

修平科技大學

資訊網路技術系

專題題目：無線智慧花園

指導老師：張瑞淇老師

學生：BN101083 張竣凱

BN101081 陳園羽

BN101078 洪煒翔

BN101079 吳觀璋

BN101102 許世昕

中華民國 105 年 1 月 13 日

修平科技大學

資訊網路技術系

無線智慧花園

指導老師：張瑞淇老師

學生：BN101083 張竣凱

BN101081 陳園羽

BN101078 洪煒翔

BN101079 吳觀璋

BN101102 許世昕

指導老師：_____老師

評審老師：_____老師

_____老師

_____老師

中華民國 105 年 1 月 13 日

摘要

本專題採用 Arduino 為主要核心元件將物聯網技術應用於農業生產，設計綠能智慧農場。除了採用太陽能發電，以自給自足的能源使用觀念，完全消除精緻農業傳統上浪費能源的習性。而且透過網路服務，使農場的管理者可以透過任何行動裝置，觀察與控制農場的環境，降低農場的人力需求。台灣未來將邁入缺乏能源與人力端短缺的國家，此專題研究，證實了我們可以應用綠能與物聯網技術，降低對能源與人力的需求，使台灣的農業能在未來持續進步。

未來的研究方向可擴大物聯網技術的應用，將農場的環境資訊收集於於資料庫。再運用大數據(Big Data)的方法，分析環境與農作物生長的關係。如此將可以更精確分析與控制植物的生長，對提升農產品的生產效率與提高農民與農業團體的收益將會有很大的助益。

目 錄

第 1 章	ARDUINO 簡介	1
1.1	ARDUINO 介紹	1
1.2	ARDUINO 的特色	1
1.3	ARDUINO 具備哪些東西	2
1.4	ARDUINO 種類	2
1.5	ARDUINO 擴充板 (SHIELDS)	3
1.6	ARDUINO 硬體介紹	4
第 2 章	ARDUINO 開發環境安裝、設定與使用	7
2.1	ARDUINO 安裝	7
2.2	開發環境	12
2.3	編譯環境	13
2.4	ARDUINO 開發環境的介面	14
第 3 章	智慧溫室的建置與相關零件	16
3.1	溫室結構	16
3.2	ARDUINO WIFI 卡	23
3.3	溫濕度感測器	25
3.4	光亮度感測器 (光敏電阻)	30
3.5	繼電器	31
3.6	風扇	32
3.7	LED 燈	36

第 4 章	智慧農場的程式開發	38
第 5 章	實際測試過程與結果	46
第 6 章	心得與討論	48
參考資料	50

圖 目 錄

圖 1-1 Arduino 硬體外觀	4
圖 2-1 搜尋 Arduino 網站	7
圖 2-2 點擊網頁	7
圖 2-3 下載專區	8
圖 2-4 選擇系統軟體	8
圖 2-5 找尋版本	9
圖 2-6 安裝流程 1	9
圖 2-7 安裝流程 2	10
圖 2-8 安裝流程 3	10
圖 2-9 安裝畫面	11
圖 2-10 安裝完成	11
圖 2-11 Arduino 介面	14
圖 2-12 函式 setup loop	15
圖 3-1 模型散熱孔與風扇裝置處	16
圖 3-2 模型正面與模型上蓋	17
圖 3-3 繼電器樣式	17
圖 3-4 繼電器接風扇與 LED 燈條線路圖	17
圖 3-5 Arduino WiFi 網路卡正面與背面	18
圖 3-6 網路卡連接 Arduino 控制面板	18
圖 3-7 溫濕度感測器(正面)與電路銜接	18
圖 3-8 架設於溫室屋頂下方	19
圖 3-9 風扇架設於溫室側面	20
圖 3-10 LED 燈條架設於溫室正上方	21
圖 3-11 線路牽線	22
圖 3-12 網路卡正面接腳與背面插腳	24
圖 3-13 WIFI 卡與 Arduino 卡連接	24
圖 3-14 溫濕度感測器 Dallas DS18B20 與 DHT11	25

圖 3-15 溫濕度感測器與單晶片開發板連接線路簡圖	27
圖 3-16 溫溼度模組	29
圖 3-17 風扇.....	32
圖 4-1 引用程式庫	38
圖 4-2 宣告物件、控制接腳、WIFI 位址	39
圖 4-3 網頁初始頁面與介紹.....	39
圖 4-4 網頁底部控制項設計.....	40
圖 4-5 初始網頁讀取資訊 零件開關頁面設計	40
圖 4-6 「關於」網頁設計	41
圖 4-7 風扇開關指令頁.....	41
圖 4-8 風扇開關程式設計頁.....	42
圖 4-9 LED 開關指令頁	43
圖 4-10 LED 開關程式設計頁面	43
圖 4-11 處理頁面程式請求 WIFI 連線設定	44
圖 4-12 顯示 WIFI 連線目前狀態	45
圖 5-1 智慧農場網頁	46
圖 5-2 綠能智慧農場控制 LED 網頁(與控制風扇的網頁畫面類似) .	47

表 目 錄

表 1-1 Arduino 規格.....	4
表 1-2 Arduino 數位特殊接腳	5
表 3-1 DHT11 規格	26
表 3-2 DHT11 溫濕度感測模組接腳圖	29
表 3-3 產品型號.....	32

第1章 Arduino 簡介

1.1 Arduino 介紹

Arduino 是一種開放授權的互動環境開發技術，互動裝置其實無所不在，像是冷氣的恆溫裝置，便是使用感測器偵測環境溫度，進行室內溫度的自動調節；還有汽車使用的倒車雷達，過於靠近物體便會發出聲音警告駕駛者。這些裝置為生活增加不少安全及便利，互動能帶給使用者驚喜，有時會以藝術品的方式呈現在生活當中。

以往要處理相關的電子設備時，需要透過工程師，逐一由單一小元件拼湊出整個電路。大多數的設計工具都是為了工程師設計，除了電路外還需要廣泛的知識，才有辦法完成電路。還好微處理器有了長足的進步，除了在使用上變得更為容易，價格上的減少更降低了學習的門檻。

1.2 Arduino 的特色

開放源碼(open-source)。不僅軟體是開放源碼，硬體也是開放的。軟體的開發環境可在網上免費下載，而 Arduino 的電路設計圖也可從官方網站自行下載，依據自身之需求進行修，但須要符合創用 CC 授權條款(創用 CC 授權條款)。

開發簡單，參考資料多。在以往的硬體環境中，要開發微控制器的

程式，開發者需要具備電子、電機及相關科系的背景，一般人需花費大量時間能有機會進入這個開發環境中。Arduino 學習門檻較為簡單，不需要電子電機相關科系的背景，也可以很容易學會 Arduino 相關互動裝置的開發。由於 Arduino 以公開共享為基礎，多數人都樂於分享自己的的創品，網路上能找的創作案子非常豐富。以此會基礎，有時只需要參考分享者的作品，依據自身的需求行調整，就可以在短時間內完成自己的創作。

1.3 Arduino 具備哪些東西

主要可區分為三大類，分別是 Arduino 硬體、Arduino 軟體及 Arduino 擴充原件。

硬體部分可購買現成的版子或是依據網路下載的電路途自行組裝，出學者建議購買現成的，根據 Arduino 官方網站所寫，台灣地區具有官方合法授權的代理商為藝科資訊 Aroboto Studio、PlayRobot 飆機器人/普特企業有限公司、iCshop (台灣)電子零件，電子材料，當然，還有別的商家，到網路上搜尋一下可找到不少，拍賣網站上也有，甚至也可以到國外網站去買。目前官方網站提供了許多不同的版子供開發者購買。

1.4 Arduino 種類

Arduino 種類繁多，各有各的應用，主要包括：Arduino Uno 、Arduino Mega、Arduino Duemilanove、Arduino Fio、Arduino Nano 3.0、

LilyPad Arduino 02 、Arduino Mini 、Arduino BT (BlueTooth) 。

1.5 Arduino 擴充板 (Shields)

Ethernet Shield -提供 Ethernet 上網能力

MP3 Shield -提供 MP3 音樂解碼能力

Input Shield -提供 Joystick 搖桿、按鍵及震動馬達能力

TouchShield -提供 OLED 觸控螢幕顯示能力

XBee Shield -提供 ZigBee 無線通訊能力

GPS Shield -提供 GPS 定位能力

WiShield -提供 Wi-Fi 上網能力

microSD Shield -提供 microSD 儲存媒體能力

AeroQuad Shield -提供三軸陀螺儀與加速計的控制

Smart Energy Groups SEGMeter -提供家庭與工業用電量測能力

1.6 Arduino 硬體介紹

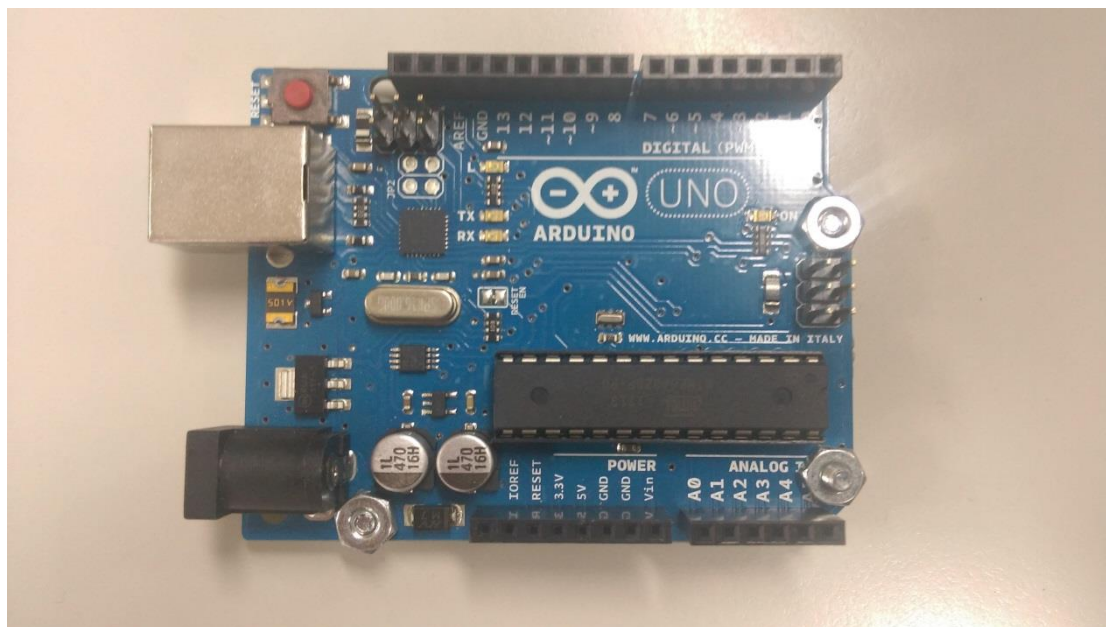


圖 1-1 Arduino 硬體外觀

微控制器	ATMEGA328
工作電壓	5V
輸入電壓（推薦）	7-12V
輸入電壓（限制）	6-20V
數字 I/O 接腳	14（6 個提供 PWM 輸出）
模擬輸入接腳	6 支
EEPROM	1 KB（ATMEGA328）
震盪速度	16 MHz

表 1-1 Arduino 規格

硬體功能說明：

有 14 支數位 I/O 接腳 可以當作 input 使用，也可以當作 output 使用，使用方法是透過 `pinMode()`, `digitalWrite()`, `anddigitalRead()` 這幾個函式。

14 支數位 I/O 接腳，其中幾支腳有特殊的功能：

Serial 通訊	0(RX) 和 1 (TX) 這兩支腳。用來接收(RX)與傳輸(TX) TTL 訊號的序列資料。這兩支腳也連接到 USB Converter 晶片中。
外部中斷	2 和 3 這兩支腳。這兩支腳可以利用外部事件觸發中斷。詳細內容請參考 <code>attachInterrupt()</code> 函式。
PWM	3, 5, 6, 9, 10 和 11 共六支腳。透過 <code>analogWrite()</code> 函式可以提供 8-bit 的 PWM 輸出。
SPI	10 (SS), 11 (MOSI), 12 (MISO) 和 13 (SCK) 這四支腳。這四支腳搭配 SPI Library 可提供 SPI 序列通訊。
LED	13。內建一顆 LED，當 pin 腳為 HIGH 時，LED 打開，當 pin 腳為 LOW 時，LED 關閉。

表 1-2 Arduino 數位特殊接腳

類比輸入 接腳:

Arduino Uno 有 6 支類比輸入腳，標記為 A0 到 A5，每支腳都可提供 10 位元的解析 (即 1024 種不同的數值)。這些腳位所用的參考電壓預設為 0 到 5V，不過參考電壓也是可以更改的，方法是透過 AREF 腳和 `analogReference()`

I2C	4 (SDA) 和 5 (SCL) 這兩支腳。透過 Wire library 可以提供 I2C 通訊。
-----	---

其它:

AREF	類比輸入的參考電壓，搭配 <code>analogReference()</code> 函式一起使用。
Reset	當 Reset 腳為 LOW 時，微控制器會重置。

第2章 Arduino 開發環境安裝、設定與使用

2.1 Arduino 安裝

1. 在搜尋輸入 Arduino



圖 2-1 搜尋 Arduino 網站

2. 點入 Arduino-HomePage



圖 2-2 點擊網頁

3. 進入下載專區(Download)



圖 2-3 下載專區

4. Windows 下載(選擇自己的電腦系統的軟體)

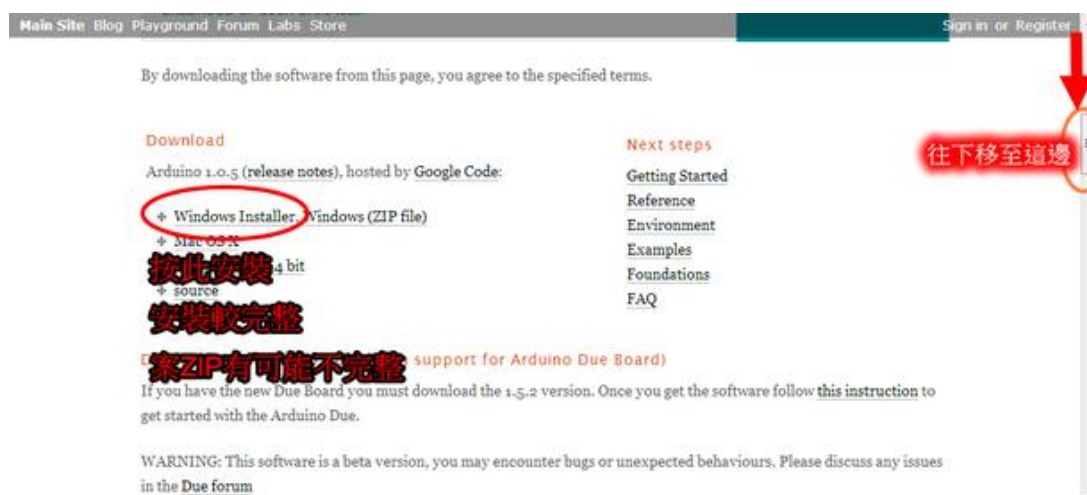


圖 2-4 選擇系統軟體

5. 下載專區

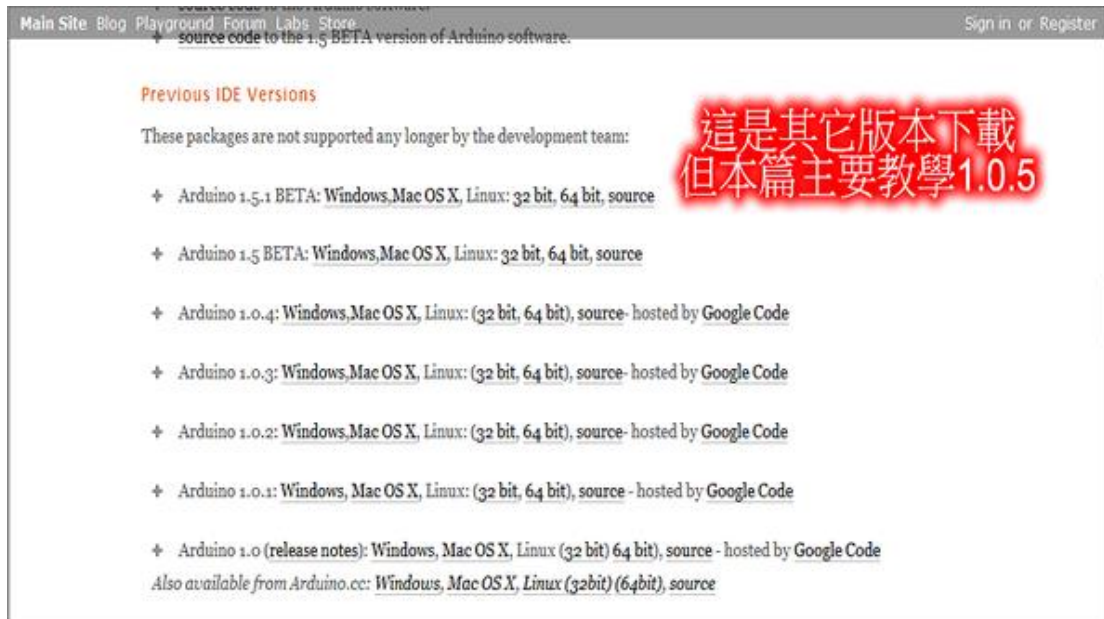


圖 2-5 找尋版本

6. 下載流程介紹

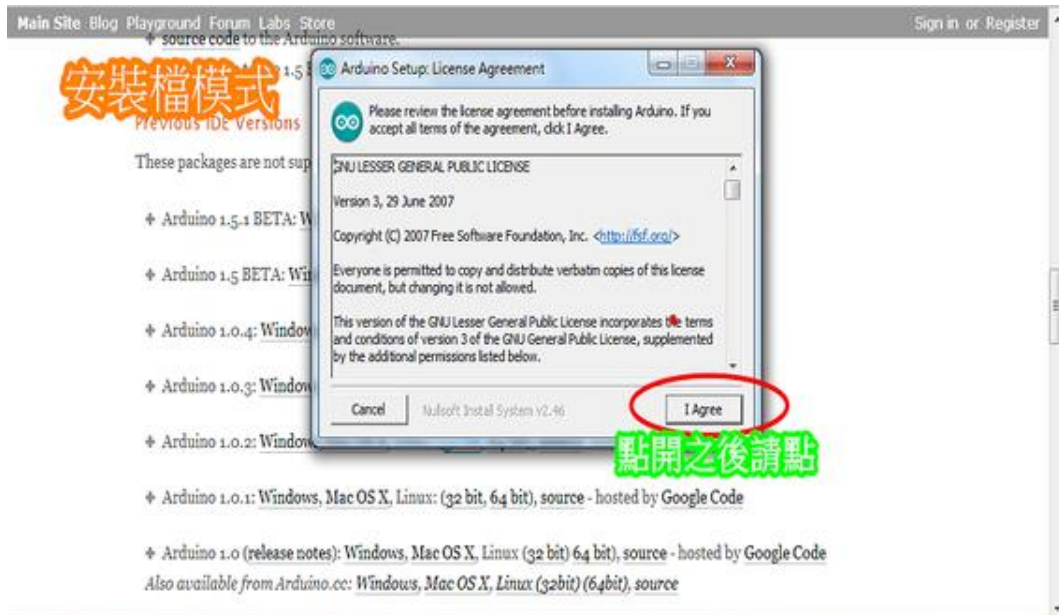


圖 2-6 安裝流程 1

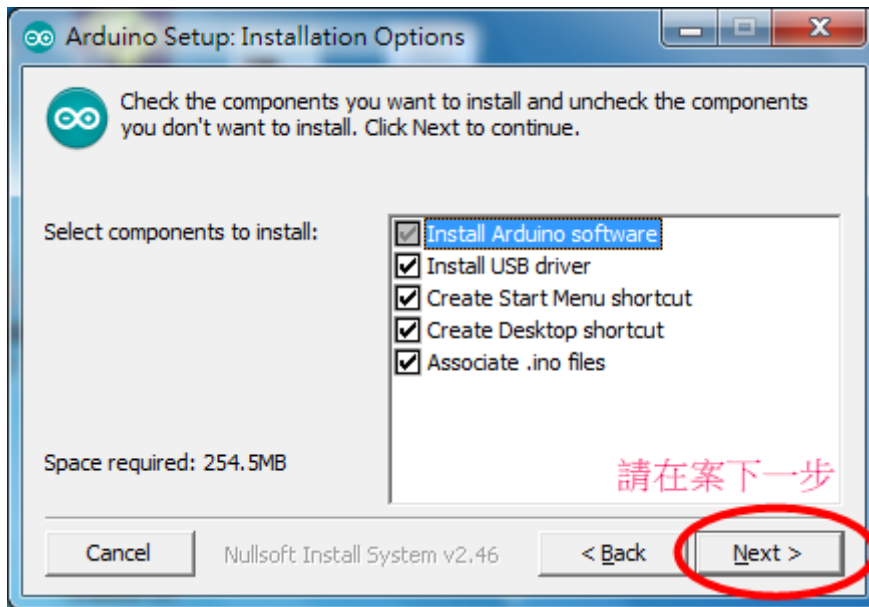


圖 2-7 安裝流程 2

7. 儲存於你想要的位置

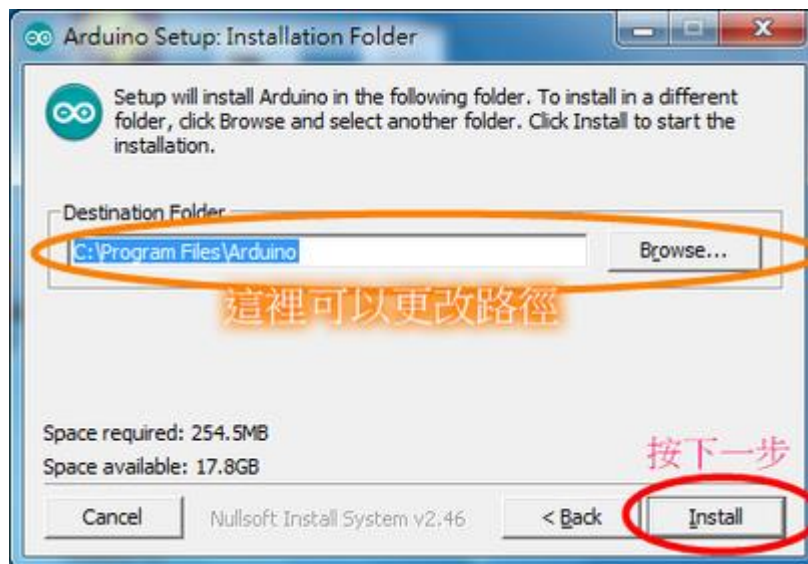


圖 2-8 安裝流程 3

8. 安裝畫面

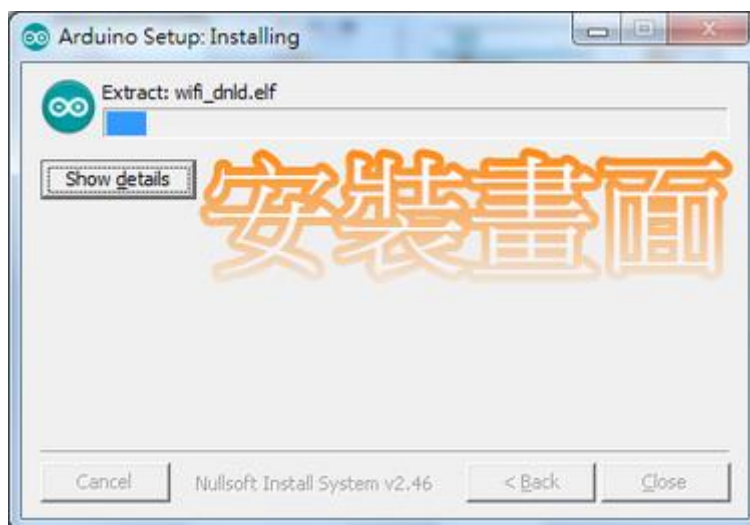


圖 2-9 安裝畫面

9. 安裝完成

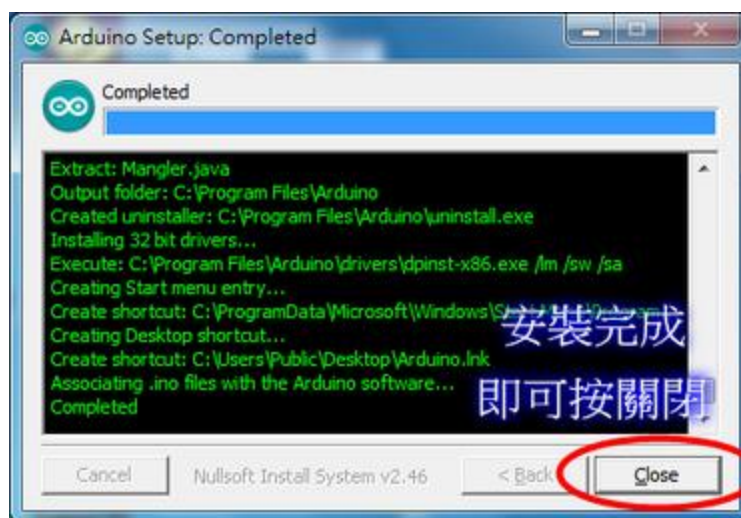


圖 2-10 安裝完成

2.2 開發環境

Arduino 的軟體開發環境是開放源碼，可以在 Arduino 官網免費下載，它所用的程式語言語法類似於 C/C++，具備文字編輯介面、常用工具欄、圖形化控制介面及錯誤編輯器，運用這個工具進行城市的上傳，及各項感應器的溝通。

Arduino 軟體開發環境，裡頭包含了很多部分：

- 1.USB 轉換序列埠的驅動程式
- 2.範例程式碼（閃爍 LED、讀取數位腳位、PWM、等等）
- 3.Arduino 的核心（定義 HIGH 與 LOW 常數值、函式 pinMode()、等等）
- 4.內建程式庫（讀寫 EEPROM、控制 LCD、控制伺服馬達、步進馬達、等等）
- 5.開發工具鏈（編譯器、連結器、燒錄程式等等）
- 6.bootloader（要放進板子裡的快閃記憶體）
- 7.文件資料
- 8.整合式開發環境（以 Java 語言撰寫）
- 9.等等其他東西

2.3 編譯環境

使用的 Arduino 編寫的軟件被稱為 Sketch(腳本)。這些的腳本都寫在文本編輯器。腳本名稱就是檔案名稱。上排圖形提供了新增、上傳、除錯及腳板切換介面。而 console 介面紀錄了完整的執行訊息，更可藉由此介面監視 Arduino I/O 數值。

2.4 Arduino 開發環境的介面

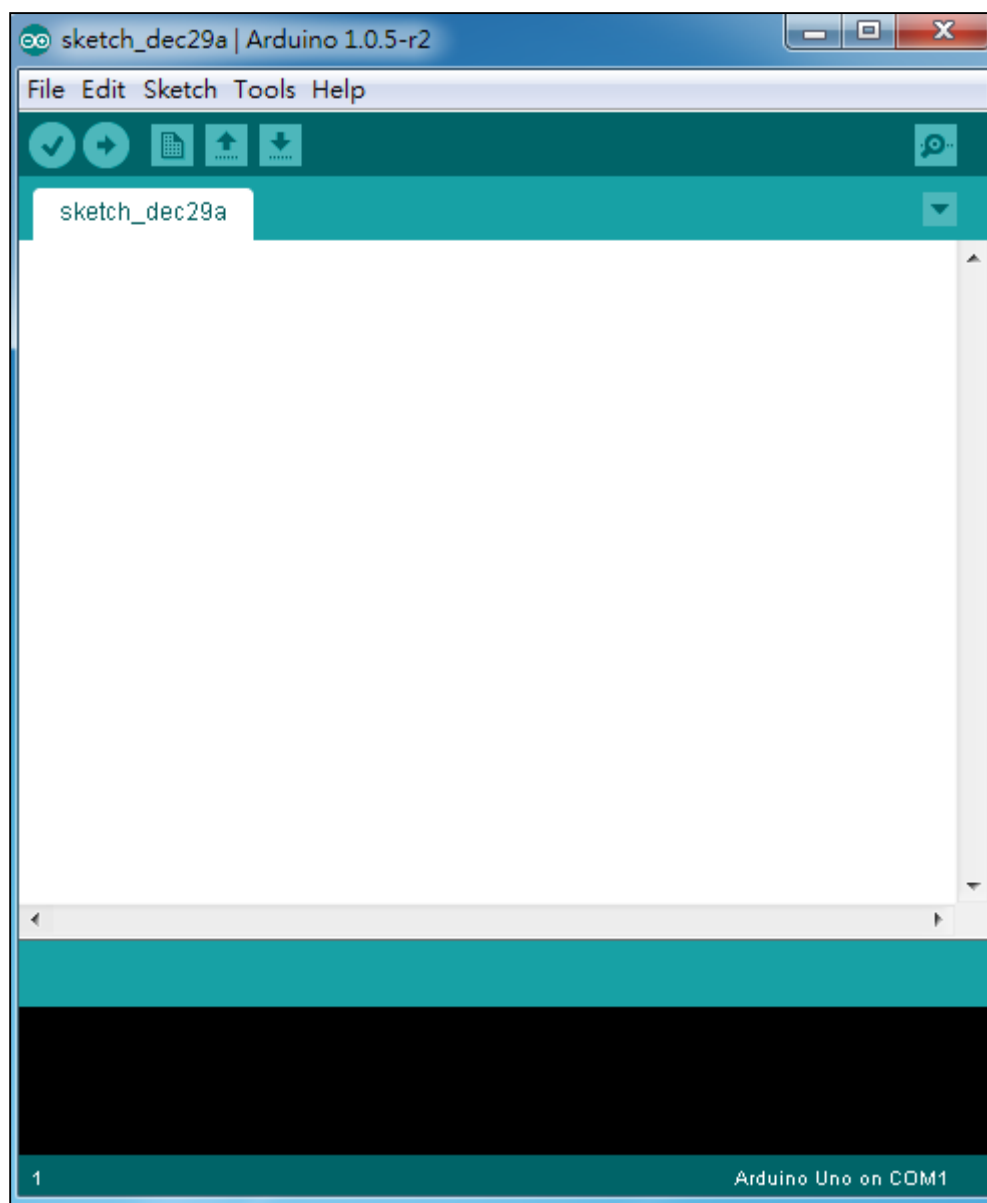


圖 2-11Arduino 介面

- ✓ Verity 檢查參數設定或引入程式是否產生錯誤。
- ➔ Upload 程式進行編譯，將程式碼透過 USB 介面燒錄至 Arduino 控

制板。



New 產生新的腳本



Open 開啟腳本，顯示在同一頁面上。點擊不同腳本便顯示不同腳

本。



Save 腳本儲存。



Serial Monitor 開啟監視器頁面，監視 Arduino I/O 介面。

Arduino 程式主要由 `setup()` 和 `loop()` 這兩個函式組成：

```
void setup(){  
    //setup 初始環境設定  
}  
void loop(){  
    //loop 程式執行主體  
}
```

圖 2-12 函式 setup loop

因為 `setup()`和 `loop()`是每支 Arduino 程式都會用到的兩個函式，為了方便，Arduino 已經幫大家準備好程式基本架構了，編寫 Arduino 程式時，可直接點選 `File > Examples > 1.Basics > Digital ReadSerial` 這個範本檔，另存新檔後，然後再修改 `setup()`和 `loop()`兩個函式的內容即可。

第3章 智慧溫室的建置與相關零件

3.1 溫室結構

溫室的結構如下圖，有通風孔。並裝有風扇與 LED 燈。Arduino 透過電器控制風扇與 LED 燈。溫濕度感測器與光敏電阻也裝在溫室內。相關裝置的照面在此節中展示，各零件的說明將分數於下面各節。

溫室外部本體



圖 3-1 模型散熱孔與風扇裝置處



圖 3-2 模型正面與模型上蓋



圖 3-3 繼電器樣式

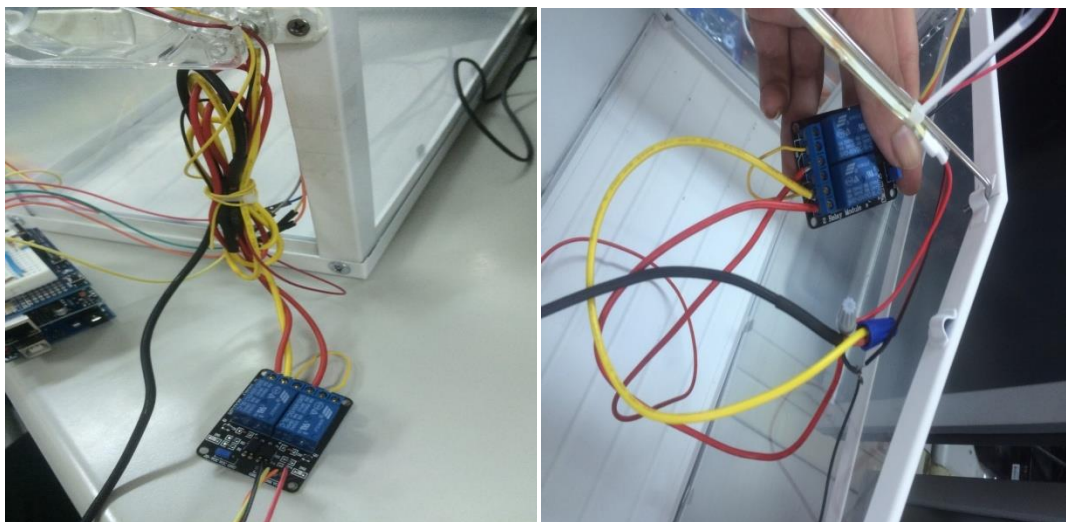


圖 3-4 繼電器接風扇與 LED 燈條線路圖



圖 3-5 Arduino WiFi 網路卡正面與背面



圖 3-6 網路卡連接 Arduino 控制面板

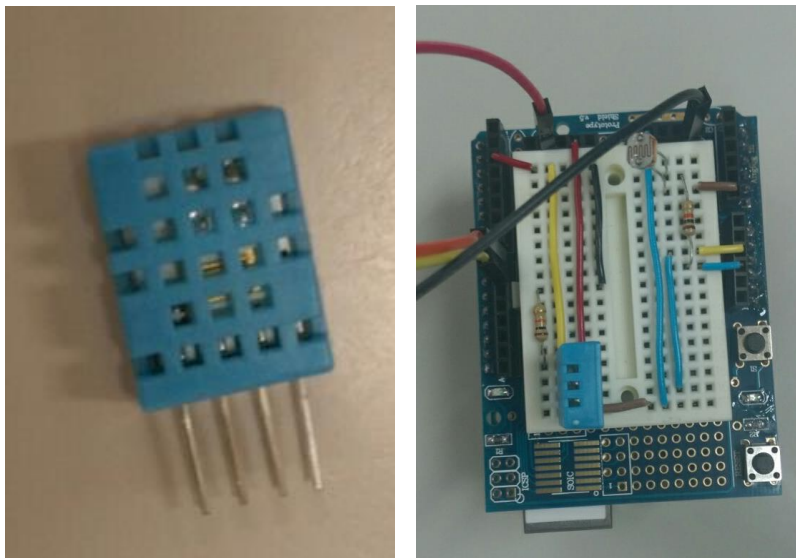


圖 3-7 溫濕度感測器(正面)與電路銜接

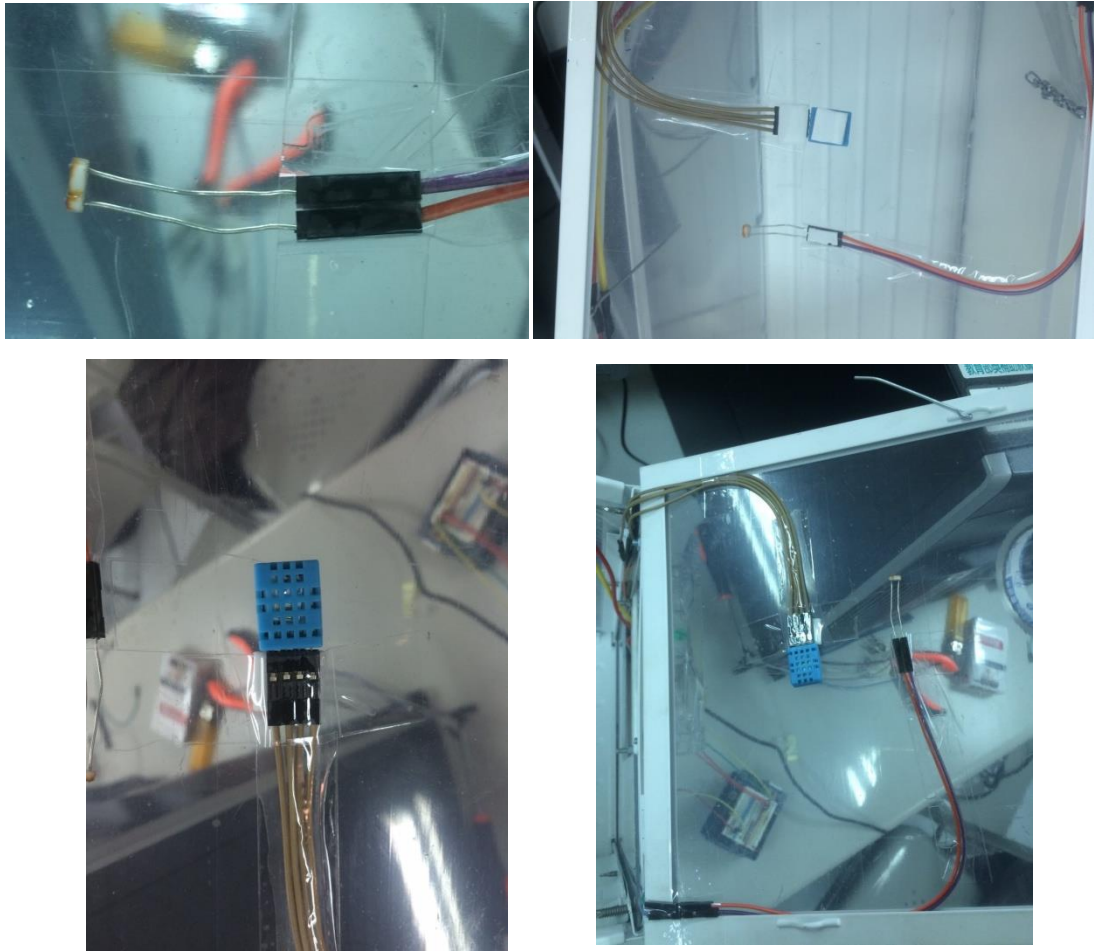


圖 3-8 架設於溫室屋頂下方

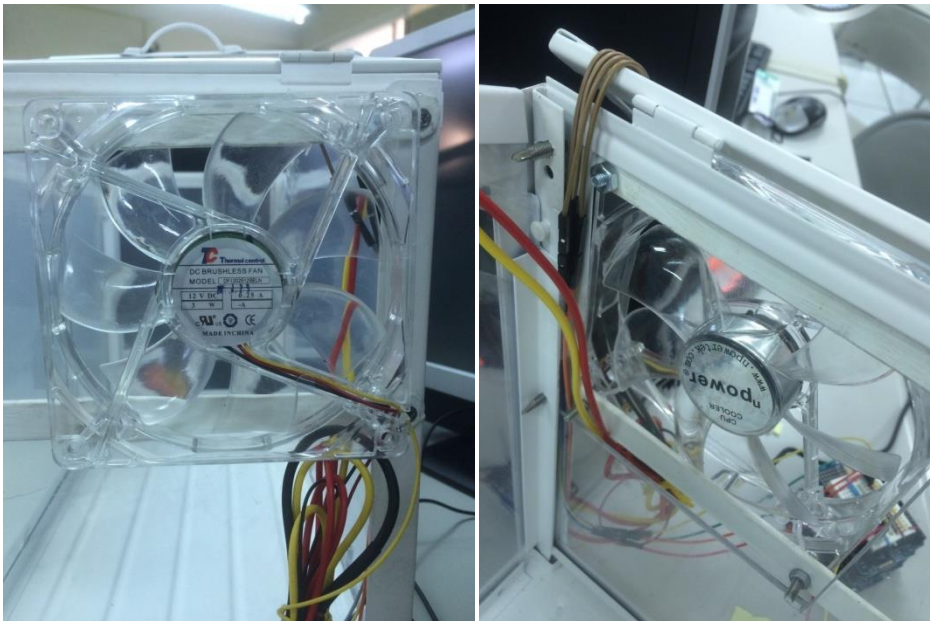


圖 3-9 風扇架設於溫室側面

LED 燈條

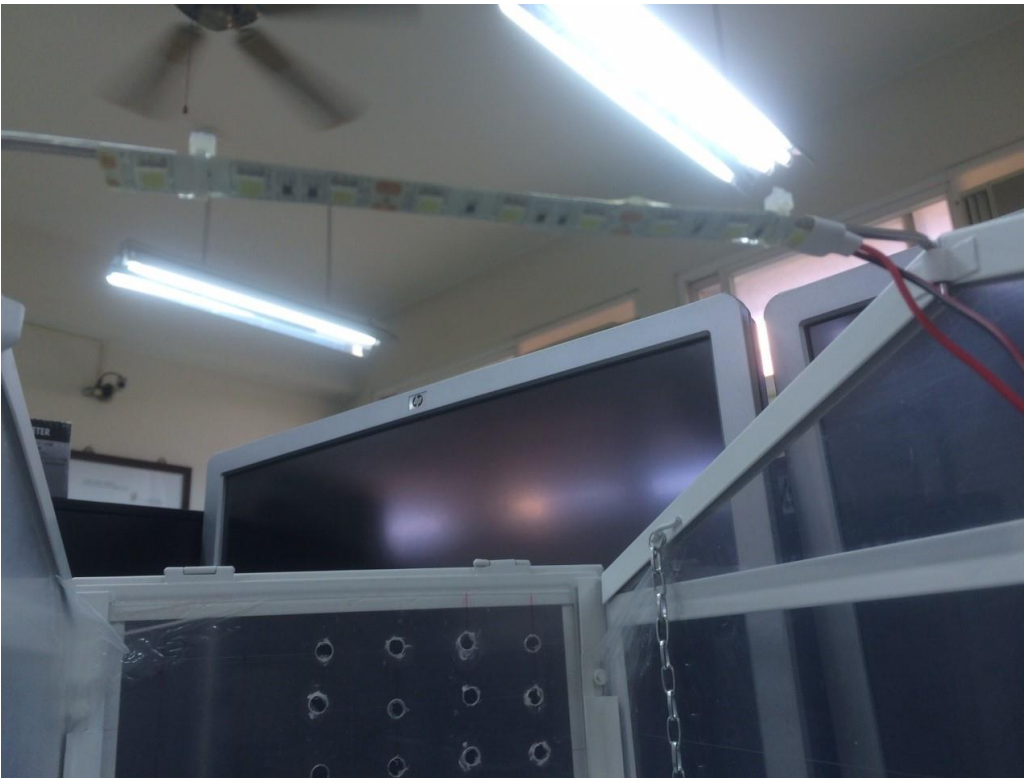


圖 3-10 LED 燈條架設於溫室正上方

線路架設

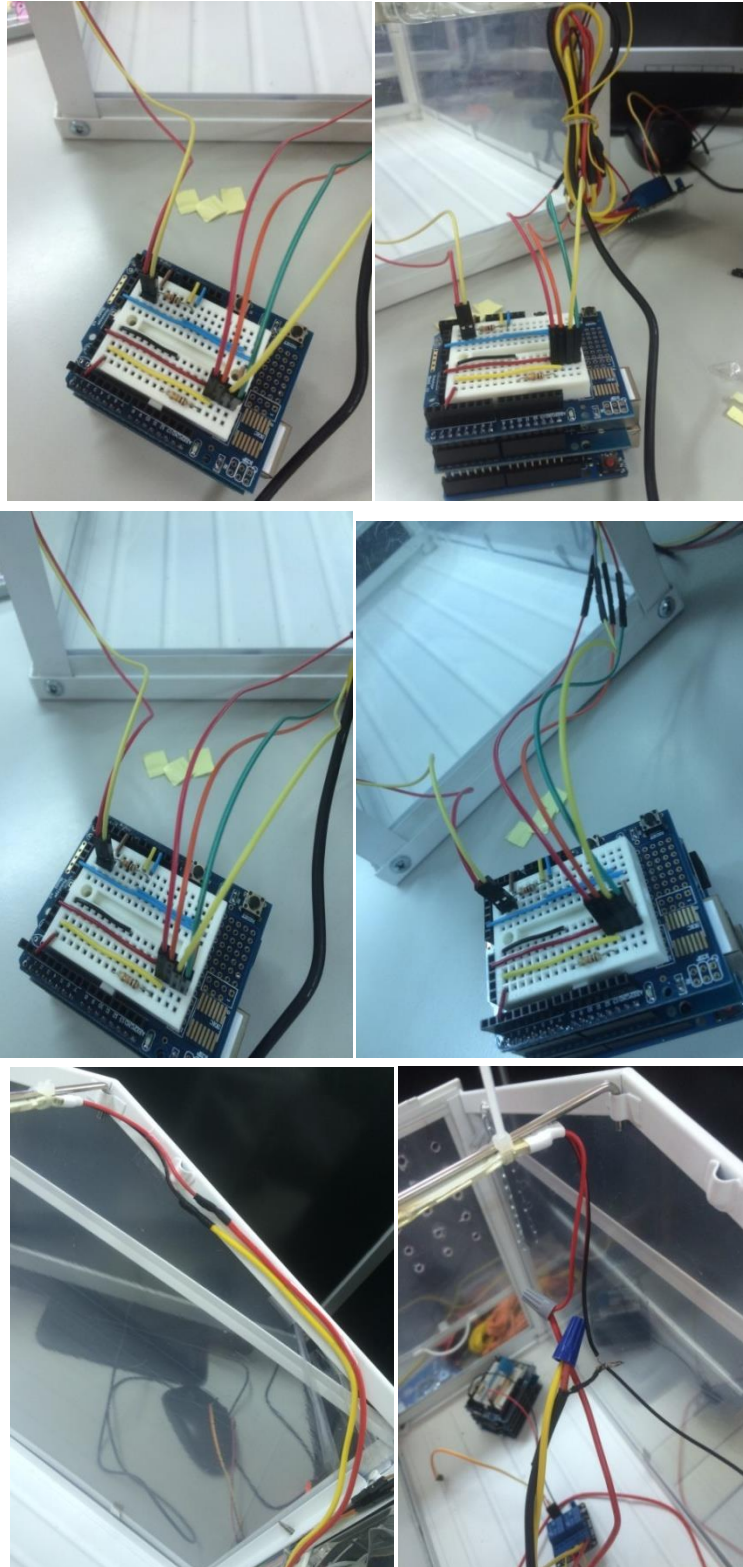


圖 3-11 線路牽線

3.2 Arduino WIFI 卡

Arduino WIFI 卡

WiFi Shield for Arduino 帶 SD 卡座 支援 UNO 和 MEGA2560

無線網路處理器：

◎IEEE 802.11 b/g

◎嵌入式 IPv4 TCP/IP

業界最佳的無線電性能：

◎Tx 功率：11Mbps 時為 +18dBm，CCK

◎Rx 靈敏度：-86dBm，8% 包差錯率 (PER)，11Mbps

與採用緊湊存儲器封裝的低每秒處理百萬條指令 (MIPS)，低成本微控制器一同工作：

◎2K 位元組閃存

◎250 位元組 RAM

經美國聯邦通信委員會 (FCC)，加拿大工業部 (IC) 和歐洲統一 (CE) 認證的晶片天線，可從德州儀器 (TI) 獲得硬件設計文件和設計指南，整合晶體和電源管理

小尺寸封裝模組

16.3mm x 13.5mm x 2mm

工作溫度範圍：

-20°C 至 70°C

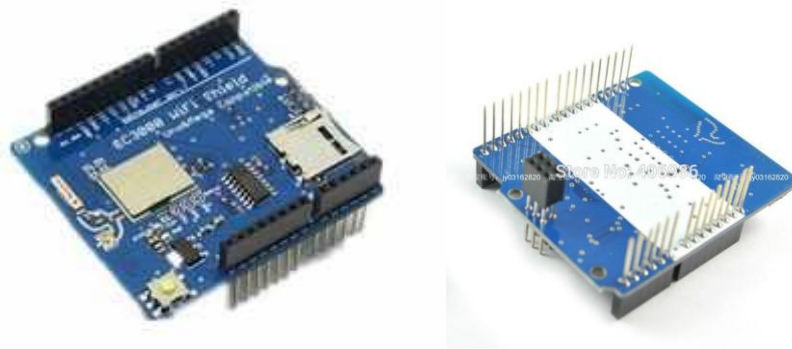


圖 3-12 網路卡正面接腳與背面插腳

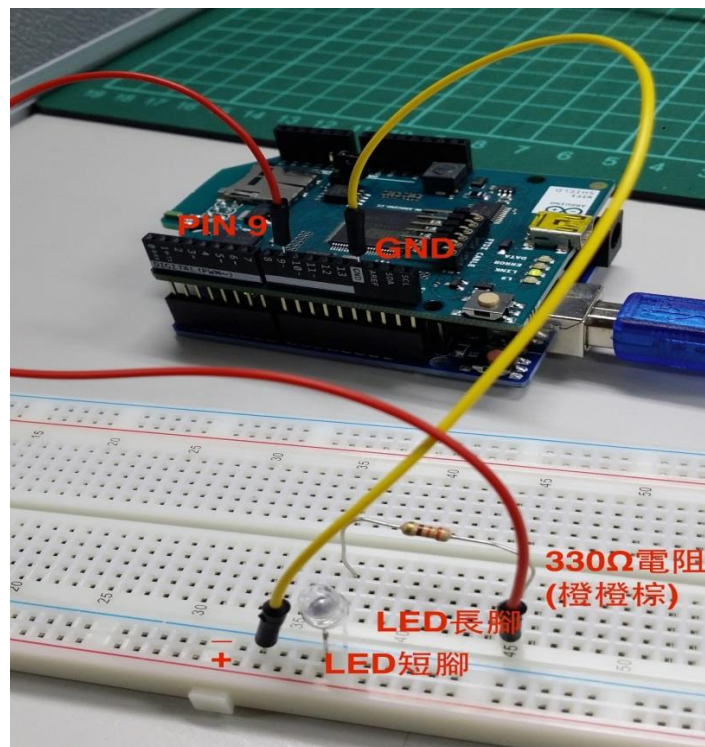


圖 3-13 WIFI 卡與 Arduino 卡連接

3.3 溫濕度感測器

目前市面上的感測器很多種，例如圖示的 Dallas DS18B20 溫度感測器和 DHT11 溫濕度感測器等等...，種類非常多種，但是我們並非使用在數百度的高溫場所，我們大部份皆使用在一般市溫場所的溫度感測器。

我們如果要可以兼備溫度與濕度量測的功能，那麼就必須採用 DHT11 溫濕度感測器，因為 DHT11 溫濕度感測器本身就具備溫度與溼度兩種感測器的功能。

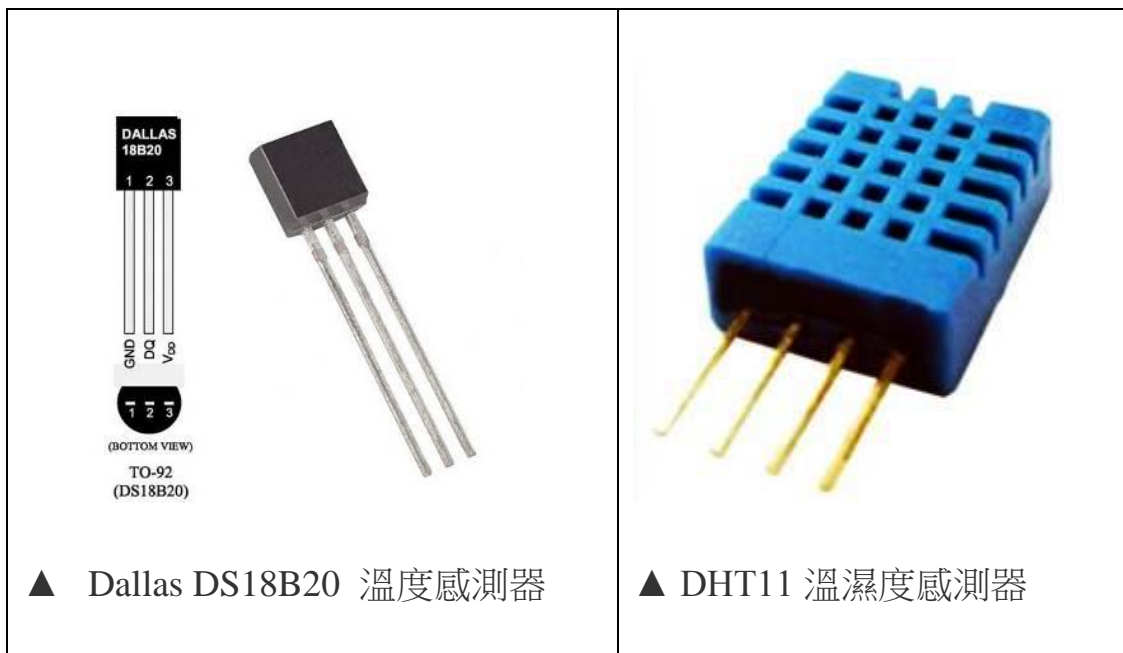


圖 3-14 溫濕度感測器 Dallas DS18B20 與 DHT11

首先各位會看到 DHT11 有 4 根接腳，但其實只有 1、2、4 有用，第 3 支腳沒有任何用處，一般在 Datasheet 上會寫 NC (No Connection 的意思)。

1 號接腳接 Vdd (最低 3V，最高 5V，所以從 Arduino 這邊供電的話，給予 3.3V 或 5V 均可)，4 號接腳接地，簡單說就是供應 DHT11 基本可運作的電力。

DHT-11 是一個結合濕度計和測溫元件量測週遭空氣環境，並與一個高性能 8 位元單晶片相連接，將所量測到的溫、濕度資料拆解成為數位訊號，再由 data pin 腳將資料送出。使用上很簡單，但是抓取資料時必須要特別注意時間的掌控，而且每筆資料的抓取時間間隔要 1~2 秒鐘，不能太快。與 DHT-22 比較，DHT-11 較不精密與準確且溫濕度量測範圍不大，雖然如此，但對於學習與熟悉溫、濕度感測卻是綽綽有餘，傳輸的距離又可長達 20m 以上，而且比 DHT-22 較小且便宜許多，是十分方便的簡易測試元件。

濕度測量範圍	20~90%RH
濕度測量精度	±5%RH
溫度測量範圍	0~50°C
溫度測量精度	±2°C
電源供應範圍	3~5V
頻率不可超過	0.5Hz (每 2 秒一次)

表 3-1 DHT11 規格

DHT11 是用電阻方式感應濕度，並用 NTC 方式感應溫度，DHT 內本來就有一顆 8 位元 MCU，MCU 的工作就是把感測到的溫度、濕度，以數位信號方式，透過 2 號腳位傳遞出去，所以其實那顆 MCU 也內建類比轉數位功效(ADC)，而溫濕度感測後的相關校準與係數，其實就燒錄在 MCU 的程式內，不過也無法更改了，因為是 OTP(One Time Program)，燒寫一次就不能再更動程式內容。

由於輸出已經是數位型式，所以 2 號接腳與 Arduino 本體，即便相隔一段距離仍可正常傳遞，傳遞過程中仍有一定程度的抗雜訊能力，一般而言能有 20 公尺以上長度。

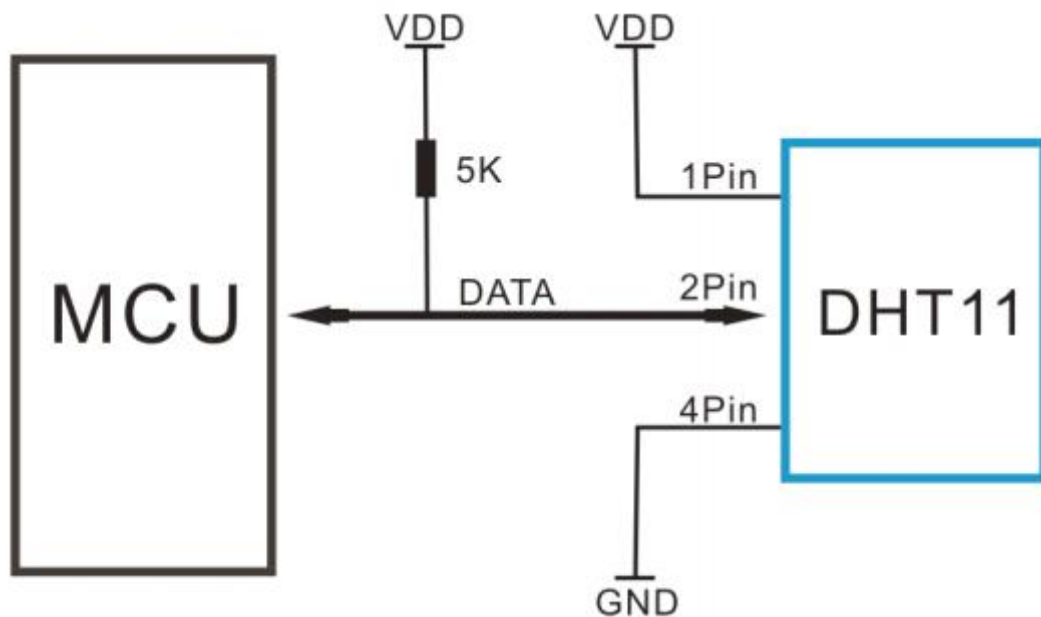


圖 3-15 溫濕度感測器與單晶片開發板連接線路簡圖

值得注意的是，DHT11 的 Datasheet 提到，若傳輸距離太短，最好在 2 號的信號輸出線路上，加上一個 5k 歐姆的外部提升電阻，由於實務上不易找到剛好 5k 歐姆的，也可以找電阻值稍大或稍小的，例如 4.7k 歐姆。

如果長度超過 20 公尺，也建議找一個合適的提升電阻，但阻值需要再行量測，依據不同的長度，以及到達 Arduino 時的邏輯電壓準位，從而選擇合適的阻值。

另外，Datasheet 也建議在 Vdd 與接地線路間接一個 100nF 的電容，讓電源可以潔淨一點，避免雜訊影響。而剛接上電源時，要等待約 1 秒，讓 DHT11 完成各項初始、校準，才能開始使用。

當然，DHT11 也涉及它的時序 (Timing)、信號準位與速度要求、協定格式等，但通常要到更細膩的開發與除錯時才會更深入探究。

進一步的，其他有些東西也會影響 DHT11，例如長時間的強烈陽光或紫外線、化學材料產生的蒸汽等，會影響 DHT11 的運作效率，而且筆者也建議傳遞信號的導線 (特別指 2 號線路) 最好能有良好的包覆，以確保信號品質，在野外使用也要避開露水。

如果要拿 DHT11 來正式製作產品，則需要進行焊接 (不再是麵包板的實習)，建議焊槍溫度不要高過攝氏 260 度，也不用接觸超過 10 秒鐘，以免焊槍的高溫把 DHT11 燒壞。

最後，其實 DHT11 是一個比較初階、平價的感測器，其實在同公司的產品中，還有更好的 DHT22 可選，若期望解析度、精準度高一點，或者是溫、濕度偵測範圍能大一點，可以考慮 DHT22。

DHT11 溫濕度感測模組

為了讓 Arduino 開發板簡化使用，我們使用了 DHT11 溫濕度感測模組，本模組只要將 Vcc 接到 Arduino 開發板 +5V 腳位，Gnd 接到 Arduino 開發板 Gnd 腳位，DataVcc 接到 Arduino 開發板 Digital Input 腳位 2，接腳圖可參考表 4 之 DHT11 溫濕度感測模組接腳圖。

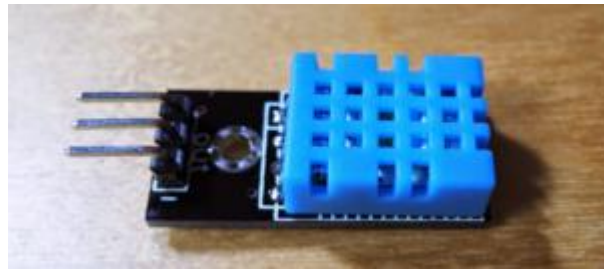


圖 3-16 溫溼度模組

DHT11 溫濕度感測模組	Arduino 開發板接腳	解說
Data	Arduino digital Input pin 2	DHT11 資料輸出 腳位
5V	Arduino pin 5V	5V 陽極接點
GND	Arduino pin Gnd	共地接點

表 3-2 DHT11 溫濕度感測模組接腳圖

3.4 光亮度感測器（光敏電阻）

- 一、設計照明設備之電路時，可利用光敏電阻對光之特性，動態偵測光度與照時 是否足夠，決定是否打開輔助照明。
- 二、光敏電阻為測量照度之元件，為測量正確之照度，所以光敏電阻器的擺設應 適當，以避免感測失真。
- 三、光敏電阻器是一種感光式的電阻器，當光敏電阻器上方的電源被遮住時，電 阻器值就會上升，反之則下降，可以利用此項特性，設計日常生活之光感測 器，以達到節能減碳之目的。
- 四、亮度愈亮，電阻值愈小，亮度愈暗，電阻值愈大。
- 五、光敏電阻量電流與照度大小、元件端電壓大小皆呈正比。
- 六、光敏電阻電流增加時，不能超過元件的容許損失功率。

3.5 繼電器

繼電器的選用

先了解必要的條件

- 1.控制電路的電源電壓，能提供的最大電流
- 2.被控制電路中的電壓和電流
- 3.被控電路需要幾組、什麼形式的觸點
- 4.選用繼電器時，一般控制電路的電源電壓可作為選用的依據。
- 5.控制電路應能給繼電器提供足夠的工作電流，否則繼電器吸合是不穩定的。

型號和規格號

查閱有關資料確定使用條件後，可查找相關資料，找出需要的繼電器的型號和規格號。若手頭已有繼電器，可依據資料核對是否可以利用。最後考慮尺寸是否合適。

注意器具的容積

若是用於一般用電器，除考慮機箱容積外，小型繼電器主要考慮電路板安裝布局。對於小型電器，如玩具、遙控裝置則應選用超小型繼電器產品。

3.6 風扇



圖 3-17 風扇

產品編號	產品名稱	電源型式
FA-22	12CM*12CM 風扇	大 4P

表 3-3 產品型號

產品介紹

特性:

1. 高效.超靜風扇
2. 容易安裝
3. 強大風力快速冷卻機箱內工作溫度，增強系統穩定性

規格:

- 1.壽命:20000 小時
- 2.額定電壓:12V DC
- 3.轉速:2500 R.P.M
- 4.風量:28.2 CFM
- 5.重量:65g
- 6.軸承型式:油封
- 7.接頭型式:大 4P
- 8.厚度:2.5CM

風扇主要性能

風扇的主要性能從以下幾個方面體現：轉速、扇葉形狀、扇葉角度和軸承系統。一般情況下，在散熱器的說明書上都標明風扇的轉速。一般來說散熱器的散熱效果有 30% 要取決於風扇的轉速。但風扇並不是轉速越高越好。正確的風扇轉速應該根據 CPU 的發熱量決定，不同規格的風機轉速選擇都應該有所區分，基本的原則就是：在產生同等風量的前提下，風機越大轉速就應該越低，噪音同樣也會較小，一般在 3500 轉至 5200 轉之間的轉速是比較合乎常規的。

功率越大，風扇風力越強勁，散熱效果也就越好。而風扇的功率與風扇的轉速又是有直接聯繫的，也就是說風扇的轉速越高，風扇也就越強勁有力。目前一般電腦市場上出售的風扇都是直流 12V 的，功率則從 0.X W 到 2.X W 不等，那麼功率的大小就需要根據您的 CPU 發熱

量來選擇了。特別提醒大家的是，不能片面的強調大功率，只需要與 CPU 本身的功率要相匹配就可以了，如果功率過大，不單冷卻效果沒有多大增強，反而可能會加重計算機的工作負荷，最終縮短 CPU 和風扇的壽命。因此，在選擇 CPU 功率大小的同時，應該量「熱」而行。較好的散熱器，邊緣不會出現毛刺現象，散熱鰭片無變形、底部光滑無裂痕、外觀整潔、造型大方、有質保、有防偽標誌或者其他特徵的商標等。另外，大家在選擇散熱器時還要注意的是散熱器的底部不能太厚，因為鋁的導熱性不太好，太厚了會影響熱量的傳遞；另外散熱器表面的導流槽應密一些，這樣可以確保散熱器能與空氣有較大的接觸面積，從而增強散熱效果。

風扇基本介紹

(1.)我們到底該如何才能為電腦防暑降溫呢？一般來說，我們可以採用風冷、水冷、半導體製冷和氟（氮）製冷等多種方法來降溫，但由於這些方法實現成本比較高，而且還可能對電腦的安全構成威脅，因此這幾種方式在國內並不是十分流行。與這些方法相比，另外一種最實效、最方便、最常用的方法就是使用風扇和散熱片。

(2.)說到 CPU 的風扇和散熱片，其實就是利用它們快速將 CPU 的熱量傳導出來並吹到附近的空氣中去，降溫效果的好壞直接與 CPU 散熱風扇、散熱片的品質有關，因此如何選擇一款合適的 CPU 風扇就成為我們非常關心的話題。

(3.)但由於目前市場上到處充斥著假冒偽劣產品，如果我們不了解有關 CPU 風扇的基本知識，那麼在購買過程中就很容易會上當受騙。因此

我們在選購風扇之前，有必要去了解一些有關 CPU 風扇的性能參數，一旦知道了這些鑒別標準，筆者相信大家肯定會選擇到一款實用而合適的風扇。下面，筆者就把有關 CPU 風扇的幾個主要性能參數介紹給大家，希望能對大家有所幫助。

風扇常見故障

(1.)轉動不靈：這是含油軸承常見的問題，原因就是軸承進了灰塵，潤滑液失去了潤滑作用。

(2.)噪音突然變得很大：遇到這種問題，首先要想的就是風扇是不是有某個葉片開裂了。

(3.)開機時噪音很大，轉動一會就安靜下來了：這種現象在冬天比較常見。原因也是潤滑液混進了灰塵，在冷狀態下粘稠度變大，就會失去潤滑作用。使用了一段時間后因為摩擦生熱，潤滑液受熱后流動性增加，就又恢復正常。

以上故障，解決的辦法都是去買個新的風扇換上，比更換整個散熱器要划算些。有些人建議是往軸承里加潤滑油，但我不推薦這麼做。因為軸承沒有清洗的話，加潤滑油也只能是當時感覺好些，但解決不了根本問題，過不了多久又會恢復原來的樣子。

(4.)風扇轉動正常，但散熱效果不好：這在小型散熱器上比較常見，原因很簡單，就是散熱鰭片之間堆滿了灰塵。這時需要把風扇拆下來，用毛刷把散熱器打掃乾淨就可以了。

3.7 LED 燈

LED 燈規格術語介紹

LED 尺寸大小：

0603、0805、1210、5050 是指 LED 燈帶上使用的發光元件----LED 的尺寸大小（英制/公制），

下面是這些規格的詳細介紹：

0603：換算為公制是 1005，即表示 LED 元件的長度是 1.0mm，寬度是 0.5mm。行業簡稱 1005，英制叫法是 0603.

0805：換算為公制是 2125，即表示 LED 元件的長度是 2.0mm，寬度是 1.25mm.行業簡稱 2125，英制叫法是 0805.

1210：換算為公制是 3528，即表示 LED 元件的長度是 3.5mm，寬度是 2.8mm。行業簡稱 3528，英制叫法是 1210.

5050:這是公制叫法，即表示 LED 元件的長度是 5.0mm，寬度是 5.0mm。行業簡稱 5050.

LED 燈數：

15 燈、30 燈、60 燈是指 LED 燈帶每米長度上焊接了多少顆 LED 元件，一般來說 1210 規格燈帶是每米 60 顆 LED，5050 規格燈帶是每米 30 顆 LED，特殊的有每米 60 顆 LED。不同 LED 數量的 LED 燈帶價格是不同的，這也是區分 LED 燈帶價格的一個重要因素。

色溫：

是指將一標準黑體加熱，溫度升高到一定程度時顏色開始由深紅 - 淺紅 - 橙黃 - 白 - 藍，逐漸改變，某光源與黑體的顏色相同時，我

們將黑體當時的絕對溫度稱為該光源之色溫。一般來說色溫不作為考核 LED 燈帶的一個指標，只是國外很多客戶因為使用環境的關係，會做一個特別的要求。

光源色溫不同，光色也不同：

色溫在 3300K 以下，光色偏紅給以溫暖的感覺；有穩重的氣氛，溫暖的感覺，通稱暖色溫。

色溫在 3000--6000K 為中間，人在此色調下無特別明顯的視覺心理效果，有爽快的感覺；故稱為"中性"色溫。

色溫超過 6000K，光色偏藍，給人以清冷的感覺，通稱冷色溫。

亮度：cd（坎德拉）

發光強度的基本單位，坎德拉是國際單位制的基本單位之一。

一般 LED 燈帶不同的顏色會有不同的發光強度，常用單位是 mcd，即毫坎德拉。數值越高，說明發光強度越大，也就是越亮。這是評定 LED 燈帶亮度的重要指標，亮度要求越高的燈帶價格越貴。這是因為高亮度的 LED 晶片價格偏貴，並且亮度越高，封裝難度越大。

發光角度：

這是指 LED 燈帶上 LED 元件的發光角度，一般通用的貼片 LED，即 SMD 元件的發光角度都是 120 度。發光角度越大，起散光效果越好，但相對的，其發光的亮度也就相應減小了。發光角度小，光的強度是上去了，但照射的範圍又會縮小。因此，評定 LED 燈帶的另一個重要指標就是發光角度。現在市面上有一些不良廠家，為了提高發光的亮度以賺取更高的利潤，故意把發光角度減小，稍有不慎，就會買到這樣的以次充好的元件。

第4章 智慧農場的程式開發

本專題的程式採用 Arduino IDE 開發環境開發，其目的在 Arduino 上建立一個網頁伺服器。透過網頁的介面，提供農場即時的溫濕度與照明資訊。並提供控制介面使可控制風扇與 LED 燈的開關。其中所採用的程式庫如圖 31 所示，其中有提供 Arduino 與 WiFi 卡通訊的 SPI、無線區域網路的 WiFi、建立網頁與網頁伺服器的 WebServer，，溫濕度感測的 dht11，處理字串的 Streaming 等程式庫。

```
// 使用Arduino Ethernet/WiFi Shield 網路卡與Webduino程式庫建立微網站
// Webduino程式庫有Wifi版與Ethernet版，
//   Ethernet版目錄: Webduino-Wired
//   Wifi版目錄: Webduino-Wifi
//編譯使用前請先做下列準備：
// 原始函式庫在Webduino-Wifi目錄，編譯前請先將整個目錄複製，並將名稱改為Webduino
#include "SPI.h"
#include "WiFi.h"
#include "WebServer.h"
#include "Streaming.h"
#include "dht11.h"
```

圖 4-1 引用程式庫

圖 32 為程式中 Arduino 相關軟硬體的基本設定，包括各種感測器與週邊裝置的接腳、WiFi 的網路名稱與密碼，網路的 ip 地址與相關設定、網頁伺服器的 Port Number 等。圖 33 為網頁的表頭設定，包括設定每三秒更新網頁一次。圖 34 為網頁的頁面底部設定，包括一頁常用按鈕。圖 35 為預設網頁的設計，會顯示溫度、濕度與照明等資訊。圖 36 為網頁中「關於」畫面的設定

```

dht11 DHT11;          // 宣告 DHT11 程式物件
const byte dataPin = 2; // 宣告 DHT11 模組的資料輸入腳位
const byte FAN_PIN = 8; // 風扇控制角位
const byte LED_PIN = 9; // LED控制角位
const byte cdsPin = A0; // 光敏電阻角位

boolean fanState = 0;
boolean ledState = 0;
char ssid[] = "RICH2F";
char pass[] = "0425600065";

int status = WL_IDLE_STATUS;

//static byte mac[] = { 0xF0, 0x7B, 0xCB, 0x4B, 0x7C, 0x9F };
IPAddress ip(192, 168, 11, 211);
IPAddress subnet(255, 255, 255, 0);
IPAddress gateway(192, 168, 11, 1);
IPAddress dnsServer(192, 168, 11, 1);

WebServer webserver("", 80);

```

圖 4-2 宣告物件、控制接腳、WIFI 位址

```

P(htmlHead) =
"<!doctype html><html>"
"<head><meta charset='utf-8'>"
"  <meta http-equiv='refresh' content='3'>"
"<meta name='viewport' content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0'>"
"<title>Arduino 智慧農場-Wifi版</title>"
"</head><body>" ;

P(htmlHead1) =
"<!doctype html><html>"
"<head><meta charset='utf-8'>"
"<meta name='viewport' content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0'>"
"<title>Arduino 智慧農場-Wifi版</title>"
"</head><body>";

```

圖 4-3 網頁初始頁面與介紹

```

P(htmlFoot) =
  "<p> <a href =\"fan_form.html\"> 控制風扇</a></p>"
  "<p> <a href =\"led_form.html\"> 控制LED燈</a></p>"
  "<p> <a href =\"about.html\"> 關於SmartFarm</a></p>"
  "<p><a href =\"/\> 回到首頁</a></p>"
  "</body></html>";

```

圖 4-4 網頁底部控制項設計

```

void defaultCmd(WebServer &server, WebServer::ConnectionType type, char *, bool)
{
  int dht = DHT11.read(dataPin);
  int light = analogRead(cdsPin);
  char buffer[5] = "";
  server.httpSuccess();
  Serial.println("httpSuccess: defaultCmd");
  if (type != WebServer::HEAD)
  {
    server.printP(htmlHead);
    if (dht == 0) {
      server << "<h1>修平智慧農場</h1>";
      server << "<p>溫度：" << dtostrf(DHT11.temperature, 5, 1, buffer)
        << "&deg;C</p>";
      server << "<p>濕度：" << dtostrf(DHT11.humidity, 5, 1, buffer)
        << "%</p>";
    } else {
      server << "<h1>無法讀取溫濕度值，請檢查設備。</h1>";
    }
    server << "<p>照度：" << dtostrf((light) / 512 * 100, 5, 1, buffer)
      << "%</p>";
    server << "<p>風扇：" << (fanState ? "開" : "關") << "</p>";
    server << "<p>LED燈：" << (ledState ? "開" : "關") << "</p>";
    server.printP(htmlFoot);
  }
}

```

圖 4-5 初始網頁讀取資訊 零件開關頁面設計

```

P(aboutPage) =
  "</head><body>這是WiFi版智慧農場網頁，由修平科大資網系開發<br>2015年12月";

void aboutCmd(WebServer &server, WebServer::ConnectionType type, char *, bool)
{
  server.httpSuccess();
  Serial.println("httpSuccess: aboutCmd");
  if (type != WebServer::HEAD)
  {
    server.printP(htmlHead1);
    server.printP(aboutPage);
    server.printP(htmlFoot);
  }
}

```

圖 4-6 「關於」網頁設計

```

P(fanFORM) =
  "<form method=\"post\" action=\"fan_sw\">"
  "風扇：<input name=\"fan\" type=\"radio\" value=\"ON\"> 開"
  "<input name=\"fan\" type=\"radio\" value=\"OFF\" checked> 關"
  "<br><br><input type=\"submit\" name=\"button\" value=\"送出\">"
  "</form>";

void fanCmd(WebServer &server, WebServer::ConnectionType type, char *, bool)
{
  server.httpSuccess();
  Serial.println("httpSuccess: fanCmd");
  if (type != WebServer::HEAD)
  {
    server.printP(htmlHead1);
    server.printP(fanFORM);
    server.printP(htmlFoot);
  }
}

```

圖 4-7 風扇開關指令頁

圖 37 為風散的開關控制使用者頁面的設定，圖 38 為風散的開關控制的伺服器端程式碼。圖 39 為 LED 的開關控制使用者頁面的設定，圖 40 為 LED 的開關控制的伺服器端程式碼。

```
void postFan(WebServer &server, WebServer::ConnectionType type, char *, bool)
{
  char name[16], value[16];
  server.httpSuccess();
  Serial.println("httpSuccess: postFan");
  if (type == WebServer::POST)
  {
    server.printP(htmlHead1);
    while (server.readPOSTparam(name, 16, value, 16)) {

      if (strcmp(name, "fan") == 0) {
        server << "<p>風扇已經";
        if (strcmp(value, "ON") == 0) {
          server << "打開◦ </p>";
          digitalWrite(FAN_PIN, HIGH);
          fanState = 1;
        } else {
          server << "關閉◦ </p>";
          digitalWrite(FAN_PIN, LOW);
          fanState = 0;
        }
      }
    }
    server.printP(htmlFoot);
  }
}
```

圖 4-8 風扇開關程式設計頁

```

P(ledFORM) =
"<form method=\"post\" action=\"led_sw\">"
"LED燈: <input name=\"led\" type=\"radio\" value=\"ON\"> 開"
"<input name=\"led\" type=\"radio\" value=\"OFF\" checked> 關"
"<br><br><input type=\"submit\" name=\"button\" value=\"送出\">"
"</form>";

void ledCmd(WebServer &server, WebServer::ConnectionType type, char *, bool)
{
    server.httpSuccess();
    Serial.println("httpSuccess: ledCmd");
    if (type != WebServer::HEAD)
    {
        server.printP(htmlHead1);
        server.printP(ledFORM);
        server.printP(htmlFoot);
    }
}

```

圖 4-9 LED 開關指令頁

```

void postLed(WebServer &server, WebServer::ConnectionType type, char *, bool)
{
    char name[16], value[16];
    server.httpSuccess();
    Serial.println("httpSuccess: postLedCmd");
    if (type == WebServer::POST)
    {
        server.printP(htmlHead1);
        while (server.readPOSTparam(name, 16, value, 16)) {

            if (strcmp(name, "led") == 0) {
                server << "<p>LED已經";
                if (strcmp(value, "ON") == 0) {
                    server << "打開 </p>";
                    digitalWrite(LED_PIN, HIGH);
                    ledState = 1;
                } else {
                    server << "關閉 </p>";
                    digitalWrite(LED_PIN, LOW);
                    ledState = 0;
                }
            }
        }
        server.printP(htmlFoot);
    }
}

```

圖 4-10 LED 開關程式設計頁面

圖 41 為 IO 接腳的起始設定與 WiFi 起始連線與網頁設定的程式碼。圖 42 為網頁伺服器執行連線處理與顯示 WiFi 連線狀態的程式碼。

```
void setup() {
  pinMode(FAN_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(9600); //初始化序列埠至9600
  while (!Serial) { } ; // wait for serial port to connect.
  // check for the presence of the shield:
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while (true);
  }
  WiFi.config(ip, dnsServer, gateway, subnet);
  // attempt to connect to Wifi network:
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid, pass);

    // wait 5 seconds for connection:
    delay(5000);
  }
  webserver.setDefaultCommand(&defaultCmd); // 處理「首頁」請求
  webserver.addCommand("about.html", &aboutCmd); // 處理「about頁面」請求
  webserver.addCommand("fan_sw", &postFan); // 處理「fan開關表單處理頁面」請求
  webserver.addCommand("fan_form.html", &fanCmd);
  webserver.addCommand("led_sw", &postLed); // 處理「led開關表單處理頁面」請求
  webserver.addCommand("led_form.html", &ledCmd);
  webserver.begin();
  printWifiStatus();
}
```

圖 4-11 處理頁面程式請求 WIFI 連線設定

```
void loop() {
  webserver.processConnection();
}
/**印出連線的WiFi狀態*/
void printWifiStatus() {
  Serial.println(WiFi.firmwareVersion());
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

圖 4-12 顯示 WIFI 連線目前狀態

第5章 實際測試過程與結果

綠能智慧農場狀態網頁如圖43 所示，我們採用標準的HTML 語法將感測器所偵測到的綠能智慧農場的即時狀態顯示於網頁上。為了及時顯示農場的状态，此網頁會每隔10 秒更新一次。



圖 5-1 智慧農場網頁

農場管理者可以透過 Arduino 控制器的網頁調整LED 與DC 風扇的運作。圖44顯示控制LED 燈的畫面，控制DC 風扇的操作畫面與圖3 類似。此網頁採用Webduino 建立表單，並且用POST 方法傳送資料表單。有別於一般的GET 方法，有較大資料傳輸量，也比較不會有資訊安全的問題。



圖 5-2 綠能智慧農場控制 LED 網頁(與控制風扇的網頁畫面類似)

第6章 心得與討論

程式設計與測試心得:

再這次的專題中，我負責的項目是打程式的部份。雖然中間有很多地方不懂，經由老師指導或查詢書本，漸漸的打出來了，再來就是利用 Arduino 作測試，如果有錯誤當下問老師要如何修改，可是老師不會直接告訴你答案，所以還是自己要去摸索，說真的那時候真的試蠻多次的，最後翻了課本找到了相關指令，打上去試試，結果還是不行，老師看到了我們試了許久，還是沒結果，老師就告訴我要去哪裡找答案，終於找到了正確的指令，也成功的運作了!當時是很許久，看到了有了成果，然後也不會影響大家的專題製作，覺得辛苦有代價了。

首先要感謝各位組員的辛苦，大家都從忙碌中撥空到學校做專題，也讓各位的有時間討論該朝哪個方向去努力，在做成品的時候還要請組員跑來跑去的去買材料，為了這個專題也看出組員的團結心，經過了慢長的時間我們的專題製作終於要完成了，因為這個專題讓我了解的 Arduino 的使用與運用，更熟悉的去操作這些東西與其他硬體配件的應用與功用，這不只是單純的畢業門檻，也是間接的讓我們吸收這些知識。

溫室設計與組裝心得:

在這次組裝上需要把溫濕度感測器以及光亮度感測器（光敏電

阻) 架設在溫室內部可以精準測量的地方，而主要線路則設在溫室本體外面，在這方面上需要思考線路要如何銜接才不會照成打結或者是浪費線材，其他設備則是假設在本體外面，在這次組裝上讓我對於線路的接法的更上一層，還有學到如何省下更多線材，不照成浪費。

電路設計與配線心得:

終於完成了這個專題製作，首先要謝謝指導老師瑞淇老師的幫忙，在專題製作的過程中也遇到很多難題，一度有想放棄的念頭但是想想之前老師說過的話，我們已經走了九十九步就差那麼一步而前功盡棄，咬緊牙關最後結果會是好的，分配好每個環節，而同學們也都很團結一致沒有那種孤軍奮鬥的感覺，最感謝我們的組長 - 陳園羽顧慮到我們平常有工作，自己擔起專題組長的責任，最後希望專題能夠順利完成。

文件與綜合支援心得:

剛開始製作文書報告時，總是思考如何製作才是對的，之後到圖書館參考各屆學長姐的作品，才讓我找到目標去進行製作，當然其中也少不了其他組員的幫忙，在報告裡我看到了我們一步步的製作過程，也讓我覺得辛苦是值得的，在其中我也會看到需要幫忙的組員前去進行幫助，很多難題總是大家一起集思廣益才想出解決辦法來，一邊幫助其他組員一邊進行報告的資料收集，並看到大家努力的過程，雖然製作報告的過程需要大量的時間，但只要看到大家一起辛苦的成果，就感覺不到任何的疲累，一份完整的報告少不了所有組員的辛苦，每個組員都是缺一不可，相信每個組員看到報告與成品都會露出微笑，因為這是好多時間累積下來的結晶。

參考資料

- [1] 趙英傑，超圖解 Arduino 互動設計入門(第二版)，旗標出版股份有限公司，台北市，2014 年 4 月
- [2] P. Seneviratne 原著，江良志譯，Arduino 物聯網專案實作，博碩文化股份有限公司，台北市，2016 年 2 月
- [3] 楊明豐，Arduino 最佳入門與應用：打造互動設計輕鬆學(暢銷經典第二版)(附光碟)，碁峰資訊股份有限公司，台北市，2015 年 4 月
- [4] 艾迪諾，Arduino 全能微處理機實習：強效解析(附範例及教學投影片光碟)，全華圖書股份有限公司，台北市，2016 年 3 月
- [5] 孫駿榮，Arduino 互動設計超入門：用 ArduBlock 圖形化控制真簡單，碁峰資訊股份有限公司，台北市，2014 年 8 月
- [6] 柯博文，Arduino 互動設計專題與實戰，碁峰資訊股份有限公司，台北市，2014 年 8 月
- [7] Mark Geddes，譯者：CAVEDU 教育團隊、曾吉弘，Arduino 自作專案大百科，碁峰資訊股份有限公司，台北市，2015 年 7 月
- [8] 陳明榮，Arduino 語音互動專題製作與應用(附光碟)，松崗電腦圖書資料股份有限公司，台北市，2016 年 3 月
- [9] 無線電編輯部，現場直擊：35 個 Arduino 大神的實作範例--第 2 版，佳魁資訊，台北市，2016 年 4 月