

自己的空氣品質自己救 – 實作 PM2.5 空氣盒子

(使用 AMEBA RTL8195AM 開發板)

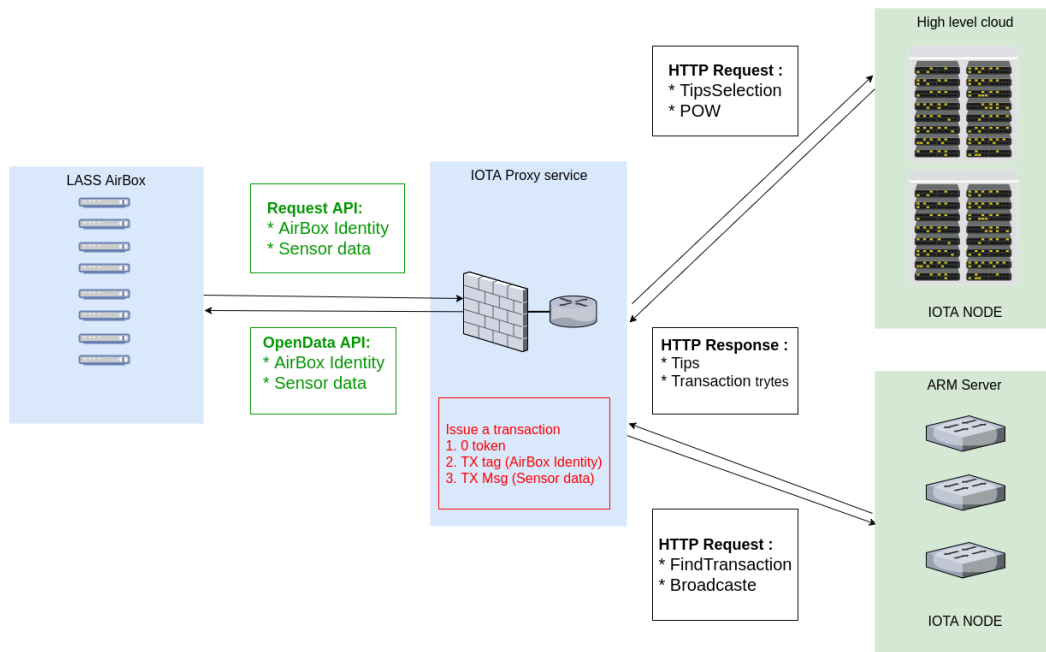
為什麼我們的 IoT 需要 IOTA ?

- 減少延遲和營運管理成本
- 降低攻擊和破壞的風險
- 確保資料完整性和存在證明 (PoE)



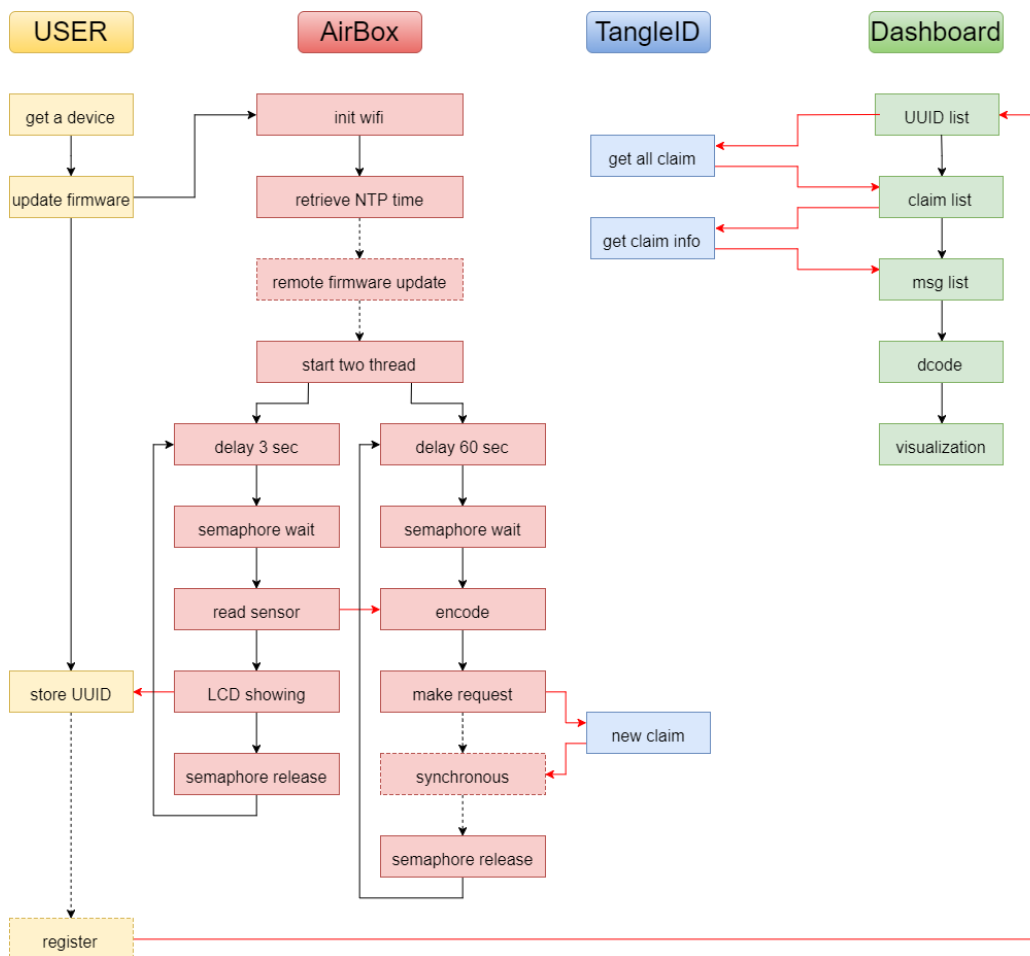
實際 PM2.5 空氣盒子

架構圖:



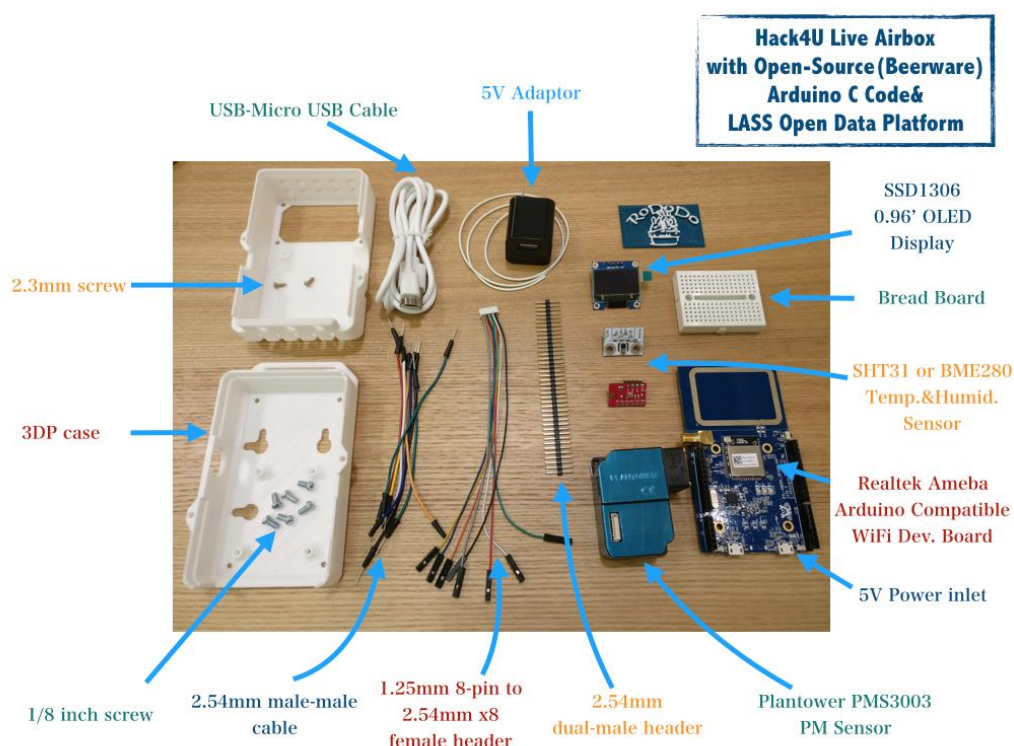
資料來源: https://hackmd.io/@twzjwang/SJnIK_8yM?type=view

技術細節:



資料來源: https://hackmd.io/@twzjwang/SJnIK_8yM?type=view

準備材料:



資料來源: <https://paper.dropbox.com/doc/HACK4U4-Hack4u-87Live--Ar8JnN4znqI7t76PL74cK8PnAg-TRmqFxxLJHXrKBo4kiR6O>

■ Ameba RTL8195AM

Ameba 是一張由 Realtek 所推出的開發板,很適合開發各種的感測器或與物聯網應用。它上面的介面有 Wifi, GPIO, NFC, I2C, UART, SPI, PWM, ADC, 這些介面可以接一些電子元件像是 LED 燈、開關、壓力計、溫濕度感測器、PM2.5 粉塵感測器等等。這些資料可以經由內建的 Wifi 上傳到 IOTA Tangle 網路或搭配手機的 App 實現物聯網的實作。

■ PMS3003 (G3) 粉塵感測器

PMS3003(G3)是一款數位式通用顆粒物濃度傳感器,可以用於獲得單位體積內空氣中懸浮顆粒物的質量,即顆粒物濃度,並以數位介面形式輸出。本傳感器可嵌入各種與空氣中懸浮顆粒物濃度相關的儀器儀表或環境改善設備,為其提供及時準確的濃度數據。傳感器採用鐳射散射原理。能夠得到空氣中 0.3 ~ 10 微米懸浮顆粒物濃度,數據穩定可靠,內置風扇,整合度高。

■ OLED SSD1306 顯示器

我們使用的是 OLED SSD1306 提供 I2C 協定介面，螢幕尺寸為 0.96 吋,128*64 解析度 (I2C 介面),顯示顏色(黃藍雙色)。因為使用 I2C 介面，所以總共有四隻接腳，由左到右依序為 VCC、GND、SCL、SDA。

OLED 規格整理如下：

- 解析度: 12864 表示解析度為 128×64。除了 128×64 之外，還有 128×32 等各種不同的解析度。
- 顏色: OLED 12864 為單色螢幕，此為螢幕顯示內容的顏色。除此之外，也有彩色的 OLED 顯示器，不過價格比單色螢幕高出許多。
- 尺寸: 螢幕的尺寸，通常螢幕尺寸越大價格也越高。不過真正能顯示的內容，仍以解析度為主。在解析度一樣的情況下，螢幕尺寸越大只是每一個”點”變大，可方便辨識但無法增加可顯示的內容。
- 驅動晶片: OLED 12864 多使用 SSD1306 當作驅動元件，不過其他解析度的 OLED 螢幕有可能使用不同的驅動元件。
- 介面: 是否提供 I2C 或 SPI 協定的介面，以便於程式的控制。
- 其他: 有不少的 OLED 12864 標榜內建字型檔以解決非英文字型顯示的問題。內建字型檔的 OLED 12864 除了費用較高外，也可能需要額外的步驟才能使用字型檔。其它像 Raspberry Pi 的系統資源較一般單晶片控制板來的充足，所以可以直接使用檔案系統中的字型檔顯示非英文字型，因此可以不需要內建字型檔的版本。

■ BME280 溫溼度感測器

BME280 是一款基於博世(BOSCH) APSM 工藝和創新的電阻式測量技術的小尺寸、超高精度、高性能壓力和溫濕度感測器模組，非常適合空間有限的移動設備，如智慧手機，平板電腦、智慧手錶和可穿戴設備，BME280 特別適合電池驅動的應用場合。

BME280 的技術特性：

相對濕度範圍：0 至 100%

溫度區間為：-40°C 至+85°C

濕度測量響應時間：1S

滯後：相對濕度 2%

壓力範圍：300 至 1100 hPa

絕對壓強精確度：絕對值±1hPa（焊接後）

絕對溫度精確度：±0.5°C（25°C 時）

通訊埠：I2C、SPI

接線腳位:

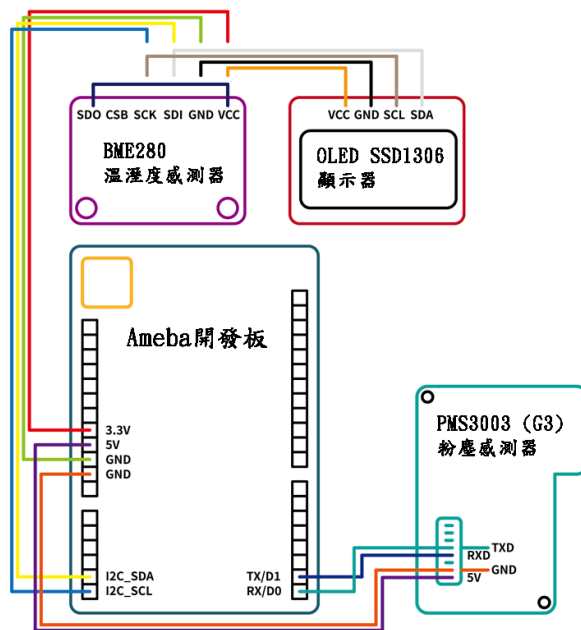
■ Ameba 開發板與 PMS3003 (G3) 粉塵感測器

Ameba 接 腳	PMS3003 (G3) 接腳
5V	5V
GND	GND
D1(TX)	RXD
D0(RX)	TXD

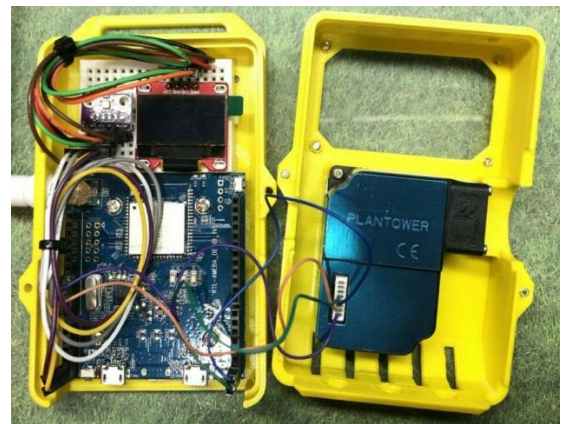
■ Ameba 開發板與 BME280 溫溼度感測器和 OLED 顯示模組

Ameba 接 腳	BME280 接 腳	OLED 接 腳
GND	GND	GND
3.3V	VCC	VCC
SDA	SDA	SDA
SCL	SCL	SCL
3.3V	SD0	

完成圖:



接線腳位圖



實際配置完成圖

組裝教學影片:

<https://www.youtube.com/watch?v=73QzM162Hm4&feature=youtu.be>

Realtek RTL8195AM:

<https://os.mbed.com/platforms/Realtek-RTL8195AM>

如果你感覺自己組裝很麻煩又不懂如何配線,其實可直接購買組裝好的空氣盒子。

https://www.icshop.com.tw/product_info.php/products_id/25456

更新 DAP 韌體(Firmware) -[win10 使用者注意]

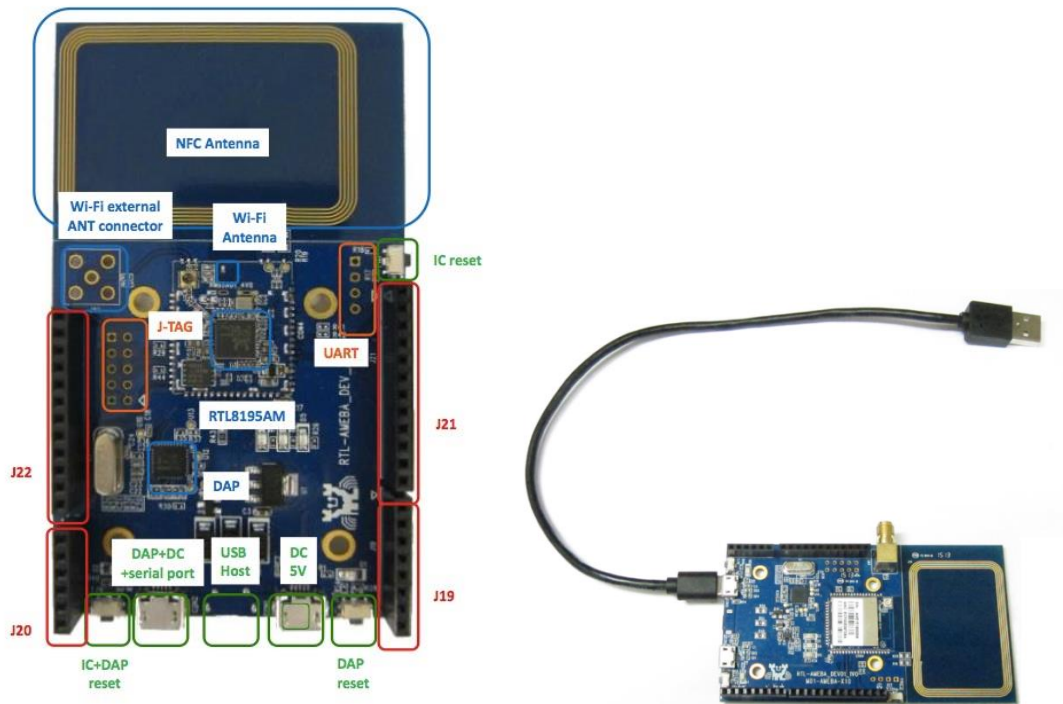
■ 步驟 1: 下載韌體

前往 Realtek 官方網站 <https://www.amebaiot.com/change-dap-firmware/> 最下方

載 DAP_FW_Ameba_V12_1_2-2M 或直接下載 DAP_FW_Ameba_V12_1_2-2M

https://github.com/Ameba8195/Arduino/raw/master/misc/dap_firmware/DAP_FW_Ameba_V12_1_2-2M.bin。

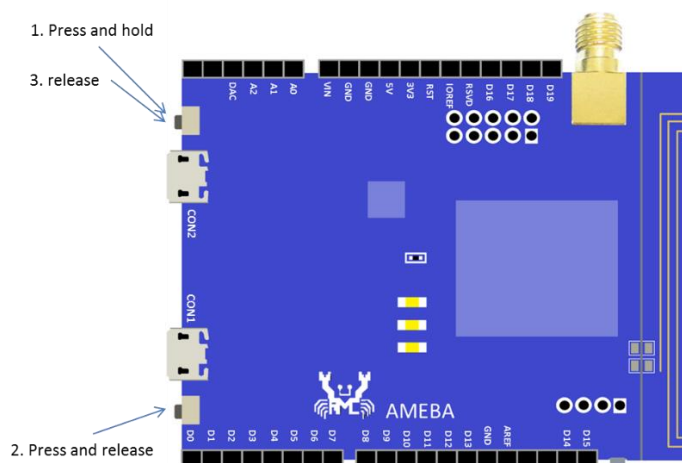
- 步驟 2: 將 Micro USB 接上 Ameba 開發板 **DAP+DC+serial port**,另一端接上電腦: 如下圖所示。



針腳設計圖(相容於 Arduino 針腳)

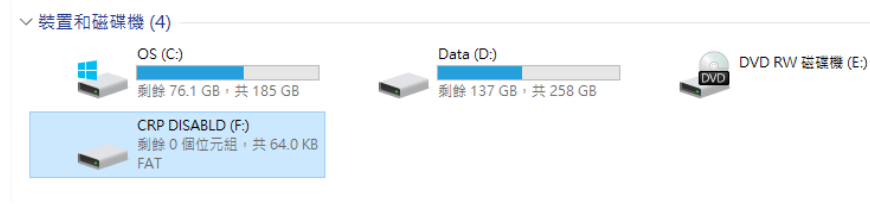
連接到 DAP+DC+serial port

- 步驟 3: 更換 DAP 韌體
 1. 先按住 CON2 旁邊的按鈕不放。
 2. 按一下 CON1 旁邊的按鈕。
 3. 最後放開在第一步按住的按鈕。

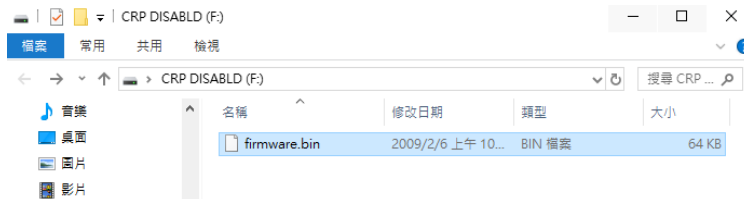


更換 DAP 韌體步驟

此時會出現一個磁碟槽，上面的標籤為 “CRP DISABLED”

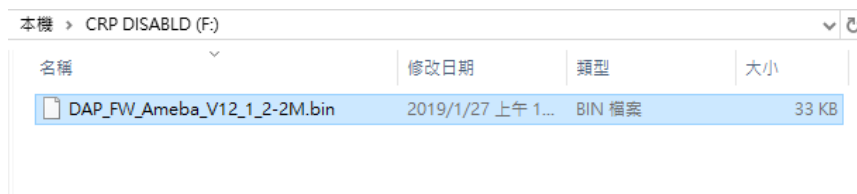


打開這個磁碟，裡面有個檔案“firmware.bin”，它是目前 Ameba 使用的 DAP firmware。



■ 步驟 4: 置換檔案

要更換韌體,可以先將這個 firmware.bin 備份起來,然後刪掉,再將新的 DAP firmware 用檔案複製的方式,以檔案拖拉方式將新韌體拉進“CRP DISABLD”磁碟槽。最後將 USB 重新插拔,新的韌體就生效了。



安裝驅動程式(Driver)

■ 步驟 1: 將 Micro USB 接上 Ameba 開發板的 DAP+DC+serial port 與另一端接上電腦

■ 步驟 2: 下載與安裝 ARM MBED CMSIS DAP 驅動程式

前往 <https://os.mbed.com/handbook/Windows-serial-configuration#1-download-the-mbed-windows-serial-port>

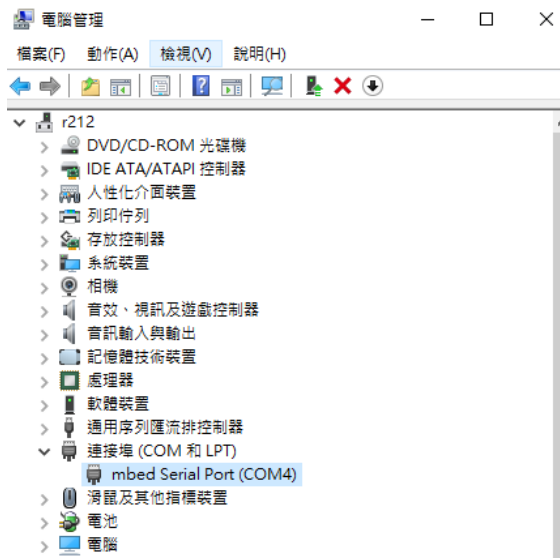
點選【Download latest driver】下載“mbedWinSerial_16466.exe”並安裝完成後,會在裝置管理員看到 mbed serial port。

您要安裝此裝置軟體嗎？

名稱: mbed 通用序列匯流排控制器
發行者: ARM Ltd

☒ 永遠信任來自 "ARM Ltd" 的軟體(A) **安裝(I)** 不要安裝(N)

! 您只應該安裝來自信任之發行者的驅動程式軟體。 [如何判定哪些裝置軟體可安全地進行安裝?](#)



顯示驅動程式

參考: Windows serial configuration

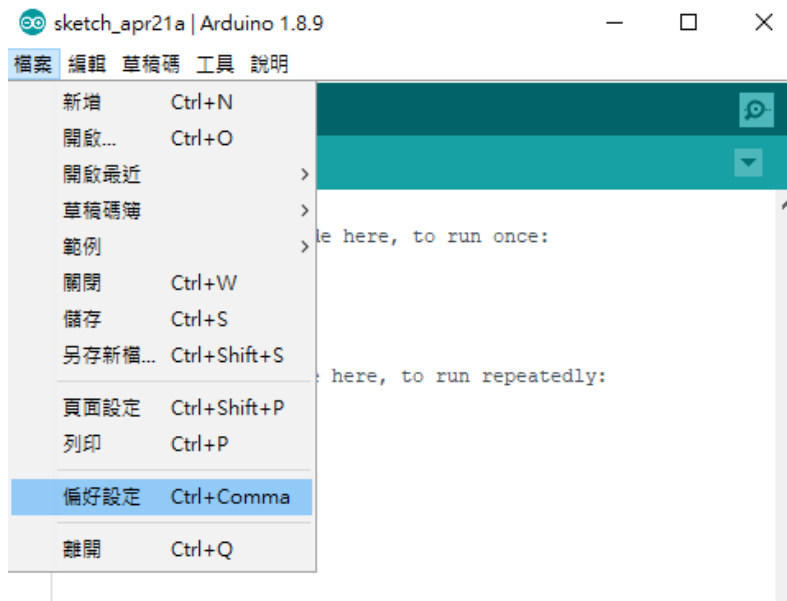
<https://os.mbed.com/handbook/Windows-serial-configuration>

安裝 Arduino IDE 開發環境

Arduino IDE 在 1.6.5 版之後，支援第三方的硬體，因此我們可以在 Arduino IDE 上開發 Ameba，並共享 Arduino 上面的範例程式。在 Arduino 官方網站上可以找到下載程式 <https://www.arduino.cc/en/Main/Software>。目前最新 Arduino IDE 版本為 1.8.9。

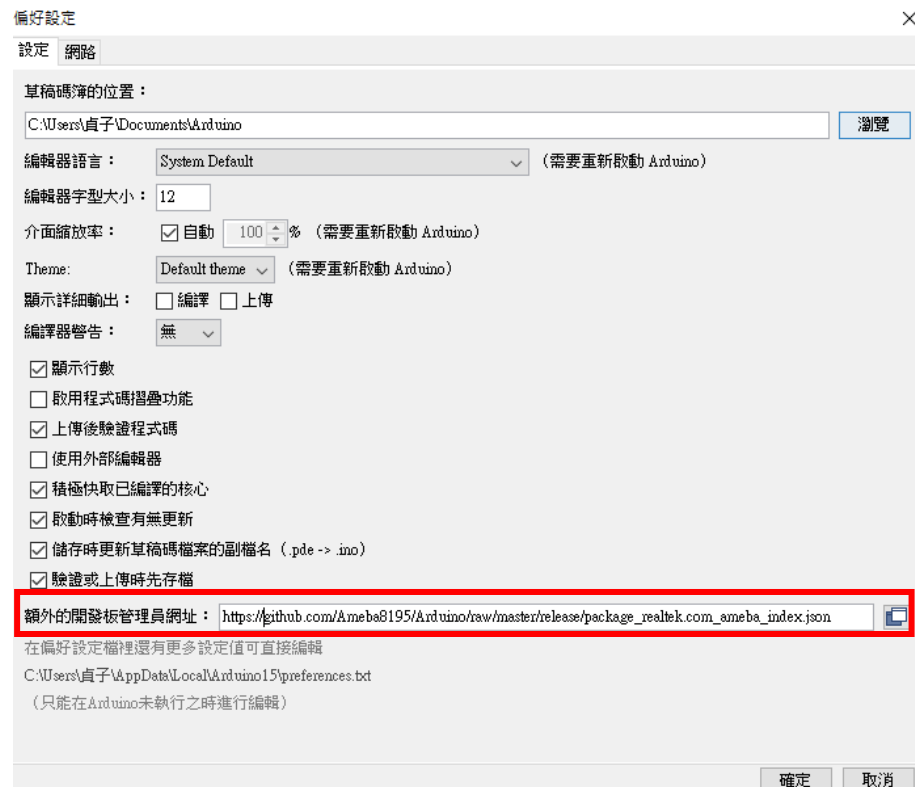


- 步驟 1: 安裝後,打開 Arduino IDE,為了讓 Arduino IDE 找到 Ameba 的設定檔，先到【檔案】→【偏好設定】。



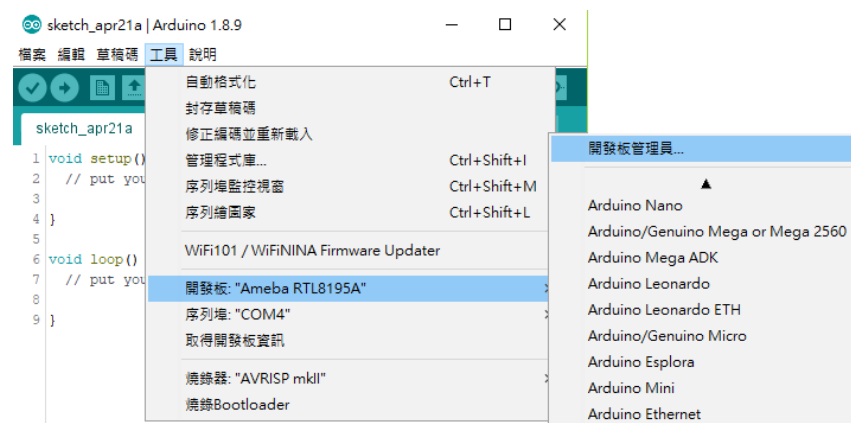
- 步驟 2: 然後在【額外的開發板管理員網址】：填入如下網址。

https://github.com/Ameba8195/Arduino/raw/master/release/package_realtek.com_ameba_index.json



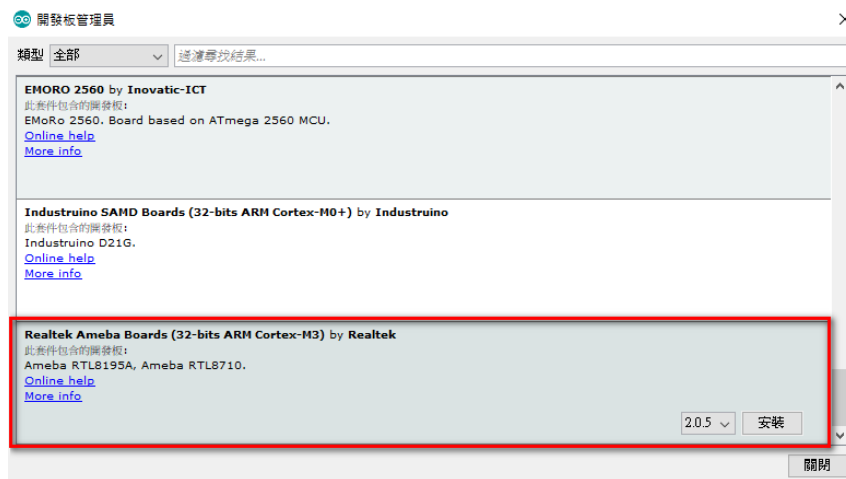
填完之後按 OK，我們將 Arduino IDE 關掉之後重開。

接著到【工具】→【開發板】→【開發板管理員】如下圖所示。

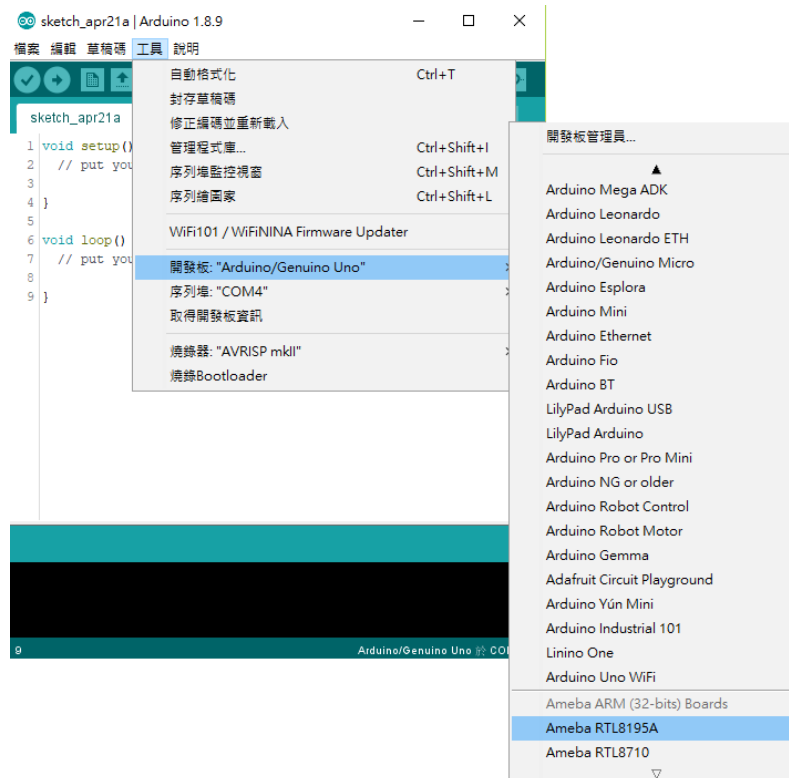


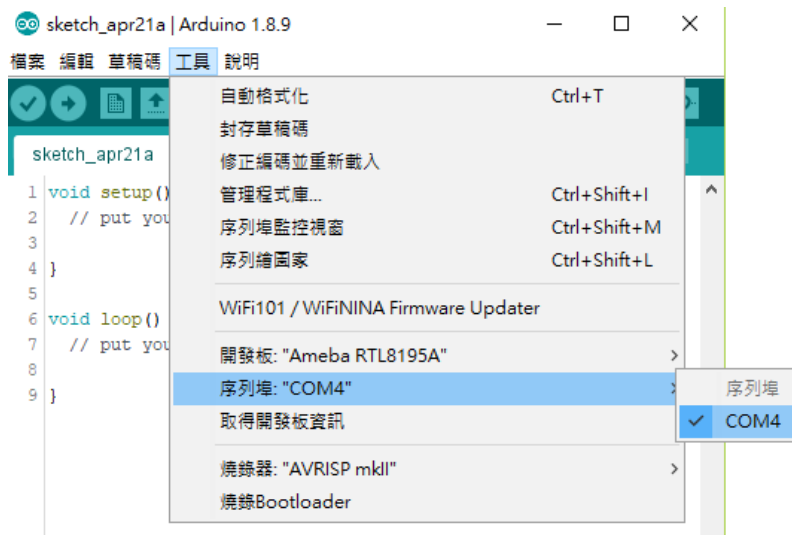
在【開發板管理員】裡，它需要約十幾秒鐘整理所有硬體檔案，如果網路狀況不好可能會等上數分鐘。每當有新的硬體設定，我們需要重開【開發板管理員】，所以我們等一會兒之後，關掉【開發板管理員】，然後再打開它，將捲軸往下拉找到【Realtek Ameba Boards】，點右邊的 Install，這時候 Arduino IDE 就根據 Ameba 的設定檔開始下載 Ameba

所需要的檔案，安裝成功後，關閉開發板管理員視窗。



- 步驟 3: 接著將板子選成 Ameba，選取【工具】→【開發板】→【Ameba RTL8195A】並正確選擇對應的序列埠。





這樣開發環境就設定完成了。

小提醒: COM4 序號埠是筆者目前的,你們可能和我不一樣,可至裝置管理員裡面→通用序列埠匯流排控制器去查看哦~

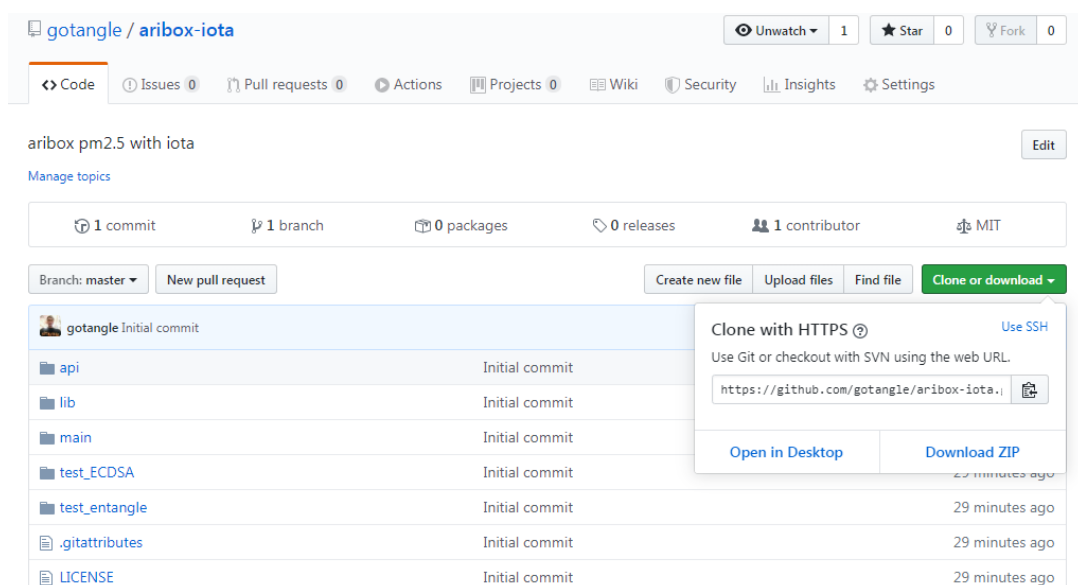
參考: Ameba Arduino: Getting Started With RTL8195

<https://www.amebaiot.com/ameba-arduino-getting-started>

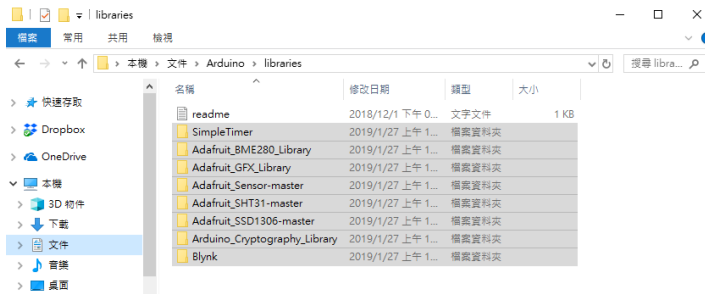
上傳程式到 Ameba

■ 步驟 1: 下載 airbox-iota 並解壓縮檔案。

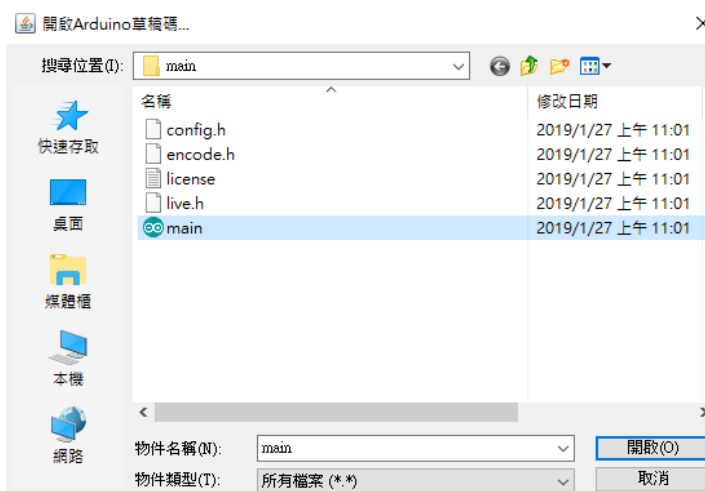
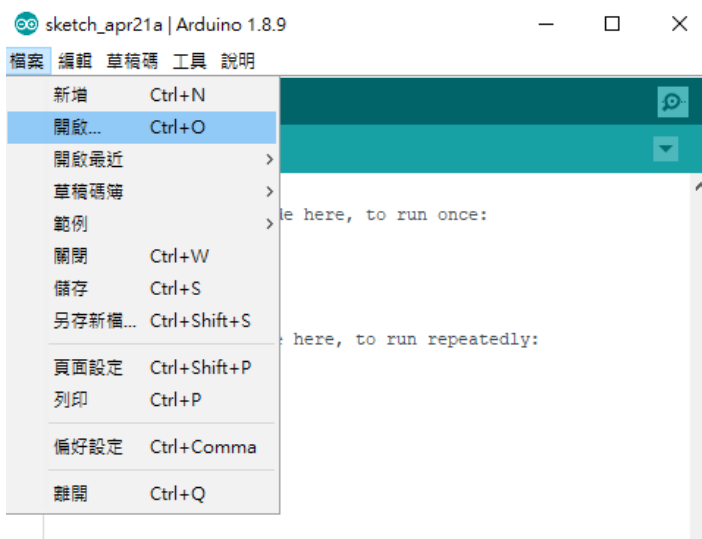
<https://github.com/gotangle/airbox-iota>



- 步驟 2: 將 aribox-iota-master 裡整個 lib 目錄的檔案到【我的文件\Arduino\library】目錄裡。



- 步驟 3: 開啟主程式【main.ino】專案。



- 步驟 4: 在【config.h】內設定 wifi SSID、wifi 密碼、緯度 gps_lat 及經度 gps_lon。設定空氣盒子要連線的網路，一般都是家裡的 WIFI 無線網路的帳號及密碼。

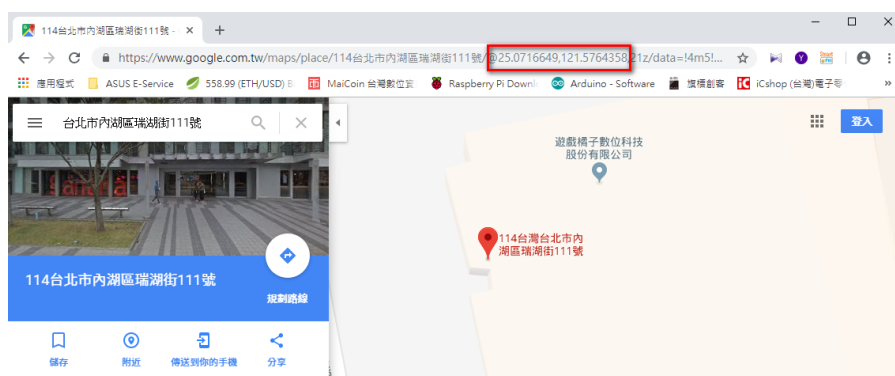


```
1 #ifndef CONFIG_HACK4U
2 #define CONFIG_HACK4U
3
4 char ssid[] = "Darwin"; // your network SSID (name)
5 char pass[] = "09131118"; // your network password
6 char gps_lat[] = "25.0716649"; // 你所在的緯度
7 char gps_lon[] = "121.5764358"; // 你所在的經度
8 const int lass_period = 60; // 幾秒想傳一次資料 建議實際上是 60
9 uint8_t iota_node[4] = {140, 116, 246, 164};
10
11 #endif
```

經緯度的設定部份,可以打開 Google MAP 選擇空氣盒子所在的地點。

可考慮使用 IOTA Area Codes(IAC) , IOTA 提出了一個新標準,該標準將使基於 IOTA 的應用程序能夠圍繞地理區域建構。經過編碼的位置代碼,可用於標記和檢索與特定位置相關的 IOTA 交易。

<https://blog.iota.org/iota-area-codes-a-proposal-to-geo-tag-iota-transactions-d3c457d1df1b>



- 步驟 5: 點選左上角 上傳 按鈕,上傳程式至開發板。

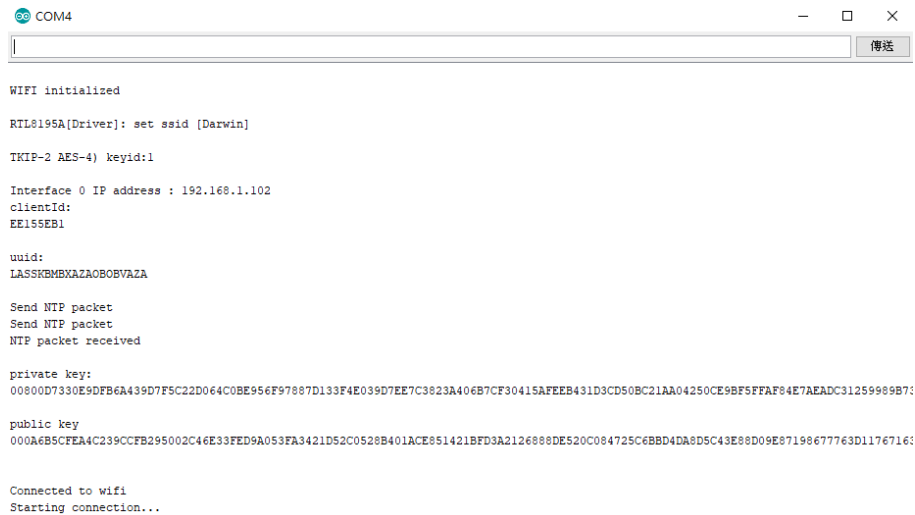


```
25 #define OTA_PORT 5000
26 #define RECOVER_PIN 18
27
```

- 步驟 6: 點選右上角【序列埠監控視窗】，開啟 serial monitor 並選擇【baud rate】為【9600】。



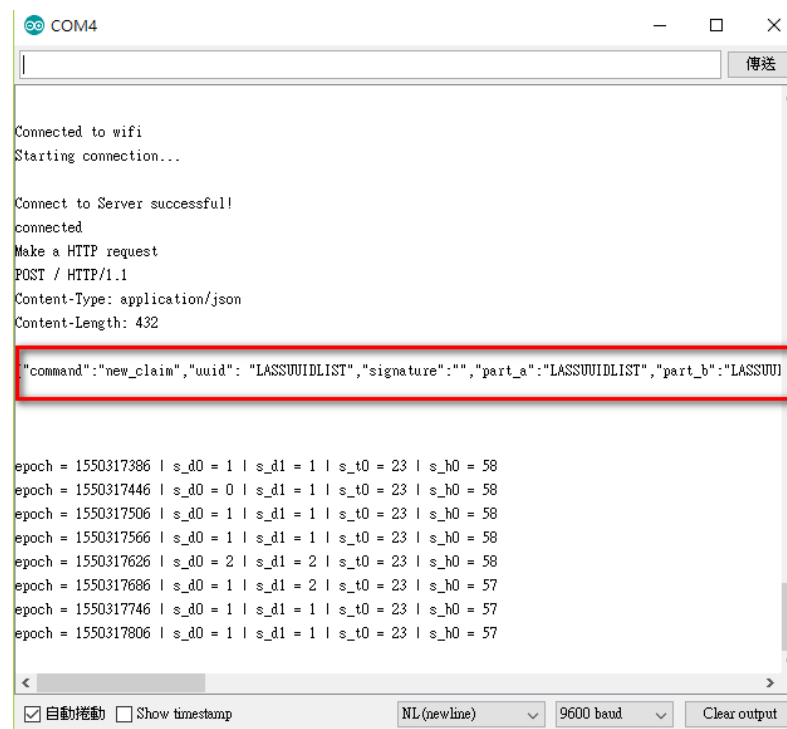
初始化 wifi 成功。



步驟 7:

clientId 為網卡 MAC 位址後 8 碼,
uuid 為 LASS + toTrytes(clientID), 記得請記下 uuid

讀取感測器(sensor)資料、編碼、發出聲明



```
COM4

Connected to wifi
Starting connection...

Connect to Server successfull!
connected
Make a HTTP request
POST / HTTP/1.1
Content-Type: application/json
Content-Length: 432

{"command": "new_claim", "uuid": "LASSUUIDLIST", "signature": "", "part_a": "LASSUUIDLIST", "part_b": "LASSUUI"}

epoch = 1550317386 | s_d0 = 1 | s_d1 = 1 | s_t0 = 23 | s_h0 = 58
epoch = 1550317446 | s_d0 = 0 | s_d1 = 1 | s_t0 = 23 | s_h0 = 58
epoch = 1550317506 | s_d0 = 1 | s_d1 = 1 | s_t0 = 23 | s_h0 = 58
epoch = 1550317566 | s_d0 = 1 | s_d1 = 1 | s_t0 = 23 | s_h0 = 58
epoch = 1550317626 | s_d0 = 2 | s_d1 = 2 | s_t0 = 23 | s_h0 = 58
epoch = 1550317686 | s_d0 = 1 | s_d1 = 2 | s_t0 = 23 | s_h0 = 57
epoch = 1550317746 | s_d0 = 1 | s_d1 = 1 | s_t0 = 23 | s_h0 = 57
epoch = 1550317806 | s_d0 = 1 | s_d1 = 1 | s_t0 = 23 | s_h0 = 57

☒ 自動捲動 ☐ Show timestamp NL(newline) 9600 baud Clear output
```

查看感測紀錄

■ IOTA Tangle explorer

uuid list

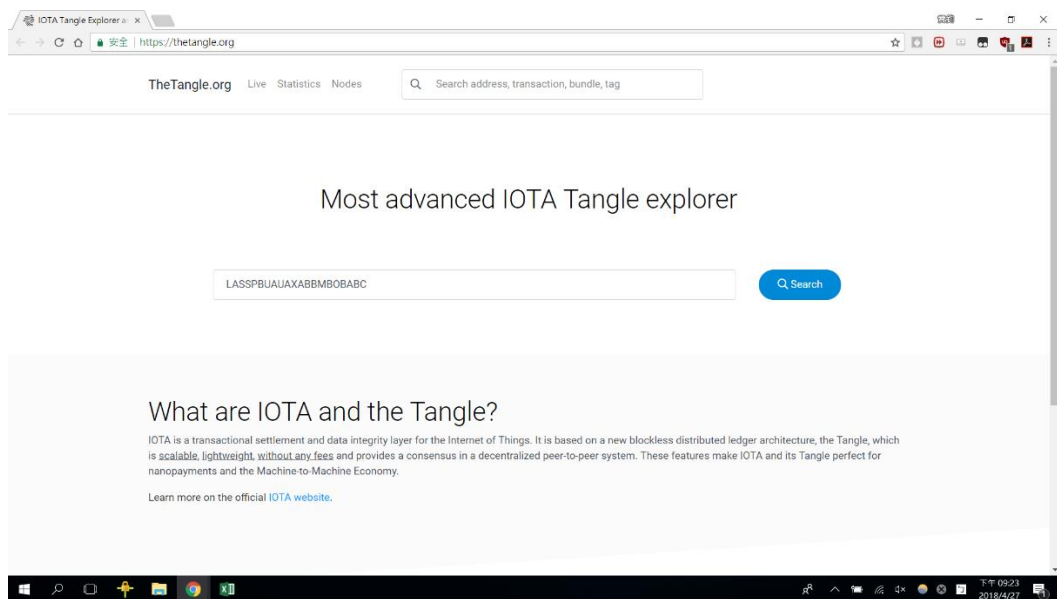
<https://thetangle.org/tag/LASSUUIDLISTC>

uuid: LASSPBUAUAXABBMBOBAB

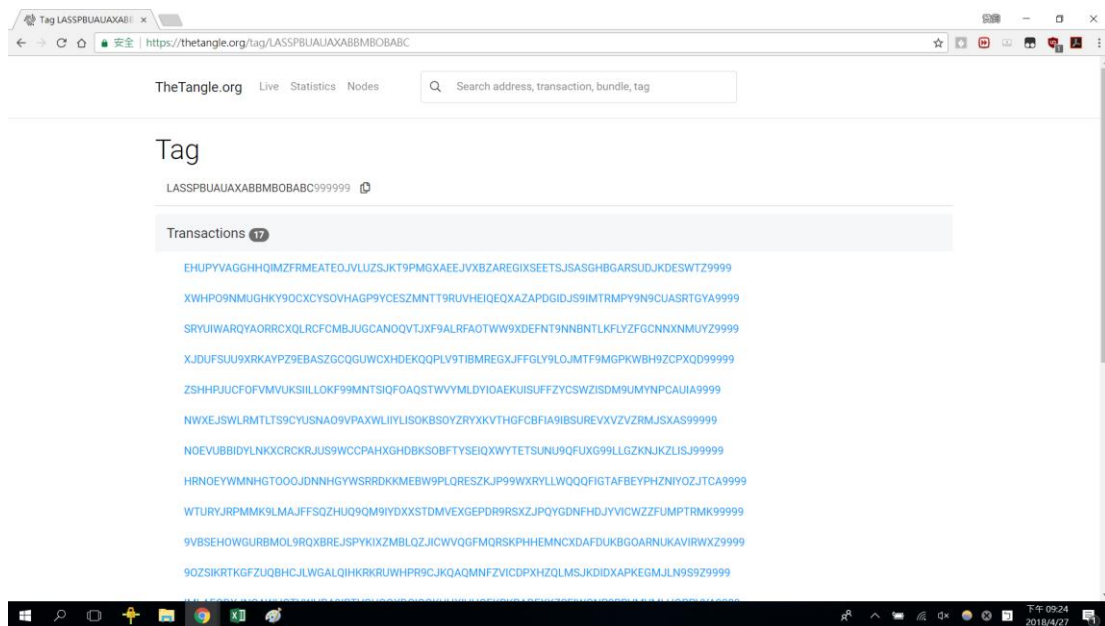
Tag (uuid + 'C') : LASSPBUAUAXABBMBOBABC

<https://thetangle.org/tag/LASSPBUAUAXABBMBOBABC>

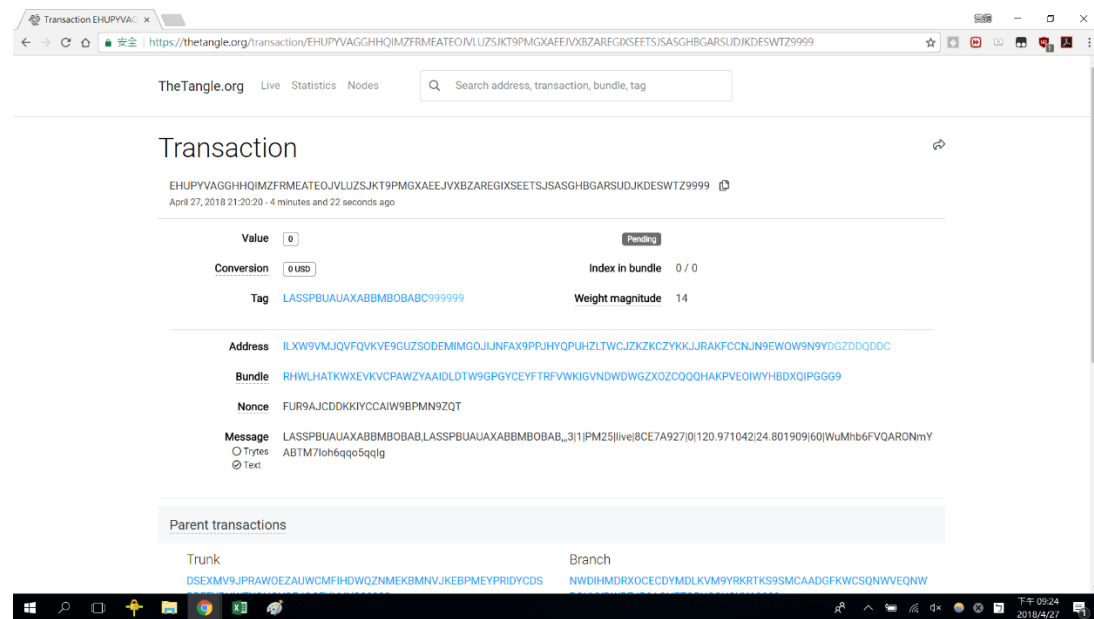
在 [Tangle explorer](https://thetangle.org) 用 Tag 搜尋



顯示結果:



Message 選 Text 可看見相關資料



APIs

- Python 3.6.5
- python airbox-with-*iota*/api/test.py

■ get_uuid_list()

Search for uuids of all available airboxes.

- Input: none ()
- Return: uuid list (list)

■ get_specific_sensor_all_data()

Search for all sensor data by an uuid.

- Input: uuid (string)
- Return: sensor data (json)

■ get_specific_sensor_latest_data()

Search for latest sensor data by an uuid.

- Input: uuid (string)
- Return: sensor data (json)

■ `get_specific_sensor_history_data()`

Search for sensor data in one week by an uuid.

- Input: uuid (string)
- Return: sensor data (json)

■ `get_specific_sensor_date_data()`

Search for latest sensor data by an uuid and a date.

- Input: uuid (string), yyyy-mm-dd (string)
- Return: sensor data (json)

■ `get_all_sensor_latest_data()`

Search for all latest sensor data of each airbox.

- Input: none ()
- Return: sensor data (json)

■ 可參考網站:

- PM2.5 開放資料入口網站: <https://pm25.lass-net.org>
- g0v 零時空汙觀測網: <https://airmap.g0v.asper.tw/v5/#/map>
- LASS/Airbox and IOTA Integration 網站: https://hackmd.io/s/SJnIK_8yM
- DLTcollab/airbox-with-iota: <https://github.com/DLTcollab/airbox-with-iota>
- IOTA Area Codes BLOG: <https://blog.iota.org/iota-area-codes-a-proposal-to-geo-tag-iota-transactions-d3c457d1df1b>