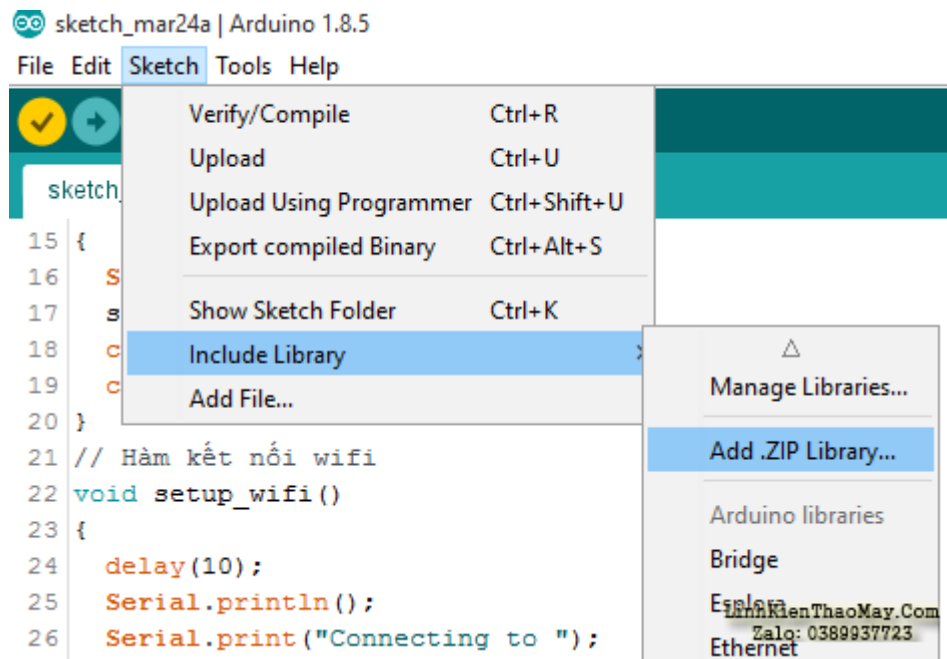


Giao thức MQTT phù hợp nhất cho các Project IoT thương mại, nó đáp ứng tốc độ tốt, băng thông ít, độ tin cậy cao. Tài liệu về giao thức MQTT thì các bạn tham khảo ở các trên mạng hoặc 1 số bài sau :

- <https://smartfactoryvn.com/technology/internet-of-things/giao-thuc-mqtt-la-gi-nhung-un-g-dung-cua-mqtt-nhu-the-nao/>
- <https://esp8266.vn/nonos-sdk/mqtt/what-is-mqtt/>

Mình sẽ không nhắc lại phần lí thuyết nữa vì trên mạng có rất nhiều rồi. mình sẽ đi vào thực hành làm thử 1 project với giao thức MQTT luôn

Đầu tiên, phía esp8266 các bạn tải thư viện Pubsubclient



ADD thư viện vào

Giao thức MQTT cần có 1 server ( gọi là broker) để làm trung tâm của mọi luồng dữ liệu, trong các bài viết sau mình sẽ hướng dẫn các bạn tự build server, còn trong bài này mình sẽ sử dụng server miễn phí không bảo mật là broker.hivemq.com để demo

Các bạn copy chương trình cho esp8266

C

```
#include
<ESP8266WiFi.h>
```

```
#include
<PubSubClient.h>
```

```
// Thông tin v□ wifi
```

```
#define ssid  
"dieukhien"
```

```
#define password  
"12345678"
```

```
#define mqtt_server  
"broker.hivemq.com"
```

```
const uint16_t  
mqtt_port = 1883;
```

```
WiFiClient  
espClient;
```

```
PubSubClient  
client(espClient);
```

```
void setup()
```

```
{
```

```
Serial.begin(115200)  
;
```

```
setup_wifi();
```

```
client.setServer(mqt  
t_server,
```

```
client.setCallback(c  
allback);
```

```
}
```

```
// Hàm k□t n□i wifi
```

```
void setup_wifi()
```

```
{
```

```
delay(10);
```

```
Serial.println();
```

```
Serial.print("Connec  
ting to ");
```

```
Serial.println(ssid)  
;
```

```
WiFi.begin(ssid,
password);
while (WiFi.status()
!= WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi
connected");
Serial.println("IP
address: ");
Serial.println(WiFi.
localIP());
}
// Hàm call back đ
nhận dữ liệu
void callback(char*
topic, byte*
{
Serial.print("Co tin
nhan moi tu
Serial.println(topic
);
for (int i = 0; i <
length; i++)
Serial.print((char)p
ayload[i]);
Serial.println();
}
// Hàm reconnect
thực hiện k
t n
i
void reconnect()
{
```

```
while
(!client.connected())
{
// Thực hiện kết nối
với mqtt user và
if
(client.connect("ESP
{
Serial.println("Đã
kết nối:");
client.subscribe("Io
T47_MQTT_Test");
}
else
{
Serial.print("Lỗi:,
rc=");
Serial.print(client.
state());
Serial.println(" try
again in 5
// Đợi 5s
delay(5000);
}
}
}
unsigned long t;
void loop()
{
if
(!client.connected())
```

```
reconnect();
```

```
client.loop();
```

```
if(millis() - t >  
500) //nếu 500 mili
```

```
{
```

```
t=millis();
```

```
Serial.print("Gui  
tin nhan "Xin chao"
```

```
client.publish("IoT4  
7_MQTT_Test", "Xin
```

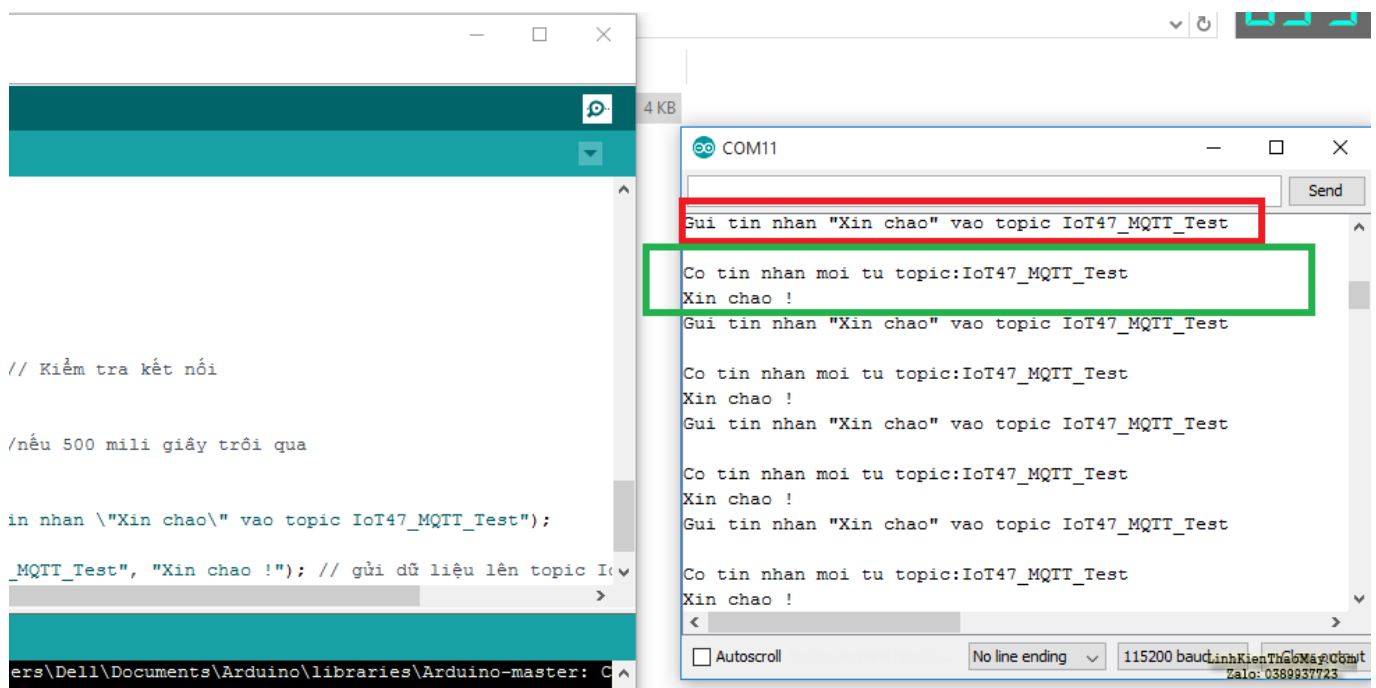
```
}
```

```
}
```

```
1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3
4
5 // Thông tin về wifi
6 #define ssid "dieukhien"
7 #define password "12345678"
8 #define mqtt_server "broker.hivemq.com"
9 const uint16_t mqtt_port = 1883; //Port của CloudMQTT TCP
10
11 WiFiClient espClient;
12 PubSubClient client(espClient);
13
14 void setup()
15 {
16   Serial.begin(115200);
17   setup_wifi();
18   client.setServer(mqtt_server, mqtt_port);
19   client.setCallback(callback);
20 }
21 // Hàm kết nối wifi
22 void setup_wifi()
23 {
24   delay(10);
25   Serial.println();
26   Serial.print("Connecting to ");
27   Serial.println(ssid);
28   WiFi.begin(ssid, password);
29   while (WiFi.status() != WL_CONNECTED) {
30     delay(500);
31     Serial.print(".");
32   }
33   Serial.println("");
34   Serial.println("WiFi connected");
35   Serial.println("IP address: ");
36   Serial.println(WiFi.localIP());
37 }
38 // Hàm call back để nhận dữ liệu
39 void callback(char* topic, byte* payload, unsigned int length)
40 {
41   Serial.print("Co tin nhan moi tu topic:");
42   Serial.println(topic);
43   for (int i = 0; i < length; i++)
44     Serial.print((char)payload[i]);
45   Serial.println();
46 }
47 // Hàm reconnect thực hiện kết nối lại khi mất kết nối với MQTT Broker
48 void reconnect()
49 {
50   while (!client.connected()) // Chờ tới khi kết nối
51   {
52     // Thực hiện kết nối với mqtt user và pass
53     if (client.connect("ESP8266_id1", "ESP_offline", 0, 0, "ESP8266_id1_offline")) //kết nối vào broker
54     {
55       Serial.println("Đã kết nối.");
56       client.subscribe("IoT47_MQTT_Test"); //đăng kí nhận dữ liệu từ topic IoT47_MQTT_Test
57     }
58     else
59     {
60       Serial.print("Lỗi:, rc=");
61       Serial.print(client.state());
62       Serial.println(" try again in 5 seconds");
63       // Đợi 5s
64       delay(5000);
65     }
66   }
67 }
68 unsigned long t;
69 void loop()
70 {
71   if (!client.connected())// Kiểm tra kết nối
72     reconnect();
73   client.loop();
74   if(millis() - t > 500) //nếu 500 mili giây trôi qua
75   {
76     t=millis();
77     Serial.print("Gui tin nhan "Xin chao" vao topic IoT47_MQTT_Test");
78     client.publish("IoT47_MQTT_Test", "Xin chao !"); // gửi dữ liệu lên topic IoT47_MQTT_Test
79   }
80 }
```

- Ở dòng 5 và 6 các bạn đổi thành wifi của mình.
- Hàm callback là hàm gọi lại khi có dữ liệu gửi đến topic mà mình đăng kí
- Hàm client.subscribe dùng để đăng kí 1 topic
- Hàm client.publish dùng để gửi dữ liệu lên 1 topic
- Hàm client.connect để kết nối vào MQTT Broker, với các tham số  
 ESP8266\_id1 : Id của thiết bị đăng kí vào (có thể chỉnh sửa bất thành bất kì)  
 ESP\_offline : khi thiết bị (esp8266) mất mạng (offline) thì broker sẽ xuất bản  
 1 tin nhắn vào topic này  
 ESP8266\_offline : Nội dung của tin nhắn offline

Sau khi kết nối thành công ở dòng 53 mình đăng kí topic IoT47\_MQTT\_Test và trong hàm loop xuất bản tin nhắn "Xin chào" vào chính topic IoT47\_MQTT\_Test .Như vậy mình sẽ nhận lại được chính tin nhắn mà ta đã xuất bản !



Kết quả: Mình đã thấy chính tin nhắn mà mình xuất bản lên

## Thiết kế giao diện web gửi tin nhắn cho ESP8266

Để có thể kết nối tới MQTT broker, mình sẽ sử dụng ngôn ngữ JavaScript và thư viện PahoMQTT

Các bạn tạo 1 file tên là index.html và thêm mã code web

C

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Demo  
MQTT</title>
```

```
<meta  
name="viewport"
```

```
<meta  
charset="utf-8">
```

```
<script  
src="https://cdnjs.c
```

```
<script type =  
"text/javascript"
```

```
var max,at_OK;
```

```
function makeid()
```

```
{
```

```
var text = "";
```

```
var possible =  
"ABCDEFGHIJKLMNOPS
```

```
for (var i = 0; i <  
5; i++)
```

```
text +=  
possible.charAt(Math
```

```
return text;
```

```
}
```

```
// Create a client  
instance
```

```
var client = new  
Paho.MQTT.Client("br
```

```
// set callback  
handlers
```

```
client.onConnectionL  
ost =
```

```
client.onMessageArri  
ved =
```

```
var options = {
```

```
  useSSL: false,
```

```
  userName: "",
```

```
  password: "",
```

```
  onSuccess: onConnect,
```

```
  onFailure: doFail
```

```
}
```

```
console.log("Connect  
to
```

```
// connect the  
client
```

```
client.connect(options);
```

```
function doFail(e){
```

```
  console.log(e);
```

```
}
```

```
function onConnect()  
//sự kiện kết nối
```

```
{
```

```
  console.log("Connect  
OK");
```

```
  client.subscribe("Io  
T47_MQTT_Test");
```

```
}
```

```
// called when the  
client loses its
```

```
function  
onConnectionLost(res
```

```
{
```

```
if  
(responseObject.erro  
{  
console.log(response  
Object.errorMessage)  
}  
}
```

```
// called when a  
message arrives
```

```
function  
onMessageArrived(mes
```

```
{
```

```
console.log(message.  
destinationName +
```

```
}
```

```
function public  
(topic,data)
```

```
{
```

```
message = new  
Paho.MQTT.Message(da
```

```
message.destinationN  
ame = topic;
```

```
client.send(message)  
;
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

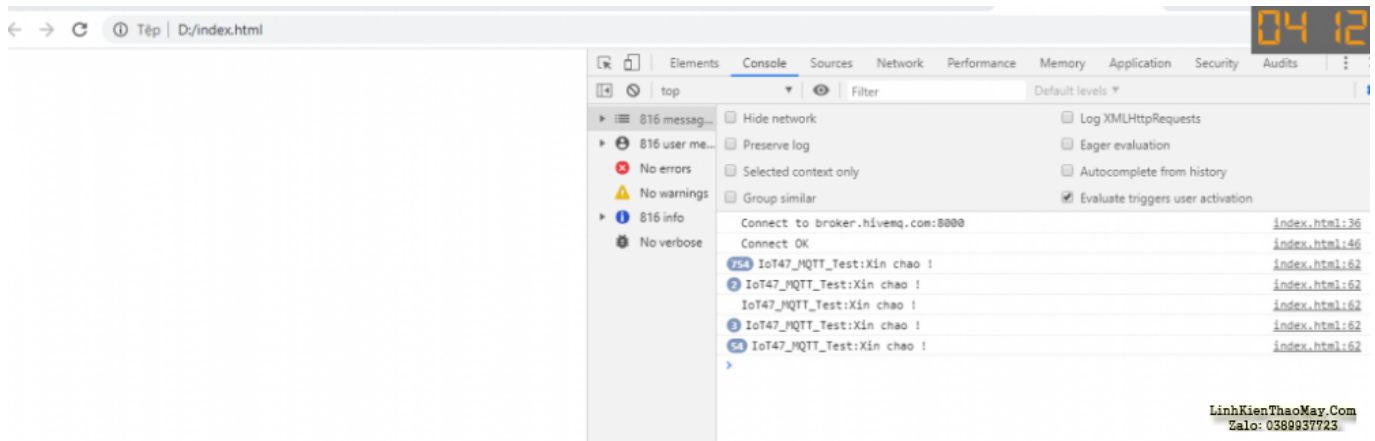
```
</body>
```

```
</html>
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Demo MQTT</title>
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <meta charset="utf-8">
7 <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
8 <script type = "text/javascript" language = "javascript">
9 var max,at_OK;
10 function makeid()
11 {
12 var text = "";
13 var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
14
15 for (var i = 0; i < 5; i++)
16 text += possible.charAt(Math.floor(Math.random() * possible.length));
17
18 return text;
19 }
20 // Create a client instance
21 var client = new Paho.MQTT.Client("broker.hivemq.com", 8000, makeid());
22
23 // set callback handlers
24 client.onConnectionLost = onConnectionLost;
25 client.onMessageArrived = onMessageArrived;
26
27 var options = {
28 useSSL: false,
29 userName: "",
30 password: "",
31 onSuccess:onConnect,
32 onFailure:doFail
33 }
34
35
36 console.log("Connect to broker.hivemq.com:8000");
37 // connect the client
38 client.connect(options);
39
40 function doFail(e){
41 console.log(e);
42 }
43
44 function onConnect() //sự kiện kết nối thành công
45 {
46 console.log("Connect OK");
47 client.subscribe("IoT47_MQTT_Test"); //đăng kí kênh
48 }
49
50 // called when the client loses its connection
51 function onConnectionLost(responseObject)
52 {
53 if (responseObject.errorCode !== 0)
54 {
55 console.log(responseObject.errorMessage);
56 }
57 }
58
59 // called when a message arrives
60 function onMessageArrived(message)
61 {
62 console.log(message.destinationName + ":" +message.payloadString);
63 }
64 function public (topic,data)
65 {
66 message = new Paho.MQTT.Message(data);
67 message.destinationName = topic;
68 client.send(message);
69 }
70 </script>
71 </head>
72 <body>
73
74 </body>
75 </html>
```

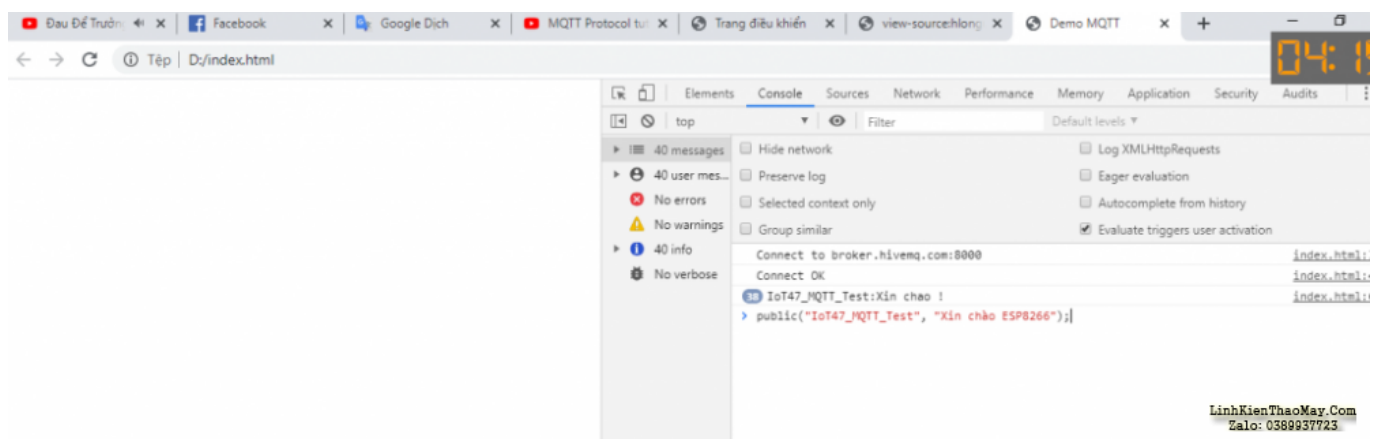
- Hàm makeID có nhiệm vụ tạo ra 1 id ngẫu nhiên để web kết nối tới MQTT broker tránh bị trùng id
- Mình sẽ kết nối vào broker.hivemq.com qua cổng 8000 vì cổng 8000 là cổng dành cho các kết nối thông qua Socket, trong khi cổng 1883 dành cho các kết nối qua TCP
- Hàm public dùng để xuất bản 1 tin nhắn tới 1 topic nào đó
- Hàm client.subscribe dùng để đăng kí nhận tin nhắn từ 1 topic nào đó
- Hàm onMessageArrived là hàm callback khi có 1 tin nhắn từ 1 topic đã đăng kí gửi tới

Mình sẽ lưu lại và mở file này bằng trình duyệt, sau đó ấn F12 để xem dữ liệu debug in ra màn hình console

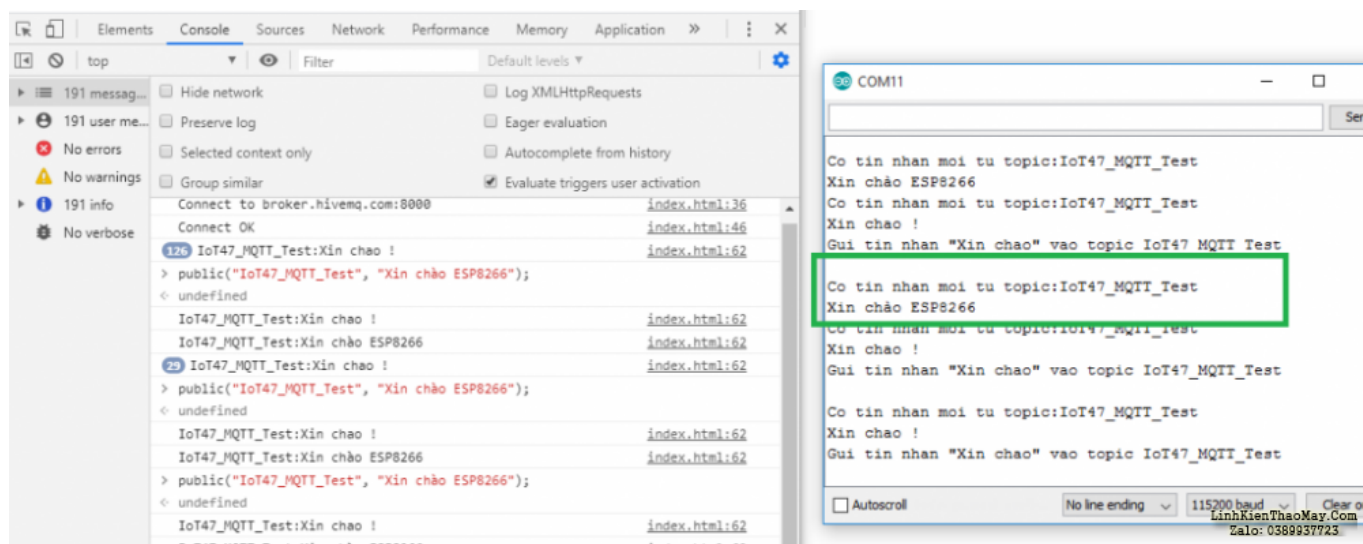


Các bạn có thể thấy sau khi thông báo Connect OK thì mình đã thấy tin nhắn “Xin chao !” mà esp8266 liên tục gửi lên mỗi 500 mili giây

Cũng trong màn hình debug, mình gõ hàm public để gửi thử dữ liệu đến esp8266 nhé !



Gọi hàm public gửi dữ liệu



ESP8266 đã thấy tin nhắn xin chào

## Thiết kế giao diện nút nhấn điều khiển thiết bị cho giao thức mqtt

mình sẽ tạm dừng lại giao diện đã xài ở bài 5 nhé

C

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Demo  
MQTT</title>
```

```
<meta  
name="viewport"
```

```
<meta  
charset="utf-8">
```

```
<script  
src="https://cdnjs.c
```

```
<script type =  
"text/javascript"
```

```
var max,at_OK;
```

```
function makeid()
```

```
{  
var text = "";  
var possible =  
"ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
  
for (var i = 0; i <  
5; i++)  
text +=  
possible.charAt(Math  
  
return text;  
}  
  
// Create a client  
instance  
var client = new  
Paho.MQTT.Client("br  
  
// set callback  
handlers  
client.onConnectionL  
ost =  
client.onMessageArri  
ved =  
  
var options = {  
useSSL: false,  
userName: "",  
password: "",  
onSuccess:onConnect,  
onFailure:doFail  
}  
  
console.log("Connect  
to
```

```
// connect the  
client
```

```
client.connect(optio  
ns);
```

```
function doFail(e){
```

```
console.log(e);
```

```
}
```

```
function onConnect()
```

```
//sự kiện kết nối
```

```
{
```

```
console.log("Connect  
OK");
```

```
client.subscribe("ES  
P8266_sent_data");
```

```
}
```

```
// called when the  
client loses its
```

```
function  
onConnectionLost(res
```

```
{
```

```
if  
(responseObject.erro
```

```
{
```

```
console.log(response  
Object.errorMessage)
```

```
}
```

```
}
```

```
// called when a  
message arrives
```

```
function  
onMessageArrived(mes
```

```
{
```

```
console.log(message.  
destinationName +
```

```
document.getElementById("tinnhan").inner
```

```
}
```

```
function public  
(topic,data)
```

```
{
```

```
message = new  
Paho.MQTT.Message(da
```

```
message.destinationN  
ame = topic;
```

```
client.send(message)  
;
```

```
}
```

```
</script>
```

```
<style>
```

```
.b{width:  
100px;height:
```

```
.t{width:  
100px;height:
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div style="width:  
330px;height:
```

```
<h1  
align="center">Đi□u
```

```
<p align="center"  
id="tinnhan">Tin
```

```
<table  
align="center">
```

```
<tr>
```

```
<td><button  
class='b'
```



```
onclick="public('ESP  
8266_read_data', 'Bật  
1')">Bật  
1</button><td>
```

```
<td><button  
class='t'
```

```
<tr>
```

```
<tr>
```

```
<td><button  
class='b'
```

```
<td><button  
class='t'
```

```
<tr>
```

```
<tr>
```

```
<td><button  
class='b'
```

```
<td><button  
class='t'
```

```
<tr>
```

```
<tr>
```

```
<td><button  
class='b'
```

```
<td><button  
class='t'
```

```
<tr>
```

```
</table>
```

```
</div>
```

```
</body>
```

```
</html>
```

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Demo MQTT</title>
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <meta charset="utf-8">
7 <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
8 <script type="text/javascript" language="javascript">
9 var max_at_OK;
10 function makeid()
11 {
12 var text = "";
13 var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
14
15 for (var i = 0; i < 5; i++)
16 text += possible.charAt(Math.floor(Math.random() * possible.length));
17
18 return text;
19 }
20 // Create a client instance
21 var client = new Paho.MQTT.Client("broker.hivemq.com", 8000, makeid());
22
23 // set callback handlers
24 client.onConnectionLost = onConnectionLost;
25 client.onMessageArrived = onMessageArrived;
26
27 var options = {
28 useSSL: false,
29 userName: "",
30 password: "",
31 onSuccess: onConnect,
32 onFailure: doFail
33 }
34
35
36 console.log("Connect to broker.hivemq.com:8000");
37 // connect the client
38 client.connect(options);
39
40 function doFail(e){
41 console.log(e);
42 }
43
44 function onConnect() //sự kiện kết nối thành công
45 {
46 console.log("Connect OK");
47 client.subscribe("ESP8266_sent_data"); //đăng kí kênh
48 }
49
50 // called when the client loses its connection
51 function onConnectionLost(responseObject)
52 {
53 if (responseObject.errorCode !== 0)
54 {
55 console.log(responseObject.errorMessage);
56 }
57 }
58
59 // called when a message arrives
60 function onMessageArrived(message)
61 {
62 console.log(message.destinationName + ":" + message.payloadString);
63 document.getElementById("tinnhan").innerHTML = "Tin nhắn từ esp8266: " + message.payloadString;
64 }
65 }
66 function public (topic,data)
67 {
68 message = new Paho.MQTT.Message(data);
69 message.destinationName = topic;
70 client.send(message);
71 }
72 </script>
73 <style>
74 .b{width: 100px;height: 40px;font-size: 21px;color: #FFF;background-color:#4caf50;border-radius: 10px;}
75 .t{width: 100px;height: 40px;font-size: 21px;color: #FFF;background-color:#f44336;border-radius: 10px;}
76 </style>
77 </head>
78 <body>
79 <div style="width: 330px;height: auto;margin: 0 auto;margin-top: 70px">
80 <h1 align="center">Điều khiển thiết bị qua WIFI - MQTT</h1>
81 <p align="center" id="tinnhan">Tin nhắn từ esp8266: ... </p>
82 <table align="center">
83 <tr>
84 <td><button class="b" onclick="public('ESP8266_read_data','Bật 1')">Bật 1</button><td>
85 <td><button class="t" onclick="public('ESP8266_read_data','Tắt 1')">Tắt 1</button><td>
86 <tr>
87 <tr>
88 <td><button class="b" onclick="public('ESP8266_read_data','Bật 2')">Bật 2</button><td>
89 <td><button class="t" onclick="public('ESP8266_read_data','Tắt 2')">Tắt 2</button><td>
90 <tr>
91 <tr>
92 <td><button class="b" onclick="public('ESP8266_read_data','Bật 3')">Bật 3</button><td>
93 <td><button class="t" onclick="public('ESP8266_read_data','Tắt 3')">Tắt 3</button><td>
94 <tr>
95 <tr>
96 <td><button class="b" onclick="public('ESP8266_read_data','Bật 4')">Bật 4</button><td>
97 <td><button class="t" onclick="public('ESP8266_read_data','Tắt 4')">Tắt 4</button><td>
98 <tr>
99 </table>
100 </div>
101 </body>
102 </html>

```

Demo MQTT

# Điều khiển thiết bị qua WIFI - MQTT

Tin nhắn từ esp8266: ...

Bật 1 Tắt 1

Bật 2 Tắt 2

Bật 3 Tắt 3

Bật 4 Tắt 4

Khi ấn nút ( sự kiện OnClick) mình sẽ cho xuất bản tin nhắn tương ứng tới ESP8266 qua topic ESP8266\_read\_data. Do vậy ở code esp8266 mình sẽ sửa lại code cho nó đăng kí vào topic ESP8266\_read\_data và cho web đăng kí vào kênh ESP8266\_sent\_data để nhận tin nhắn esp8266 gửi lên

```
while (!client.connected()) // Chờ tới khi kết nối
{
  // Thực hiện kết nối với mqtt user và pass
  if (client.connect("ESP8266_id1","ESP_offline",0,0,"ESP8266_id1_offline")) //kết nối vào broker
  {
    Serial.println("Đã kết nối:");
    client.subscribe("ESP8266_read_data"); //đăng kí nhận dữ liệu từ topic ESP8266_read_data
  }
  else
  {
    Serial.print("Lỗi:, rc=");
    Serial.print(client.connect_return_code());
  }
}
```

LinhKienThaoMay.Com  
Zalo: 0389937723

Sửa lại topic đăng kí của esp8266 thành ESP8266\_read\_data

Mình cũng thêm đoạn mã đọc data từ cổng Serial để gửi lên cho web

```
if(Serial.available() > 0)
{
  delay(30);
  char inputString[30]="";
  int i=0;
  while(Serial.available() > 0)
  {
    char inChar = (char)Serial.read();
    inputString[i++] = inChar;
  }
  client.publish("ESP8266_sent_data", inputString); // gửi dữ liệu lên topic ESP8266_sent_data
}
```

LinhKienThaoMay.Com  
Zalo: 0389937723

TRUNG TÂM SỬA CHỮA ĐIỆN TỬ QUẢNG BÌNH

MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình

GIÁ RẺ

NHANH CHÓNG

LINH KIỆN CHÍNH HÃNG

SANYO ELEC  
Panasonic TOSHIBA  
SAMSUNG  
BISHI



## TRUNG TÂM SỬA CHỮA ĐIỆN TỬ XÔ NGUYỄN

- Dịch vụ sửa chữa điện tử tại nhà
- Cung cấp linh kiện điện tử
- Tư vấn lắp đặt nhà thông minh

Đc: Quảng Thuận, tx Ba Đồn,  
tỉnh Quảng Bình - 0901.679.359

Full code cho esp8266

C

```
#include  
<ESP8266WiFi.h>
```

```
#include  
<PubSubClient.h>
```

```
// Thông tin v□ wifi
```

```
#define ssid  
"dieukhien"
```

```
#define password  
"12345678"
```

```
#define mqtt_server  
"broker.hivemq.com"
```

```
const uint16_t  
mqtt_port = 1883;
```

```
WiFiClient  
espClient;
```

```
PubSubClient  
client(espClient);
```

```
void setup()
```

```
{
```

```
Serial.begin(115200)  
;
```

```
setup_wifi();

client.setServer(mqt
t_server,

client.setCallback(c
allback);
}

// Hàm k t n i wifi

void setup_wifi()

{

delay(10);

Serial.println();

Serial.print("Connec
ting to ");

Serial.println(ssid)
;

WiFi.begin(ssid,
password);

while (WiFi.status()
!= WL_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi
connected");

Serial.println("IP
address: ");

Serial.println(WiFi.
localIP());

}

// Hàm call back đ 
nhận dữ liệu
```

```
void callback(char*
topic, byte*
{
Serial.print("Co tin
nhan moi tu
Serial.println(topic
);
for (int i = 0; i <
length; i++)
Serial.print((char)p
ayload[i]);
Serial.println();
}
// Hàm reconnect
thực hiện kết nối
void reconnect()
{
while
(!client.connected())
{
// Thực hiện kết nối
với mqtt user và
if
(client.connect("ESP
{
Serial.println("Đã
kết nối:");
client.subscribe("ES
P8266_read_data");
}
else
{
Serial.print("Lỗi:,
rc=");
```

```
Serial.print(client.  
state());  
Serial.println(" try  
again in 5  
// Đợi 5s  
delay(5000);  
}  
}  
}  
void loop()  
{  
if  
(!client.connected())  
reconnect();  
client.loop();  
if(Serial.available(  
) > 0)  
{  
delay(30);  
char  
inputString[30]="";  
int i=0;  
while(Serial.availab  
le() > 0)  
{  
char inChar =  
(char)Serial.read();  
inputString[i++] =  
inChar;  
}
```



```
client.publish("ESP8  
266_sent_data",
```

```
}
```

```
}
```

```
1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3
4
5 // Thông tin về wifi
6 #define ssid "dieukhien"
7 #define password "12345678"
8 #define mqtt_server "broker.hivemq.com"
9 const uint16_t mqtt_port = 1883; //Port của CloudMQTT TCP
10
11 WiFiClient espClient;
12 PubSubClient client(espClient);
13
14 void setup()
15 {
16   Serial.begin(115200);
17   setup_wifi();
18   client.setServer(mqtt_server, mqtt_port);
19   client.setCallback(callback);
20 }
21 // Hàm kết nối wifi
22 void setup_wifi()
23 {
24   delay(10);
25   Serial.println();
26   Serial.print("Connecting to ");
27   Serial.println(ssid);
28   WiFi.begin(ssid, password);
29   while (WiFi.status() != WL_CONNECTED) {
30     delay(500);
31     Serial.print(".");
32   }
33   Serial.println("");
34   Serial.println("WiFi connected");
35   Serial.println("IP address: ");
36   Serial.println(WiFi.localIP());
37 }
38 // Hàm call back để nhận dữ liệu
39 void callback(char* topic, byte* payload, unsigned int length)
40 {
41   Serial.print("Co tin nhan moi tu topic:");
42   Serial.println(topic);
43   for (int i = 0; i < length; i++)
44     Serial.print((char)payload[i]);
45   Serial.println();
46 }
47 // Hàm reconnect thực hiện kết nối lại khi mất kết nối với MQTT Broker
48 void reconnect()
49 {
50   while (!client.connected()) // Chờ tới khi kết nối
51   {
52     // Thực hiện kết nối với mqtt user và pass
53     if (client.connect("ESP8266_id1", "ESP_offline", 0, 0, "ESP8266_id1_offline")) //kết nối vào broker
54     {
55       Serial.println("Đã kết nối.");
56       client.subscribe("ESP8266_read_data"); //đăng kí nhận dữ liệu từ topic ESP8266_read_data
57     }
58     else
59     {
60       Serial.print("Lỗi, rc=");
61       Serial.print(client.state());
62       Serial.println(" try again in 5 seconds");
63       // Đợi 5s
64       delay(5000);
65     }
66   }
67 }
68 void loop()
69 {
70   if (!client.connected())// Kiểm tra kết nối
71     reconnect();
72   client.loop();
73   if (Serial.available() > 0)
74   {
75     delay(30);
76     char inputString[30]="";
77     int i=0;
78     while (Serial.available() > 0)
79     {
80       char inChar = (char)Serial.read();
81       inputString[i++] = inChar;
82     }
83     client.publish("ESP8266_sent_data", inputString); // gửi dữ liệu lên topic ESP8266_sent_data
84   }
85 }
```

Demo kết quả

hay

### Các bài viết tương tự:

1. [\[IoT\] Bài 3: ESP8266 Demo ứng dụng điều khiển LED từ xa qua internet bằng tập lệnh AT esp8266](#)
2. [\[IoT\] Bài 4: Giới thiệu ngôn ngữ HTML và mô hình http request/reponse](#)
3. [Bài 1: Làm quen với module WiFi Esp8266](#)
4. [Bài 2: Giao tiếp với module WiFi ESP8266 bằng tập lệnh AT](#)
5. [Các mạch nạp không thể thiếu dành cho dân chế cháo](#)
6. [dau thu ky thuat so thanh thanh - thu cac chuong trinh tot co hinh ro net nhung mat tieng](#)
7. [Hướng dẫn Flash firmware Tasmota cho SONOFF một cách đơn giản nhất](#)
8. [LG \[L177WSB-PSF - hình bị nhòe và mờ](#)
9. [Mạch sạc pin 18650 chống ngược cực](#)
10. [MSI 7267 - không cài dc win,máy báo màn hình xanh ghost vẫn ok.\(đã thiết lập trong bios\)](#)
11. [panasonic THL-32C5V - máy bị hong bo chinh](#)
12. [samsuns 32e420 - nhảy hình phía trên nhỏ các bác chi diem cat](#)