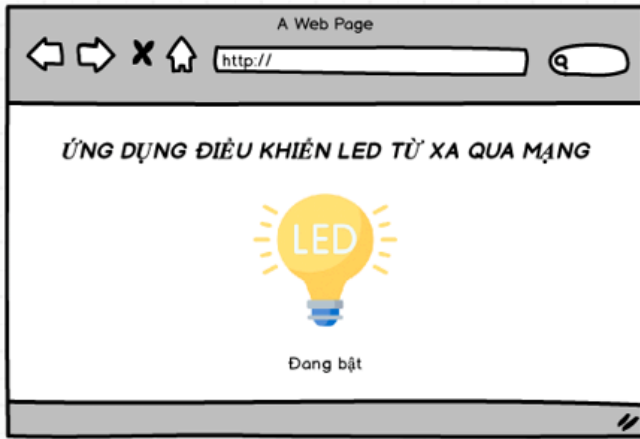
c. Thực hiện đấu nối 4 LED điều khiển bật/tắt LED sáng/tắt cùng lúc sau mỗi 3s.

## Bài 2: Lập trình đấu nối LED với bo mạch ESP32 thực hiện điều khiển bật/tắt LED đó thông qua MQTT trên ứng dụng WEB ?

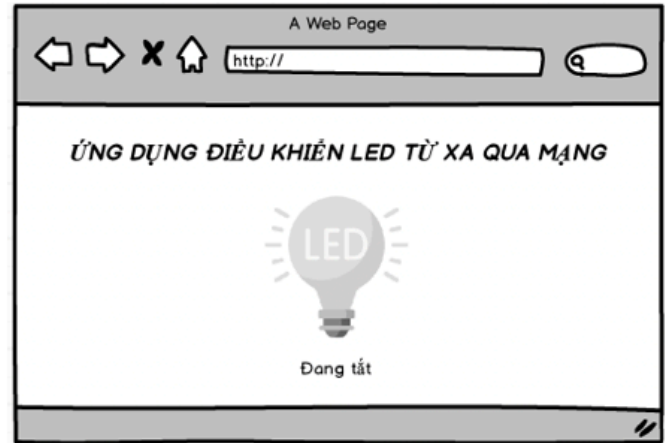
a. Thực hiện lại các yêu cầu 1a, 1b, 1c thông qua MQTT trên ứng dụng WEB.

b. Điều khiển bật/tắt LED đó thông qua MQTT trên ứng dụng WEB. Gợi ý:

```
1 <script>  
2   var anh_bat = "https://i.imgur.com/pPqLQx4.png";  
3   var anh_tat = "https://i.imgur.com/yRMydZv.png";  
4   $('anh').click(function(){  
5     var anhht = $('anh').attr('src');  
6     if(anhht == anh_bat){  
7       $('anh').attr('src', anh_tat);  
8       message = new Paho.MQTT.Message("off");  
9       message.destinationName = "VLUTE/LED";  
10      client.send(message);  
11    }  
12  });  
13 </script>
```



*Giao diện LED bật*



*Giao diện LED tắt*