

國立暨南國際大學電機系

專題(上)期末報告

智慧溫室

Base on IoT Smart Greenhouse

學生：109323032 陳旻偉

109323036 黃柏翰

指導教授：郭耀文 教授

中華民國 112 年 1 月

目錄

第 1 章 緒論

第 2 章 IoT Unit

2-1 ESP8266 D1

2-2 BME 280

2-2-1 I2C 簡介

2-3 Moisture sensor

2-3-1 目前採集之數據

2-4 BH1750

2-4-1 目前採集之數據

2-5 Relay

2-5-1 Relay

2-5-2 LED

2-5-3 Pump

2-5-4 fan

第 3 章 Tasmota

3-1 Tasmota 介紹

3-2 如何連上開發版

3-3 如何連上 modules

第 4 章 網路協議(MQTT)

4-1 MQTT 介紹

4-2 Public MQTT

4-3 Homeassistant MQTT

第 5 章 使用者介面(Homeassistant)

5-1 Raspberry Pi

5-2 如何蒐集觀測數據(Tasmota)

5-3 如何自動化

5-4 SSH

第 6 章 下學期進度

第 1 章 緒論

我們正在設計一個基於物聯網的植物溫室監控系統，以避免多肉植物(註一)的徒長(圖一)問題，造成徒長的因素有以下兩點，第一點「水太多」以及第二點「陽光太少」。這個系統可以透過物聯網技術，在任意地點隨時監控溫室內的光線、溫溼度、給水等環境因素，並由系統自動調整這些參數，以確保在台灣的任何環境(包括室內、平地、高海拔地區)中，都能滿足觀賞植物(多肉植物)的存活和成長需求。我們的目標是建立一個小型的溫室，能夠在任何情況下都能提供植物適當的環境條件。



圖 1 多肉植物徒長現象(左)

基於這個目標，我們可以將系統設計成能夠監控多種環境因素，並根據植物的需求調整這些因素。例如，系統可以監控光線強度，並在光線不足時自動打開燈具。它也可以監控溫度和濕度，並在需要的時候自動調整加溫或加濕的設備。此外，系統可以監控土壤濕度，並在需要的時候自動給植物灌水。

透過自動監控和調整這些環境因素，我們可以確保植物能夠在適當的環境中生長，避免造成徒長的問題。

註一：多肉植物，通指景天科、番杏科、龍舌蘭科、大戟科等，具肥大化之莖、葉或根之植物。其多具有耐旱怕澇、對溫度光照有一定要求等特質，且易從外表觀測是否成功滿足植株之存活需求，故而採用之作為模擬對象。

第 2 章 IoT Unit

2-1 ESP8266 D1

ESP8266 D1 R1(WeMos D1 R1) 是 WeMos 基於 ESP8266EX 開發的兼容 Arduino UNO (IDE 整合開發環境) 之物聯網開發板。ESP8266Ex 處理器本身自帶 Wi-Fi，亦可直接用 micro USB 供電與傳輸，並有 11 個數位 I/O

接腳與 1 個類比輸入接腳。

支援 UART、SDIO、SP、I2C、I2S、IR Remote Control 等各種通訊介面以及 GPIO(通用接腳)、PWM (脈衝調製模擬類比)等，自帶 Wi-Fi、容易上手、低功耗、兼容性高且物美價廉，是 IoT 開發者的不二選擇。

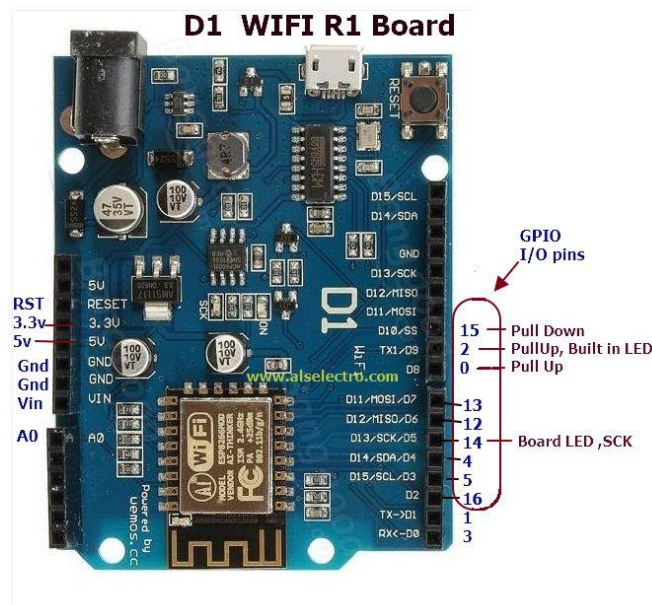


圖 2 ESP8266 D1

2-2 BME 280

GY-BME280，為高精度溫度、濕度、大氣壓強傳感器。工作電壓 3.3V-5V，由 I2C 或 SPI 串接序列。將 VCC 接上 5V 或是 3.3V，GND 接 GND，SDA 接 SDA，SCL 接 SCL。

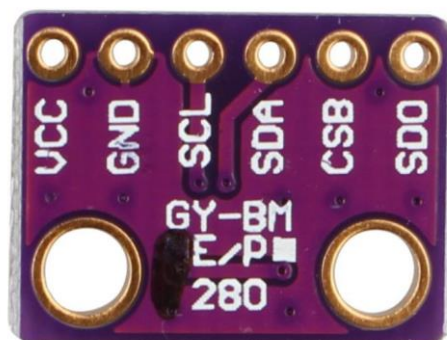


圖 3 BME280

2-2-1 I2C 簡介

I²C (Inter-Integrated Circuit) 字面上的意思是積體電路之間，是一種使用多主從架構之串列通訊匯

流排。使用兩個雙向開漏線：串列資料線（SDA）和串列時脈線（SCL）便可完成通訊，匯流排上有兩種主從（master&slave）結點，由地址對應特定主從；主為發出 SCL 並與從通訊，從則是接收 SCL 並回應主節點，透過回應(ACK)確認相應主從是否獲得有效訊號。

由匯流排各自高低電頻變化，產生幾個基礎訊號：

1. 有效：當 SCL 與 SDA 皆為 high
2. 無效：其中一個為 low
3. 起始(Start)：當 SCL 為 high，SDA 由低到高
4. 暫停(Stop)：當 SCL 為 high，SDA 由高到低
5. 應答(ACK)/ 非應答(NACK)

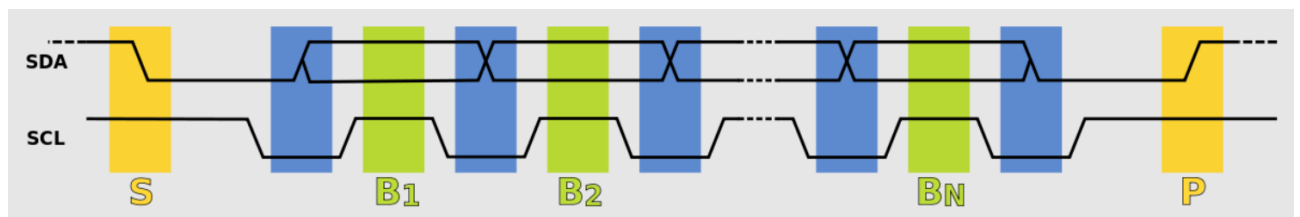


圖 4 I2C 實體層訊號狀態

當發送完 8bits 後，發送端(不分主從)需讀取接收端的 ACK(low)或 NACK(high)回應

Ex:

指令：主讀到 ACK，則繼續傳輸

主讀到 NACK，指從無回應，放棄指令

資料：從讀到 ACK(主:收到資料)，則繼續傳輸

從讀到 NACK(要求結束)，則傳輸 1 結束訊號

I2C 常用在簡單應用上如改變音量、溫度或風扇轉速等。由於接腳少，能減少尺寸、重量與電源的消耗，對行動電話等小型產品尤為重要。

2-3 Moisture sensor

Grove-Moisture sensor v.1.4，工作電壓 3.3V-5V，透過電阻率之變化測量土壤濕度。VCC 接 3.3V-5V，GND 接 GND，注意接腳 sig 需接上 A0 類比接口，Tasmota module 亦須注意。

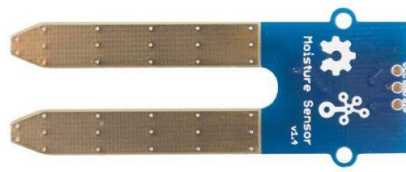


圖 5 Moisture Sensor

2-3-1 目前採集之數據

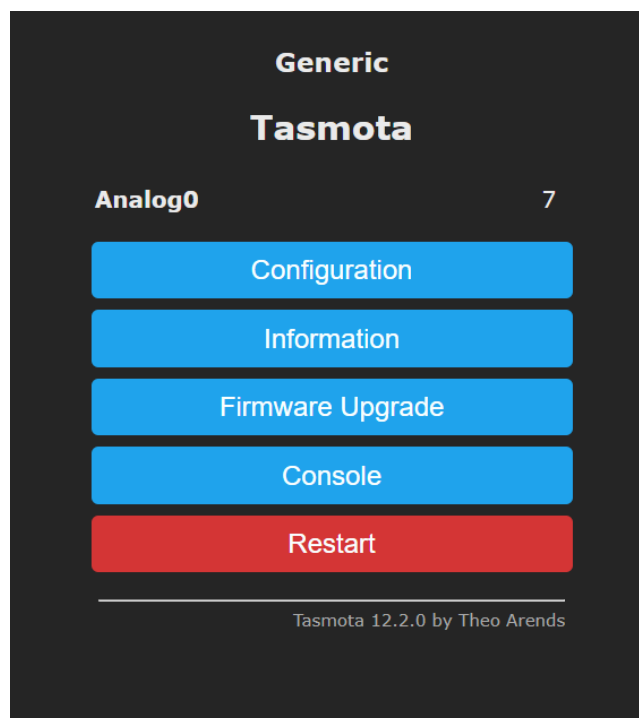


圖 6 測試普通空氣電阻率變化進而推導之濕度結果

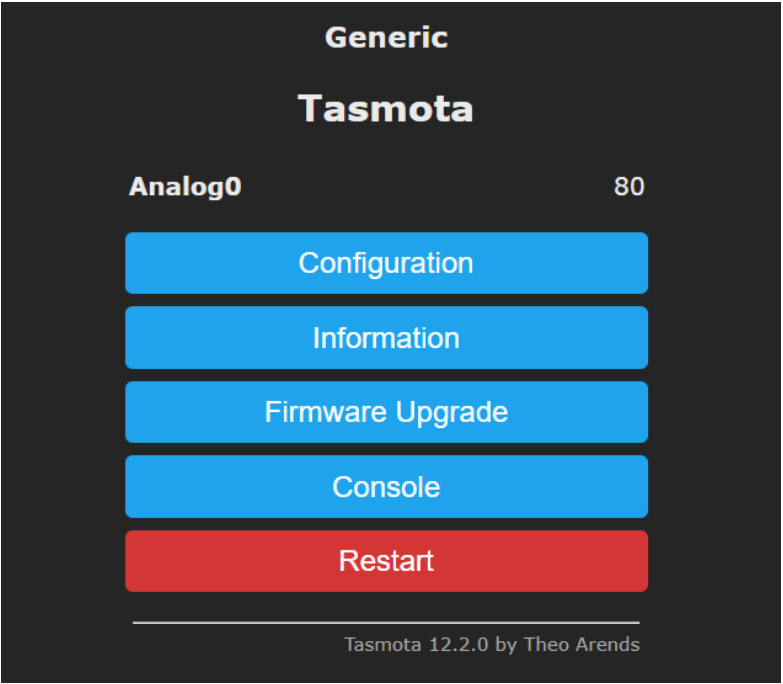


圖 7 手指測試結果

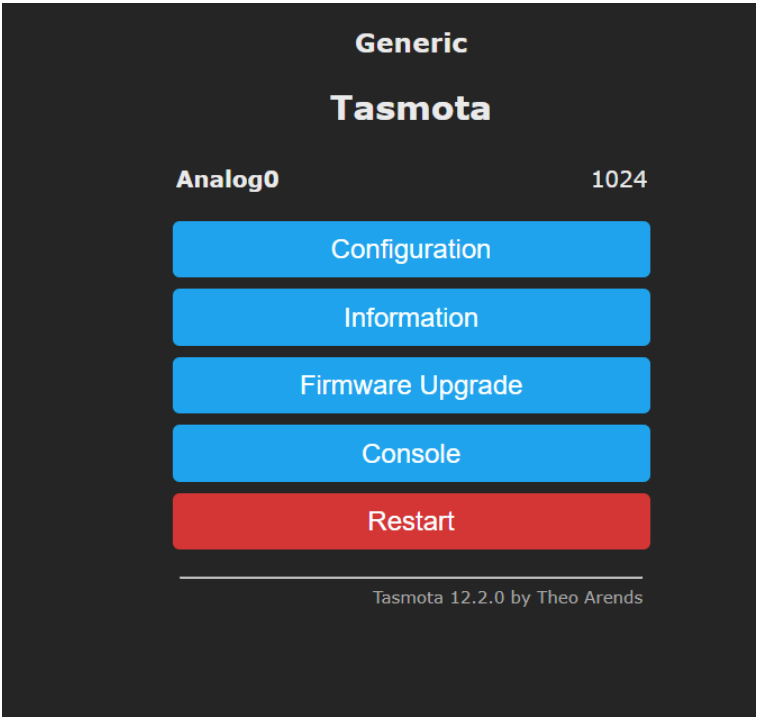


圖 8 電線連接之結果

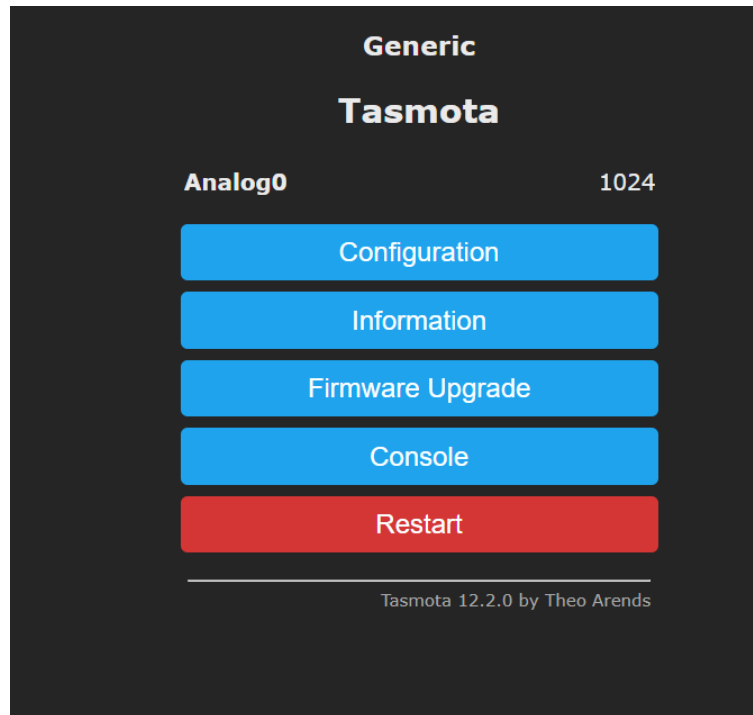


圖 9 水中測試結果

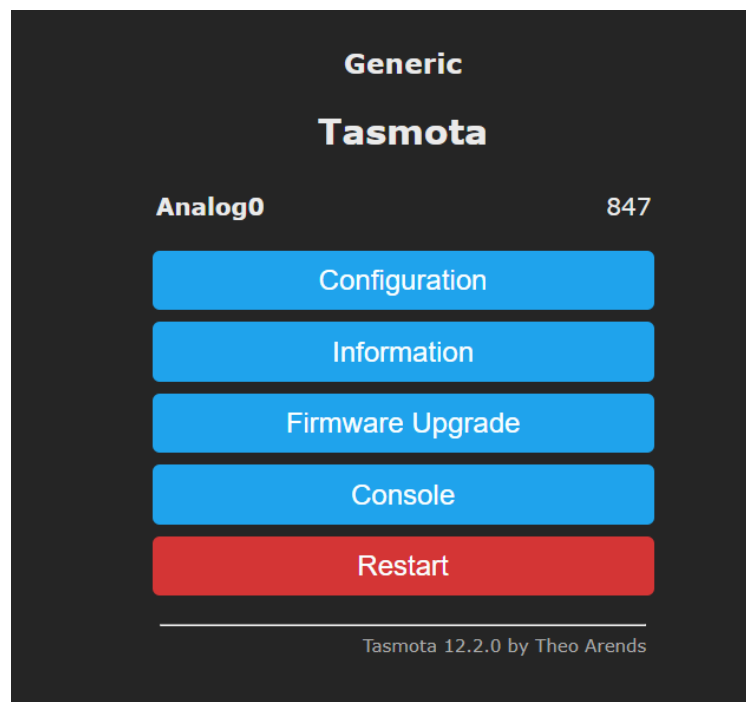


圖 10 飽和水分之泥炭土濕度

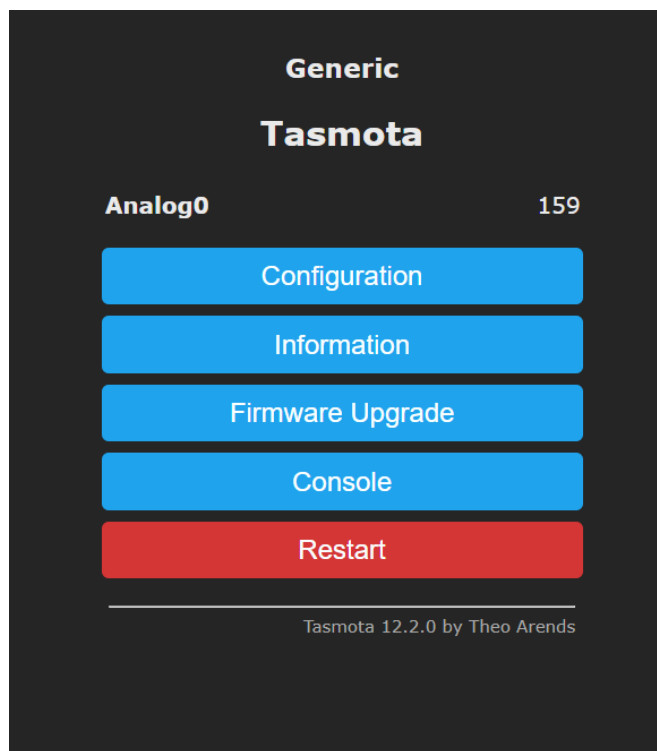


圖 11 乾燥顆粒土溼度(電阻率)測試結果

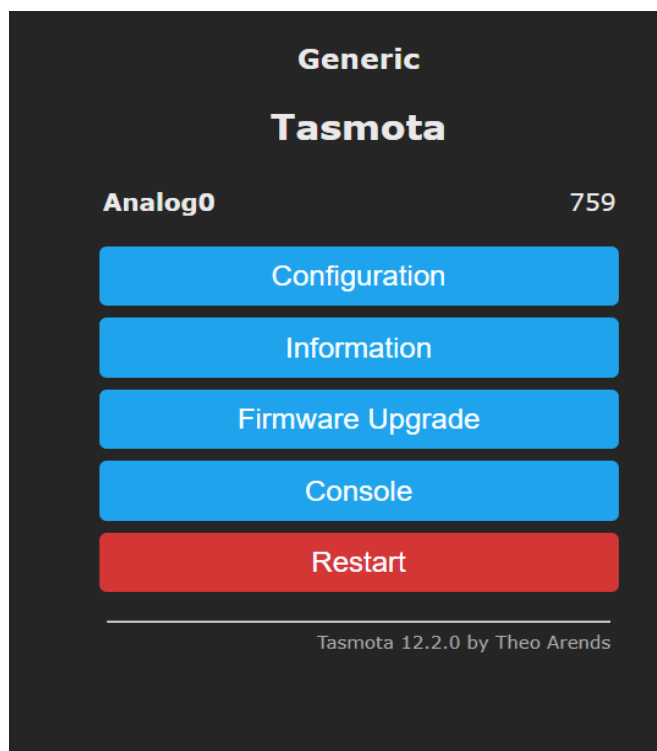


圖 12 水分飽和之顆粒土溼度測試結果

透過採集溼度之上下限，可以簡單擬定啟動沉水馬達(澆水)或風扇(通風乾燥)之閾值。

各類植物對溼度有不同之喜好，以耐旱怕濕的多肉植物來說，將飽和數值之 70% 至 15% 定義為合適濕度範圍較妥當。

2-4 BH1750

GY-302 BH1750，為光強度與照度感測模組，工作電壓為 3.3V-5V，感測範圍從 1 至 65535lux，內置 16bit A/D 轉換器，可配合 I2C(SDA、SCL)輸出數位訊號。將 VCC 接上 5V 或是 3.3V，GND 接 GND，SDA 接 SDA，SCL 接 SCL。

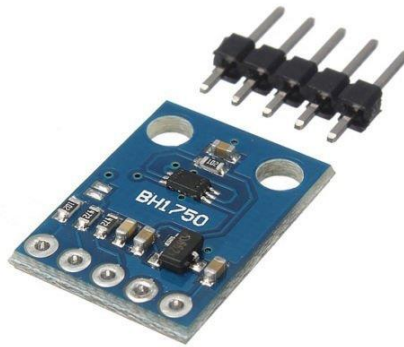


圖 13 BH1750

2-4-1 採集數據

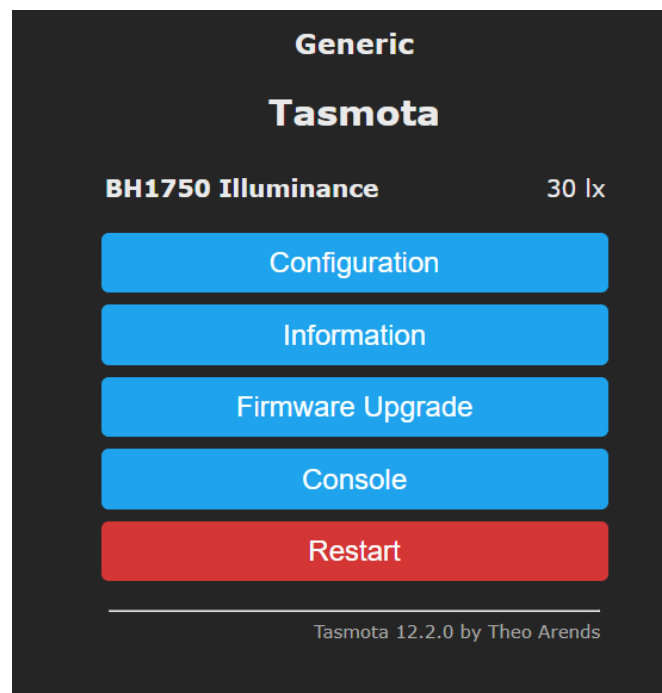


圖 14 室內散光

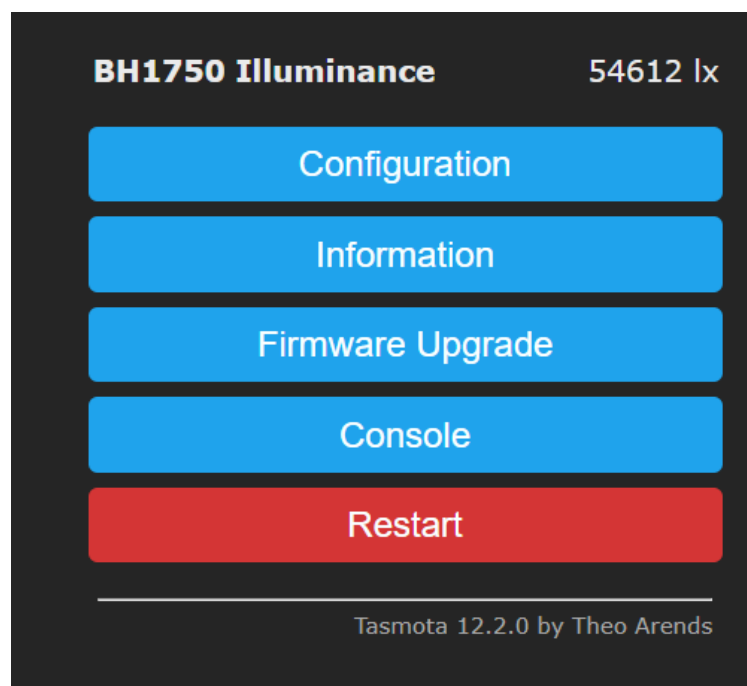


圖 15 上午 9 點太陽直射光

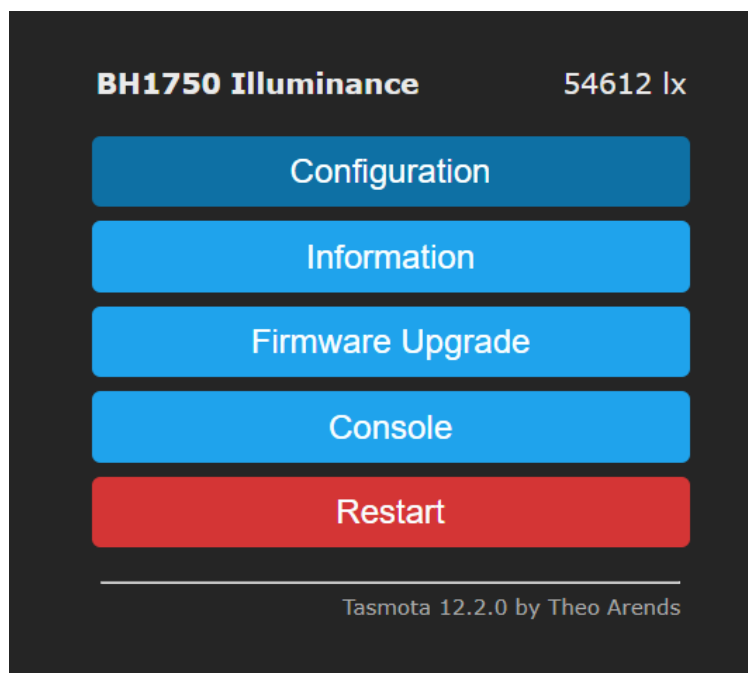


圖 16 正午太陽直射光

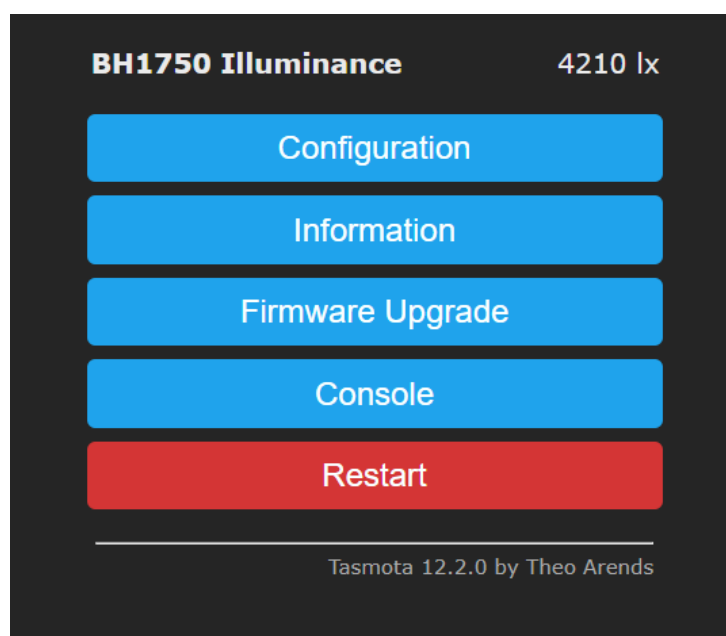


圖 17 一般手機手電筒

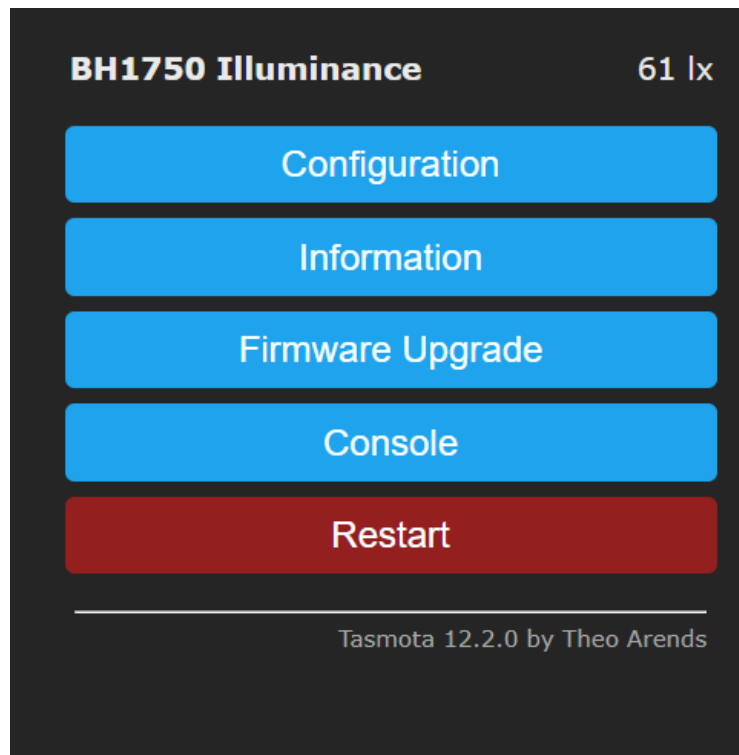


圖 18 一般室內日光燈(距離約 3 公尺)

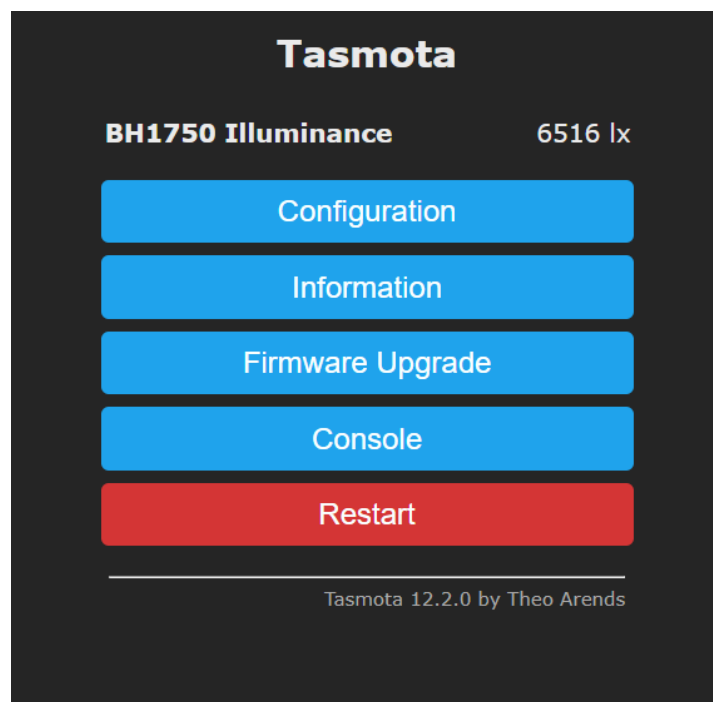


圖 19 室外樹蔭

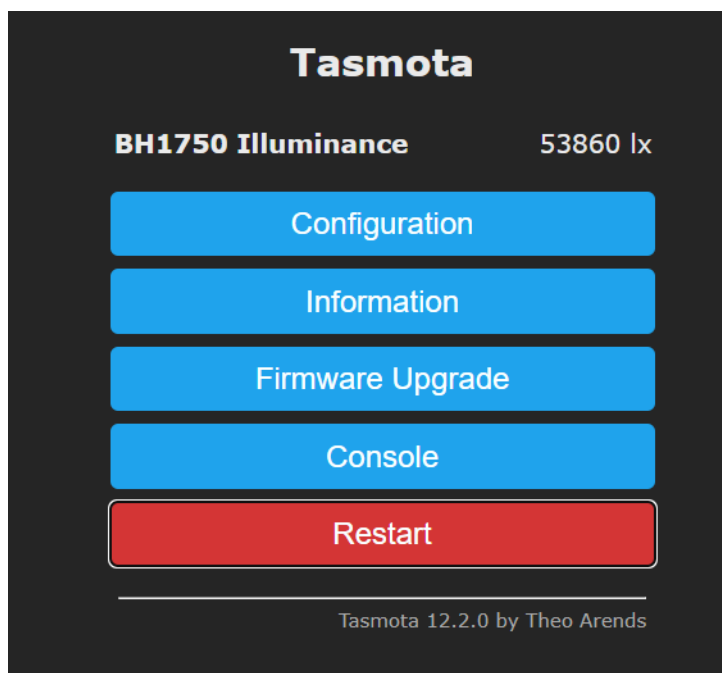


圖 20 室內隔著玻璃之日照

透過數據我們了解，太陽光之直接照射強度(無論時間)可以輕易突破 50000 流明，非直射之光線則遠遠不及此；以及人造光源之距離對光強度影響甚大，採集 LED 燈強度後將可擬定開啟補光之照度閾值。

2-5 Relay

2-5-1 Relay

在本專題中繼電器負責控制 LED、Pump 以及風扇的開關，將 IN 的接角接在 wemos 的 GPIO2 跟 GPIO14(兩者皆為 wemos 板子上的 LED)，後方 NO 接上電壓，在 COM 的地方接上要控制的裝置。



圖 21 繼電器

2-5-2 LED

要使用 LED 需要藉由 LED 驅動器將輸入的電壓和電流調整到適合 LED 燈的工作電壓和電流，LED 需要特定電壓和電流範圍才能正常運作，而 LED 驅動器可以確保 LED 燈得到適當的電壓和電流。

將驅動器的 LED+和 LED-分別接上 LED 的正負極，VIN+接在繼電器的 COM，VIN-則是接地。



圖 22 LED 驅動器

2-5-3 Pump

抽水馬達主要用於自動控制土壤濕度。當土壤濕度低於一定值時，程序就會改變繼電器的開關，打開抽水馬達，使土壤得到澆水。抽水馬達通常需要 5V 電壓供電，非常適合用於智能澆水系統或智能花園系統中。

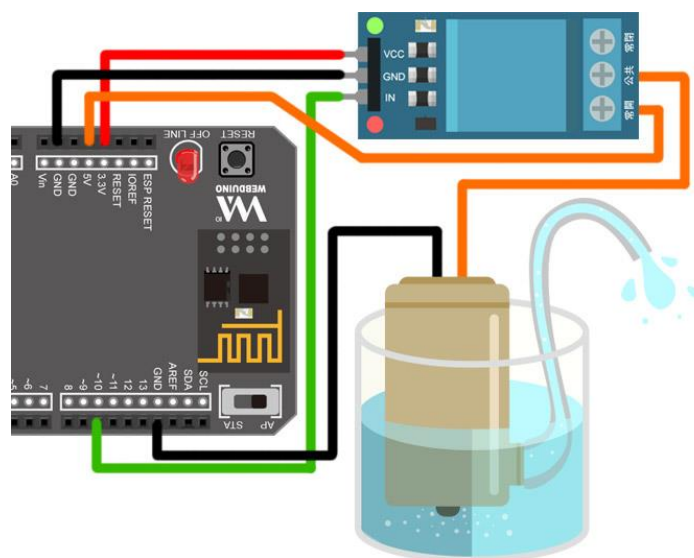


圖 23 接線圖

2-5-4 fan

準備中

第 3 章 Tasmota

3-1 Tasmota 介紹

Tasmota 是一個開源的智能設備韌體，用意是簡化使用物聯網(IoT)設備的過程。它可以在各種智能設備上安裝，包括智能插座、智能開關、智能燈泡等。

Tasmota 的優點在於，它可以輕鬆地與各種物聯網協議集成，如 MQTT、HTTP、KNX 等。這使得用戶可以輕鬆地將智能設備連接到任何支持這些協議的系統中，例如智能家居平台、物聯網網關等。此外，Tasmota 還提供了豐富的功能，如定時器、觸發器、腳本等，使用戶能夠創建複雜的智能場景。

總的來說，Tasmota 可以幫助用戶輕鬆地使用物聯網設備。

3-2 如何連上開發版

我們所採用的開發版為 ESP8266 D1，在安裝 Tasmota 之前，需要先在電腦上安裝 USB 驅動程式 CH340G(檔名為 CH341SER)。

安裝好驅動程式後，就可以使用 Micro USB 傳輸線將 ESP8266 D1 mini 連接到電腦。接下來，在 <https://tasmota.github.io/install/> (圖 24)網站上，您可以在兩個下拉欄中選擇適合的選項。左邊的欄位用於選擇 Tasmota 韌體的種類，例如要使用 Sensor 的可以選擇 Tasmota Sensors，如果您使用的是繼電器，則可以選擇 Tasmota(english)。右邊的欄位用於選擇您使用的開發版，例如本專題使用的 ESP8266。

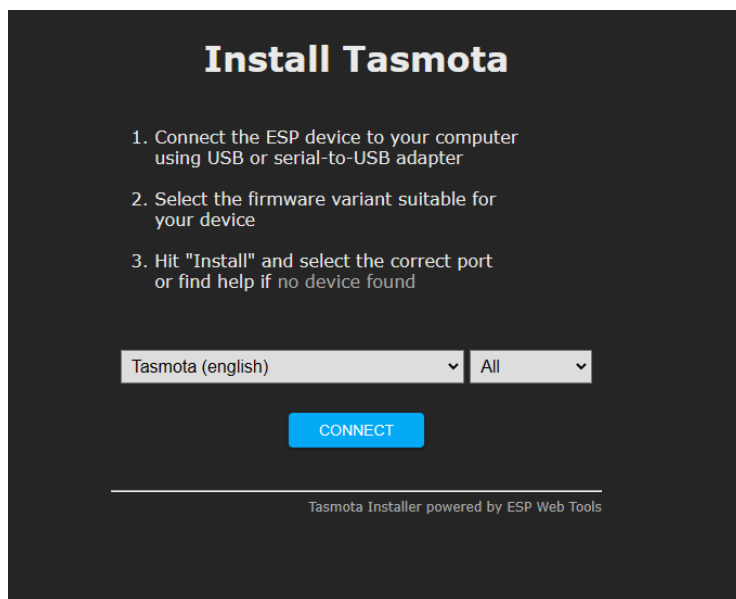


圖 24 Tasmota install 頁面

兩個欄位都選擇好後就可以點選 CONNECT，此時網頁會跳出一個視窗詢問你想要連線的序列埠是哪一個

(圖 25)，想要知道 ESP8266 的 COM 是哪一個的話，前往裝置管理員(對你的 windows icon 右鍵可以發現)，點開連接埠(COM 和 LPT)可以觀察到所有連上你電腦的 USB 接口，找到 CH340 的那一項就是 ESP8266，以圖 26 為例就是 COM4

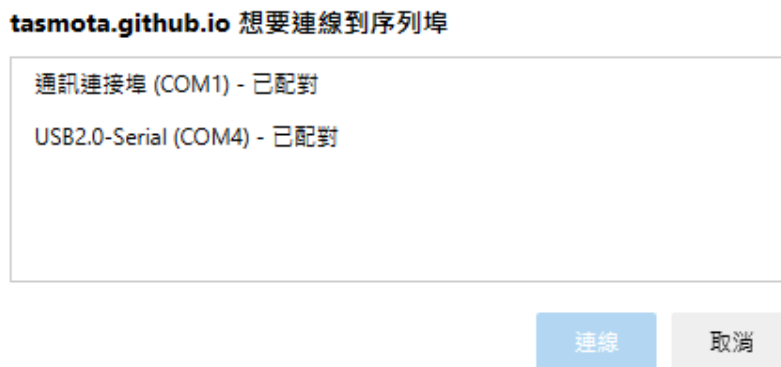


圖 25 點擊 CONNECT 後會跳出的視窗

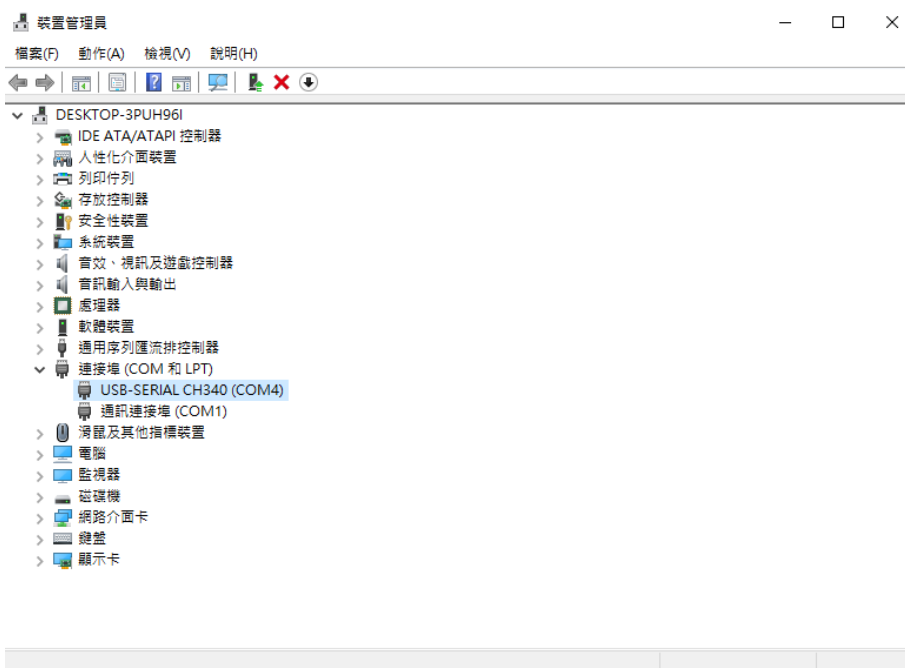


圖 26 如何找到裝置的序列埠

回到 Tasmota install 的頁面後，在圖 25 中選擇剛剛找到的 USB2.0-Serial(COM4)，之後會出現 Device Dashboard，選擇 INSTALL TASMOTA，安裝成功之後也要配置 Wi-Fi，配置 Wi-Fi 成功後就可以點選 visit device(圖 27)進入介面(圖 28)

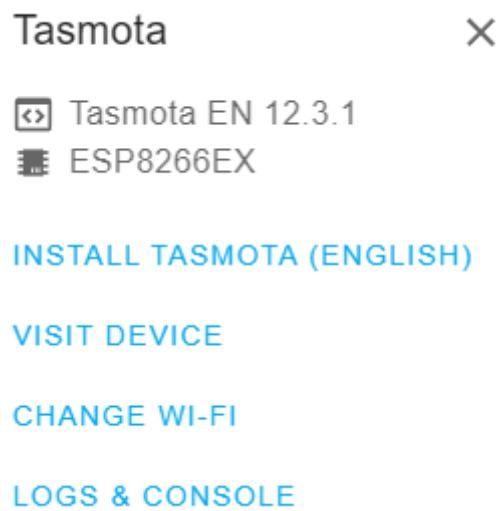


圖 27 成功配置 Wi-Fi 過後

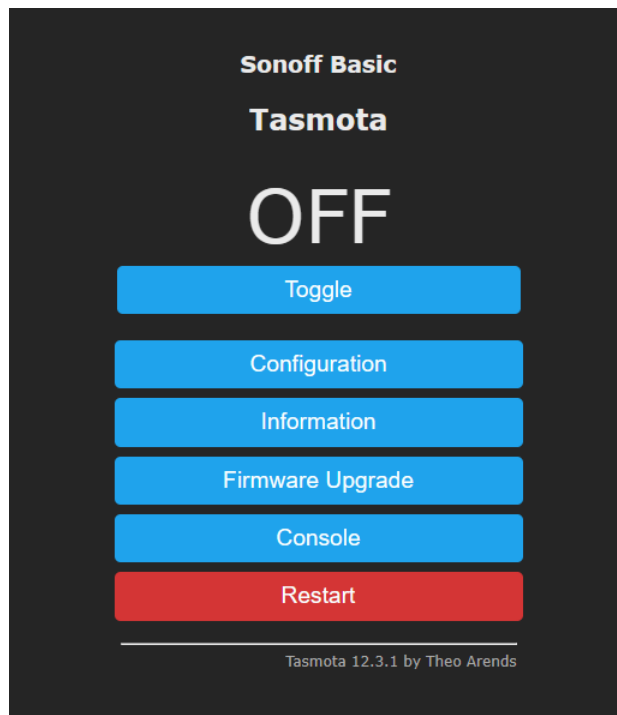


圖 28 VISIT DEVICE

3-3 如何連上 modules

進入到 UI 介面後點選 Configuration，接著點選 Configure Module，可以看到有很多 Module Type 可以做選擇，這邊以繼電器為例子選擇 Generic(18)，我們將繼電器的 VCC、GND 接在 ESP8266 相對應的位置再將 IN 接在 GPIO14(LED)上面，回到 Tasmota 的介面將 GPIO14 的地方選擇 Relay，點擊 Save 回到主介面後就可以透過 Toggle 去控制繼電器的開關。

如果是要使用 Sensor 的話 Module Type 一樣選擇 Generic(18)，看你的 Sensor 接角是 I2C 還是類比接角，在你接的 GPIO 上做選擇，I2C 應該要選擇 I2C SCL 跟 I2C SDA，類比接角就要選擇 ADC Input。(注意 sensor

在前面安裝 tasmota 的時候要選擇 tasmota sensor)

3-4 如何連上 MQTT

進入到 UI 介面後點選 Configuration，接著點選 Configure MQTT，並修改格式符合系統的需求(圖 29)。

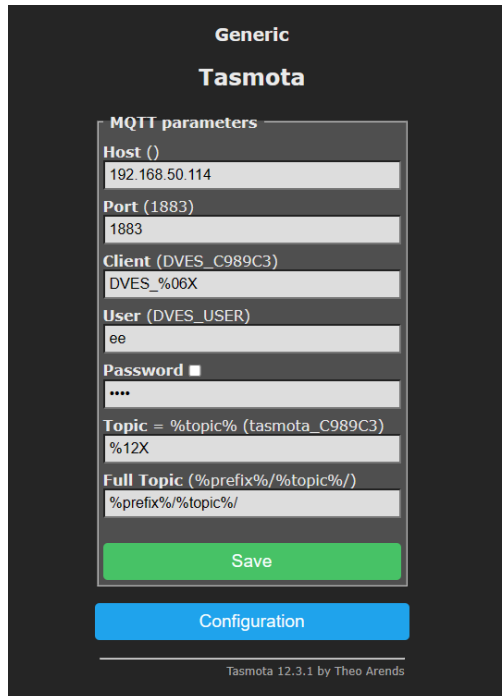


圖 29 範例

第 4 章 網路協議(MQTT)

4-1 MQTT 介紹

MQTT (Message Queuing Telemetry Transport) 是一種網路協議，專為資源受限的設備和低帶寬、高延遲的網路連接而設計。MQTT 使用發布/訂閱模型，允許設備和應用程序異步地發送和接收消息。這使得它非常適合於物聯網 (IoT) 應用程序。

MQTT 的特性包括：

1. 輕量級：消息格式簡單，協議棧小，可以在資源受限的設備上運行。
2. 發布/訂閱模型：讓設備和應用程序可以異步地發送和接收消息。
3. QoS (Quality of Service) 等級：支持三種不同的消息發送質量，以保證消息能夠及時和正確地到達目標。

4-2 Public MQTT

我們可以使用 free MQTT broker，free server: public.mqttq.com，下載 mqttbox:

<https://chrome.google.com/webstore/detail/mqttbox/kaajoficamnjjhkeomgfljpicifbkaf?hl=zh-TW>

需要使用 chrome 才可以使用。

打開 mqtt box 後點擊 Create MQTT Client，將 Host 的地方改成 public.mqttq.com，Protocol 更改為 mqtt/tcp(圖 30)。

MQTT Client Name <input type="text" value="test"/>	MQTT Client Id <input type="text" value="33f091e0-c6d4-4550-816b-6eb1f8"/>
Protocol <input type="text" value="mqtt / tcp"/>	Host <input type="text" value="public.mqttthq.com:1883"/>
Username <input type="text" value="Username"/>	Password <input type="password" value="Password"/>
Reconnect Period (milliseconds) <input type="text" value="1000"/>	Connect Timeout (milliseconds) <input type="text" value="30000"/>
Will - Topic <input type="text" value="Will - Topic"/>	Will - QoS <input type="text" value="0 - Almost Once"/>
<input type="button" value="Save"/>	

圖 30 Create MQTT Client

建立好用戶之後就可以 subscribe 你想要的內容，舉例來說我想要獲得 sensor 的資料，要先將板子連上 tasmota，接下來將 tasmota 連上 MQTT(圖 31)，回到 tasmota 的主頁面點擊 console，可以觀察到板子每隔 5 分鐘(預設)會發送一次消息給 MQTT，我們要訂閱的就是這台的 MAC 地址，圖 32 為例，我們訂閱的是 #tele/A4CF12C989C3，就可以在 MQTT box 上訂閱，就可以在上面看到發送過來的訊息(圖 33)。

Generic

Tasmota

MQTT parameters

Host ()

Port (1883)

Client (DVES_C989C3)

User (DVES_USER)

Password ■

Topic = %topic% (tasmota_C989C3)

Full Topic (%prefix%/%topic%/)

Tasmota 12.1.1 by Theo Arends

圖 31 free mqtt broker

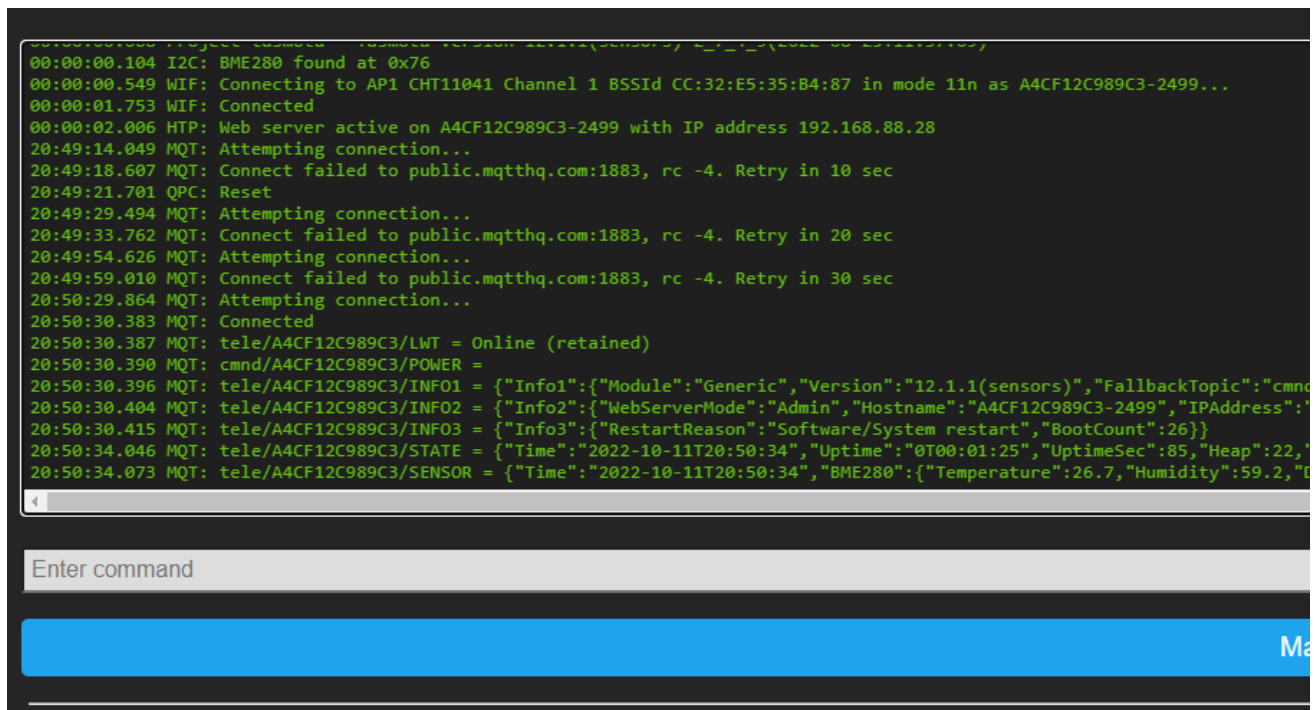


圖 32 MAC 地址

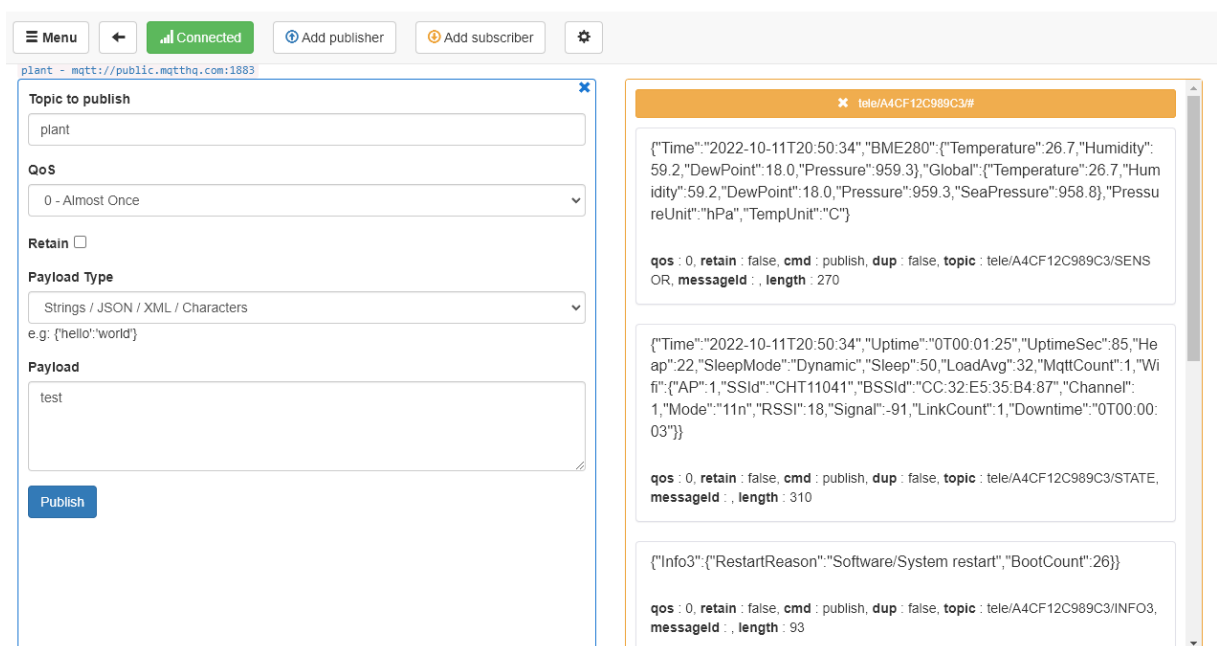


圖 33 訂閱#tele/A4CF12C989C3

4-3 Home Assistant MQTT

本專題使用的 MQTT broker 是在 Home Assistant 上建立的，在 Home Assistant 上點擊設定，附加元件，右下角附加元件商店中搜尋 Mosquitto broker 並且安裝，安裝成功過後點開 Mosquitto broker，設定，可以設定你的用戶名稱跟密碼(圖 34)，設定好過後重啟 Mosquitto broker，就可以接收其他用戶的訊息(圖 35)。

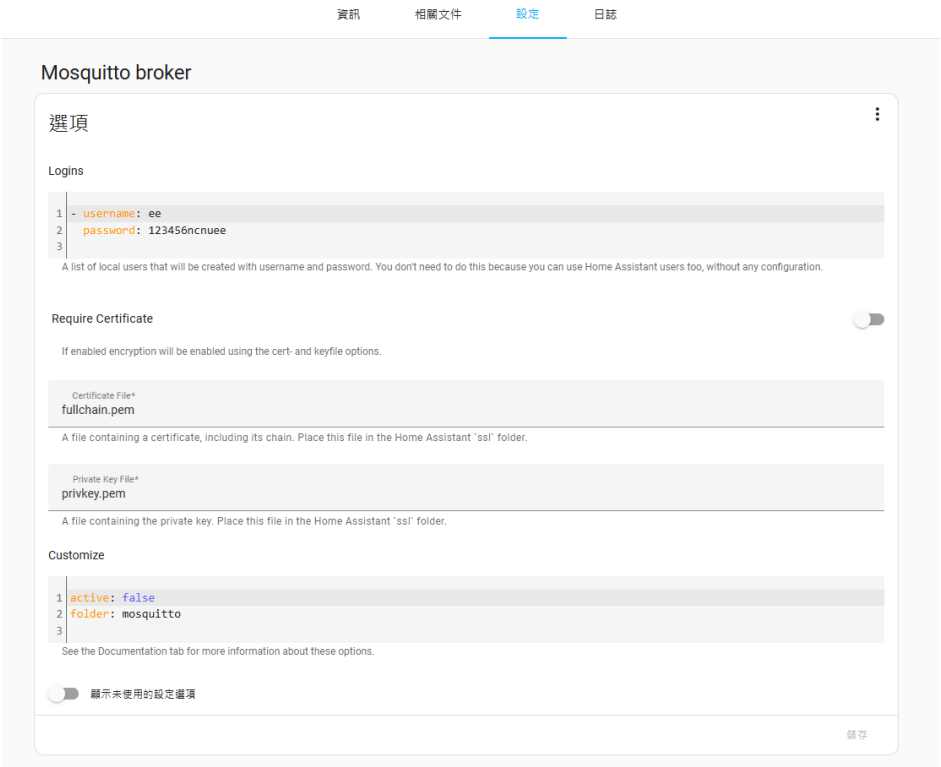


圖 34 設定用戶名稱跟密碼

圖 35 連接 homeassistant mqtt 範例

Host 的地方需要填入你 homeassistant 主機的 IP 位置，有兩種方式可以查詢，第一種如果你是使用路由器的話可以到路由器的管理介面，找到用戶的閱覽名單(圖 36)，就可以找到客戶端的 IP 位置。

圖示	客戶端名稱	客戶端 IP 位址	客戶端 MAC 位址	介面
	A4CF12C989C3-2499	192.168.50.37	A4:CF:12:C9:89:C3	
	46:6E:75:EE:39:37	192.168.50.38	46:6E:75:EE:39:37	
	homeassistant	192.168.50.114	B8:27:E8:83:0D:E5	
	EA:EE:8E:18:00:83	192.168.50.139	EA:EE:8E:18:00:83	
	DESKTOP-3PUH96I	192.168.50.215	18:31:BF:51:E9:77	

圖 36 管理介面

第二種方式，到商店下載 Net Analyzer，點選 LAN，左上角 Scan 也可以查看你的 IP 位置(圖 37)。

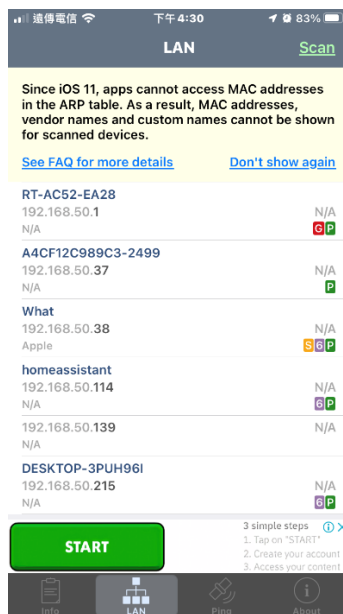


圖 37 Net Analyzer

第 5 章 使用者介面(Home Assistant)

5-1 Raspberry Pi

我們專題所採用的使用者介面是 Home Assistant 並且運行在樹莓派上，因為 Home Assistant 提供了一個可視化的用戶界面，也可以使用它來查看和控制各種設備，最重要的是可以在設備之間創建自動化工作流程，以完成我們的智慧溫室。下面提供了安裝 Home Assistant 在 Pi 上的方式。

[學習日誌十六：Raspberry Pi 4 安裝 Home Assistant | by Chun-Li 春麗 | 彼得潘的 Swift iOS App 開發教室 | Medium](#)

[樹莓派 - 家庭助理 \(home-assistant.io\)](https://home-assistant.io/)

5-2 如何蒐集觀測數據(Tasmota)

在確定 MQTT 已經建立好的情況下(Tasmota 跟 HA)，透過網址(<http://homeassistant.local:8123/>)進入 Home Assistant 頁面，如果無法進入的話先確認主機的網路跟 Pi 的網路是否是在同一個區域網下，在 Home Assistant 的頁面中點選設定，裝置與服務，右下角新增整合搜尋 Tasmota，屆時你可以看到他會顯示實體以及裝置(圖 38)，裝置是你的開發版(ESP8266)，實體則是像是繼電器或是感測器等等，點開實體可以去更改繼電器跟感測器的名稱以便更好識別(圖 39)。改的是名稱不是實體 ID!



圖 38 HA 整合 Tasmota

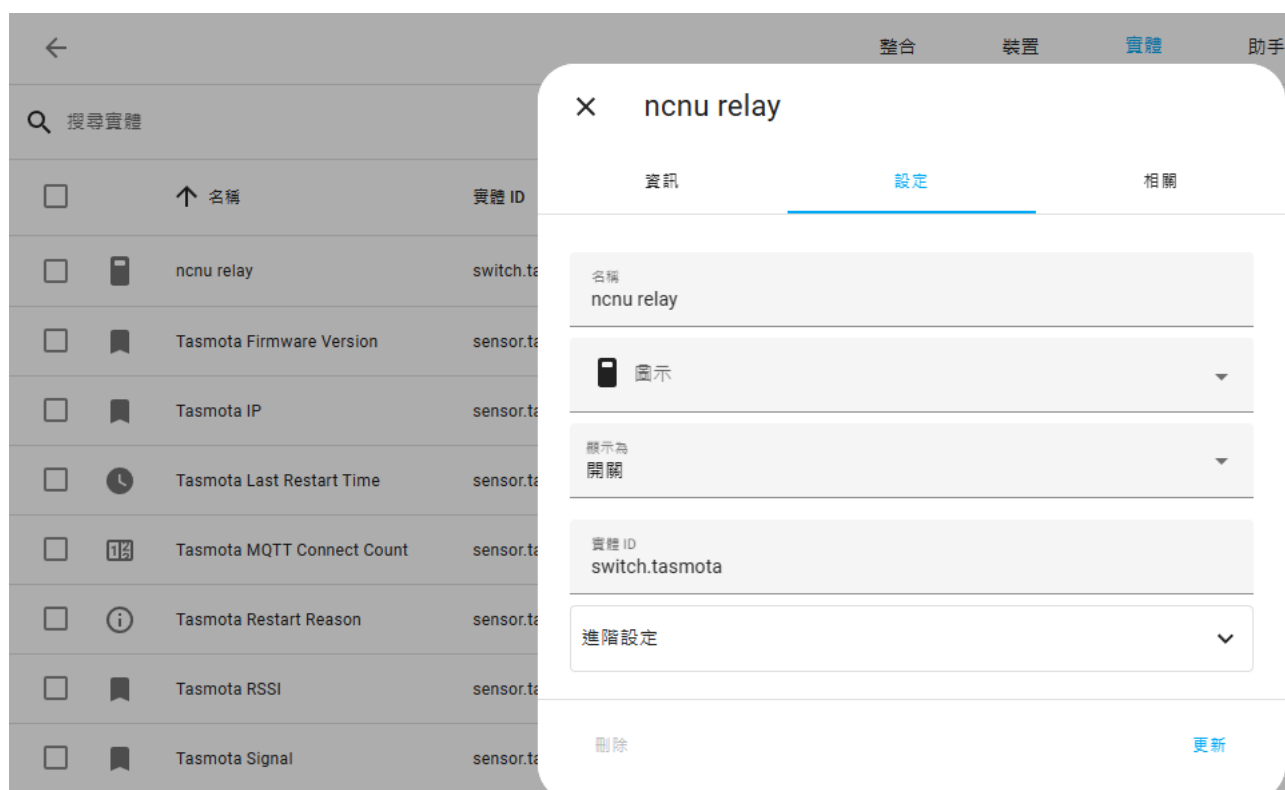


圖 39 更改名稱

接下來可以到 Home Assistant 總覽的頁面，右上角可以編輯主面板，新增面板，再選擇你需要的面板，例如繼電器就會選擇按鈕的開關控制面板，感測器可以選擇感測器面板，示你的要求而定(圖 40)。如此以來就可以在 Home Assistant 的首頁上觀測你的 sensor 數據以及控制 relay。

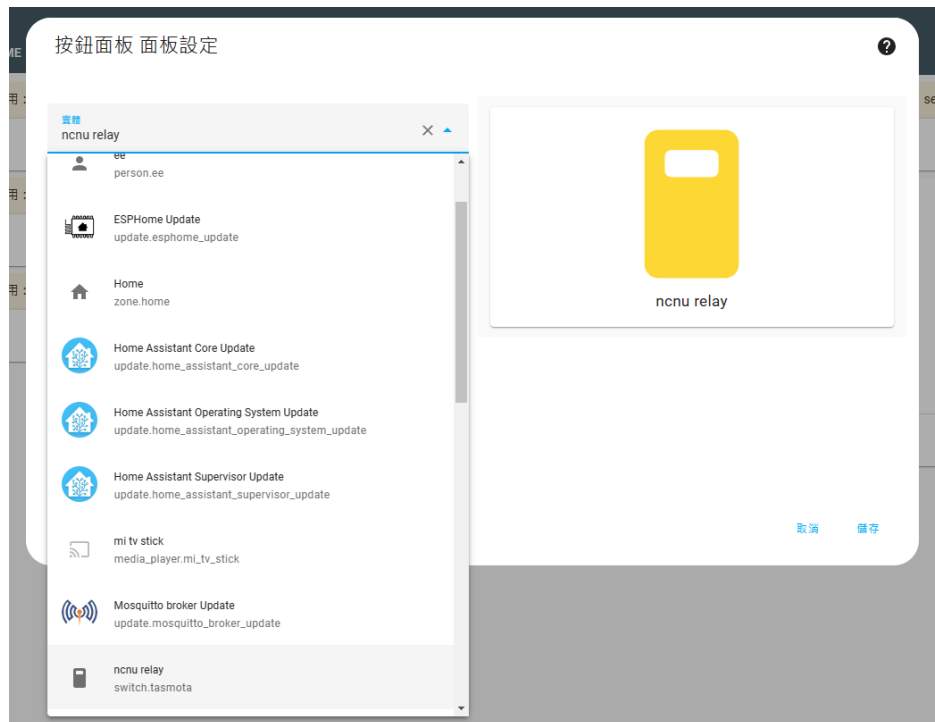


圖 40 主控版範例

5-3 如何自動化

我們可以使用 Home Assistant 的自動化功能來創建工作流程，以達到自動化家居的目的。

要創建一個自動化，您需要定義觸發器和動作。觸發器是指當什麼發生時會觸發自動化。動作是指當觸發器觸發時要執行的操作。

Home Assistant 支持多種不同類型的觸發器，包括時間、地理位置、傳感器讀數等等。您可以使用這些觸發器來定義您的自動化。例如，您可以定義一個觸發器，當您回家時觸發，然後設置一個動作來打開燈。

進到 Home Assistant 的介面點選設定，自動化與場景，右下角新增自動化，選擇以空白自動化開始，就可以建立一個新的自動化程序(圖 41)，根據你的需求去新增觸發條件跟觸發判斷還有觸發動作，舉例來說，當我的土壤濕度檢測低於 8 的時候，我會切換 switch 讓抽水馬達澆水，Delay 一分鐘過後再將他切換關閉。

建立新的自動化

觸發

+ 新增一個觸發

觸發判斷

+ 新增一個判斷式

觸發後動作

+ 新增一個動作

圖 41 自動化

觸發

123 When sensor.tasmota_analog_a0_2 is below 8

+ 新增一個觸發

觸發判斷

+ 新增一個判斷式

觸發後動作

開關: Turn off Tasmota

記錄
開關: Turn off

Turn a switch off

目標
此服務使用之目標分區、裝置或實體。

Tasmota

+ 選擇分區

+ 選擇裝置

+ 選擇實體

Delay for 1:00

持續時間

hh:mm:ss

0:01:00

開關: Turn on Tasmota

記錄
開關: Turn on

Turn a switch on

目標
此服務使用之目標分區、裝置或實體。

Tasmota

+ 選擇分區

+ 選擇裝置

+ 選擇實體

圖 42 一個簡單的自動化澆水範例

5-4 SSH

SSH 是一種傳輸的協議，常用於電腦之間的加密登入。SSH 使用的是一對密碼，一個公鑰一個私鑰，訊息透過公開的公鑰加密後傳遞，中途即使其他人可以看到加密後的訊息，但是因為沒有私鑰解密，仍無法知道訊息的內容，最後收到的人再拿私鑰解密，完成一次單方向的傳遞訊息。

若要雙方溝通，則需要兩對密碼，發信的人除了擁有對方的公鑰之外，也要產生另一份公鑰，發送給對方。

在 Home Assistant 的設定，附加元件，右下角附加元件商店搜尋 SSH & Web Terminal，安裝成功後在點腦上打開 power shell 輸入產生金鑰的指令：

```
ssh-keygen -t rsa -b 4096 -C "Marcus"
```

// -t: SSH 金鑰的類型

// -b: SSH 金鑰的長度

// -C: 金鑰的備註，後面的" " 裡的内容

執行成功後到 C:\Users\你的電腦名稱.ssh 下找到公鑰: id_rsa.pub，將其打開並將裡面的內容複製下來。

回到 SSH & Web Terminal 的設定頁面，將公鑰複製到 authorized_keys，並將下面的網路通訊埠更改為 22。
儲存過後重啟 Home Assistant，回到終端視窗(power shell)輸入 "ssh root@homeassistant.local" 就可以直接進入 Home Assistant。

有些時候當你進入不了 Home Assistant 的頁面時可以透過以上方法查看 Pi 的螢幕畫面(如果有接螢幕就不必使用此方法)確認一下出了什麼問題，我在實作的過程中偶爾會發生進入不了頁面的問題，需要輸入 banner 的指令才有能成功進入。

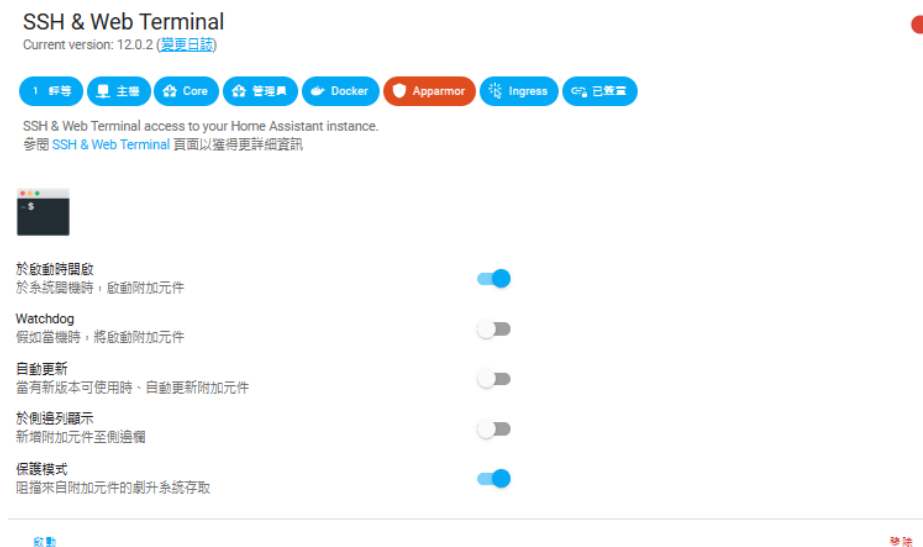


圖 43 SSH & Web Terminal

```
OpenSSH SSH client
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\gaunwei> ssh root@homeassistant.local

Home Assistant

Welcome to the Home Assistant command line.

System information
IPv4 addresses for eth0: 192.168.50.114/24
IPv6 addresses for eth0: fe80::9af4:2875:7269:369c/64
IPv4 addresses for wlan0:

OS Version:           Home Assistant OS 9.0
Home Assistant Core:   2022.9.5

Home Assistant URL:    http://homeassistant.local:8123
Observer URL:          http://homeassistant.local:4357
```

圖 44 終端視窗

第 6 章 下學期進度

這學期對 IoT unit 和相關元件有了進一步的了解。在下一學期，我們可以將這些模組整合起來並實現自動化，然後進行實驗，看看我們的裝置對多肉植物是否有實質上的用處。