

# 大安高工電子科 分散式智慧環境盒 專題製作報告書

指導老師：

楊仁元 老師

組長：

電子三甲 俞飛揚

組員：

電子三甲 高瑋鴻

電子三甲 洪丞禾

電子三甲 段凱杰

電子三甲 溫哲睿

## 摘要

二十一世紀是個網路資訊蓬勃發展的時代，任何人都離不開網路，從個人需求的即時通訊、資料查詢，到大公司的伺服器建置等等，皆與網路息息相關。可想而知。

在不久的將來，網路會更加深入每個人的生活，不再只是局限於手機、平板等設備而已，而是能透過網路控制家裡的照明、風扇、冷氣等家電，甚至是透過微電腦分析冰箱的庫存，或是能自動分析家裡的環境品質，判斷是否要開起空氣清淨機、除濕機。只要簡單的透過網路自動監控與分析，便能使家庭自動化更佳便捷，省去人力、時間的開銷，並能讓人們更加專注於其他事務。

關鍵字：分散、環境、盒子、DIEB、物聯網、3D 列印。

# Abstract

Booming in the 21st century is a network information age, anybody cannot leave the network. From the individual needs of instant messaging information query, to big company server build and so on, all is closely related to the Internet.

In the near future, Internet will be deeper in everybody's life, no longer just limited to mobile phone tablet devices, but can control many kinds of electrical appliances at home through the Internet, such as lighting system, fan or air conditioning, and even through the microcomputer analysis the inventory of refrigerator, or can automatically analyze the environment quality in the home, determine whether to open the clean air machine dehumidifier. Simple automatic monitoring and analysis over the Internet can make home automation more convenient, save human's labor and time costs, and make the people to focus on other things.

Keywords : Decentralized 、 Environmental 、 Box 、 IoT 、 3D Printing

# 目錄

摘要.....	I
Abstract.....	II
目錄.....	III
表目錄.....	IV
圖目錄.....	V
第一章 緒論.....	1
背景及目的.....	1
預期成果.....	1
第二章 理論探討.....	2
ESP8266-NodeMCU.....	2
溫溼度感測器.....	2
空氣微粒感測器.....	2
紫外線強度感測器.....	3
RTC 實時時鐘.....	4
三軸加速度感測器.....	4
矩陣式三色發光二極體.....	5
第三章 專題準備.....	6
系統方塊.....	6
甘特圖.....	6
流程圖.....	7
電路設計.....	8
軟體設計.....	10
外殼設計.....	19
第四章 專題成果.....	20
成果.....	20
問題與解決.....	21
第五章 結論與建議.....	22
結論.....	22
建議.....	22
參考文獻.....	23
附錄.....	24
附錄一 設備及材料清單.....	24
附錄二 成員簡歷.....	26

## 表目錄

▲表 1 空氣品質及溫溼度數據讀取.....	10
▲表 2 時間讀取.....	11
▲表 3 客戶端數據接收與發送.....	13
▲表 4 伺服端數據接收與發送.....	14
▲表 5 方向與模式判斷.....	16
▲表 6 矩陣顯示.....	18
▲表 7 材料清單.....	24
▲表 8 設備清單.....	25
▲表 9 成員簡歷-俞飛揚.....	26
▲表 10 成員簡歷-段凱杰.....	27
▲表 11 成員簡歷-洪丞禾.....	28
▲表 12 成員簡歷-高瑋鴻.....	29
▲表 13 成員簡歷-溫哲睿.....	30

## 圖目錄

▲圖 1 NodeMCU-ESP8266 .....	2
▲圖 2 DHT22 溫溼度偵測模組.....	2
▲圖 3 PMS3003 空氣品質偵測模組.....	3
▲圖 4 ML8511 紫外線指數偵測模組.....	3
▲圖 5 tiny RTC i2c 實時時鐘模組.....	4
▲圖 6 ADXL345 三軸加速度感測模組.....	5
▲圖 7 WS2812 三色 LED 陣列.....	5
▲圖 8 系統架構圖 .....	6
▲圖 9 專題甘特圖 .....	6
▲圖 10 專題流程圖 .....	7
▲圖 11 室外盒子原理圖 .....	8
▲圖 12 室內盒子佈線圖 .....	8
▲圖 13 室內盒子原理圖 .....	9
▲圖 14 室內盒子佈線圖 .....	9
▲圖 15 室外盒子三視圖 .....	19
▲圖 16 室外盒子上視圖 .....	19
▲圖 17 室內盒子三視圖 .....	19
▲圖 18 室內盒子上視圖 .....	19
▲圖 19 室內盒子三視圖 .....	20
▲圖 20 室外盒子俯視圖 .....	20
▲圖 21 室內盒子(前).....	20
▲圖 22 室內盒子(後).....	20
▲圖 23 演示-等待網路連線 .....	20
▲圖 24 演示-成功連至網路 .....	20
▲圖 25 演示-開啟室外盒子 .....	21
▲圖 26 演示-等待室外盒數據 .....	21
▲圖 27 演示-室外溫溼度 .....	21
▲圖 28 演示-目前時間 .....	21
▲圖 29 演示-室內溫溼度 .....	21
▲圖 30 演示-空氣品質及日期 .....	21

# 第一章 緒論

## 1-1 背景及目的

鑑於關心我們生活周遭的環境，而測量出現在環境的各項因素即是最佳的關注方式，我們選擇了「居家環境盒」當作我們這一次的專題。同時並增加不同區域的環境數據檢測(如室內、戶外、潮濕空間等)，因此以「居家環境盒」作為基礎，再延伸分散偵測數據之能力。

分散式智慧環境盒 (Decentralized Intelligent Environmental Box 【DIEB】，以下簡稱本專題)，主要目的即是創造出一個能結合不同測量資訊，例如：空氣品質、溫度檢測、濕度感測、日期和時間的標示等，也能蒐集來自不同區域的環境數據。

「分散式」表示能從不同的區域蒐集資料，並且統一傳送至雲端數據庫，以便於存取、顯示。

本專題也結合了遠端 WI-FI 遙控、矩陣式 LED 燈和 3D 列印外盒。遠端 WI-FI 遙控是為了幫助使用者即使在一段距離外，仍能獲得本專題所提供的環境品質資訊。矩陣式 LED 使本專題在顯示資訊時能不受圖示的限制，以平易近人的圖示顯示出使用者可以清楚了解的樣式。外盒以近來很具話題性的 3D 列印來製作，為營造簡潔有力而能清楚標示每個不同面顯示的資訊，3D 列印盒上 4 面有列印上英文名稱來標示當面顯示不同種資料。

希望經由此專題，我們都能從中學習到不同的知識和合作的精神，更加了解一個專題的產生方式。

## 1-2 預期成果

我們預計完成整項專題，也就是能監測空氣品質、溫度檢測、濕度感測、日期和時間，也能利用 3D 列印所列印出來的盒子，結合 LED 矩陣，在翻轉至不同面時，達到顯示各項數值的功能，同時也能利用 Wi-fi 在有一段的距離下，監看不同區域的數值。

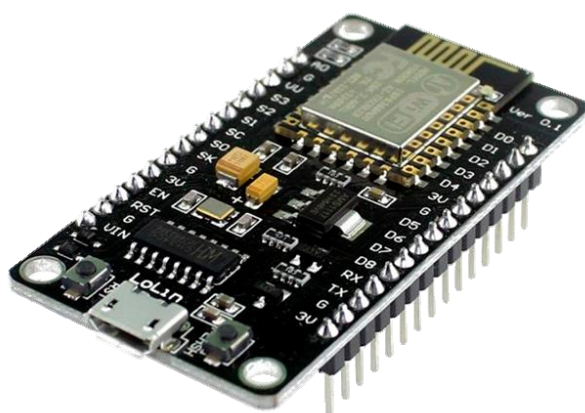
一般天氣預測是針對大範圍進行偵測，而小區域時常會顯得不夠精準，而「分散式智慧環境盒」能對小地區的環境更精確的偵測及獲取資料，並藉由 WiFi 物聯網的方式即時傳送數據給使用者。如果能擴大整體系統運用範圍，從一般的時間、溫度等擴大到經緯度、海拔等，同時結合衛星雲圖的預測，最後由數個伺服器分散演算，再做出更大範圍的精準分析，即時做出準確的天氣預測。

## 第二章 理論探討

### 2-1 ESP8266-Nodemcu

ESP8266 是一個微控制器，具有體積小、造價低廉、以及易取得之特性，並以 Arduino IDE 進行函式庫的撰寫，因此降低使用門檻，簡化了撰寫難度，使初學者更易於開發。本專題即是以 ESP8266 配合 Arduino IDE 撰寫。

ESP8266-12E 具有 13 可用接腳，包含了 UART PORT、並可支援 SPI 協定、IIC 協定，以及支援 Software Serial。內建 CP2102 燒錄器，接上 Micro USB 即可燒錄。電壓水準為 3.3V，如需 5V 則必須外接或以 USB 提供電壓，並使用 Vin 接腳。

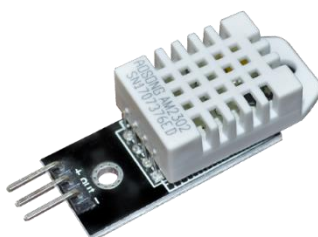


▲圖 1 NodeMCU-ESP8266

### 2-2 溫溼度感測器

AM2302(又稱 DHT22)濕敏電容數位溫濕度模組是一款含有已校準數位信號輸出的溫濕度複合感測器。採用標準單匯流排界面，使系統集成變得簡易快捷。產品為 3 引線（單匯流排排界面）連接方便。

極小的體積、極低的功耗，信號傳輸距離可達 20 米以上，使其成為各類應用甚至最為苛刻的應用場合的最佳選擇，因此本專題採用此模組偵測溫度及濕度。



▲圖 2 DHT22 溫溼度偵測模組

### 2-3 空氣微粒感測器



PMS3003 是一個數字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，並以數字接口形式輸出。此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。

本傳感器採用激光散射原理。即令激光照射在空氣中的懸浮顆粒物上產生散射，同時在某一特定角度收集散射光，得到散射光強隨時間變化的曲線。進而微處理器利用基於米氏 (MIE) 理論的算法，得出顆粒物的等效粒徑及單位體積內不同粒徑的顆粒物數量。



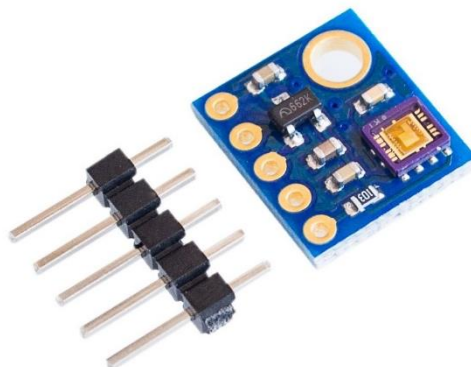
▲圖 3 PMS3003 空氣品質偵測模組

## 2-4 紫外線強度感測器

ML8511 是為日本製造商衝電氣 (OKI) 所推出的紫外線 (UV) 傳感器。該感測器內建有運算放大器，為該公司首款模擬電壓輸出、不含濾光器的 UV 傳感器，而且使用小型、薄型表面封裝方式，適合於便攜式設備用途。

由於 ML8511 採用了容易高整合度的 SOI-CMOS 技術，適合於數字及模擬電路，並加強與連接微處理器的數字輸出電路，進而與感測式亮度控制傳感器 (Ambient Light Sensor) 構成單一芯片的商品陣容。

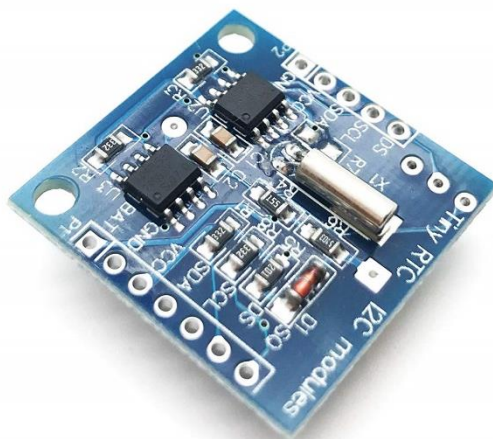
ML8511 可與 UV 光量成正比輸出模擬電壓，因此可以直接連接內建 MCU 的 A/D 數字/模擬轉換器，無需光電轉換電路。



▲圖 4 ML8511 紫外線指數偵測模組

## 2-5 RTC 實時時鐘

DS1307 I2C 實時時鐘晶片(RTC),24C32 32K I2C EEPROM 存儲器,解決 DS1307 帶備用電池不能讀寫的問題。充滿電後,能提供 DS1307 計時 1 年。引出 DS1307 的時鐘引腳,為單片機提供時鐘信號,並可以串聯其他 I2C 設備。並運用石英震盪器,產生高精度振盪頻率。



▲圖 5 tiny RTC i2c 實時時鐘模組

### 2-5-1 晶體壓電效應

壓電效應,是電介質材料中一種機械能與電能互換的現象。壓電效應有兩種,「正壓電效應」及「逆壓電效應」。而一般石英震盪器之運用為「逆壓電效應」。

「逆壓電效應」的原理,是當在壓電材料表面施加電場(電壓),會因為電場作用時電偶極矩會被拉長,壓電材料為了抵抗變化,會沿電場方向伸長。這種通過電場作用而產生機械形變的過程稱為「逆壓電效應」。逆壓電效應實質上是電能轉化為機械能的過程。

### 2-5-2 EEPROM 記憶體

EEPROM,「全稱電子抹除式可複寫唯讀記憶體」,是一種可以通過電子方式多次複寫的半導體儲存裝置。相比 EPROM,EEPROM 不需要用紫外線照射,也不需取下,就可以用特定的電壓,來抹除晶片上的資訊,以便寫入新的數據。

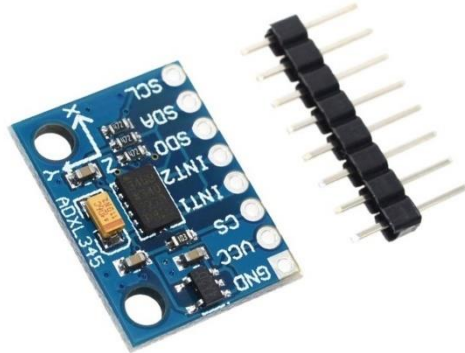
由於 EEPROM 的優秀效能,以及在線上操作的便利,它被廣泛用於需要經常擦拭的 BIOS 晶片以及快閃記憶體晶片,並逐步替代部分有斷電保留需要的 RAM 晶片,甚至取代部份的硬碟功能(見固態硬碟)。它與高速 RAM 成為當前(21 世紀 00 年代)最常用且發展最快的兩種儲存技術。

## 2-6 三軸加速度感測器

ADXL345 是三軸加速計,具有高解析度(13 位元)量測,高達  $\pm 16\text{ g}$ 。數位輸出數據經過格式化,成為 16 位元二補數,並且可透過 SPI(3 或 4 線)

或 I2C 數位介面存取。

ADXL345 非常適合行動裝置應用。此元件適合量測傾斜感測應用的重力靜態加速，以及動作或衝擊導致的動態加速。其高解析度（4 mg/LSB）能達到小於 1.0° 傾斜變化的量測。



▲圖 6 ADXL345 三軸加速度感測模組

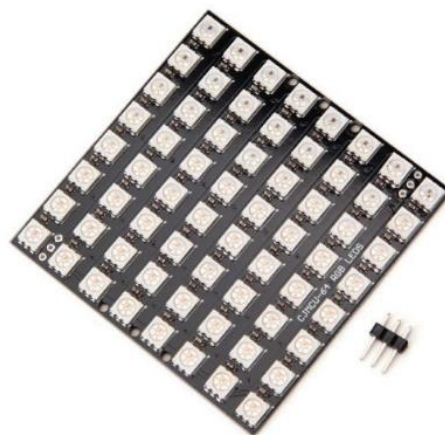
## 2-7 矩陣式三色發光二極體

傳統方式要控制多顆 RGB LED 在電路接線和程式控制方面是非常煩雜的，然而使用內建 WS2812 晶片的 RGB LED 卻是簡單又方便，不管你要控制幾顆 RGB LED，都只要使用 Arduino 3 支腳位就足夠了。

WS2812 RGB LED 是具有內建驅動晶片功能的 LED，亦是使用 5050 LED 封裝加入驅動晶片，驅動方式採串列進出，因此可獨立控制串接 LEDs 的每一顆 LED。

每個 R/G/B 顏色可獨立控制，且每一個顏色可調整 0~255 階調，因此每一個顏色需用到 8bit 控制，每一顆 LED 需用到 24bit 控制。

WS2812 的資料傳送介面，是利用「時間」來決定是為邏輯 0 或是邏輯 1。例如說拉高電位 350ns 及拉低準位 800ns 就是指邏輯 0，邏輯 1 的產生方式也可以以此類推，而這些時間都可以允許正負 150ns 的誤差。

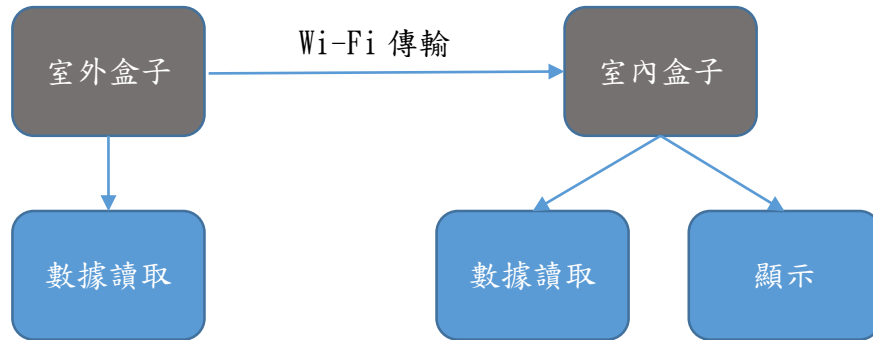


▲圖 7 WS2812 三色 LED 陣列

## 第三章 專題準備

### 3-1 系統方塊

本專題主要從室外盒子進行資料讀取後，傳送到室內盒子做顯示。數據讀取的部分，室外盒子包含 UV 值、空氣品質、溫度以及濕度；而室內盒子除了有空氣品質、溫度以及濕度之外，還會偵測翻轉方向以及實時時間。



▲圖 8 系統架構圖

### 3-2 甘特圖

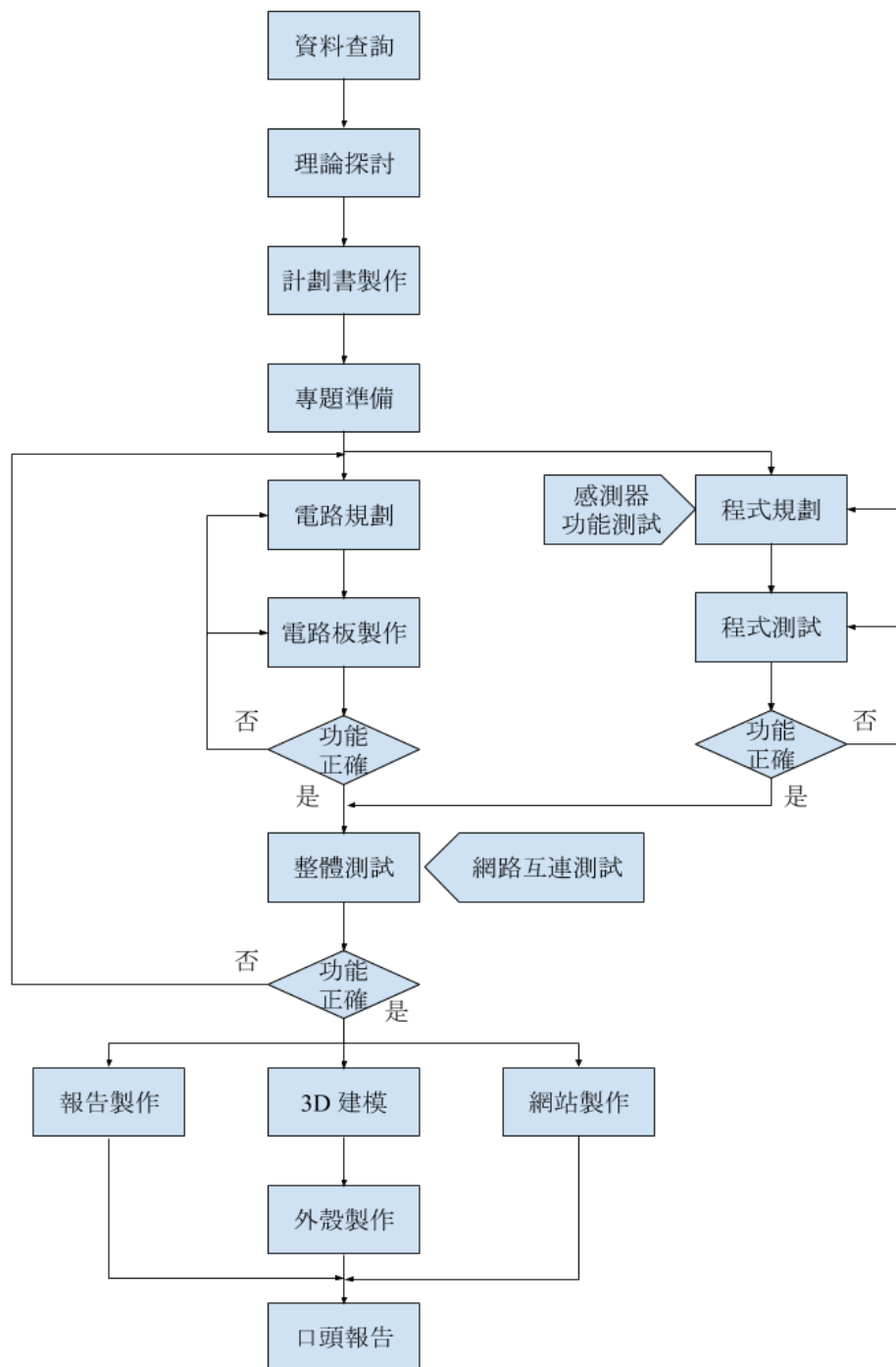
本專題在分工的部分，不採用平均分配，而是依照組員能力多寡，分配其能力所允許的工作。「負責成員」的欄位中，皆是以姓氏作為標記。

週次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	負責成員	
資料查詢	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	全員
理論探討	■	■																			全員
計劃書製作	■	■	■	■	■																全員
專題準備	■	■																			溫,段
電路規劃			■	■	■																俞
電路板製作				■	■	■	■														俞
程式撰寫-sensor						■	■														高
程式撰寫-資料傳輸							■	■													高
程式測試							■	■	■	■											高
整體測試								■	■	■	■	■	■	■							全員
3D 建模													■	■							溫,俞
外殼製作														■	■	■					溫,俞
網站架設													■	■	■	■	■	■	■		洪,段
報告撰寫							■	■	■	■	■	■	■	■	■	■	■	■	■	■	俞,段
口頭報告										■		■		■		■		■			全員
預定進度	4	8	13	20	26	30	36	41	45	50	56	62	67	77	80	85	90	98	100		累積百分比%

▲圖 9 專題甘特圖

### 3-3 流程圖

本專題步驟會預先蒐集可能會用到的資料，再去做其他的設計。而設計又可以分為「硬體」部分的電路規劃、電路板製作，以及「軟體」部分的程式規劃與測試。待整合完成後，才會根據電路大小設計外殼。



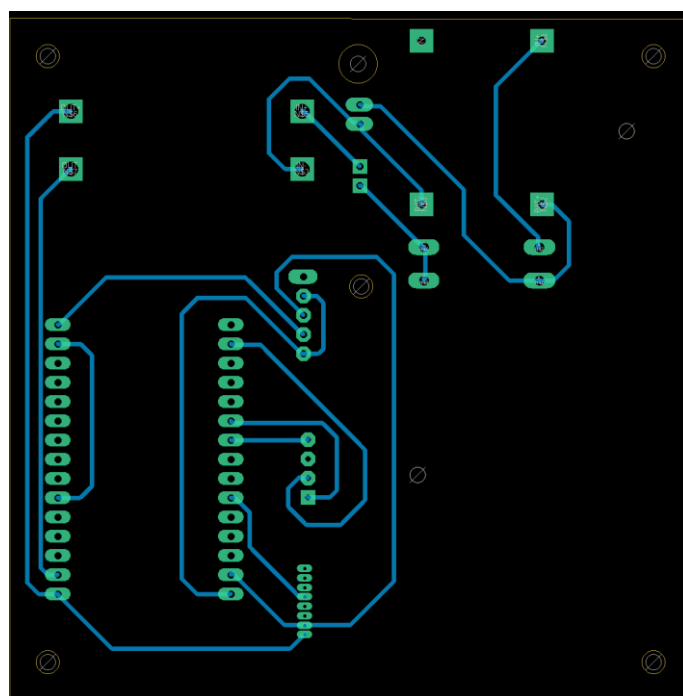
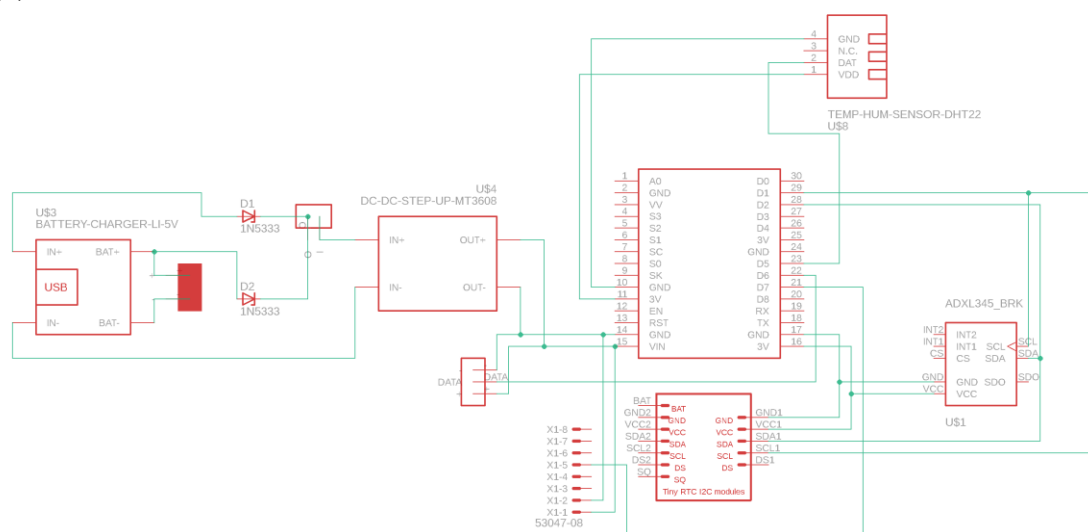
▲圖 10 專題流程圖

### 3-4 電路設計

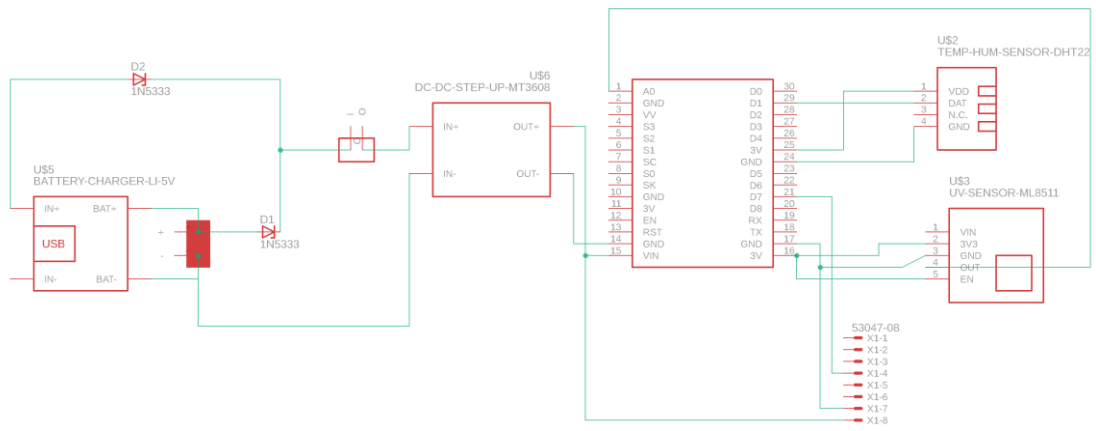
本專題電路設計的部分，是使用 Autodesk EAGLE 所製作。相對於 Altium Designer，EAGLE 介面簡潔許多，就好比 Windows 內建的小畫家比上 Adobe illustrator，雖然功能不及後者，但其基本功能就足夠完成本專題。

電源供應的部分，為了達到在接著 MicroUSB 時，能夠同時運作整個專題，並讓其中的 18650 電池進行充電動作，因此採用了兩個蕭特基二極體。原理是利用 USB 與 18650 的輸出電壓差，使得在充電與為充電交換時，能夠及時改變選擇輸入的端口，功能類似於數位邏輯中的及閘(NOR)。

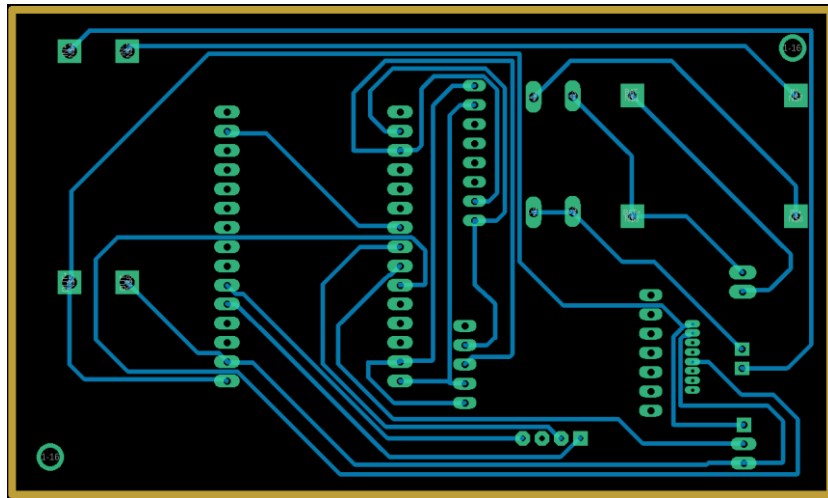
最後再利用 LM2577 直流升壓模組，供給穩定的 5 伏特給 NodeMCU 以及其他模組。



▲圖 12 室內盒子佈線圖



▲圖 13 室內盒子原理圖



▲圖 14 室內盒子佈線圖

## 3-5 軟體設計

程式撰寫的部分，是使用 Arduino IDE 進行編輯的，可以另外安裝燒錄版以及模組的函數庫，使得編輯上更為方便。主要可以分為「數據讀取」、「資料傳輸」以及「LED 顯示」三大部分，首先會經由各個感測器讀取數據後，再經由 Wi-Fi 傳送資料到室內盒子，最後再送到矩陣式 RGB LED 做顯示的作業。

### 3-5-1 數據讀取

就以空氣品質感測器以及溫溼度感測器作為舉例，先是函式庫引入，並設定其腳位，以及相關功能的設定後，再經由感測器的函式庫，讀取正確的環境數據，代入相對應之變數。

<pre>#include &lt;SoftwareSerial.h&gt; #include "PMS.h " SoftwareSerial pmsSerial(13, 15); PMS pms(pmsSerial); PMS::DATA pmsData;  #include "DHT.h " #define DHTPIN 5 #define DHTTYPE DHT22 DHT dht(DHTPIN, DHTTYPE);  while (pms.read(pmsData)) {     pm1 = pmsData.PM_AE_UG_1_0;     pm2 = pmsData.PM_AE_UG_2_5;     pm10 = pmsData.PM_AE_UG_10_0;     break; } h = dht.readHumidity(); t = dht.readTemperature();</pre>	<p>空氣品質感測器函式庫。</p> <p>//Rx, Tx 設定 PMS 序列埠 設定 PMS 物件</p> <p>溫溼度感測器函式庫 // D1 數據傳輸腳位 // DHT 22 模組版本</p> <p>} 取得空氣中 懸浮微粒濃度</p> <p>} 取得溫溼度數據</p>
--	---

▲表 1 空氣品質及溫溼度數據讀取



### 3-5-2 RTC 時間讀取

本專題的 RTC，是使用 IIC 協定進行資料傳輸，所以在程式一開始就必須先引入相關函式庫，之後才能再使用 RTC 的函式庫讀取數據，最後將讀取到的數據代入相對應的變數。

<pre>#include &lt;Wire.h&gt; #include "RTClib.h" RTC_DS1307 RTC;  Void setup(){ Wire.begin(); RTC.begin(); RTC.adjust(DateTime(F(__DATE__), F(__TIME__))); }  DateTime now = RTC.now(); d = now.day(), DEC; mo = now.month(), DEC; mi = now.minute(), DEC; hr = now.hour(), DEC;</pre>	<p>引入匯流排函式庫。</p> <p>引入 RTC 函式庫。</p> <p>設定 RTC。</p> <p>啟用 RTC 並調整時間。</p> <p>調整日期、時間。</p> <p>取得日期時間。</p> <p>代入預設的日期。</p> <p>代入預設的月份。</p> <p>代入預設的分鐘數。</p> <p>代入預設的小時數。</p>
--	--

▲表 2 時間讀取

### 3-5-3 Wi-Fi 資料傳輸-室內盒子

Wi-Fi 傳輸的部分，除了要引入函式庫，需要設定欲連接之路由器的相關數據，像是名稱、密碼、接入端口等等。本專題還有設計重新連線連線，當連線逾時一段時間，將會重新嘗試連線，所以當斷線後再重新連線就不需依靠人力去做連線的動作。

<pre>#include &lt;ESP8266WiFi.h&gt; const char* ssid = "ELECT-STUDENT"; const char* password = "student106"; WiFiClient client; WiFiServer server(2222);  if ((WiFi.status() != WL_CONNECTED) &amp;&amp; lastT - millis() == 10000)     WiFi.begin(ssid, password);     lastT = millis();     cl = 0; } if ((WiFi.status() == WL_CONNECTED) &amp;&amp; cl &lt; 1) {     Serial.print("Connected to "); Serial.println(ssid);     Serial.print(" IP Address: "); Serial.println(WiFi.localIP());     cl++; }  if (!client.connected()){     client = server.available(); } else{     if (client.available() &gt; 0){         while (client.available()){             data[ind] = client.read();             ind++;         }     }  client.flush();</pre>	<p>引入 ESP8266 函式庫。</p> <p>設定欲接入路由器名稱。</p> <p>設定欲接入路由器密碼。</p> <p>設定接入端口。</p> <p>判斷逾時連線。</p> <p>重新嘗試連線。</p> <p>連線成功。</p> <p>顯示提示與 IP 位址。</p> <p>嘗試與 Client 連線。</p> <p>讀取 Client 端發送的字元。</p>
--	---

<pre>if (ind &gt;= 5) {     t2 = (data[0] - '0') * 10 + (data[1] - '0');     h2 = (data[2] - '0') * 10 + (data[3] - '0');     pm1 = (data[4] - '0') * 10 + (data[5] - '0');     ind = 0; } } }</pre>	<p>將接收到的字元，轉換成數值，並代入預設的變數。</p>
--	--------------------------------

▲表 3 客戶端數據接收與發送

### 3-5-4 Wi-Fi 資料傳輸-室外盒子

室內盒子網路的部分與室內盒子大同小異，差異幾乎只在於設定接收或是發送功能的不同。

<pre>#include &lt;ESP8266WiFi.h&gt; const char* ssid = "ELECT-STUDENT"; const char* password = "student106"; WiFiClient client; const char *host = "192.168.100.166"; WiFiServer server(2222);  void setup(){   WiFi.mode(WIFI_STA);   WiFi.begin(ssid, password); }  if (!client.connect(host, 2222)) {   Serial.println("connection failed");   count1 = 0; } else if (count &lt; 1) {   Serial.println("connected succed");   count1++; }  client.print(int(t)); client.print(int(h)); client.print(pm2);</pre>	<p>引入 ESP8266 函式庫。</p> <p>設定欲接入路由器名稱。</p> <p>設定欲接入路由器密碼。</p> <p>設定欲連至的 IP 位址。</p> <p>設定接入端口。</p> <p>設定 Wi-Fi 模式，並啟用。</p> <p>嘗試連線至伺服器端。</p> <p>} 傳送所得的環境數據</p>
--	---

▲表 4 伺服器端數據接收與發送

### 3-5-5 ADXL345 偵測擺放方向

擺放的方不同，決定要顯示哪一種數據。本專題主要是讀取 X、Y、Z 軸的方向(即上、下、左、右)，判斷擺放的方向，而判斷完方向後，會去比照安排好的模式內容，決定要顯示甚麼數據。

<pre>X0 = getData(0x32); X1 = getData(0x33); X = ((X1 &lt;&lt; 8) + X0) / 256.0;  Y0 = getData(0x34); Y1 = getData(0x35); Y = ((Y1 &lt;&lt; 8) + Y0) / 256.0;  Z0 = getData(0x36); Z1 = getData(0x37); Z = ((Z1 &lt;&lt; 8) + Z0) / 256.0;  if (Z &gt; 0.5 &amp;&amp; Z &lt; 2) {     m1 = 3; } else if (X &lt; 0.1    X &gt; 255.5) {     m1 = 0; } else if (X &lt;= 1 &amp;&amp; X &gt; 0.5) {     m1 = 1; } else if (X &lt; 255.5 &amp;&amp; X &gt; 254) {     m1 = 2; }  if (Y &gt; 254) {     m2 = 0; } else {     m2 = 1; }</pre>	<pre>//取得 X 軸低位元資料 //取得 X 軸高位元資料  //取得 Y 軸低位元資料 //取得 Y 軸高位元資料  //取得 Z 軸低位元資料 //取得 Z 軸高位元資料</pre> <p>計算所得 X, Y, Z 軸資料， 得知擺放為橫向或直向及 哪一面朝上。</p>
---	--

```
switch (m1) {
  case 0:
    if (m2 == 0) {
      model = 1;
    } else {
      model = 3;
    }
    break;
  case 1:
    model = 2;
    break;
  case 2:
    model = 4;
    break;
  case 3:
    model = model;
  default:
    break;
}
```

參考所得參數，設定  
model，切換所顯示之數  
據。

▲表 5 方向與模式判斷

### 3-5-6 WS2812B RGB LED 顯示控制

WS2812 矩陣式 RGB LED 只需要三支接腳即可動作，即 VCC、GND 以 Data 腳位，所以再設置腳位的地方只需設置個數據讀取接腳即可。其為一陣列顯示器，因此設置陣列函數以方便進行顯示作業。以下為設定和基本控制並以溫度顯示為範例：

<pre>#include &lt;Adafruit_NeoPixel.h&gt;; #define PIN 12 Adafruit_NeoPixel pixel = Adafruit_NeoPixel(72, PIN, NEO_GRB + NEO_KHZ800); pixel.setPixelColor(i, pixel.Color(r, g, b));  char tempten[10][13] = {     {9, 10, 11, 12, 14, 33, 35, 36, 38, 57, 58, 59, 73},     {10, 13, 34, 37, 58, 73, 73, 73, 73, 73, 73, 73, 73},     {9, 10, 11, 12, 33, 34, 35, 38, 57, 58, 59, 73, 73},     {9, 10, 11, 14, 33, 34, 35, 38, 57, 58, 59, 73, 73},     {9, 14, 33, 34, 35, 36, 38, 57, 59, 73, 73, 73, 73},     {9, 10, 11, 14, 33, 34, 35, 36, 57, 58, 59, 73, 73},     {9, 10, 11, 12, 13, 14, 33, 34, 35, 36, 57, 58, 59},     {9, 14, 33, 36, 38, 57, 58, 59, 35, 73, 73, 73, 73},     {9, 10, 11, 12, 14, 33, 34, 35, 36, 38, 57, 58, 59},     {9, 10, 11, 14, 33, 34, 35, 36, 38, 57, 58, 59, 73} };</pre>	<p>引入函式庫。 設定腳位為 12。 設定 LED 數量和基本參數。</p> <p>控制某一顆 LED 顯示某個顏色。</p> <p>設定溫度十位數 0-9 部分</p>
--	--

<pre> char tempone[10][13] = {     {5, 6, 7, 16, 18, 29, 31, 40, 42,     53, 54, 55, 73},     {6, 17, 30, 41, 54, 73, 73, 73, 73,     73, 73, 73, 73},     {5, 6, 7, 16, 29, 30, 31, 42, 53,     54, 55, 73, 73},     {5, 6, 7, 18, 29, 30, 31, 42, 53,     54, 55, 73, 73},     {5, 18, 29, 30, 31, 40, 42, 53, 55,     73, 73, 73, 73},     {5, 6, 7, 18, 29, 30, 31, 40, 53,     54, 55, 73, 73},     {5, 6, 7, 16, 18, 29, 30, 31, 40,     53, 54, 55, 73},     {5, 18, 29, 31, 40, 42, 53, 54, 55,     73, 73, 73, 73},     {5, 6, 7, 16, 18, 29, 30, 31, 40,     42, 53, 54, 55},     {5, 6, 7, 18, 29, 30, 31, 40, 42,     53, 54, 55, 73} }; char templ[] = {0, 1, 2, 21, 26, 25, 24, 44, 73, 73, 73, 73, 73};  void displayINtemp() {     pixel.clear();     for (int i = 0; i &lt; 13; i++) {         pixel.setPixelColor(tempten[int(tl) / 10][i], pixel.Color(5, 5, 5));         pixel.setPixelColor(tempone[int(tl) % 10][i], pixel.Color(5, 5, 5));         pixel.setPixelColor(templ[i], pixel.Color(5, 5, 5));     }     pixel.show(); } </pre>	<p>設定溫度個位數 0-9 部分</p> <p>設定顯示°C</p> <p>清除先前保持的顯示狀態</p> <p>顯示溫度十位數</p> <p>顯示溫度個位數</p> <p>顯示°C</p>
---	--

▲表 6 矩陣顯示



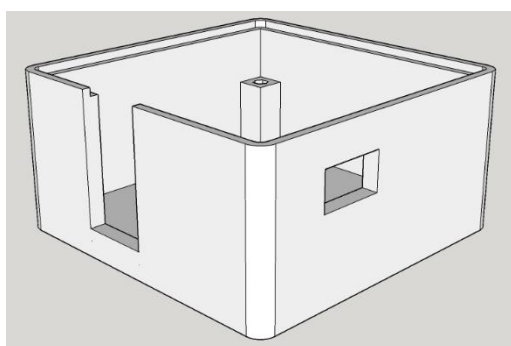
### 3-6 外殼設計

兩個盒子，皆是使用 Google sketchup 所設計。此軟體介面簡單，操作方便，並且有額外的空衝模組可供使用，極為方便。

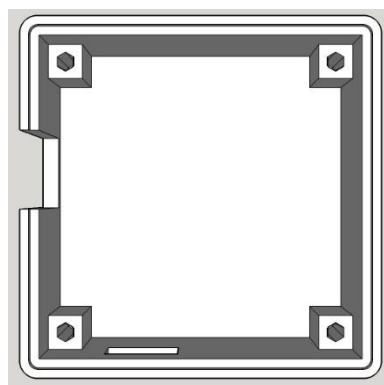
以下兩個上視圖中，可見到些許六角形孔，目的是為了安裝六角銅柱，以固定電路板，防止晃動而造成內部模組損壞。上緣的部分有稍微下凹 2.5mm，目的是要方便透明壓克力板及矩陣式 RGBLED 的安裝及檢修。

室外盒的側邊有兩開口，較常者適用於放置 PMS3003 及 DHT22，以便偵測正確數據；較小者則為 MicroUSB 充電插槽。

適合底部開口的部分，較為方大的為 PMS3003 風扇的放置口，右側為 DHT22 的放置口，而右下為開關的放置處，方便啟動與關閉。



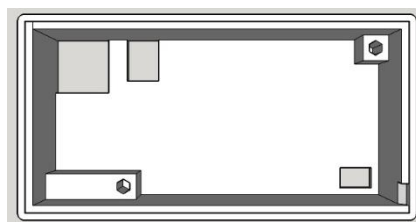
▲圖 15 室外盒子三視圖



▲圖 16 室外盒子上視圖



▲圖 17 室內盒子三視圖

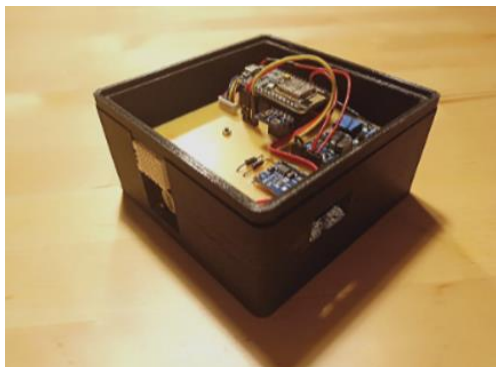


▲圖 18 室內盒子上視圖

## 第四章 專題成果

### 4-1 成果

下面四張圖為本專題最後完成的「室內盒子」以及「室外盒子」。兩個盒子的外殼列印出來後，有比預期大了一些，但基本上不影響整體功能。室內盒子的矩陣式 RGB LED，因為亮度高，直視容易造成眼部受損，因此以白紙覆蓋之，以減少亮度，同時去除原件上的雜色，使得閱讀數值更加容易。



▲圖 19 室內盒子三視圖



▲圖 20 室外盒子俯視圖



▲圖 21 室內盒子(前)



▲圖 22 室內盒子(後)

演示的部分，為了配合前置的連線作業，以及後續四個不同面的數據顯示，直接使用圖像作為功能演示。圖片中，紅色的盒子為「室內盒子」，其作手邊黑色的盒子則為「室外盒子」。(為了能夠清楚顯示矩陣式 RGB LED 上的數據，圖片中的背景光線會稍顯得昏暗。)



▲圖 23 演示-等待網路連線



▲圖 24 演示-成功連至網路



▲圖 25 演示-開啟室外盒子



▲圖 26 演示-等待室外盒數據



▲圖 27 演示-室外溫溼度



▲圖 28 演示-目前時間



▲圖 29 演示-室內溫溼度



▲圖 30 演示-空氣品質及日期

## 4-2 問題與解決

### 一、語音模組：

原本希望研究 WF8266RV 非特定人聲中文語音離線語音模組，並應用於本專題，但因為 wf8266rv 不支援 Arduino IDE 撰寫，只能另外透過其他程式運作，因此暫時不向語音方面發展。

### 二、電源供應：

原本使用的 LM2577 直流升壓模組，會時常有電壓不穩定，及因為輸出小於輸入的問題，後來才考慮改成使用 7805 穩壓 IC，但因為要用 7805，必須重新設計電路及製作電路板，加上發現問題時，距離專題發表的時間已經不多了，因此暫時使用行動電源作為代替。

## 第五章 結論與建議

### 5-1 結論

這個題目與「物聯網」息息相關，發展空間沒有極限，不過目前能做到的僅僅是兩個單體之間資料傳送，一個盒子只能單純搜集資料並上傳；另一個則是能同時搜集資料，並顯示另一顆單體的數據及本身的數據。

數據整合之後，在派段盒子的放置方向，進一步透過矩陣式 RGB LED 去顯示不同的環境數據。整體上來看，完整度粗估有 95%，而剩下的 5%則是一些尚未解決的問題，例如：RGB LED 不規律閃爍、實時時鐘時間不正確以及數據錯誤等等。

### 5-2 建議

專題原為一個簡單的獨立系統，只能單純顯示時間、溫濕度、空氣品質，希望能延伸到遠端傳送數據資料到數據庫，再從數據庫取出資料傳到行動裝置及其他室內配備。甚至能擴大大範圍環境運用，從一般的時間、溫度等擴大到經緯度、海拔等，最後由數個伺服器分散演算，再做出更精準的氣象分析與預測。

除了大範圍的數據外，還可另外加入語音模組，像是 Google Assistant 或是 Siri 等等諸如此類的語音助理，應用對象就可以包括盲人及行動不便者。

另外電源供應方面，建議可直接採用 7805 穩壓 IC，而不採用升壓模組，但如果改成使用穩壓 IC，則電池必須使用兩顆以上的 18650 串聯，才可符合 7805 輸入最低電壓。

## 參考文獻

- 一、UVA-30A 紫外線傳感器模組。2018 年 8 月 21 日，取自  
<https://www.taiwaniot.com.tw/product/uvm-30a%E7%B4%AB%E5%A4%96%E7%B7%9A%E5%82%B3%E6%84%9F%E5%99%A8%E6%A8%A1%E7%B5%84/>
- 二、PM2.5 空氣感測器。2018 年 8 月 21 日，取自  
<https://vmaker.tw/archives/3812>
- 三、楊仁元、張顯盛、林家德 (2014)。專題製作理論與呈現技巧。台北市：台科大圖書股份有限公司
- 四、以 ESP8266 存取 IoT 數據資料庫。2018 年 8 月 21 日，取自  
<http://www.funbroad.tw/2017/06/esp8266-iot.html>
- 五、蔡朝洋、蔡承佑 (2018)。電子學 I。台北市：全華圖書股份有限公司。
- 六、DHT22 Temperature & Humidity sensor on NodeMCU using Arduino IDE。2018 年 8 月 21 日，取自 <https://roboindia.com/tutorials/DHT11-NodeMCU-arduino>
- 七、ML8511。2018 年 8 月 21 日，取自  
<https://learn.sparkfun.com/tutorials/ml8511-uv-sensor-hookup-guide/all>
- 八、PMS3003。2018 年 8 月 21 日，取自  
<http://jjpaid.blogspot.com/2017/02/ywrobot-ems-pms3003-pm25-sensor.html>

# 附錄

## 附錄一 設備及材料清單

### 材料清單

類別	設備、軟體名稱	應用說明
NodeMCU	ESP8266	連接 Wi-Fi 熱點，傳遞資料。
充電電池	18650	電力來源。
三軸加速度	ADXL335	偵測設備放置方向。
時間	tiny RTC i2c	輸出實際時間。
紫外線	GYML8511	偵測室外紫外線強度。
溫濕度感測	DHT11	偵測室內外溫濕度。
pm2.5	PMS3003	偵測室內外空氣品質。
開關	4 向 2 線	設備總開關。
耗材	電路板	製作電路用。
耗材	杜邦線	偵測器接至電路板。
耗材	ABS 線料	3D 列印材料。
耗材	焊錫	製作電路用。

▲表 7 材料清單

## 設備清單

類別	設備、軟體名稱	應用說明
設備	桌上型電腦	1. 撰寫程式語言 2. 製作企劃書 3. 製作簡報 4. 查詢相關知識 5. 數據庫建立
設備	智慧型手機	1. 紀錄專題製作的過程 2. 查詢相關知識
設備	Skymaker R2 3D印表機	1. 外殼列印
軟體	Arduino 1.8.5	1. 撰寫程式碼
軟體	Autodesk EAGLE	1. 電路圖設計 2. 電路版繪製
軟體	Google Sketchup 2017	1. 外殼設計 2. 外殼繪製
軟體	MakerBot Print v3.2	1. 外殼模型支架建立 2. 檔案轉換
軟體	Google 雲端硬碟	1. 文書處理

▲表 8 設備清單

## 附錄二 成員簡歷

姓名	俞飛揚	班級	電子108甲	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學與實習</li> <li>2. 電子學與實習</li> <li>3. 數位邏輯與實習</li> <li>4. 微處理機與實習</li> <li>5. 電子電路與實習</li> <li>6. CPLD邏輯實習</li> <li>7. Visual Basic程式設計實習</li> <li>8. Altium Designer電路圖與印刷電路板設計</li> </ol>			
參與專題工作項目	<ol style="list-style-type: none"> <li>1. 整體計畫擬定</li> <li>2. 計畫書撰寫</li> <li>3. 電路設計與配置</li> <li>4. 3D列印外殼與設計</li> <li>5. 研究與討論</li> <li>6. 資料彙整</li> <li>7. ThingSpeak數據庫建立</li> </ol>			
經歷簡介	<ol style="list-style-type: none"> <li>1. 擔任105學年度圖書幹事</li> <li>2. 擔任106學年度圖書幹事</li> <li>3. 擔任106學年度內掃衛生幹事</li> <li>4. 擔任106學年度工場場長以及材料幹事</li> <li>5. 擔任106學年度班聯會企劃長</li> <li>6. 擔任107學年度畢聯會秘書長</li> <li>7. 工業電子丙級技術士合格</li> <li>8. 20180829多益總成績700分</li> </ol>			

▲表 9 成員簡歷-俞飛揚



姓名	段凱杰	班級	電子三甲	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學及實習</li> <li>2. 電子學及實習</li> <li>3. 數位邏輯及實習</li> <li>4. 程式設計實習(VB, arduino)</li> <li>5. Altium Designer 電路圖與印刷電路板設計</li> <li>6. CPLD 邏輯實習</li> <li>7. 電子電路及實習</li> <li>8. 微處理機及實習</li> </ol>			
參與專題工作項目	<ol style="list-style-type: none"> <li>1. 撰寫計劃書</li> <li>2. 蒐集與整理資料</li> <li>3. 口頭報告</li> <li>4. 網站設計</li> <li>5. 製作簡報</li> <li>6. 研究與討論</li> </ol>			
經歷簡介	<ol style="list-style-type: none"> <li>1. 工業電子丙級技術士合格</li> </ol>			

▲表 10 成員簡歷-段凱杰

姓名	洪丞禾	班級	電子三甲	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學與實習</li> <li>2. 電子學與實習</li> <li>3. 數位邏輯與實習</li> <li>4. 微處理機與實習</li> <li>5. 電子電路與實習</li> <li>6. CPLD 邏輯實習</li> <li>7. Visual Basic 程式設計實習</li> <li>8. Altium Designer 電路圖與印刷電路板設計</li> </ol>			
參與專題工作項目	<ol style="list-style-type: none"> <li>1. 資料查詢</li> <li>2. 整體測試</li> <li>3. 計劃書製作</li> <li>4. 製作簡報</li> <li>5. 理論探討</li> <li>6. 網站架設</li> </ol>			
經歷簡介	<ol style="list-style-type: none"> <li>1. 工業電子丙級技術士合格</li> <li>2. 擔任 105 學年度輔導幹事</li> <li>3. 擔任 106 學年度總務幹事</li> <li>4. 擔任 106 學年度紫錐花社總務幹事</li> <li>5. 擔任 107 學年度工場安全衛生幹事</li> </ol>			

▲表 11 成員簡歷-洪丞禾

姓名	高瑋鴻	班級	電子三甲	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學及實習</li> <li>2. 電子學及實習</li> <li>3. 數位邏輯及實習</li> <li>4. 程式設計實習(VB, arduino)</li> <li>5. Altium Designer 電路圖與印刷電路板設計</li> <li>6. CPLD 邏輯實習</li> <li>7. 電子電路及實習</li> <li>8. 微處理機及實習</li> </ol>			
參與專題工作項目	<ol style="list-style-type: none"> <li>1. 計劃書製作</li> <li>2. 蒐集與整理資料</li> <li>3. 口頭報告</li> <li>4. 程式撰寫</li> <li>5. 研究與討論</li> </ol>			
經歷簡介	<ol style="list-style-type: none"> <li>1. 工業電子丙級技術士合格</li> <li>2. 擔任 106 學年度班長</li> <li>3. 擔任 106 學年度體育股長</li> </ol>			

▲表 12 成員簡歷-高瑋鴻

姓名	溫哲睿	班級	電子三甲	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學及實習</li> <li>2. 電子學及實習</li> <li>3. 數位邏輯及實習</li> <li>4. Visual Basic 程式設計實習</li> <li>5. Altium Designer 電路圖與印刷電路板設計</li> <li>6. CPLD 邏輯實習</li> <li>7. 電子電路及實習</li> <li>8. 微處理機及實習</li> </ol>			
參與專題工作項目	<ol style="list-style-type: none"> <li>1. 撰寫計劃書</li> <li>2. 蒐集與整理資料</li> <li>3. 口頭報告</li> <li>4. 3D 列印外殼與設計</li> <li>5. 研究與討論</li> </ol>			
經歷簡介	<p>擔任 105 學年度工場場長</p> <p>擔任 106 學年度戶外實境設總務</p>			

▲表 13 成員簡歷-溫哲睿