Arduino 安裝說明書

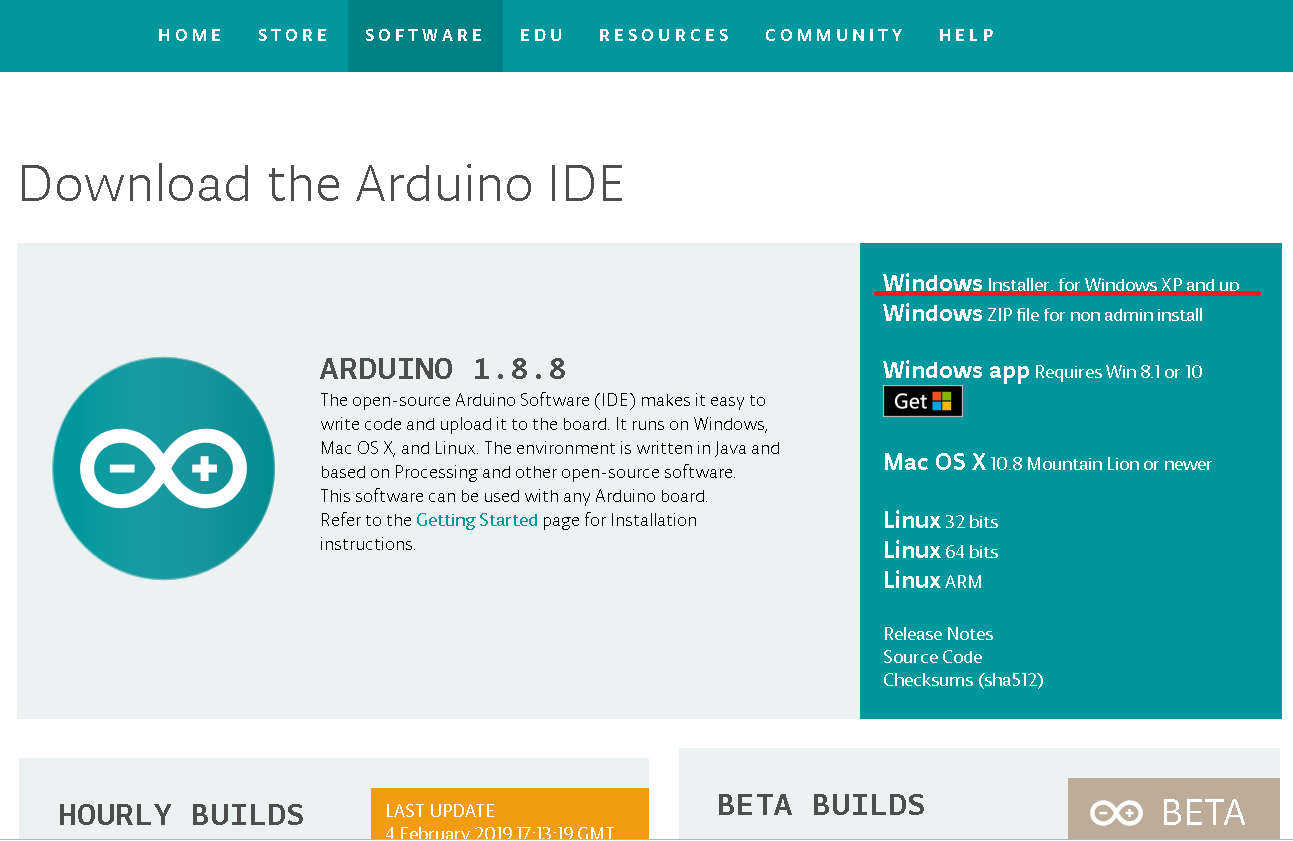
**一、下載非官方開發板的支援網址：**

**二、安裝ESP8266(系列) 驅動函式庫：**

**三、將開發板ESP8266連接到電腦：**

官方網站：<https://www.arduino.cc/>

官網安裝檔下載：<https://www.arduino.cc/en/Main/Software>

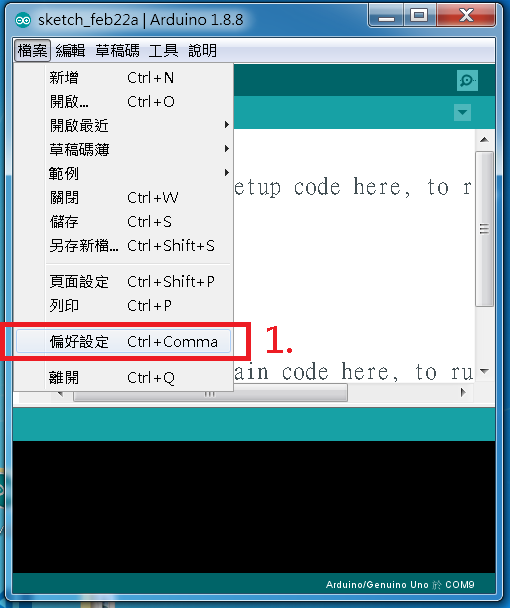




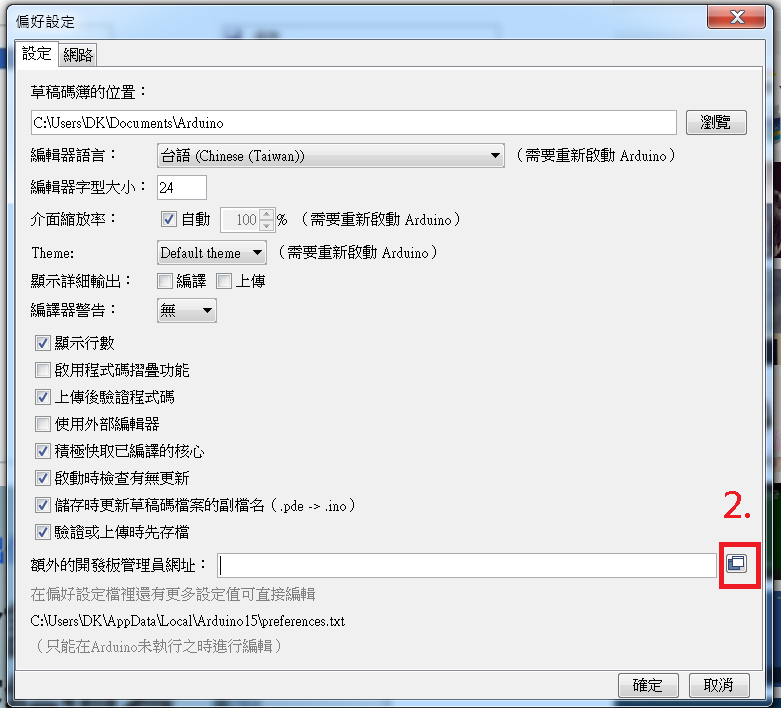
安裝完畢後，點選桌面程式啟動

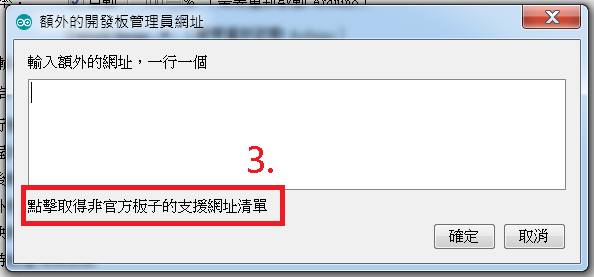
**一、下載非官方開發板的支援網址：**

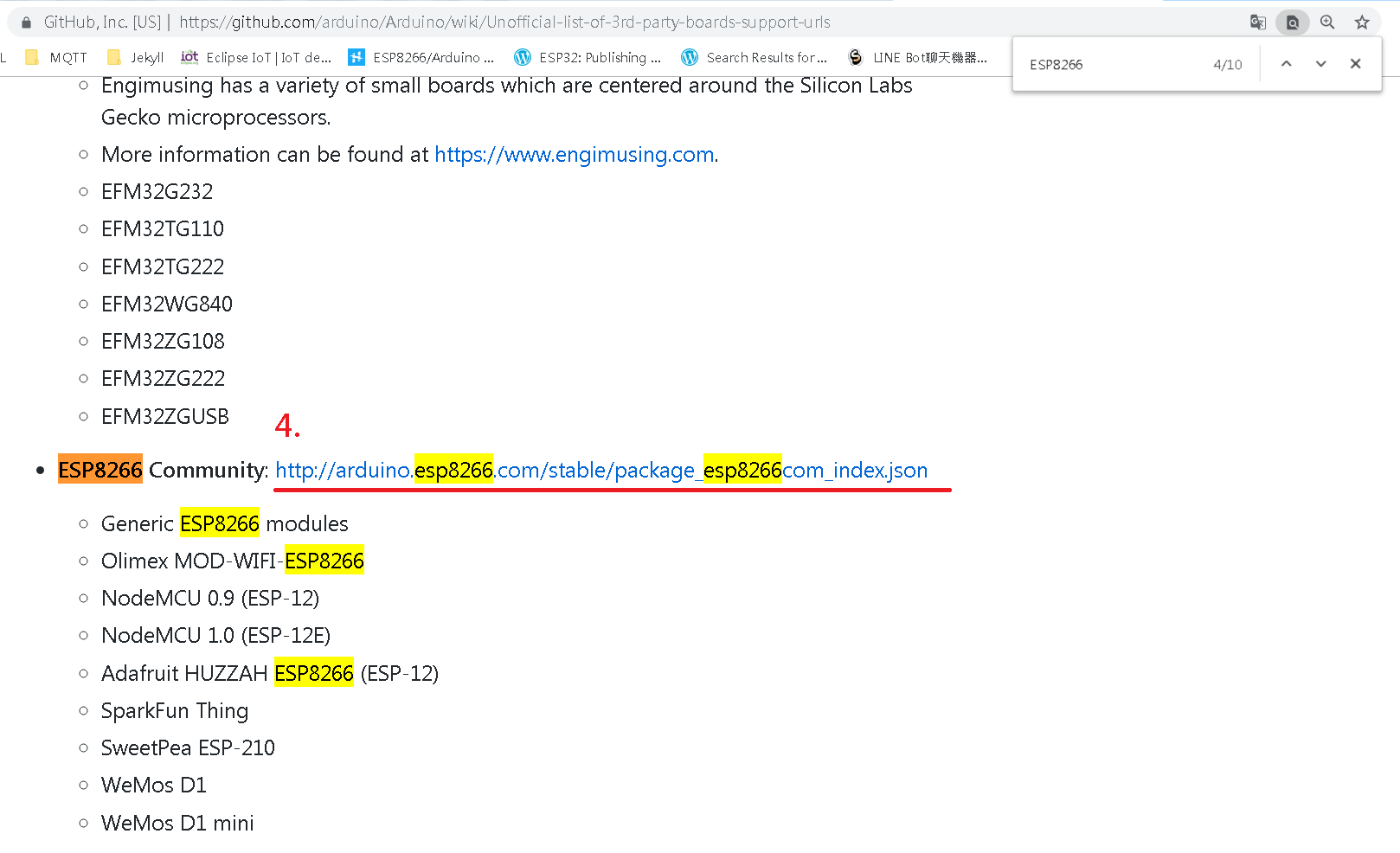
1. 檔案→偏好設定→
2. 額外的開發板管理員網址→
3. 點擊取得非官方版子的支援網址清單→
4. 在GitHub網頁：搜尋開發板名稱(ESP8266)並找到「.json」的網址
5. 將開發板(ESP8266)的網址貼到上面 儲存



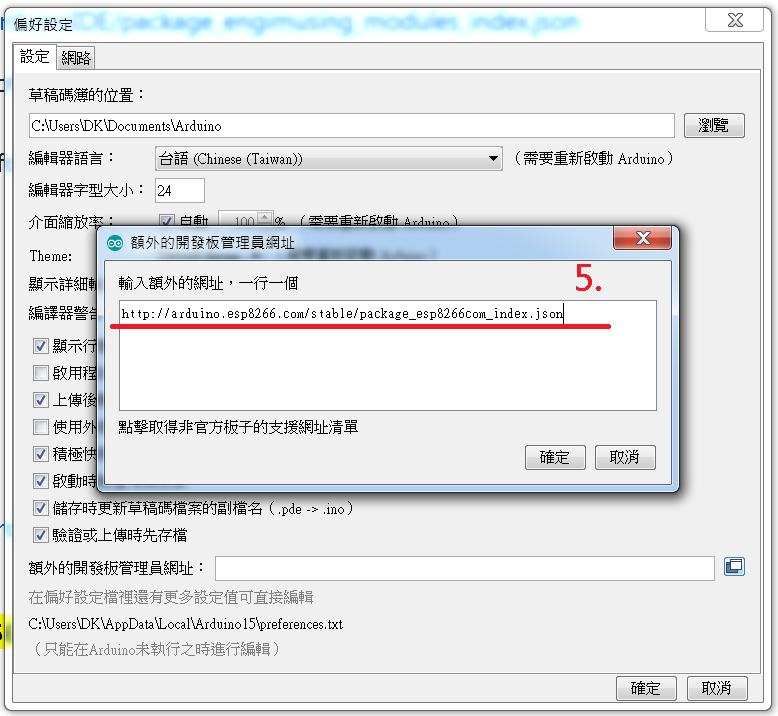








搜尋開發板名稱(ESP8266)並找到「.json」的網址



**二、安裝ESP8266驅動函式庫：**

1. 點取工具列的「工具」

2. 開發板

3. 開發板管理員

4. 搜尋「esp8266」並安裝

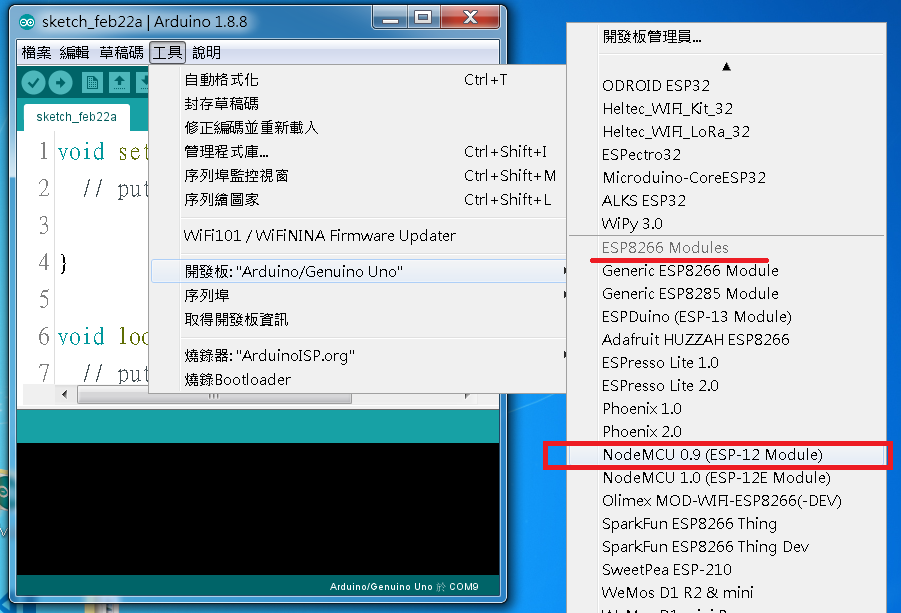


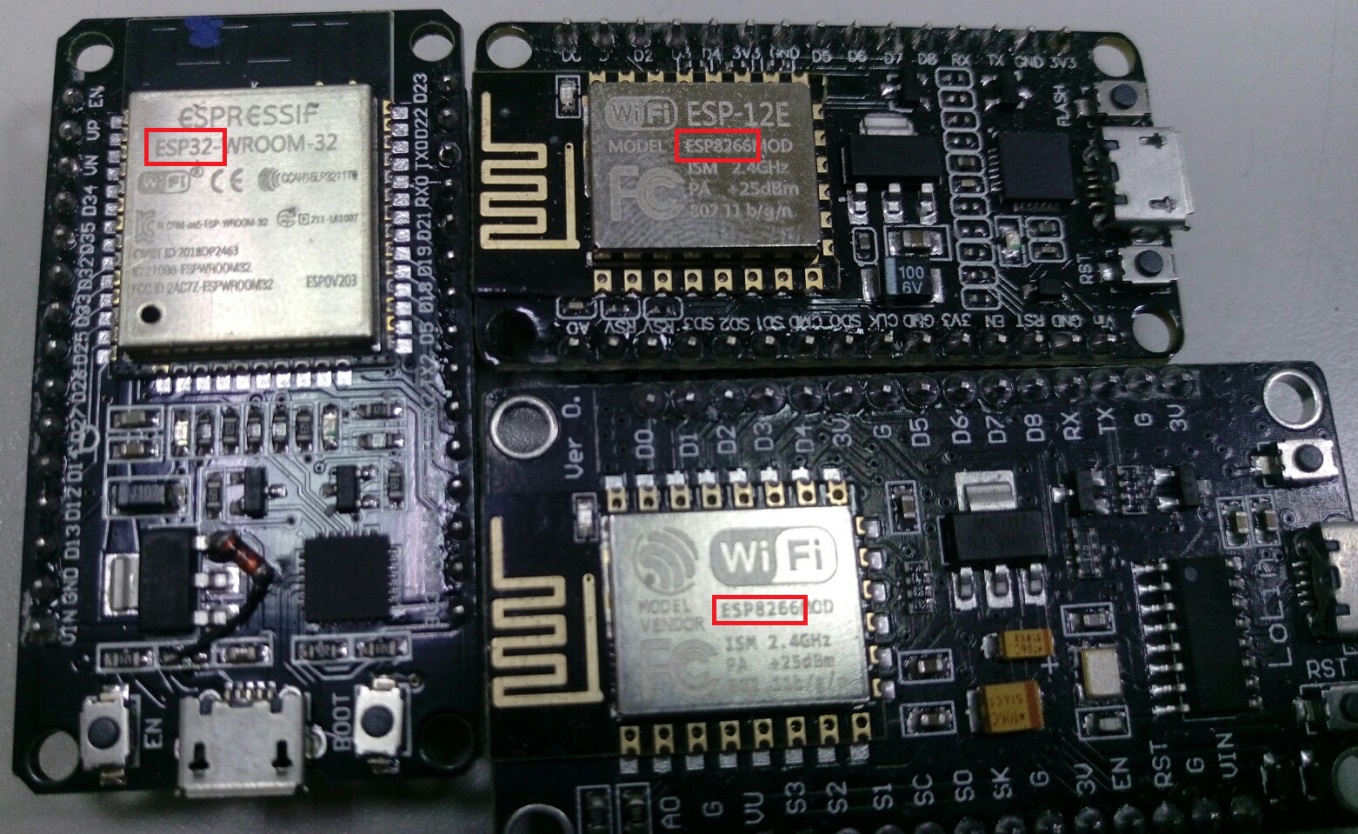


**三、將開發板ESP8266連接到電腦：**

1. 確認板子型號
2. 安裝ESP8266驅動 (二、步驟已安裝)
3. 將板子接上電腦之後，「工具」的「開發板」的裡面，會出現ESP8266 Modules的板子類型
4. 選取NodeMCU 0.9(開發板型號)
5. 設定序列埠COM x



**開發板 (正面):**



**開發板 (背面):**



**開發板 (正面):**

ESP32晶片系列的開發板，跟ESP8266晶片系列的開發板。

晶片上面有清楚寫晶片的系列:

例如：ESP8266晶片系列的第ESP-12E版本

**開發板 (背面):**

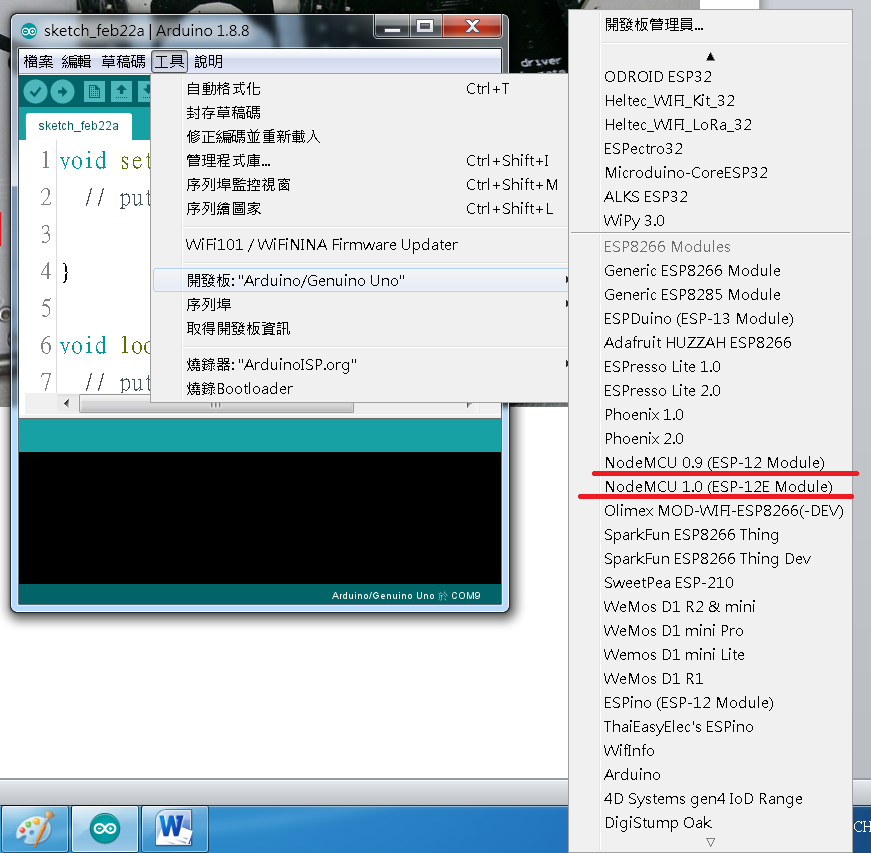
因為Ardiono的硬體也是開源的，所以有很多種類型的開發板

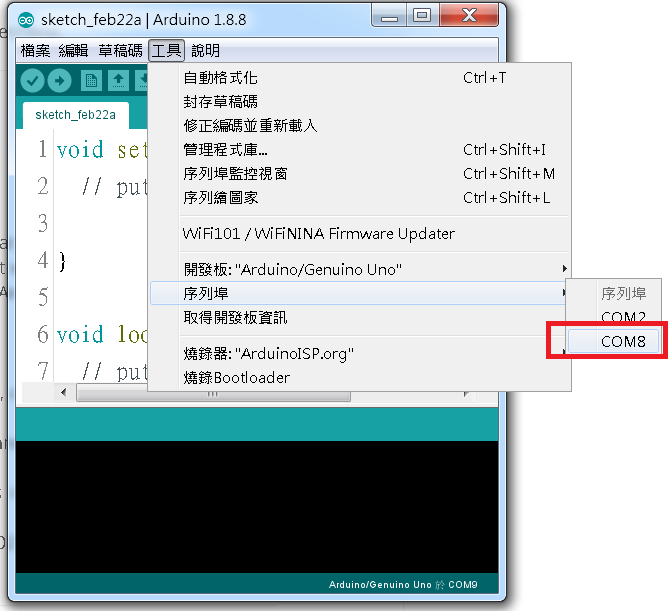
註：會因為不同廠商生產，完全不一樣

例如下面圖片：

ESP8266(晶片)系列的開發板：NodeMCU →廠商：AMICA

ESP8266(晶片)系列的開發板：NodeMCU →廠商：LOLIN

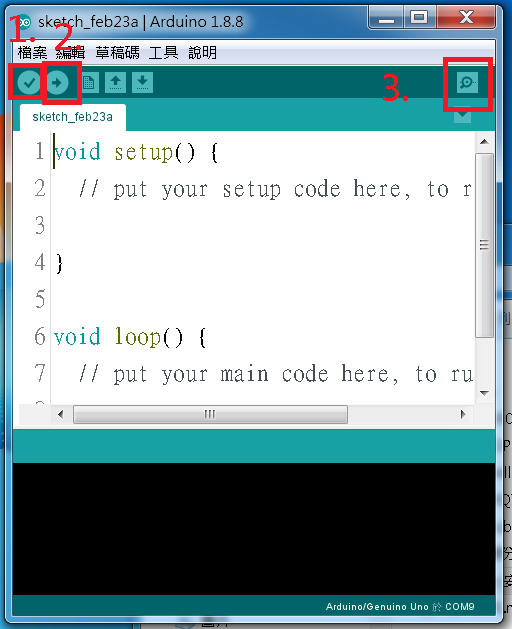




Arduino 操作方式：編譯與燒錄

介面說明：

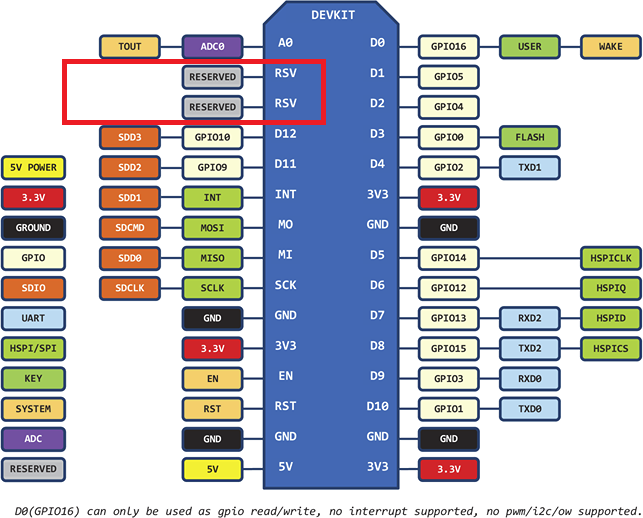
1. 驗證
2. 上傳
3. 序列埠監控視窗

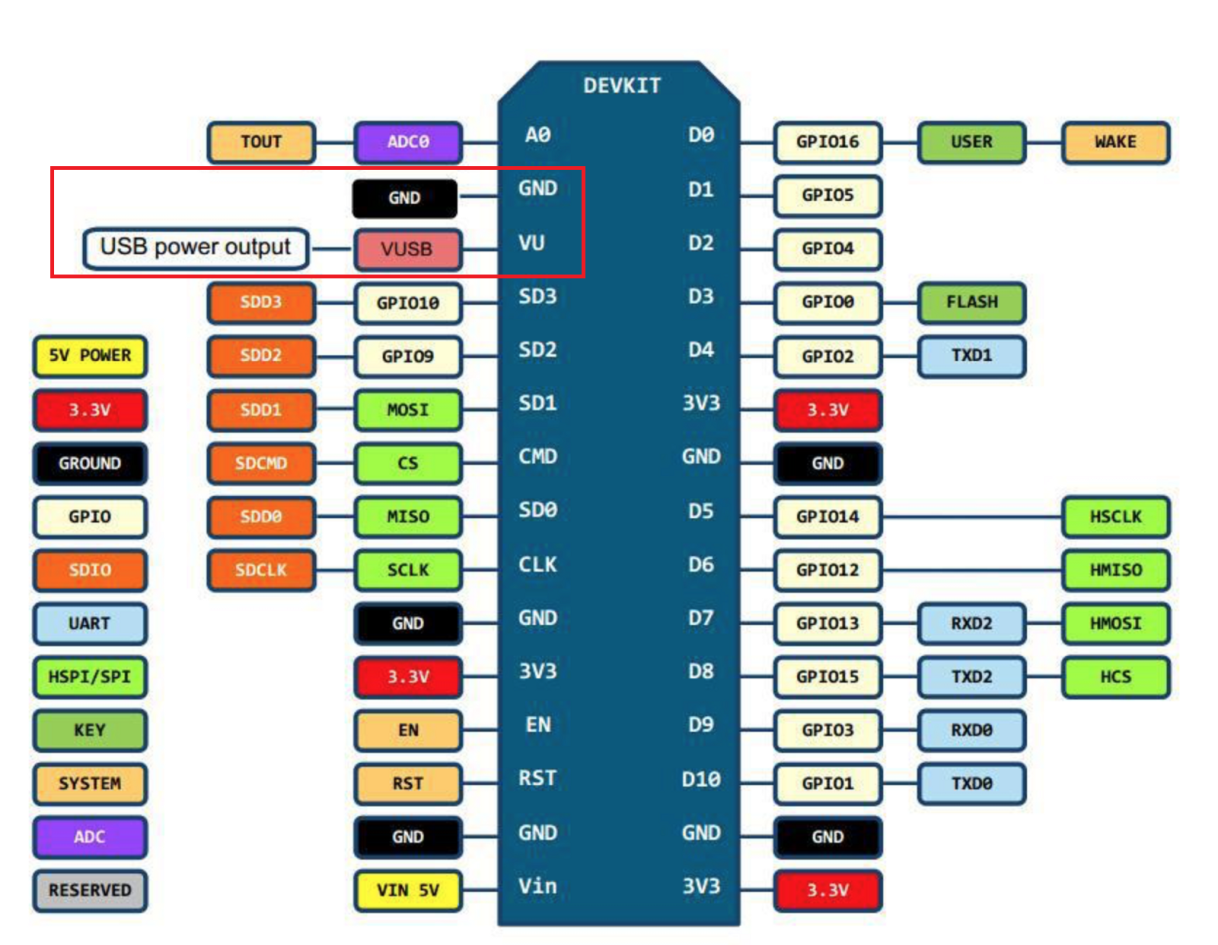


使用第一步：從工具選單，選擇開發板型號，以及序列埠

ESP8266 兩種開發板：腳位的不同

上網查詢方式：搜尋「ESP8266 GPIO」或「ESP8266 datasheet」

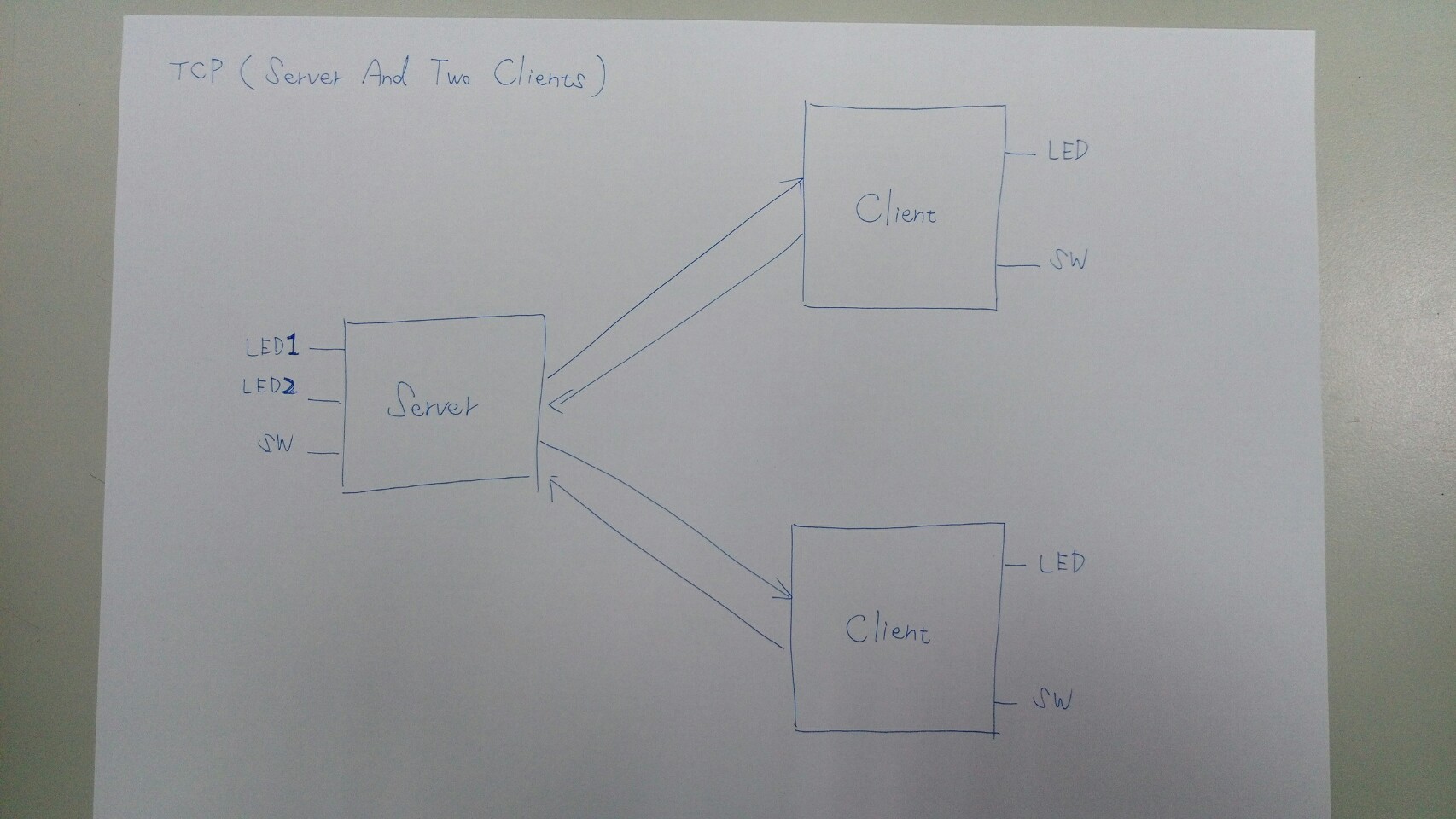




**通訊架構**

**總共3種!**

**第一種：開發板 對 開發板 傳無限信號**



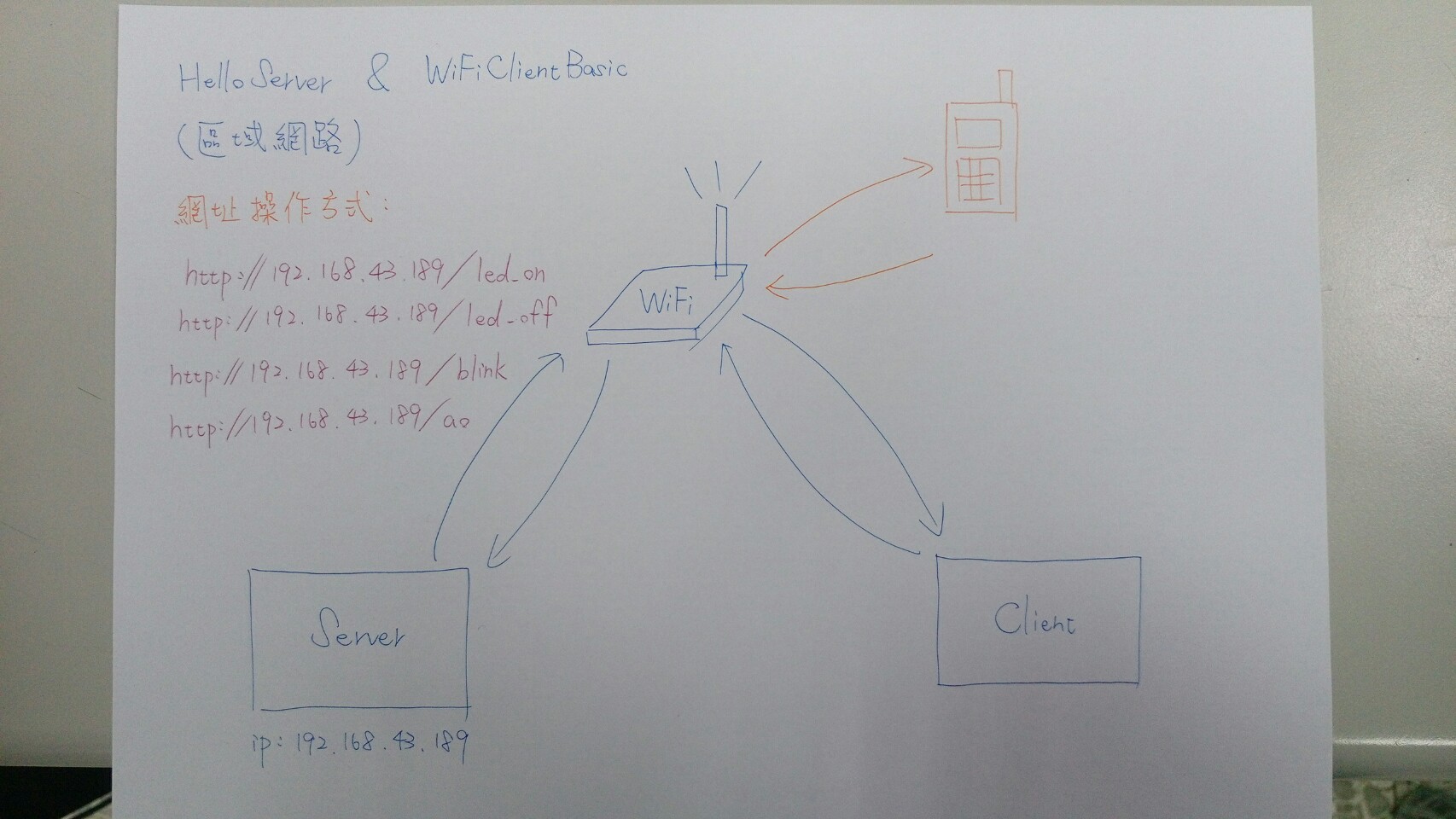
說明

WiFi晶片直接對另一個WiFi晶片傳遞資料。

YouTube影片操作：

<https://www.youtube.com/watch?v=a7wEuzupfQc>

**第二種：區域網路**



**說明：**

在同一個網域，外面連不進來、裡面也傳不出去。

**例如：**

連線至同一個WiFi分享器，有連線的開發板、或電腦、或手機，都可以發送資料到指定ip位置。每一個裝置都可以對它下命令。

**命令方式：輸入網址**

<http://192.168.43.189/led_on>

<http://192.168.43.189/led_of>

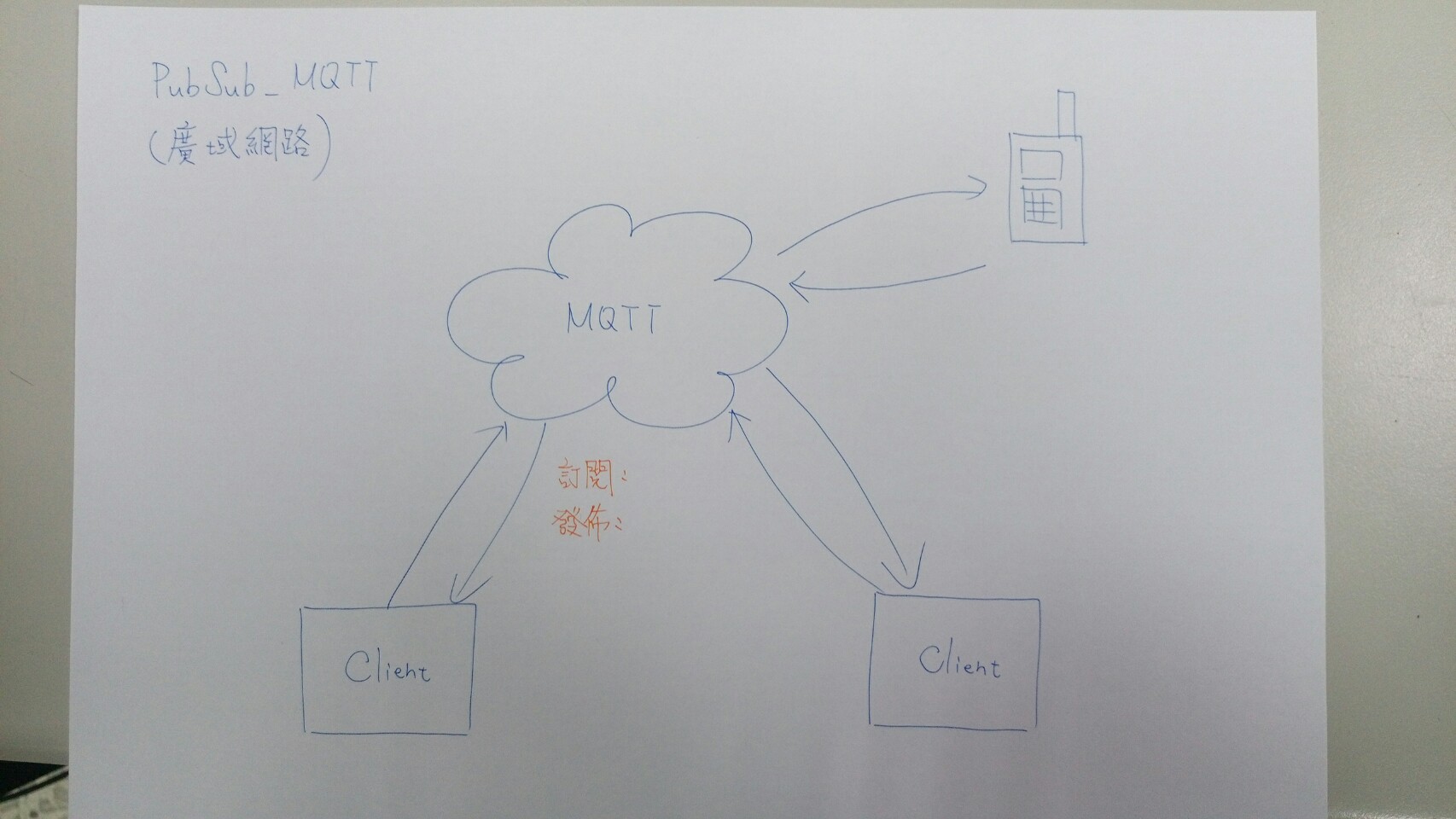
<http://192.168.43.189/blink>

<http://192.168.43.189/a0>

「/led\_on」 跟 「/blink」 是對開發板子的LED控制

「/a0」 是讀取開發板 腳位a0的電壓

**第三種：MQTT廣域網路**



詳細教學網址：<https://swf.com.tw/?p=1002>

說明：

所有的資料都 **不是** 直接傳送到指定開發板，每一筆資料必須「發佈」在MQTT的Broker上面。然後只要有「訂閱」此名稱的，都可以收到資料。

任意有線/無線網路的裝置：板子、手機、電腦、網頁都是跟MQTT公司的Broker連接。這個Broker就是一個必經的「中間者」。

「發佈資料」要向MQTT Broker發佈資料，

「訂閱資料」要向MQTT Broker訂閱資料。

**MQTT代理商(Broker)**

1. 加密方式：

每一家身分驗證方式都完全不一樣。

1. 撈資料的方式：
2. API應用：

有一些代理商有現成的儀表板應用。

代理商1：CloudMQTT

<https://www.cloudmqtt.com/plans.html>

代理商2：MQTTRoute

<https://www.bevywise.com/mqtt-broker/>

**各種MQTT效能使用參考網站：**

MQTT通訊協議簡介：

1. <http://designer.mech.yzu.edu.tw/articlesystem/article/compressedfile/(2016-07-15)%20%E7%AC%AC%E4%B8%89%E7%AB%A0%20MQTT%E9%80%9A%E8%A8%8A%E5%8D%94%E8%AD%B0.aspx?ArchID=2621>
2. <https://github.com/mqtt/mqtt.github.io/wiki/servers>

Arduino Client for MQTT：

<https://pubsubclient.knolleary.net/api.html>

C# MQTT資料的抓取

1. <http://misccp3.cnu.edu.tw/myblog/blogMessage.aspx?blog_id=297>
2. <https://yazelin.github.io/usc2017nsp/week2.html>

MQTT性能測試

1. <https://blog.csdn.net/educast/article/details/78352641>
2. <https://dotblogs.com.tw/justacoder/2016/06/01/mqttfirstmeet>

**各種MQTT代理商特性比較**

GitHub：<https://github.com/mqtt/mqtt.github.io/wiki/server-support>

MQTT学习（六）--各类MQTT代理服务器特性对比（译文）：<https://blog.csdn.net/lordwish/article/details/85061687>

| **Server** | **QoS 0** | **QoS 1** | **QoS 2** | **auth** | **bridge** | **$SYS** | **SSL** | **dynamic topics** | **cluster** | **websockets** | **plugin system** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2lemetry | ✔ | ✔ | ✔ | ✔ | ✔ | § | ✔ | ✔ | ✔ | ✔ | ✘ |
| Apache ActiveMQ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Apache ActiveMQ Artemis | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Bevywise IoT Platform | ✔ | ✔ | ✔ | ✔ | rm | ✔ | ✔ | ✔ | ✔ | ✔ | rm |
| emitter | ✔ | § | ✘ | ✔ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✘ |
| emqttd | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| flespi | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✘ |
| GnatMQ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✘ | ✔ | ✘ | ✘ | ✘ |
| HBMQTT | ✔ | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ |
| HiveMQ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| IBM MessageSight | ✔ | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | § | ✔ | ✘ |
| JoramMQ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Mongoose | ✔ | ✔ | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| moquette | ✔ | ✔ | ✔ | ✔ | ? | ? | ✔ | ? | rm | ✔ | ✘ |
| mosca | ✔ | ✔ | ✘ | ✔ | ? | ? | ? | ? | ✘ | ✔ | ✘ |
| mosquitto | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | § | ✔ | ✔ |
| MQTT.js | ✔ | ✔ | ✔ | § | ✘ | ✘ | ✔ | ✔ | ✘ | ✔ | ✘ |
| MqttWk | ✔ | ✔ | ✔ | ✔ | ✔ | ? | ✔ | ✔ | ✔ | ✔ | ✘ |
| RabbitMQ | ✔ | ✔ | ✘ | ✔ | ✘ | ✘ | ✔ | ✔ | ? | ? | ? |
| RSMB | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✔ | ✘ | ✘ | ? |
| Software AG Universal Messaging | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✔ | ✔ | ✔ | rm | ✘ |
| Solace | ✔ | ✔ | ✘ | ✔ | § | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ |
| SwiftMQ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✘ | ✔ |
| Trafero Tstack | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✔ | ✔ | ✘ | ✘ | ✘ |
| VerneMQ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| WebSphere MQ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ? | ? | ? |

CloudMQTT <https://www.cloudmqtt.com/>

|  |  |
| --- | --- |
| 價格(美金) | 功能 |
| 免費 | * 5 users/acl rules/connections * 10 Kbit/s |
| 5 | * 25 users/acl rules/connections * 20 Kbit/s * 3 bridges * Support by e-mail |
| 19 | * Up to 100 connections * No artificial limitations * Support by e-mail |
| 99 | * Up to 1 000 connections * No artificial limitations * Support by e-mail |
| 299 | * Up to 10 000 connections * No artificial limitations * Support by e-mail * Support by phone |

MQTTRoute <https://www.bevywise.com/mqtt-broker/pricing.html>

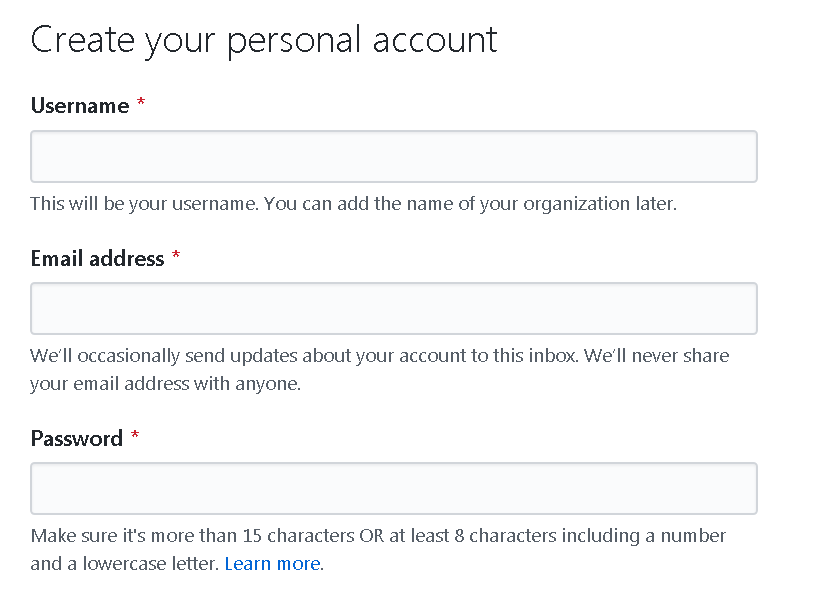
|  |  |
| --- | --- |
| 價格(美金) | 功能 |
| 699 | * Upto 1000 Clients * SSL * Rule Engine * Multiple Username/password * - * Email Support * 24 Hours response time * 1 year FREE Support & Updates |
| 1999 | * Upto 1000 Clients * SSL * Rule Engine * Multiple Username/password * - * Email Support * 24 Hours response time * 1 year FREE Support & Updates |
| 3499 | * Upto 1000 Clients * SSL * Rule Engine * Multiple Username/password * - * Email Support * 24 Hours response time * 1 year FREE Support & Updates |
| Contact for Price | * Upto 1000 Clients * SSL * Rule Engine * Multiple Username/password * - * Email Support * 24 Hours response time * 1 year FREE Support & Updates |

**網頁GitHub使用**

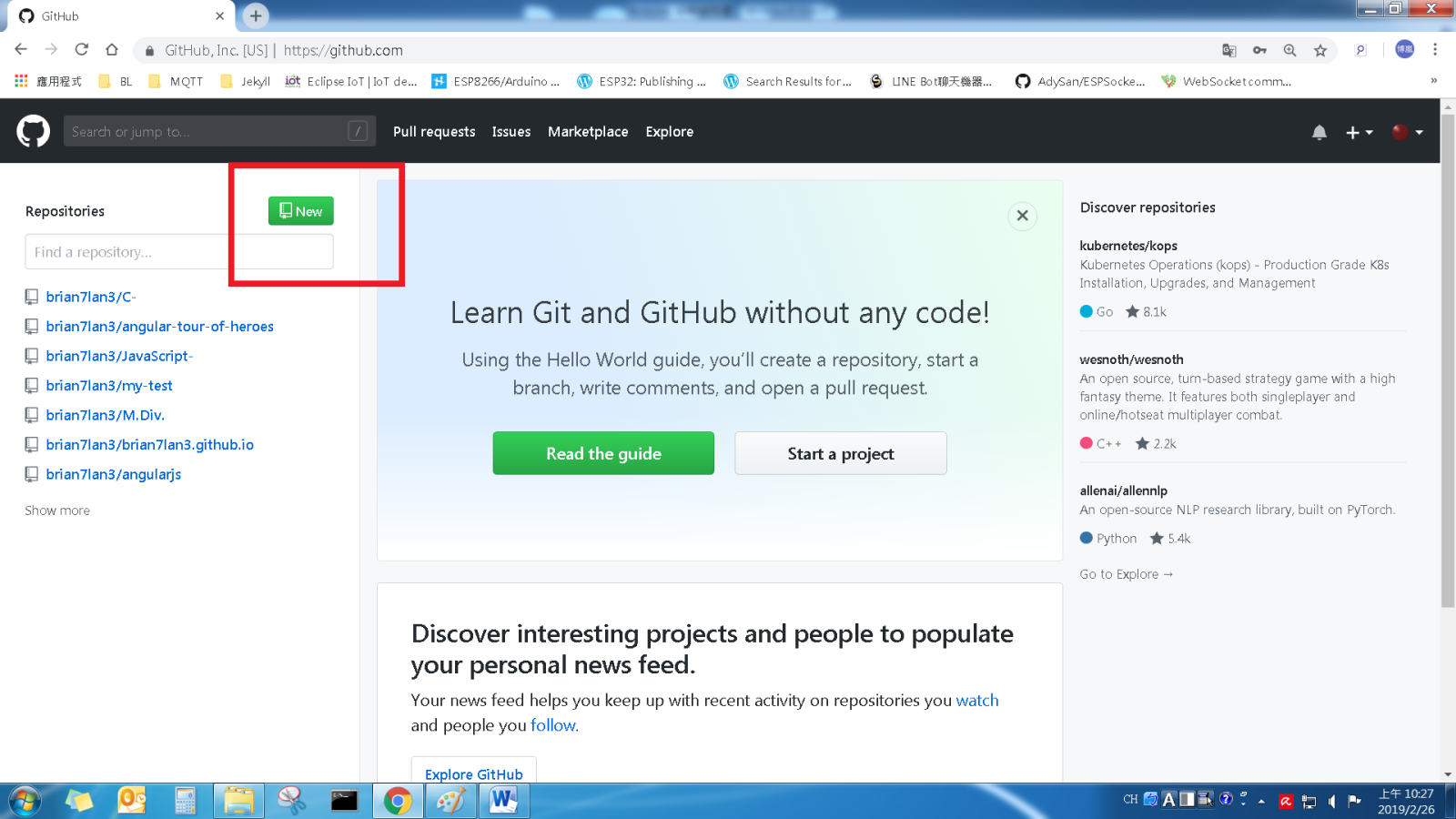
**一、註冊GitHub**

1. 搜尋GitHub，首先進入官網，並註冊為會員

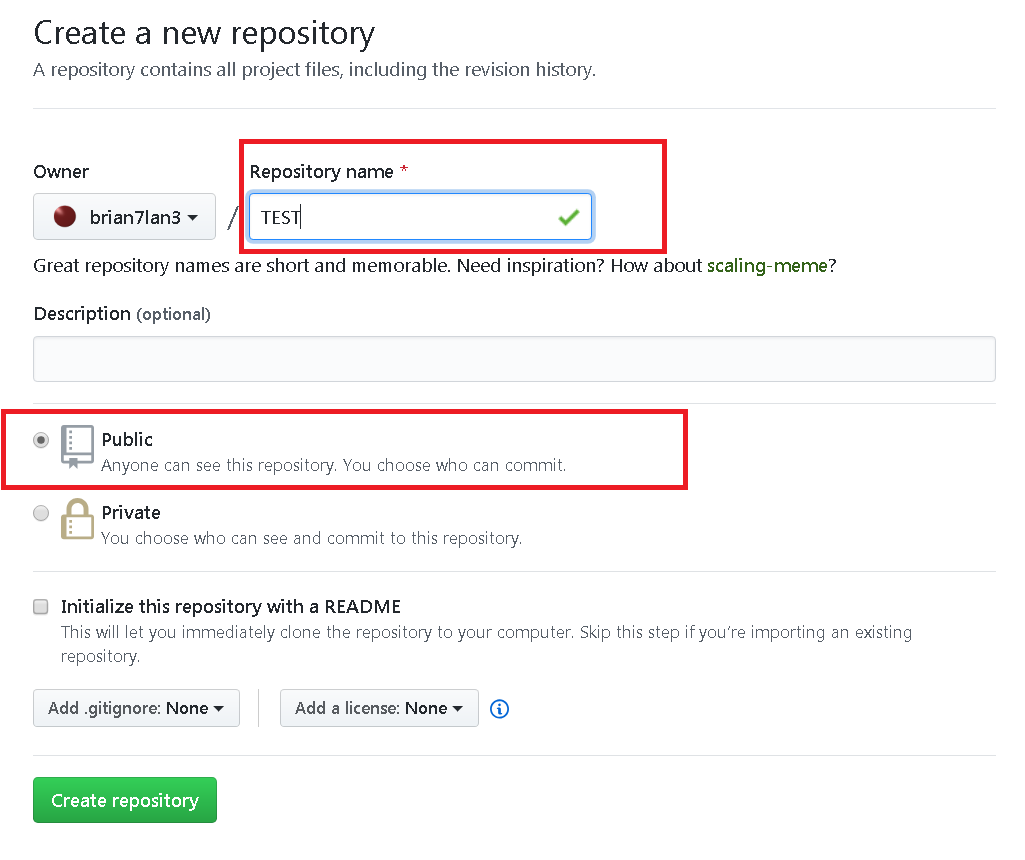




2. 登入帳號之後，點擊「New」創建新專案

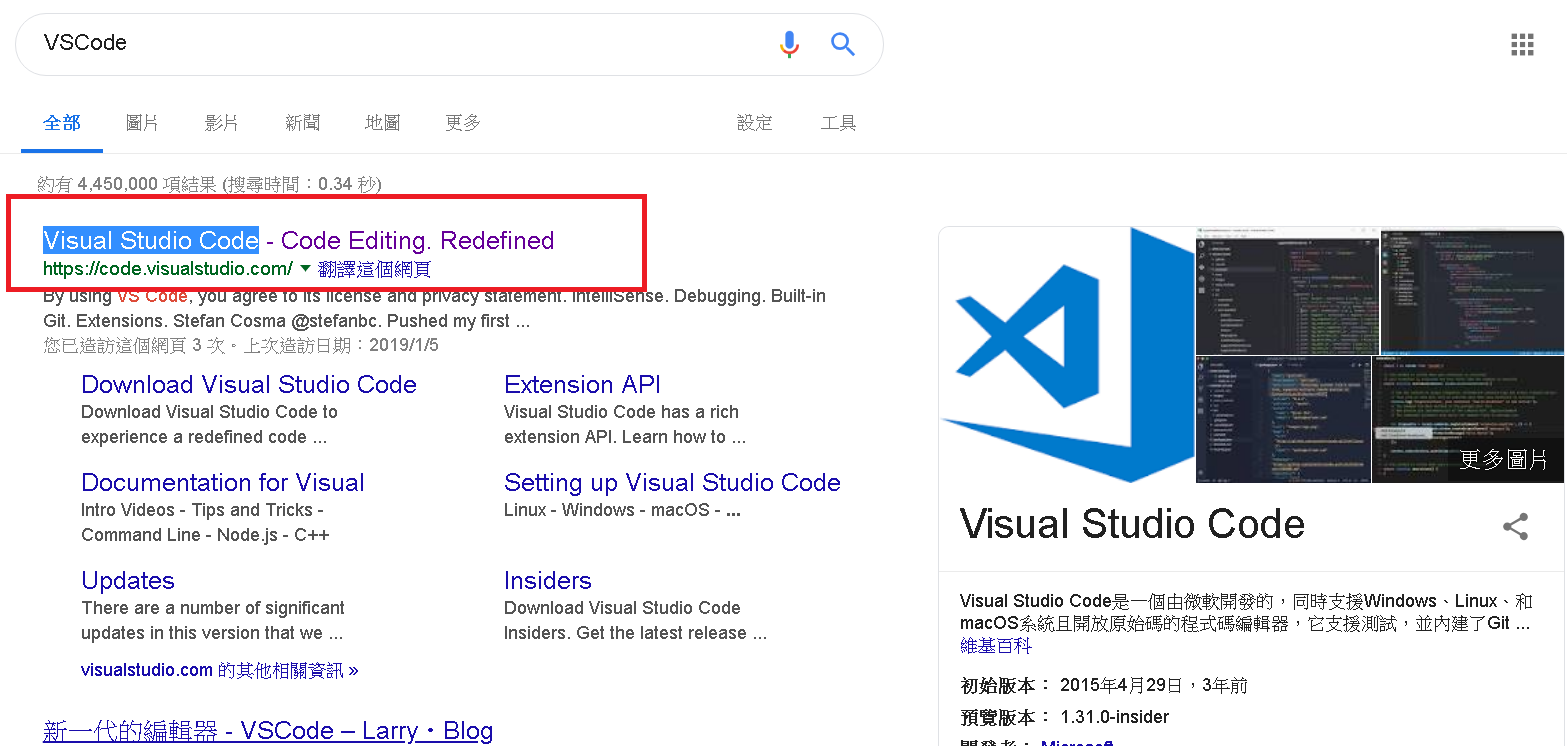


3. 輸入專案名稱，並設定公開Public就可以了。

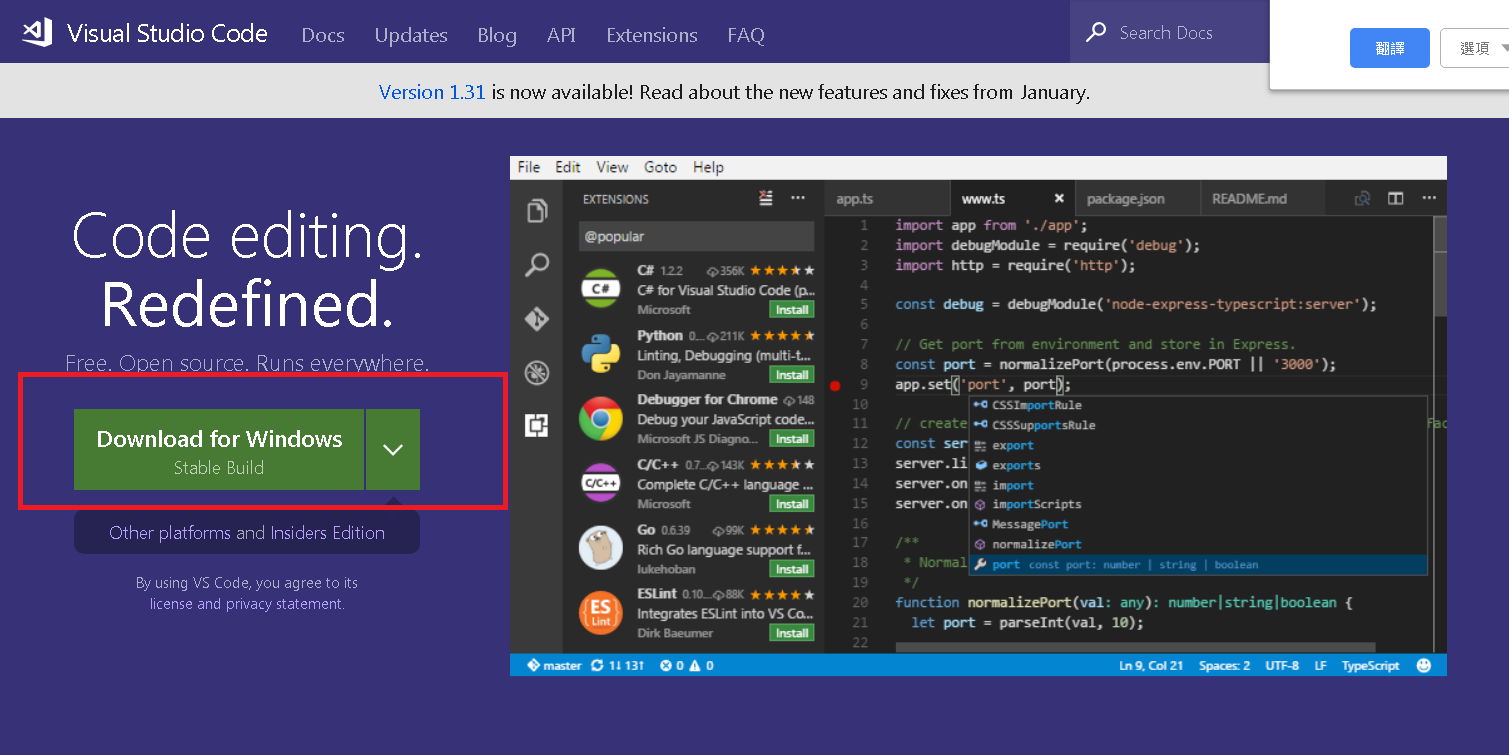


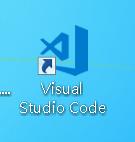
**二、安裝Visual Studio Code (VSCode)**

1. 搜尋「VSCode」，並進入官網



2. 下載並安裝 VSCode



桌面圖示：

**三、安裝Git**

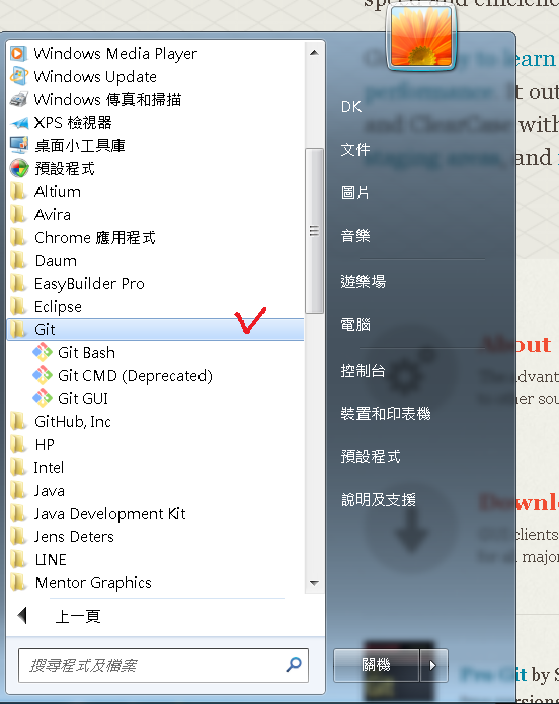
1. 搜尋「Git」並進入官網



2. 下載及安裝Git



3. 安裝完成之後，電腦應用程式就有「Git」及「GitHub」



說明：

A. 什麼是「Git」：Git就是「版本控制軟體」

當程式碼做到某一時刻，可以記錄「當下」程式碼的狀態，例如：今天早上程式碼寫了加法器功能，就可以紀錄當下目前的程式碼，並任意記錄目前程式碼的「描述」。(下面有圖解說明)

B. 什麼是「GitHub」：GitHub就是「存放程式碼的地方」

它的特色呢? 是可以在上面看到程式碼，並且看到程式碼使用Git在每一個階段時的程式碼狀態。

舉例說明：

今天早上我的程式碼，寫了加法器功能，我就用Git記錄當下的狀態，並且自己描述：「新增加法器功能 顯示方式還需要調整」

然後過了30分鐘之後，我的加法器功能寫的比較完善了，我就再用Git紀錄當下的程式碼狀態，描述：「加法器功能OK 完善」

這很方便的可以看到每一個階段的程式碼與描述，並且在GitHub上面可以觀看、取得每一個階段的程式碼，看別人的功能是怎麼寫的，每一個步驟都有。

**四、程式碼上傳至GitHub**

接下來用VSCode把程式碼上傳至GitHub

1. 先開啟VSCode並寫好程式碼

2. 在VSCode有Git的輔助外掛必須先安裝

3.

**參考書籍**

書籍：物聯網技術理論與實作

