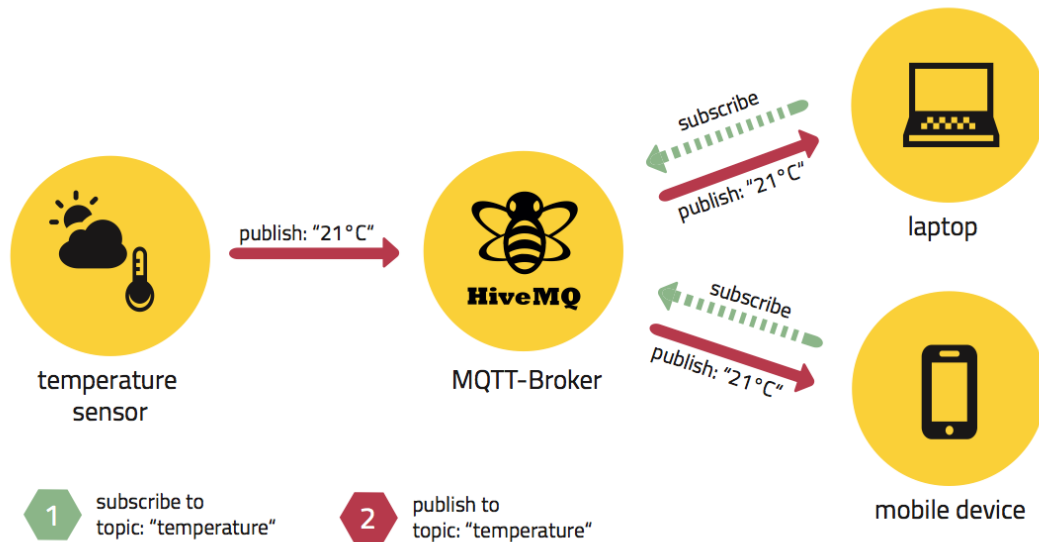


What is MQTT?

[http://www.eclipse.org/community/eclipse\\_newsletter/2014/october/article2.php](http://www.eclipse.org/community/eclipse_newsletter/2014/october/article2.php)

Visit above web site to know more about MQTT.

---



The above figure shows the system architecture. The messages are handled by the MQTT broker. You need a program at the sensor to publish data, and another program as a client to process the data. The messages are forwarded according to the "TOPIC" field.

You can install mqttbox (win10) to subscribe the message. Let's take Arduino for example.

Some free mqtt brokers.

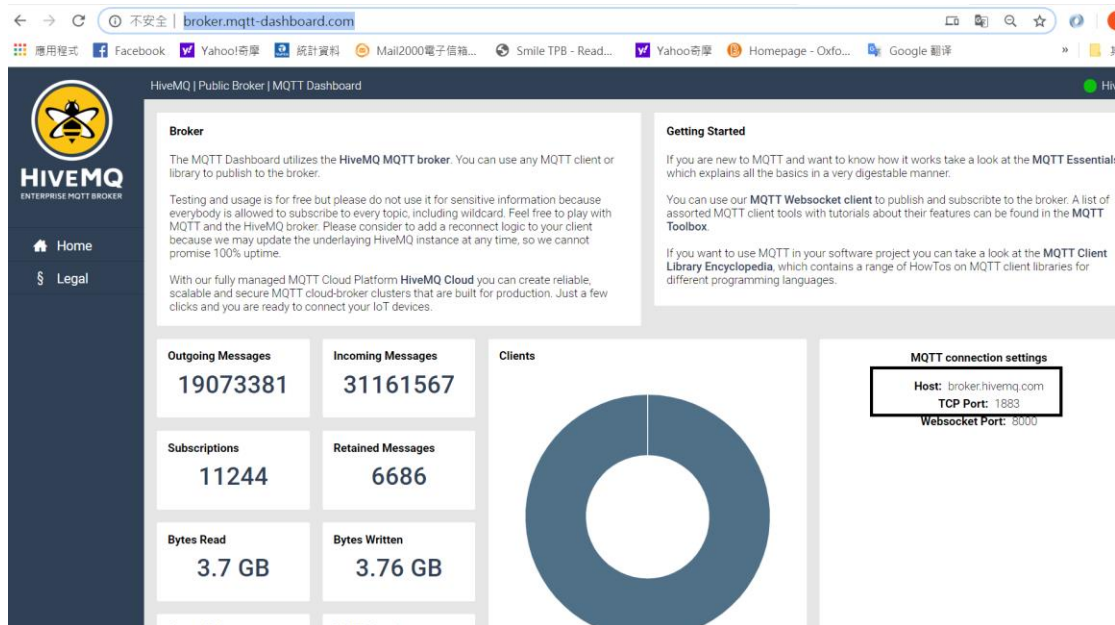
<https://diyprojects.io/8-online-mqtt-brokers-iot-connected-objects-cloud/#.XpZgg8gzbb0>

1. Free MQTT broker.

Visit <http://broker.mqtt-dashboard.com/>

**Host:** broker.hivemq.com

**TCP Port:** 1883



## 2. Install mqttbox

[http://workswithweb.com/html/mqttbox/installing\\_apps.html#install on windows](http://workswithweb.com/html/mqttbox/installing_apps.html#install_on_windows)

[WS](#)

### Installing on Windows

You can run MQTTBox apps on Windows in two ways.

1. From Windows App Store (Recommended)
2. Installing from .exe file

#### Installing from Windows Store (Recommended)

MQTTBox is available on Windows store for free. (Available only from windows 10 anniversary update and above)



#### Installing from .exe file

Installing MQTTBox on windows is straightforward. [Click here to download MQTTBox-win.exe file](#) and double click on .exe file to install.

Run mqttbox and create a new

MQTTBox

MQTTBox Edit Help

Menu MQTT CLIENT SETTINGS

<b>MQTT Client Name</b>	<b>MQTT Client Id</b>
free	d3edaf25-0c88-4839-b6cb-441a42
<b>Protocol</b>	<b>Host</b>
mqtt / tcp	broker.mqtt-dashboard.com:1883
<b>Username</b>	<b>Password</b>
Username	Password
<b>Reconnect Period (milliseconds)</b>	<b>Connect Timeout (milliseconds)</b>
1000	30000
<b>Will - Topic</b>	<b>Will - QoS</b>
Will - Topic	0 - Almost Once

You can see that MQTT connected the broker.

MQTTBox

MQTTBox Edit Help

Menu Connected Add publisher Add subscriber Settings

free - mqtt://broker.mqtt-dashboard.com:1883

<b>Topic to publish</b>	<b>Topic to subscribe</b>
Topic to publish	Topic to subscribe
<b>QoS</b>	<b>QoS</b>
0 - Almost Once	0 - Almost Once
<b>Retain</b> <input type="checkbox"/>	
<b>Payload Type</b>	
Strings / JSON / XML / Characters	
e.g: {'hello':'world'}	
<b>Payload</b>	<b>Subscribe</b>

Try to deliver a message. You can try any topic name, but remember to use as special as possible. Click “subscribe” to subscribe the topic you just type.

MQTTBox

MQTTBox Edit Help

Menu

←

Connected

Add publisher

Add subscriber

free - mqtt://broker.mqtt-dashboard.com:1883

Topic to publish

mytest1234

QoS

0 - Almost Once

Retain

Payload Type

Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

Topic to subscribe

mytest1234

QoS

0 - Almost Once

Subscribe

click

Send a message.

MQTTBox

MQTTBox Edit Help

Menu

←

Connected

Add publisher

Add subscriber

free - mqtt://broker.mqtt-dashboard.com:1883

Topic to publish

mytest1234

QoS

0 - Almost Once

Retain

Payload Type

Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

{'hello':'world'}

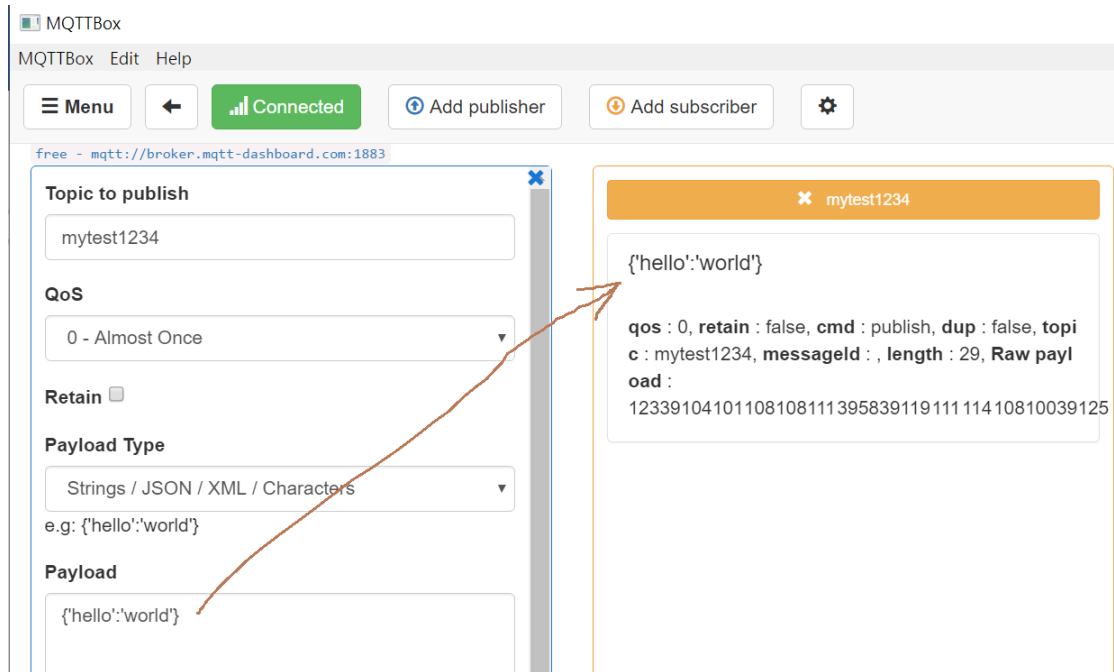
type a message here

Publish

click publish

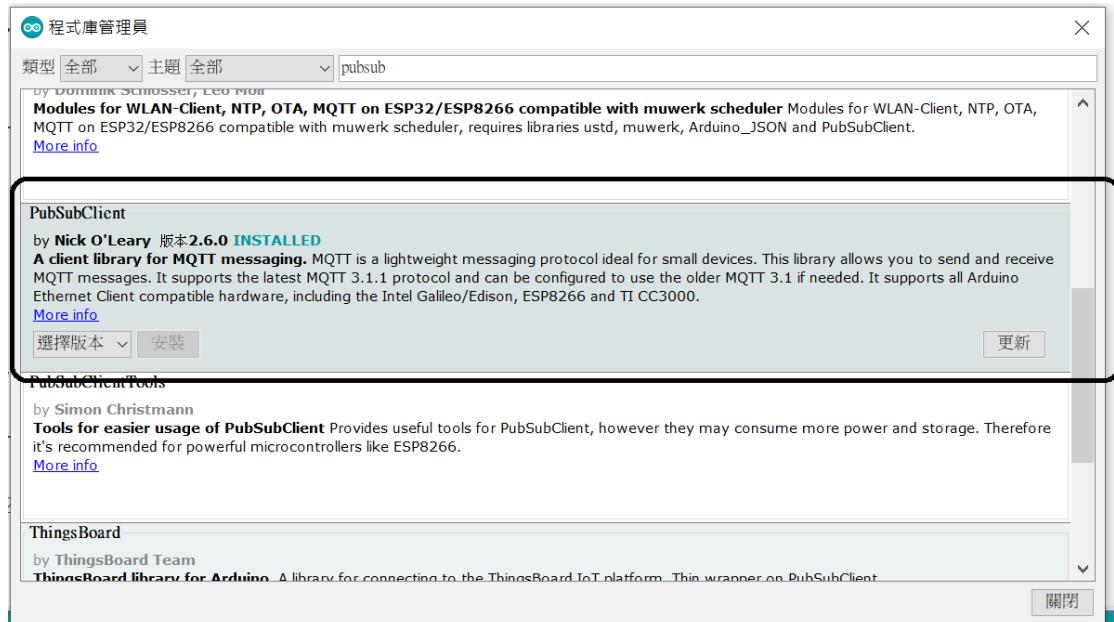
mytest1234

You will see that message returns at left.



In this experiment, we use mqttbox to publish a message with topic *mytest1234*. We can see the message at the right hand side because mqttbox also subscribe the same topic.

### 3. Install Arduino library, PubSubClient.



Let take mqtt\_8266 for example. This an example file of pubsubclient.

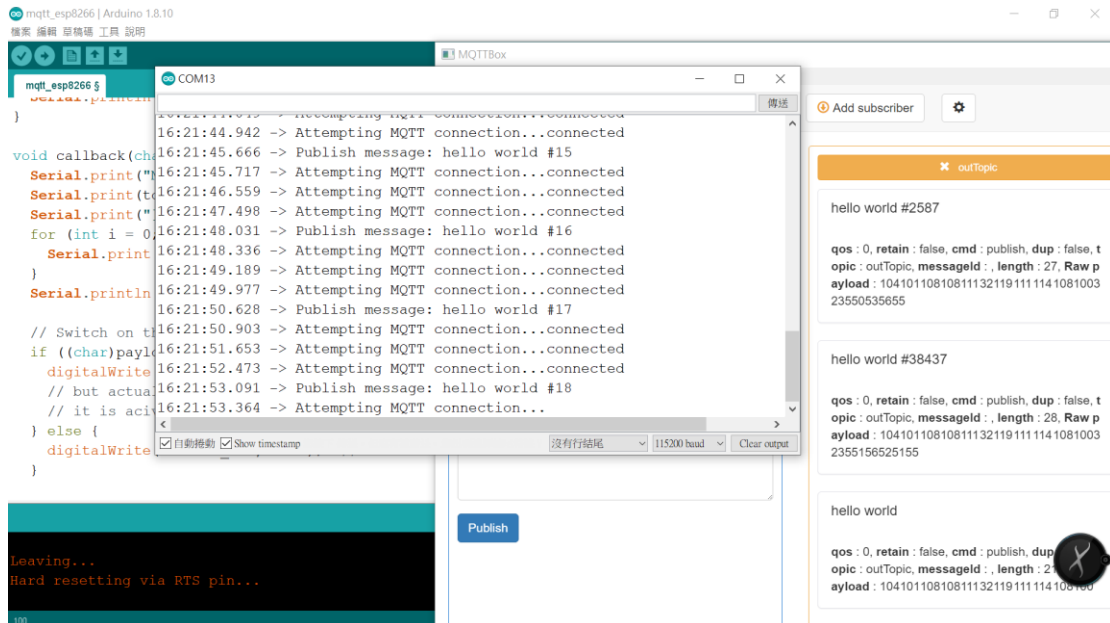
- Modify ssid, password, and mqtt\_server.

The default topic to publish: outTopic

The default topic to be subscribed: inTopic

You can modify the them.

Open mqttbox and subscribe "outTopic". You will see many messages arrive.



This is because many other users run this example at the same time. If you modify the topic, you can observe your own messages only. I just append "ncnu" with the old topics.

\*\*\*\*\* Important remark \*\*\*\*\*

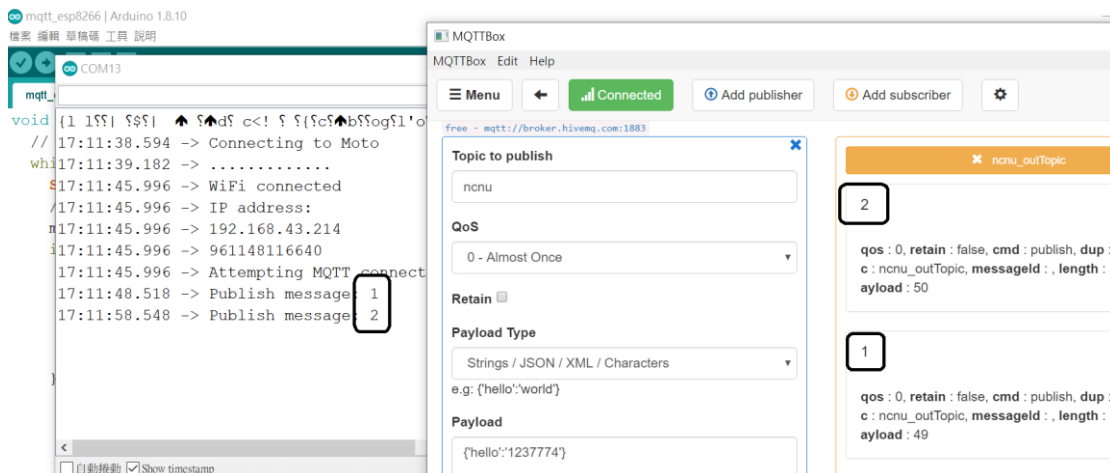
client.connect("ESP8266Client")

This example uses "ESP8266Client" as the session ID. That's why we see that 8266 always need to reconnect to the broker because many other users use the same ID.

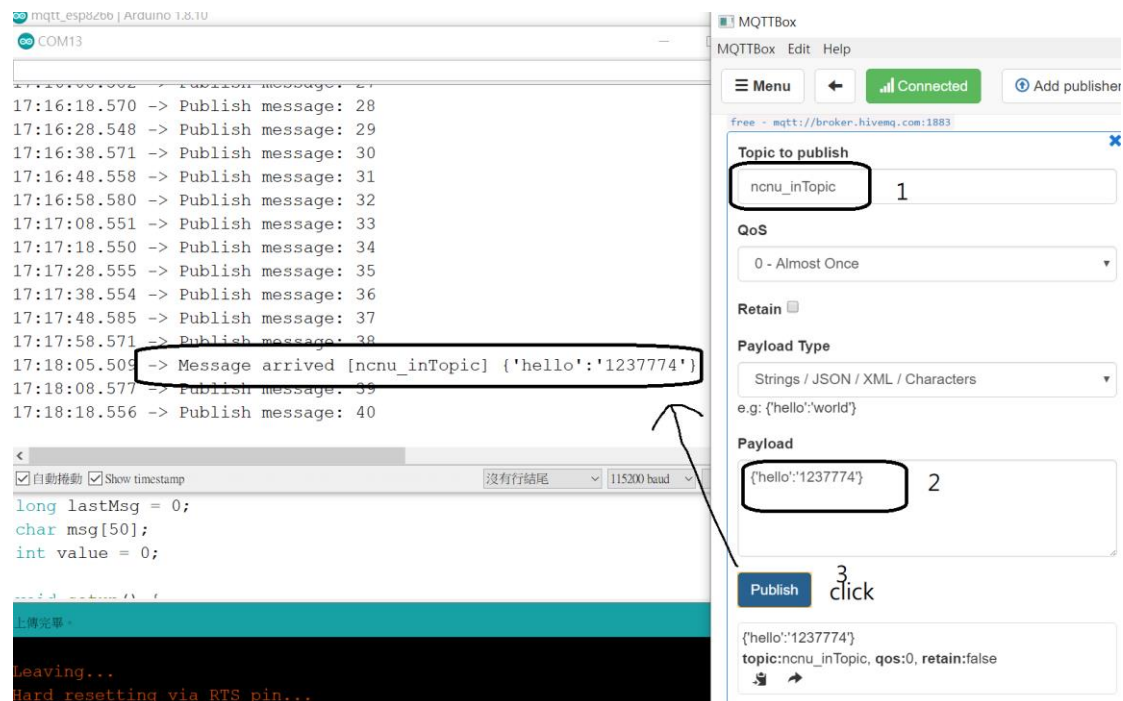
The best way is to use MAC address of 8266 as the ID. So we can change it to client.connect(msg)

where msg is a string containing MAC address.

Then we can have a clean result. You won't see the reconnecting message and mqttbox won't get any strange messages.



So far, we know how to send a message from a device to the broker. Next step is to know the way to receive a message from the broker. That message can be generated by mqttbox.



You can see the message displayed in the terminal. The message is handled by the *callback* function. You can implement your own task to process the message.

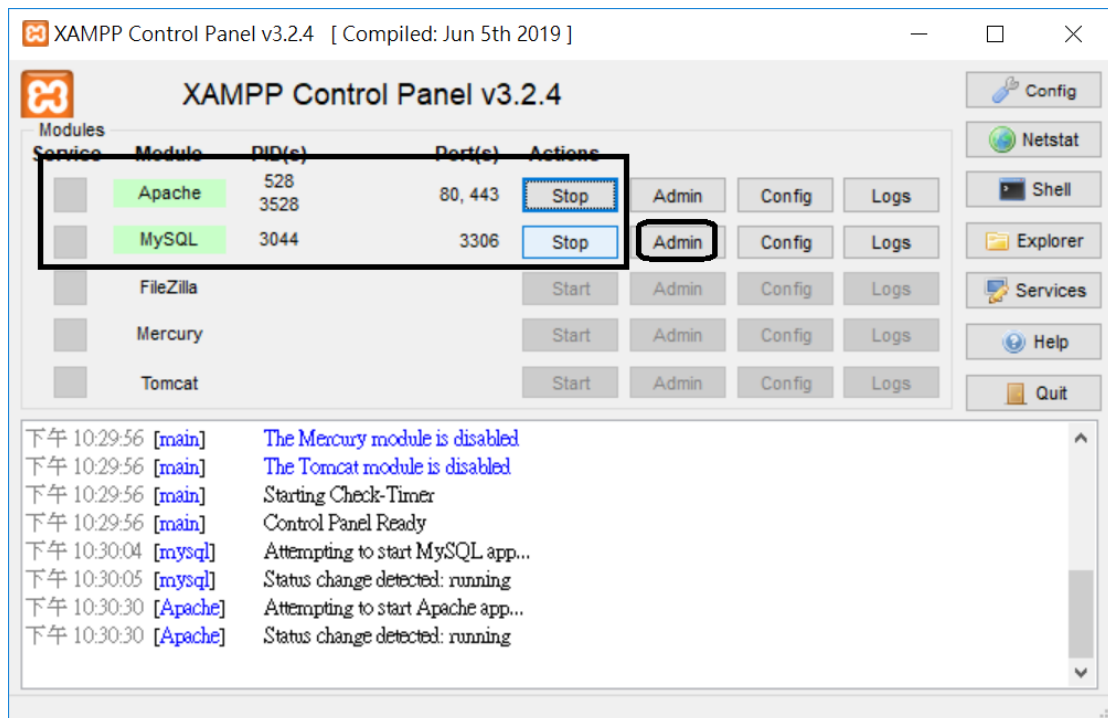
```
void callback(char* topic, byte* payload, unsigned int length)
```

\*\*\*\* important remark \*\*\*\*

The message is handled by "client.loop();" We need to put this inside the loop of Arduino. Remember don't put a LONG busy waiting in your loop. Otherwise, the response time of callback will be very long.

#### 4. XAMPP: [https://www.apachefriends.org/zh\\_tw/download.html](https://www.apachefriends.org/zh_tw/download.html)

Install it for database. Just install xampp.



Start the two service and click mysql admin to open phpMyAdmin. Search web pages to learn how to create tables.

TEMP.sql: create a table called temp to store temperature values.

TEMP\_CHECK.sql: create a table called temp\_check to store query information which can be used as footprint tracking.

5. Install python. I use python 3.7.  
<https://www.python.org/downloads/windows/>
6. Information flow of auto-thermometer  
 Raspberry pi publish (submit.py) → broker → temp\_submit.py (store data to "temp" table.
7. Information flow of card screener  
 WeMos mini publish () → broker → temp\_ask.py (store data to "temp\_check" table and query "temp" table.  
 Temp\_ask.py publish → broker → WeMos mini sub (trigger callback function)
- 8.