

臺北市立大安高級工業職業學校  
電子科

專題報告

智慧電風扇  
Intelligent fan

學生 組長：陳毓堉  
組員：王主恩  
組員：王經安  
組員：王維熙

指導老師：簡靖哲老師

中華民國 109 年一月

## 中文摘要

電風扇是以前冷氣還不普及時夏日家中最常使用的家電，即使到現今電扇也被作為主要的空調設備之一。

隨著科技進步，市面上的電扇產品也日新月異，像是人手一支的手持風扇，或是攜帶方便的 USB 風扇等等。隨著全球暖化日加嚴重，冷氣的需求大增，尤其到了夏日電費更是貴得驚人，因此相較於耗電的冷氣也能夠節省電費，此外也不會使身體過度受寒。

此專題主要是搭配紅外線感測器感測物體移動，再藉由伺服馬達轉動的特性，使電扇能夠隨著人體移動而改變其風向。也可透過手機藍芽遙控控制風速，讓使用性更加提升。

關鍵字：藍芽、風扇、自動偵測

## **Abstract**

Electric fans are the most commonly used household appliances in summer when air-conditioning was not popular. Even now, electric fans have been used as one of the main air-conditioning equipment.

With the advancement of science and technology, the electric fan products on the market are also changing with each passing day, such as a hand-held fan or a portable USB fan. With the increasing global warming, the demand for air-conditioning has increased greatly, especially in summer. Electricity bills are even more expensive. Therefore, compared to power-consumption air-conditioning, electricity bills can be saved, and the body will not be excessively cold.

This topic is mainly matched with the infrared sensor to detect the movement of the object, and then the characteristic of the servo motor rotation, so that the electric fan can change its wind direction as the human body moves. The wind speed can also be controlled remotely through the mobile phone's Bluetooth, which improves the usability.

# 目錄

中文摘要.....	I
Abstract.....	II
目錄.....	III
圖目錄.....	IV
表目錄.....	V
第一章前言.....	1
1-1 專題製作背景.....	1
1-2 預期成果.....	1
第二章理論探討.....	2
2-1 MEGA2560.....	2
2-2 HC-05 藍芽模組.....	3
2-3 軟體部分.....	4
2-4 E18-D80NK工作原理.....	5
2-5 MG996R工作原理.....	6
2-6 直流馬達工作原理.....	7
2-7 L298N.....	8
第三章專題設計.....	9
3-1 系統架構.....	9
3-2 專題流程.....	10
3-3 甘特圖.....	11
3-4 電路設計圖.....	12
第四章專題成果.....	14
第五章結論和建議.....	18
5-1 結論.....	18
5-2 問題與解決.....	18
5-3 建議.....	18
參考文獻.....	19
附錄.....	20
附錄一設備清單.....	20
附錄二材料清單.....	21
附錄三程式碼.....	22

## 圖目錄

圖 1	Arduino MEGA2560 板.....	3
圖 2	HC-05.....	3
圖 3	Arduino 介面.....	4
圖 4	APP INVENTER.....	4
圖 5	E18-D80NK.....	5
圖 6	MG996R.....	6
圖 7	馬達工作原理.....	7
圖 8	馬達工作原理.....	7
圖 9	馬達工作原理.....	8
圖 10	系統架構圖.....	9
圖 11	專題流程圖.....	10
圖 12	甘特圖.....	11
圖 13	電路設計圖.....	12
圖 14	電路設計圖.....	13
圖 15	完成品.....	14
圖 16	電扇關閉未動作.....	14
圖 17	數值顯示.....	15
圖 18	伺服馬達轉動.....	15
圖 19	自然風模式.....	16
圖 20	風速變換.....	16
圖 21	風速變換.....	17
圖 22	關閉所有電扇功能.....	17

## 表目錄

表 1	Arduino MEGA2560 參數表.....	2
表 2	E18-D80NK 參數表.....	5
表 3	MG996R 參數表.....	6
表 4	L298N 參數表.....	8
表 5	參考文獻.....	19
表 6	設備清單.....	20
表 7	材料清單.....	21

## 第一章 前言

### 1-1 專題製作背景

目前市面上的電風扇固定性的轉動會讓其效率降低，每個人吹到的時間很短，為了將電扇的使用效率大為提升，我們善用所學，將高一至高三所學習的知識及專業技巧濃縮起來，嘗試做出一台能夠依循人體移動而改變其移動方向的電扇，也能藉由藍芽控制其轉動風速。

### 1-2 預期成果

可控制各種模式，風速「強、中、弱、自然風」及「自動擺頭、偵測人擺頭、定時」等功能，並運用「MIT App Inventor2」製作 APP 與藍芽模組「HC-05」連動控制風扇各種模式。

## 第二章 理論探討

### 2-1 MEGA2560

Arduino Mega2560 是一塊以 ATmega2560 為核心的微控制器開發板，本身具有 54 組數位 I/O input/output 端（其中 14 組可做 PWM 輸出），16 組模擬比輸入端，4 組 UART (hardware serial ports)，使用 16 MHz crystal oscillator。由於具有 bootloader，因此能夠通過 USB 直接下載程式而不需經過其他外部燒入器。供電部份可選擇由 USB 直接提供電源，或者使用 AC-to-DC adapter 及電池作為外部供電。

由於開放原代碼，以及使用 Java 概念（跨平臺）的 C 語言開發環境，讓 Arduino 的周邊模組以及應用迅速的成長。而吸引 Artist 使用 Arduino 的主要原因是可以快速使用 Arduino 語言與 Flash 或 Processing...等軟體通訊，作出多媒體互動作品。Arduino 開發 IDE 介面基於開放原代碼原則，可以讓您免費下載使用於專題製作、學校教學、電機控制、互動作品等等。

表 1 Arduino MEGA2560 參數表

控制器	ATmega2560
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 口	54（含 15 路 PWM 輸出）
模擬輸入口	16
I/O 口直流電流	40 mA
3.3v 口直流電流	50 mA
快閃記憶體（Flash Memory）	256 KB
靜態儲存器（SRAM）	8 KB
EEPROM	4 KB
時脈	16 MHz





圖 1 Arduino MEGA2560 板

## 2-2 HC-05 藍芽模組

HC-05 是一款藍芽模組，市面上較常見的除了 HC-05 之外，還有 HC06，兩者的差異就是 HC-05 可以自行設定為主端或從端，然而 HC-06 在出廠前就已經被設定為從端或者是主端，無法自行更改，然而所謂的主端就是發送指令，從端則是接收命令。

透過 AT 模式可以將已設定為主端或從端的模式進行更改，一般的 HC05 總共有六隻接腳，分別是 VCC, GND, TXD, RXD, STATE, KEY。

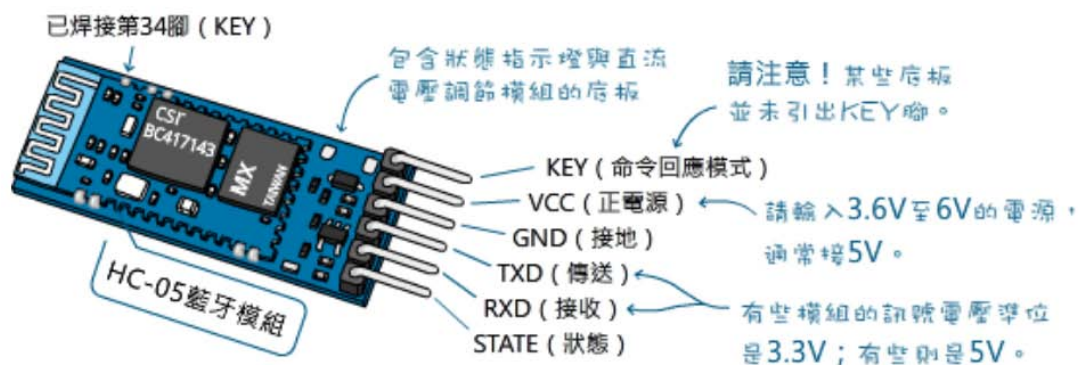


圖 2 HC-05

## 2-3 軟體部分

Arduino 的軟體開發環境是開放源始碼，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，利用這項軟體進程式的上傳，以便於和各項感應器的溝通。

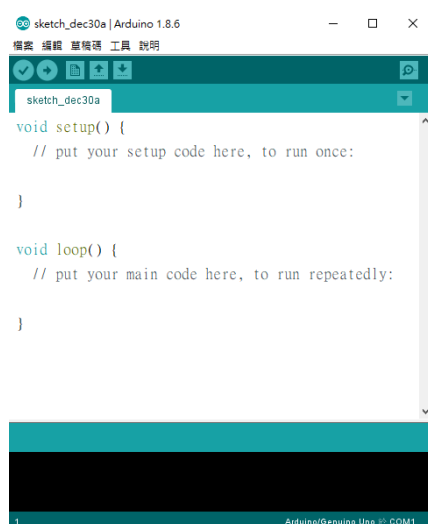


圖 3 Arduino 介面

它可以讓任何熟悉或不熟悉程序設計的人來創造基於 Android 作業系統的應用軟體。它使用圖形化界面，非常類似於 Scratch 語言和 StarLogo TNG 用戶界面。用戶可以拖放圖形對象來創造一個運行在安卓系統上的應用，它就可以在許多手機設備上運行。



圖 4 APP INVENTER

## 2-4E18-D80NK 工作原理

紅外線是一種波長介於微波與可見光之間的電磁波，其波長約在 760 奈米至 1 毫米之間，是波長比紅光長的非可見光，對應頻率約在 430 THz 到 300 GHz 的範圍內。室溫下物體所發出的熱輻射大多都在此波段。紅外線也可應用在軍事、工業、科學及醫學的領域。

E18-D80NK 是一種集發射與接收於一體的光電感測器。檢測距離可以根據要求進行調節。此感測器具有探測距離遠、受可見光干擾小、價格便宜、易於裝配、使用方便等特點，可以廣泛應用於機器人避障、流水線計件等眾多場合。

表 2 E18-D80NK 參數表

控制器	E18-D80NK
工作電壓	5V
工作電流	<25mA
有效探測距離	3~80cm
響應時間	<2ms
指向角	$\leq 15^\circ$
工作環境溫度	$-25^\circ\text{C} \sim 55^\circ\text{C}$
外殼材料	塑料
傳感器線長	45mm
引線長度	45cm
檢測物體	透明或不透明體
適用	生產線貨物自動計數設備、多功能提醒器、走迷宮機器人



圖 5 E18-D80NK

## 2-5 MG996R 工作原理

伺服馬達的動作特性是進行位置定位控制和動作速度控制，其主要特點是轉速可以精確控制，速度控制範圍廣，可以安定平順等速運轉之外，還可以根據需求隨時變更速度。在極低速度也可以穩定轉動。能迅速做出正轉與逆轉，也能迅速加減速。在由靜態改為動態運作或由動態改為靜態運作所需費時極短，而且即便有外力附加仍可以保持位置。並在額定容量範圍內瞬間產生大轉矩，輸出功率大且效率也高。

MG996R 舵機為原 MG995 的升級產品，速度，拉力和精確度都比原裝產品有改進，是目前市場上性價比最高的大扭力舵機之一。

表 3 MG996R 參數表

控制器	MG996R
工作電壓	4.8~7.2V
空載工作電流	120mA
堵轉工作電流	1450mA
運行速度	0.17 秒/60 度(4.8V 空載)
運行速度	0.13 秒/60 度(6.0V 空載)
響應脈寬時間	≤5usec
角度偏差	回中誤差 0 度，左右各 45° 誤差≤3°



圖 6 MG996R

## 2-6 直流馬達工作原理

直流馬達的好處為在控速方面比較簡單，只須控制電壓大小即可控制轉速，但此類馬達不宜在高溫、易燃等環境下操作，而且由於馬達中需要以碳刷作為電流變換器的部件，所以需要定期清理炭刷磨擦所產生的污物。無碳刷之馬達稱為無刷馬達，相對於有刷，無刷馬達因為少了碳刷與軸的摩擦因此較省電也比較安靜。製作難度較高、價格也較高。一般工業用直流馬達之電壓 DC 110V 和 DC 220V 兩種。

直流馬達的基本構造包括「電樞」、「場磁鐵」、「集電環」、「電刷」。

電樞:可以繞軸心轉動的軟鐵芯纏繞多圈線圈。

場磁鐵:產生磁場的強力永久磁鐵或電磁鐵。

集電環:線圈約兩端接至兩片半圓形的集電環，隨線圈轉動，可供改變電流方向的變向器。每轉動半圈，線圈上的電流方向就改變一次。

電刷:通常使用碳製成，集電環接觸固定位置的電刷，用以接至電源。

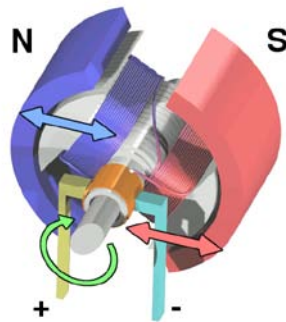


圖 7 馬達工作原理

此為一個簡單的直流電電動機。當線圈通電後，轉子周圍產生磁場，轉子的左側被推離左側的磁鐵，並被吸引到右側，從而產生轉動

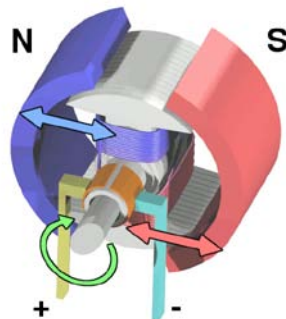


圖 8 馬達工作原理

轉子依靠慣性繼續轉動

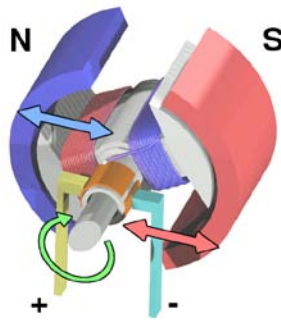


圖 9 馬達工作原理

當轉子運行至水平位置時電流變換器將線圈的電流方向逆轉，線圈所產生的磁場亦同時逆轉，使這一過程得以重複。

## 2-7 L298N

L298N 是 ST 公司生產的一種高電壓、大電流電機驅動晶片。此晶片採用 15 腳封裝。主要特點是：工作電壓高，最高工作電壓可達 46V；輸出電流大，瞬間峰值電流可達 3A，持續工作電流為 2A；額定功率 25W。內含兩個 H 橋的高電壓大電流全橋式驅動器，可以用來驅動直流馬達和步進馬達、繼電器線圈等感性負載；採用標準邏輯電平信號控制；具有兩個使能控制端，在不受輸入信號影響的情況下允許或禁止器件工作有一個邏輯電源輸入端，使內部邏輯電路部分在低電壓下工作；可以外接檢測電阻，將變化量反饋給控制電路。使用 L298N 晶片驅動馬達，該晶片可以驅動一台兩相步進馬達或四相步進馬達，也可以驅動兩台直流馬達。

表 4 L298N 參數表

主控晶片	L298N
電壓	5V
驅動電壓	5V~35V
電流	0mA~36mA
驅動電流	2mA
工作溫度	-20°C ~ 135°C
工作環境溫度	-25°C ~ 55°C
最大功率	25W

L298N 作為主驅動晶片，具有驅動能力強，發熱量低，抗干擾能力強的特點，內置的 78M05 通過驅動電源部分取電工作，但是為了避免穩壓芯片損壞，當使用大於 12V 驅動電壓的時候，需使用外置的 5V 邏輯供電。

### 第三章 專題設計

#### 3-1 系統架構

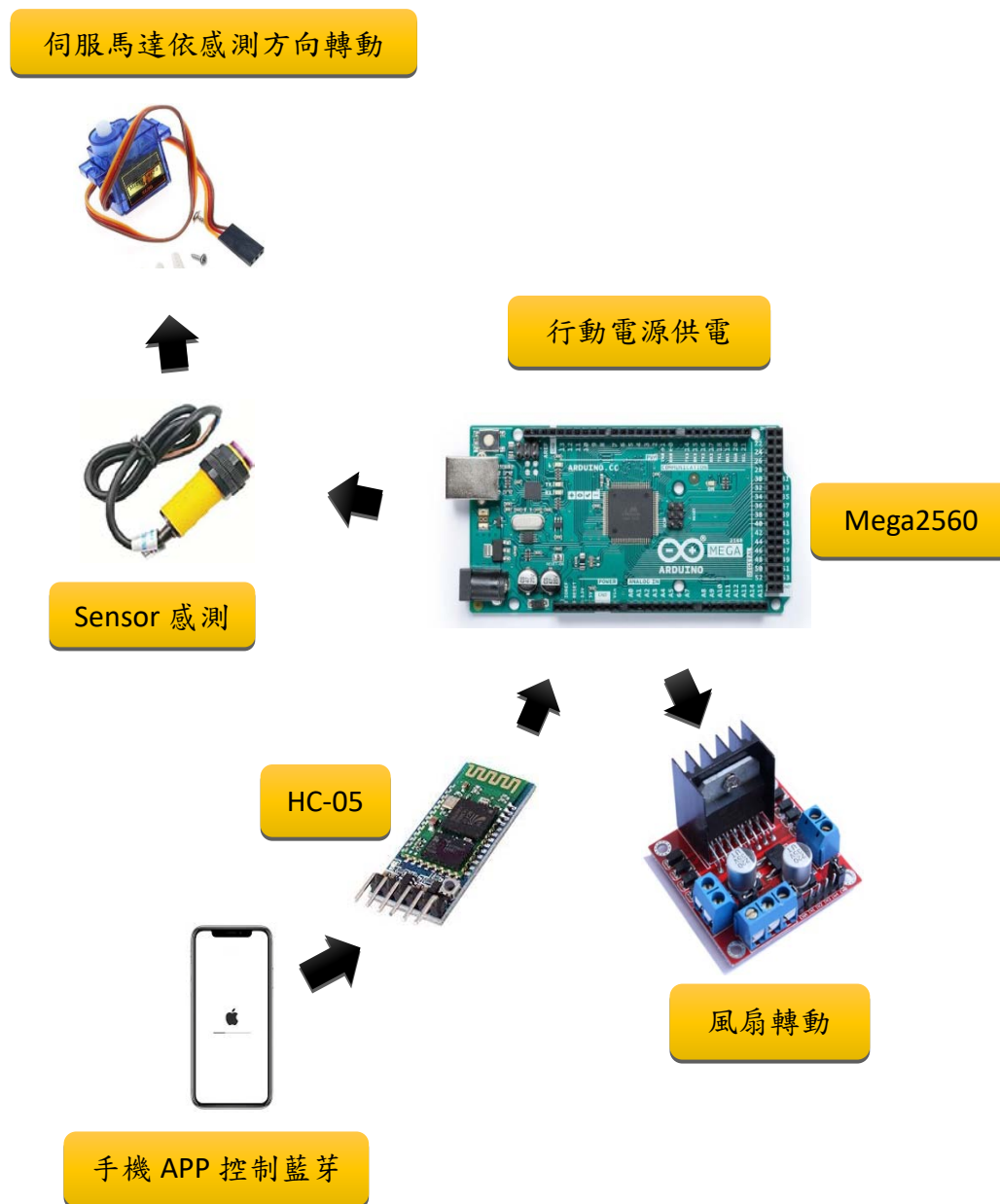


圖 10 系統架構圖

### 3-2 專題流程

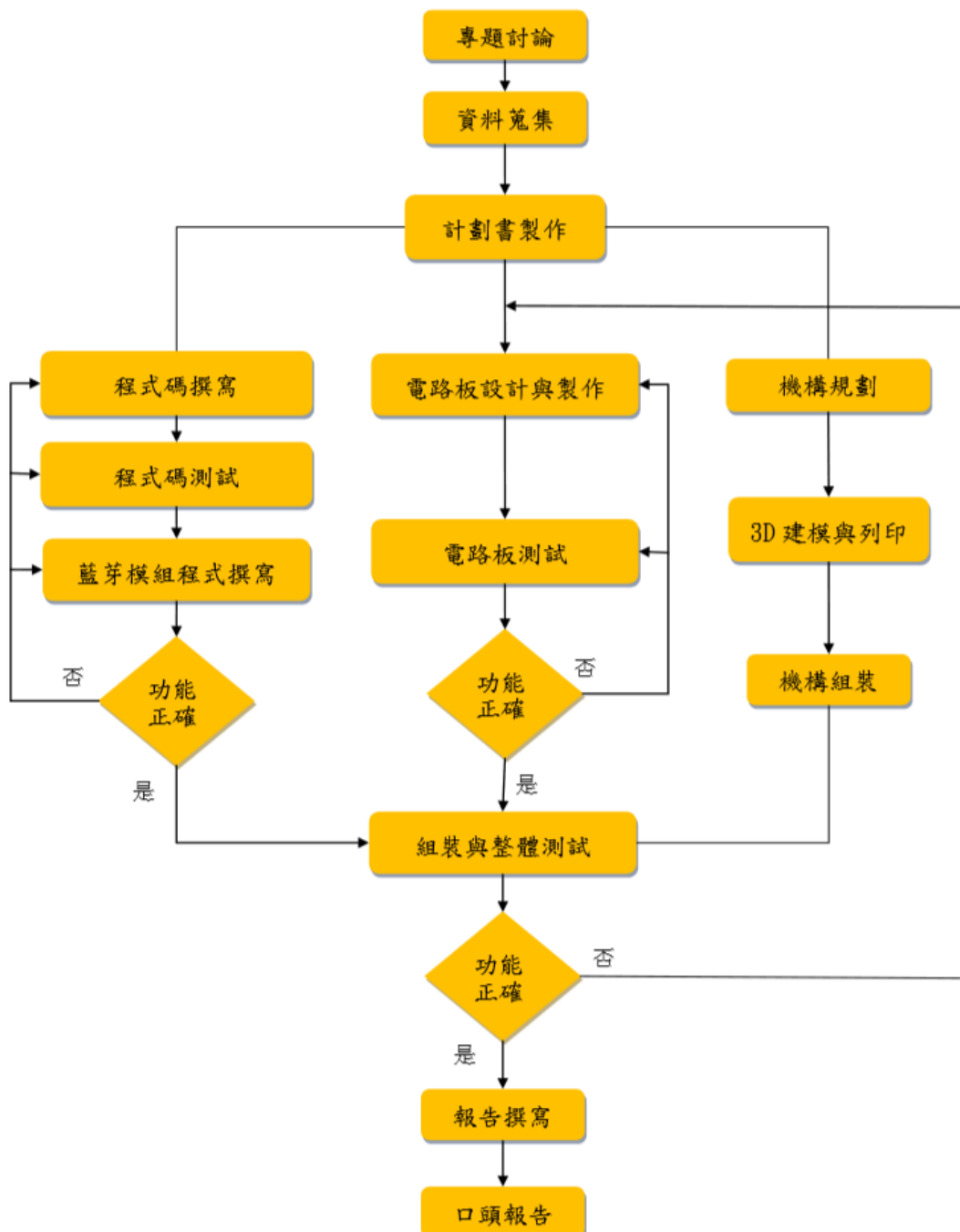


圖 11 專題流程圖



### 3-3 甘特圖

工作項目	週次 (日期)																				負責成員
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
專題討論	■	■	■	■																	全
資料蒐集	■	■	■	■	■	■	■														全
計劃書製作				■																	全
程式碼撰寫	■	■	■	■	■	■	■	■	■												陳
程式碼測試					■	■	■	■	■	■											陳
藍芽模組程式撰寫					■	■	■	■	■	■											陳
電路板設計與製作					■	■	■	■	■	■	■	■									恩、安
電路板測試							■	■	■	■	■	■	■								恩、安
機構規劃									■	■	■	■	■	■	■						熙、安
3D 建模及列印											■	■	■	■	■	■					熙、安、恩
機構組裝													■	■	■	■	■				熙、安、恩
組裝與整體測試																			■	■	全
報告撰寫					■			■	■		■		■		■	■		■		■	全
口頭報告						■				■		■		■	■		■				全
預定進度	0	5	10	15	20	25	30	35	50	55	57	60	65	70	75	80	85	90	95	100	累 積 百分比%

圖 12 甘特圖

### 3-4 電路設計圖

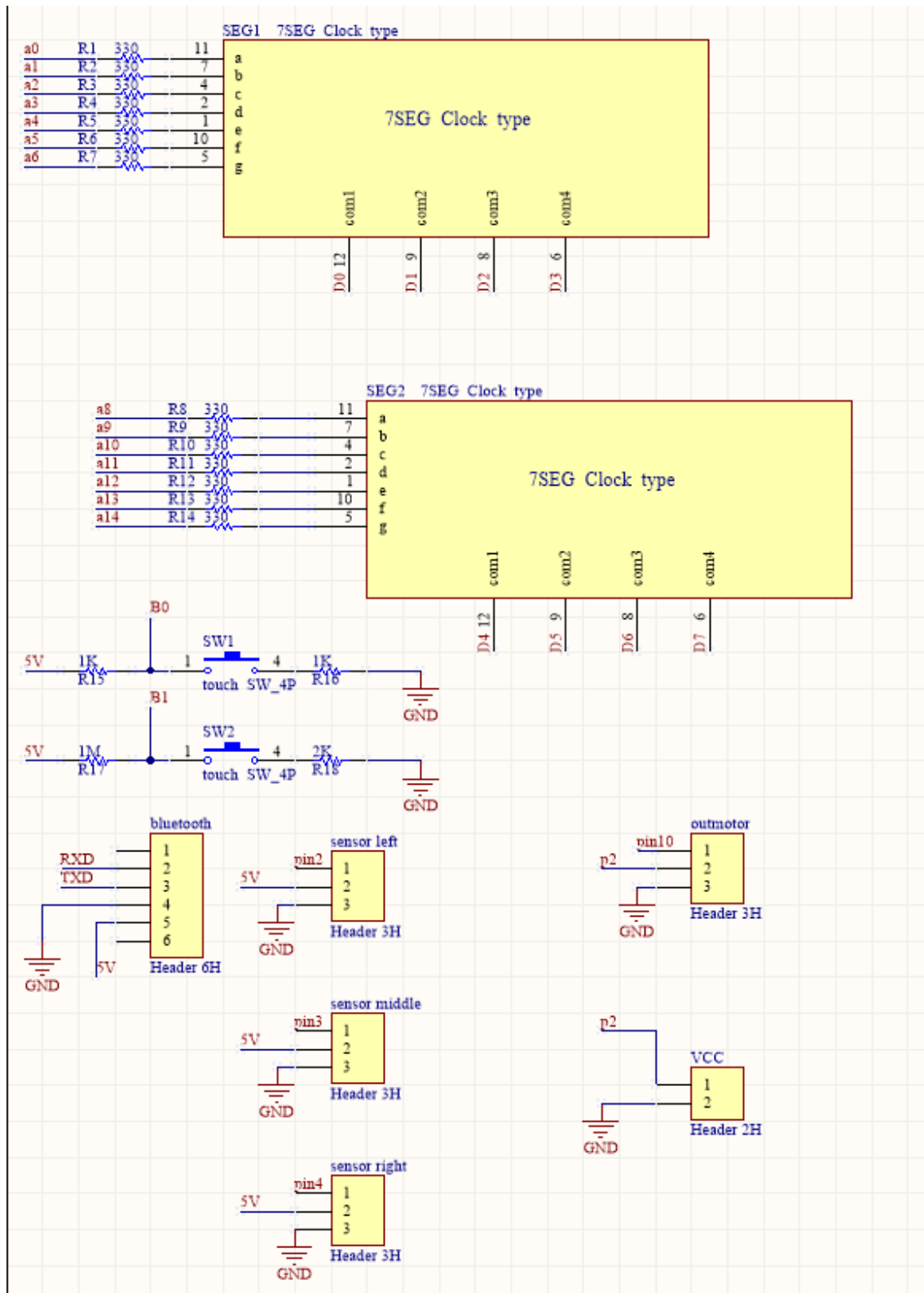


圖 13 電路設計圖

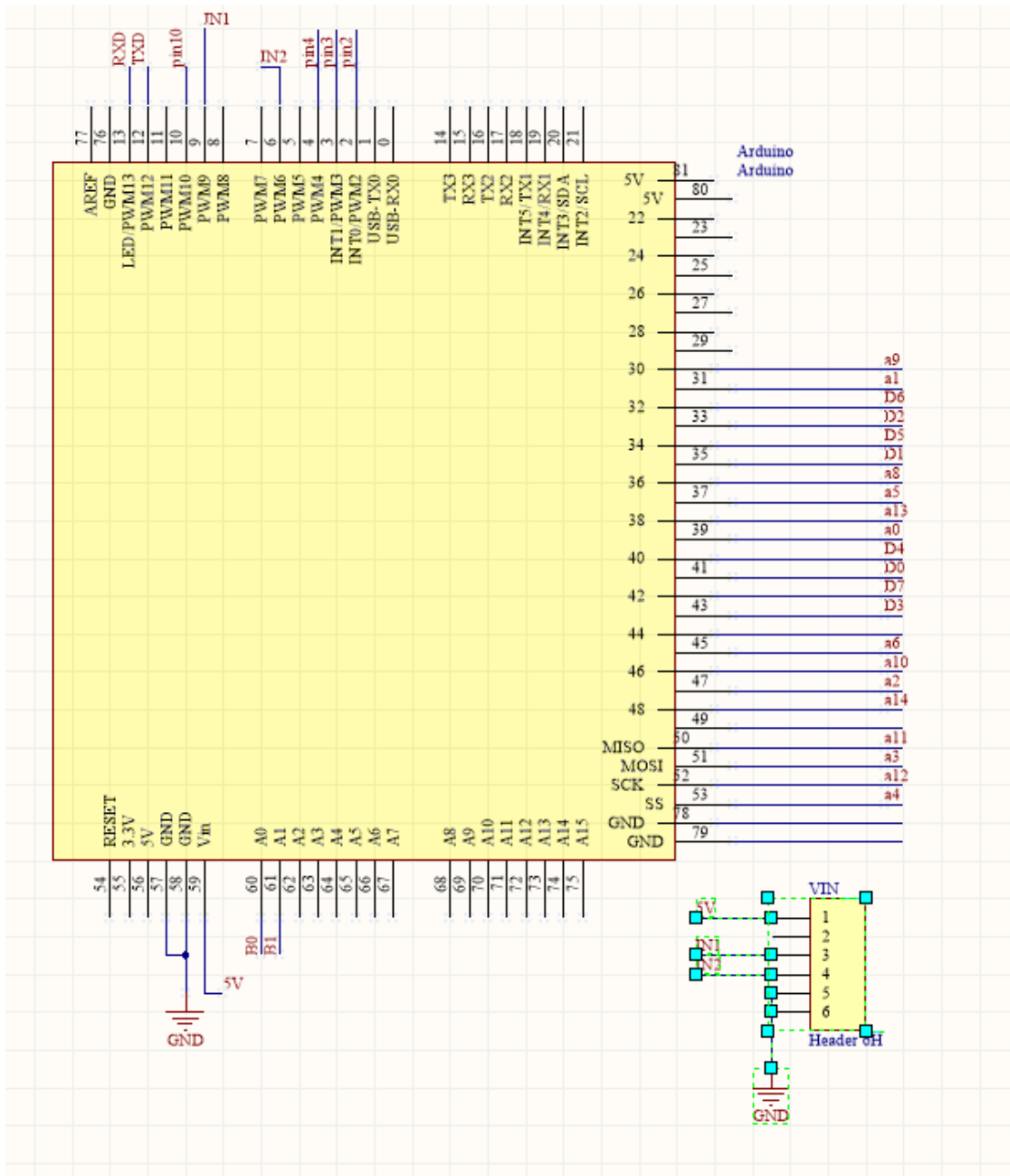


圖 14 電路設計圖

#### 第四章 專題成果



圖 15 完成品

電扇未開時，感測器不會動作，伺服馬達也不會轉動



圖 16 電扇關閉未動作

七段能正常顯示風速及定時數值



圖 17 數值顯示

伺服馬達依照感測器偵測方向轉動



圖 18 伺服馬達轉動



圖 19 自然風模式

藉由手機藍芽連線控制風速(L40)



圖 20 風速變換

將風速由 L 40 一路加到 L 58



圖 21 風速變換

當定時時間到達，關閉所有電扇功能



圖 22 關閉所有電扇功能

## 第五章 結論和建議

### 5-1 結論

除了可利用手動模式控制電扇轉速，也可讓現代手機不離身的人們，透過藍芽遙控定時，並控制伺服馬達轉向，在操作上大為提升其方便性。雖然功能並不完整，但該有的基本功能都有實現，如果時間夠充裕，或許能夠做出更多獨特功能的電扇。

### 5-2 問題與解決

AD 的操作全部忘光光，使得電路板的設計常有漏洞，因此板子洗出來的時候常常會出現問題，例如：接線斷掉，電路接觸不良等等…，不過再經過自己研讀 AD 操作課本，再加上老師的幫忙，終於將成品完成，雖然有小部分的七段問題，但後來發現是自己腳位設錯，改正過來後就可正常動作。

### 5-3 建議

與市面上的電扇產品重疊性過高，雖然利用 Sensor 來感測人體移動，但其偵測範圍最多也只能到 80cm，而且只要感測到物體，Sensor 就會動作，希望能夠利用更精良的感測器，讓物體及人體的辨識率大為提升。另外外殼體積過大，使得攜帶性大為減少，可以改善電路接線過多的問題，使內部空間可以縮小，使電扇更為精巧。



## 參考文獻

楊仁元、張顯盛、林家德(2008)：專題製作理論與呈現技巧 Office2010 板。新北：台科大圖書股份有限公司。
孫駿榮、吳明展、盧聰勇(2012)：最簡單的互動設計 Arduino 一試就上手(第二版)。台北：基峯資訊股份有限公司。
張議和(2013)。新例說 ALTIUM DESIGNER。台北。新文京開發出版股份有限公司。
Cubie(民 103 年 11 月 01 日)。HC-05 與HC-06 藍牙模組補充說明(一)。民 108 年 8 月 10 日，取自： <a href="https://swf.com.tw/?p=693">https://swf.com.tw/?p=693</a>
SourceForge.SourceForge Mega 2560 Footprint(Sep 27, 2019).From： <a href="https://sourceforge.net/directory/os:windows/?q=altium+mega+2560">https://sourceforge.net/directory/os:windows/?q=altium+mega+2560</a>
潘建宏。L298N馬達驅動模組介紹。民 108 年 10 月 25 日，取自： <a href="http://web.htjh.tp.edu.tw/B4/105-2robot/">http://web.htjh.tp.edu.tw/B4/105-2robot/</a>
Popular mechanics(Apr 4,2016).How To Get Started In 3D Printing.(Dec 15,2019).From： <a href="https://www.popularmechanics.com/technology/gadgets/a19698/get-started-3d-printing/">https://www.popularmechanics.com/technology/gadgets/a19698/get-started-3d-printing/</a>
維基百科。構造實體幾何。民 108 年 12 月 15 日，取自： <a href="https://zh.wikipedia.org/zhtw/%E6%9E%84%E9%80%A0%E5%AE%9E%E4%BD%93%E5%87%A0%E4%BD%95">https://zh.wikipedia.org/zhtw/%E6%9E%84%E9%80%A0%E5%AE%9E%E4%BD%93%E5%87%A0%E4%BD%95</a>
維基百科。直流馬達。民 108 年 12 月 29 日，取自： <a href="https://zh.wikipedia.org/zhtw/%E7%9B%B4%E6%B5%81%E7%94%B5%E5%8A%A8%E6%9C%BA">https://zh.wikipedia.org/zhtw/%E7%9B%B4%E6%B5%81%E7%94%B5%E5%8A%A8%E6%9C%BA</a>
TAIWANIOT。E18-D80NK 紅外線避障感測器。民 108 年 12 月 29 日，取自： <a href="https://www.taiwaniot.com.tw/product/e18d80nk%E7%B4%85%E5%A4%96%E7%B7%9A%E9%81%BF%E9%9A%9C%E6%84%9F%E6%B8%AC%E5%99%A8/">https://www.taiwaniot.com.tw/product/e18d80nk%E7%B4%85%E5%A4%96%E7%B7%9A%E9%81%BF%E9%9A%9C%E6%84%9F%E6%B8%AC%E5%99%A8/</a>
維基百科。伺服馬達。民 108 年 12 月 29 日，取自： <a href="https://zh.wikipedia.org/wiki/%E4%BC%BA%E6%9C%8D%E9%A6%AC%E9%81%94">https://zh.wikipedia.org/wiki/%E4%BC%BA%E6%9C%8D%E9%A6%AC%E9%81%94</a>
普特企業有限公司。MG996R 大扭力金屬標準伺服機。民 108 年 12 月 29 日，取自： <a href="https://www.playrobot.com/20kgf-cm/1150-standard-servo-mg996r.html">https://www.playrobot.com/20kgf-cm/1150-standard-servo-mg996r.html</a>

表 5 參考文獻

## 附錄

### 附錄一 設備清單

類別	設備、軟體名稱	應用說明
硬體	電腦	用於撰寫程式、查詢資料
硬體	雷射切割機	切割外殼架構
硬體	3D 列印機	列印感測器固定架
軟體	Word2013	撰寫計劃書與文字報告
軟體	PowerPoint2013	製作口頭報告需要的簡報
軟體	App inventor	控制及撰寫藍芽模組的應用程式

表 6 設備清單

## 附錄二 材料清單

類別名稱	材料名稱	單位	數量	應用說明	備註
3D 列印耗材		捲		印製電風扇相關的零件	
感應器	E18-D80NK	個	3	人體感測裝置	
馬達	直流馬達	個	1	電風扇用	
伺服馬達	MG996R	個	1	底座平台轉動用	
Arduino	Mega2560	個	1	控制核心	
線材	杜邦線	條	若干	連接控制器與馬達	
顯示器	共陰型七段顯示器	個	2	顯示段數，風數及定時功能	
開關	按鈕開關	個	2	調整段數及功能切換	
馬達模組	L298N	個	1	推動電風扇馬達	
分壓電阻	電阻	個	4	電路分壓用	
藍芽模組	HC-05	個	1	藉由手機遙控電扇	

表 7 材料清單

### 附錄三 程式碼

```
#include <SoftwareSerial.h>          //資料庫
#include <Wire.h>
#include<Servo.h>
#define sensor_left  2              //左邊的 sensor
#define sensor_mid   3
#define sensor_right 4              //右邊的 sensor
#define in_p1 9                // in_p1 in_p2
#define in_p2 6                // 1 0 正轉

void(* resetFunc) (void) = 0;
// 0 1 逆轉
SoftwareSerial BTSerial(12, 13);    //RX,TX 但接板子時 12 接 HC05 的 TX 13 RX
Servo myservo;

boolean mid_trigger = false;        //中間 sensor 偵測
boolean turn_left = false;         //轉右
boolean turn_right = false;        //轉左
boolean turn_right_auto = true;    //自動擺頭
boolean timer_enable = false;      //是否計時
boolean stop_sensor = false;       //是否選擇不自動偵測人 --藍芽
boolean stop_sensor_turnleft = false; //控制左轉
boolean stop_sensor_turnright = false; //控制右轉
boolean auto_turnhead = false;     //自動擺頭
boolean turn_right_appbtn = false; //app 長按擺頭
boolean turn_left_appbtn = false;  //同上

const int seg7[] = {39, 31, 47, 51, 53, 37, 45}; //a -> f
const int seg7_2[] = {36, 30, 46, 50, 52, 38, 44};
const int scan[] = {43, 33, 35, 41}; //左 ->右
const int scan_2[] = {40, 34, 32, 42};

char blue_speed_show[10], blue_time_show[10];
//藍芽資料傳輸速度 //時間
```

```

char TAB[] = { 0x40, 0x79, 0x24, 0x30, 0x19, 0x12, 0x02, 0x78, 0x00, 0x18, 0x7F, 0x47,
0x46, 0x09}; //解碼 0~9 格數 -低態動作
//          0    1    2    3    4    5    6    7    8
9    10   11   12   13
char TAB_2[] = { 0x40, 0x79, 0x24, 0x30, 0x19, 0x12, 0x02, 0x78, 0x00, 0x18, 0x7F,
0x47, 0x46, 0x09, 0x08, 0x41, 0x07}; //解碼 0~9 格數 -低態動作
//          0    1    2    3    4    5    6    7    8
9    10   11   12   13    14(A) 15(U) 16(T)

int Data[] = {10, 10, 10, 10}; //顯示在七段的文字
int Data_2[] = {10, 10, 10, 10}; //同上
int ii, jj = 0; //變數
int time_set = -1; //定時 mode 60m / 120m /240m /480m
int delaytime = 1; //延遲時間 7 段掃描
int btn, btn2 = 0; //按鈕 1 開 2 設定時間 // (1M 跟 2K)
分壓分壓(1k 1k 分)
int deb1, deb2, deb3 = 0; //防彈跳
int fan_mode = 0; // 0 關 1 弱風 2 中風 3 強風 4 自然風
int time_set_number, time_set_number_save = 0; //倒數計時 //儲
存並計算的值
int pos = 0; //風扇角度

float timer = 0; //計時用 millis()
float debounce = 0; //防彈跳
float auto_timer = 0; //自然風計時所需時間
float analogfan_speed = 0; //設定大小風扇速度 ~255 >255 用 255 算
float analogfan_speed_show = 0; //顯示數值 analog 1:X = 99:255 X=2.56

void setup() {
  for (ii = 0; ii < 7; ii++)pinMode(seg7[ii], OUTPUT); //7 段 1 a~g 設腳位 o/p
  for (ii = 0; ii < 7; ii++)pinMode(seg7_2[ii], OUTPUT); //7 段 2 a~g 設腳位 o/p
  for (ii = 0; ii < 4; ii++)pinMode(scan[ii], OUTPUT); //7 段 1 掃描 0~3
  for (ii = 0; ii < 4; ii++)pinMode(scan_2[ii], OUTPUT); //7 段 1 掃描 0~3
  pinMode(sensor_left, INPUT_PULLUP); //左邊的偵測器
  pinMode(sensor_mid, INPUT_PULLUP);

```

```

pinMode(sensor_right, INPUT_PULLUP);
pinMode(in_p1, OUTPUT);
pinMode(in_p2, OUTPUT);
  randomSeed(analogRead(A5));          //亂數種子
myservo.attach(10, 0, 180);           //xx.attach(pin,min,Max)
  myservo.write(0);                   //預設開始為零度
  BTSerial.begin(38400);              //藍芽胞率 hc-50 預設 38400 at 模式改
為 9600 at + uart = 9600,0,0
  Serial.begin(9600);                 //胞率 9600

}

void loop() {

  bluetoothClient();                 //藍芽接收
  seg7btn();                          //七段
  fan_speed();                        //速度快慢
  if (fan_mode != 0 && stop_sensor != true && auto_turnhead != true) {      //
風扇開才開始偵測
    servo();                          //sensor 偵測
  }//(fan_mode != 0 && stop_sensor != true && auto_turnhead != true)
  if (auto_turnhead == true && fan_mode != 0) {      //自動擺頭
    if (turn_right_auto == true) {
      pos += 1;
      if (pos == 180) {
        turn_right_auto = false;
      }// pos == 180
    } else if (turn_right_auto == false) {
      pos -= 1;
      if (pos == 0) {
        turn_right_auto = true;
      }// pos == 0
    }//else if
    Serial.println(pos);
  }
  myservo.write(pos);
  delay(10);
  }//auto_turnhead == true && fan_mode != 0

```

```

if ((turn_left_appbtn == true or turn_right_appbtn == true )&& fan_mode != 0) {
    pos_pluses_app_btn();
}

segscan(); //七段掃描

}

void segscan() {
    for (int q = 0; q <= 3; q++) { //七段掃描 4 位元各做一次
        Seg7Decoder(TAB[Data[q]]); //Data[q] 先取陣列中的數字 ex = 7
        變 --> TAB[7] (轉換) '0xXX' -->丟給 Seg7Decoder[0xXX] (解碼)
        digitalWrite(scan[q], 1); //交替
    }
    delay(1);
    digitalWrite(scan[q], 0);
}

for (int c = 0; c <= 3; c++) { //七段掃描
    Seg7Decoder_2(TAB_2[Data_2[c]]);
    digitalWrite(scan_2[c], 1);
}
delay(1);
digitalWrite(scan_2[c], 0);
}
}

void bluetoothClient() {
    byte cmmd[20];
    int insize; //輸入資料
    if ((insize = (BTSerial.available())) > 0) { //手機藍芽發送訊號 abc
    等
        //Serial.print("input size = ");
        // Serial.println(insize);
        for (int i = 0; i < insize; i++) { //看有幾位元讀幾次值
            cmmd[i] = char(BTSerial.read());
            //Serial.print(); //讀取藍芽的 ASCII 值並儲存
            //Serial.print("\n");
        }
        switch (cmmd[0])
    {
        //swich-case 判斷輸入為何做
    }
}

```

什麼事

```
//讀取第一個字元
case 97: //'a' 加速
    if (fan_mode != 0 && fan_mode != 4) {
        fan_mode = 5;
        analogfan_speed += 2.57; //1 : x =
99 : 255 X = 2.56
        analogfan_speed_show = analogfan_speed / 2.56; //除現在風
速得數值
        dtostrf(analogfan_speed_show, 2, 0, blue_speed_show);
        if ((int)(analogfan_speed_show) >= 100) { //取整數 ;若
超過 99 則
            analogfan_speed = 255; //風速固
定 255(Max)
            analogfan_speed_show = 99; //七段顯
示 99
        }
        //Serial.println(analogfan_speed_show);
        Data_2[2] = (int)analogfan_speed_show / 10; //高位元 ex
98 / 10 = '9' ...8 顯示 9
        Data_2[3] = (int)analogfan_speed_show % 10; //低位元 ex
98 / 10 = 9 ... '8' 顯示 8 (%10 取餘數
        fan_speed_add_chack(); //判斷調
速時狀態(fan_mode)改變
    }

    break;
case 98: //'b' 減速
    //Serial.println("bb");
    if (fan_mode != 0 && fan_mode != 4) { //不在 (還沒
開關&&自然風) 時動作
        fan_mode = 5;
        analogfan_speed -= 2.57;
        analogfan_speed_show = (int)analogfan_speed / 2.56;
        dtostrf(analogfan_speed_show, 2, 0, blue_speed_show);
        if ((int)(analogfan_speed_show) <= 1) {
            analogfan_speed = 2.56;
            analogfan_speed_show = 1;
```



```

    }
    //Serial.println(analogfan_speed_show);
    Data_2[2] = analogfan_speed_show / 10;
    Data_2[3] = (int)analogfan_speed_show % 10;
    if ((int)Data_2[2] == 0) Data_2[2] = 10;
    fan_speed_minus_chack();
}

break;
case 99:/'c' 定時
    //Serial.println("cc");
    if (fan_mode != 0) {
        if (time_set == 3) time_set = -1;
        else {
            time_set++;
        }
        if (time_set == -1) {
Data[3] = 10;
Data[2] = 10;
Data[1] = 10;
            timer_enable = false;
            time_set_number = 0;
            time_set_number_save = 0;
        } else if (time_set == 0) {
            time_set_number = 60;
            timer_enable = true;
        } else if (time_set == 1) {
            time_set_number = 120;
            timer_enable = true;
        } else if (time_set == 2) {
            time_set_number = 240;
            timer_enable = true;
        } else if (time_set == 3) {
            time_set_number = 480;
            timer_enable = true;
        }
        time_set_number_save = time_set_number ;
    }
    dtostrf(time_set_number_save, 2, 0, blue_time_show);

```

```

        for (int e = 1; e <= 3; e++) {
            Data[e] = time_set_number_save % 10;
            time_set_number_save = time_set_number_save / 10;
        }
        if (time_set == -1) {
            Data[3] = 10;                //關起不顯示
Data[2] = 10;
Data[1] = 10;
        } else if (time_set == 0) {
Data[3] = 10;
        }
    }

    break;
case 100:// 'd' 自然風
    Serial.println("dd");
    fan_mode = 4;
    Data_2[0] = 14;                //A
    Data_2[1] = 15;                //U
    Data_2[2] = 16;                //T
    Data_2[3] = 0;                //O
    break;
case 101:// 'd' 四段開關

    judge_mode();                //判斷現在模式 (為加減速)
    if (fan_mode >= 3) {
        fan_mode = 0;
        for (int a = 0; a < 4; a++) Data_2[a] = 10;
    }
    else {
        fan_mode ++;
    }
    Serial.println(fan_mode);
    Data_2[0] = 10 + fan_mode ;
    if (fan_mode == 0) {
        for (int t = 0; t < 4; t++) {
            Data[t] = 10;
            time_set = -1;

```

```

        timer_enable = false;
        time_set_number = 0;
        dtostrf(time_set_number, 2, 0, blue_time_show);
    }
}
break;

case 102:/'f' 偵測器暫停
    stop_sensor = true;
    break;

case 103:/'g' 右轉
    if (fan_mode != 0) {
        stop_sensor_turnright = true;
        if (pos >= 170) {} else {
            pos += 5;
        }
    }
myservo.write(pos);
}
break;
case 104:/'h' 左轉
    if (fan_mode != 0) {
        stop_sensor_turnleft = true;
        if (pos <= 0) {} else {
            pos -= 5;
        }
    }
myservo.write(pos);
}
break;

case 105:/'i' 偵測器啟動
    stop_sensor = false;
    Serial.println(stop_sensor);
    break;

case 106:/'j' arduino 重啟
resetFunc();
break;

```

```

case 107:/'k' 自動擺頭
    auto_turnhead = true;
    break;

case 108:/'l' 停止自動擺頭
    auto_turnhead = false;
    break;
case 109:/'m' 長按持續擺頭左
    turn_left_appbtn = true;
    pos_pluses_app_btn();
    break;

case 110:/'n' 長按持續擺頭右
    turn_right_appbtn = true;
    pos_pluses_app_btn();
    break;

case 111:/'o' 鬆開取消持續擺頭左
    turn_left_appbtn = false;
    break;

case 112:/'p' 鬆開取消持續擺頭右
    turn_right_appbtn = false;

    break;

case 122:
    if (fan_mode == 0) {
        BTSerial.write("close ");
        BTSerial.write(" ");
        BTSerial.write(" ");
    } else if (fan_mode == 1) {
        BTSerial.write("low ");
        blue_speed_show_for();
        blue_time_show_for();
    } else if (fan_mode == 2) {
        BTSerial.write("mid ");

```

```

        blue_speed_show_for();
        blue_time_show_for();
    } else if (fan_mode == 3) {
        BTSerial.write("high ");
        blue_speed_show_for();
        blue_time_show_for();
    } else if (fan_mode == 4) {
        BTSerial.write("nature");
        blue_speed_show_for();
        blue_time_show_for();
    } else if (fan_mode == 5) {
        BTSerial.write("adjust");
        blue_speed_show_for();
        blue_time_show_for();
    }

    break;

} //Switch
}
}
void blue_speed_show_for() {
    for (int p ; p < 2 ; p++) {
        BTSerial.write(blue_speed_show[p]);
    }
}

void blue_time_show_for() {
    for (int c ; c < 3 ; c++) {
        BTSerial.write(blue_time_show[c]);
    }
}

void pos_pluses() { //轉動
    if (pos < 180 && turn_right == true) { //右轉
        pos += 1;
    } else if (pos > 0 && turn_left == true) { //左轉
        pos -= 1;
    }
}

```

```

    }
}

//中間偵測到後停止
void servo_stop() {
    if (digitalRead(sensor_mid) == 0) {
        mid_trigger = true;
        turn_left   = false;
        turn_right  = false;
        //    if (digitalRead(sensor_mid) == 1) break;
    } //if (digitalRead(sensor_mid) == 0)
} //void servo_stop()

//七段顯示風速時間
void seg7btn() {
    btn = analogRead(A0);
    btn2 = analogRead(A1);
    if (millis() - timer > 60000 && timer_enable == true) { //倒計時
        timer = millis(); //儲
    }
    存時間
    time_set_number -= 1;
    time_set_number_save = time_set_number; //
    轉為運算數值
    dtostrf(time_set_number, 2, 0, blue_time_show);
    for (int j = 1; j < 4 ; j++) { //三位數
        Data[j] = time_set_number_save % 10;
        time_set_number_save = time_set_number_save / 10;
        if (Data[3] == 0) Data[3] = 10; //遮莫
        if (Data[3] == 10 && Data[2] == 0) Data[2] = 10;
        if (Data[3] == 10 && Data[1] == 0 && Data[2] == 10) {
            Data[1] = 10;
            fan_mode = 0;
            for (int t = 0; t < 4; t++) {
                Data[t] = 10;
                time_set = -1;
                timer_enable = false;
            }
        }
    }
}

```

```

    }
  }
}
if (btn2 < 205 && btn2 > 5 && fan_mode > 0) { //外部按钮功能
//205 //5
  deb2 += 1;
  if (deb2 > 3) deb2 = 2;
  if (deb2 == 1) { //debounce
    if (time_set == 3) time_set = -1;
    else {
      time_set++;
    }
    if (time_set == -1) {
Data[3] = 10;
Data[2] = 10;
Data[1] = 10;
      timer_enable = false;
      time_set_number = 0;
      time_set_number_save = 0;
    } else if (time_set == 0) {
      time_set_number = 60;
      timer_enable = true;
    } else if (time_set == 1) {
      time_set_number = 120;
      timer_enable = true;
    } else if (time_set == 2) {
      time_set_number = 240;
      timer_enable = true;
    } else if (time_set == 3) {
      time_set_number = 480;
      timer_enable = true;
    }
    time_set_number_save = time_set_number ;

    for (int e = 1; e <= 3; e++) {
      Data[e] = time_set_number_save % 10;
      time_set_number_save = time_set_number_save / 10;

```

```

    }
    if (time_set == -1) {
Data[3] = 10;
Data[2] = 10;
Data[1] = 10;
    } else if (time_set == 0) {
Data[3] = 10;
    }
    }
} else {
    deb2 = 0;
}
if (btn < 650 && btn > 450) { //四段開關 (外 //650
//450
    deb1 += 1;
    if (deb1 > 3) deb1 = 2;
    if (deb1 == 1) {
        judge_mode();
        if (fan_mode >= 3) {
            fan_mode = 0;
            for (int a = 0; a < 4; a++) Data_2[a] = 10;
        }
        else {
            //Serial.println("按鈕錯");
            fan_mode ++;
        }
        Data_2[0] = 10 + fan_mode ;
        if (fan_mode == 0) {
            for (int t = 0; t < 4; t++) {
                Data[t] = 10;
                time_set = -1;
                timer_enable = false;
            }
        }
    }
} else {
    deb1 = 0;
}
}

```



```

}

void Seg7Decoder(byte dat) {                                //七段解碼 gfedcba
    for (jj = 0; jj < 7; jj++) {
        digitalWrite(seg7[jj], dat % 2);                //取 d2 之餘數 1 滅 0 亮
        dat = dat / 2;                                  //下一位元處理
    }
}

void Seg7Decoder_2(byte dat) {                              //七段解碼 gfedcba
    for (int uu = 0; uu < 7; uu++) {
        digitalWrite(seg7_2[uu], dat % 2);            //取 d2 之餘數 1 滅 0 亮
        dat = dat / 2;                                  //下一位元處理
    }
}

void fan_speed() {                                         //風速

digitalWrite(in_p2, 0);
    if (fan_mode == 0) {
digitalWrite(in_p1, 0);
        analogfan_speed = 0;
    } else if (fan_mode == 1) {
        analogfan_speed = 100;
        analogfan_speed_show = (int)(analogfan_speed / 2.56 );
        dtostrf(analogfan_speed_show, 2, 0, blue_speed_show);

    } else if (fan_mode == 2) {
        analogfan_speed = 200;
        analogfan_speed_show = (int)(analogfan_speed / 2.56 );
        dtostrf(analogfan_speed_show, 2, 0, blue_speed_show);

    } else if (fan_mode == 3) {
        analogfan_speed = 255;
        analogfan_speed_show = (int)((analogfan_speed - 1) / 2.56 );
        dtostrf(analogfan_speed_show, 2, 0, blue_speed_show);
}
}

```

```

    } else if (fan_mode == 4) {
        fan_speed_AUTO();           //自然風
    }
analogWrite(in_p1, analogfan_speed);

    if (fan_mode != 4 && fan_mode != 0) {           //
        Data_2[2] = analogfan_speed_show / 10;

        Data_2[3] = (int)(analogfan_speed_show) % 10;
    }
}

void fan_speed_AUTO() {
    long auto_speed = random(0, 255);    //最大 255 風速
    long auto_time = random(10000, 60000); //不定時吹風

    if (millis() - auto_timer > auto_time ) {
        if (fan_mode = 4) {
            auto_timer = millis();
            analogfan_speed = auto_speed;
dtostrf(map(analogfan_speed, 0, 255, 0, 99), 2, 0, blue_speed_show);
            Serial.println(analogfan_speed);
        } else {
            for (int a = 0; a <4 ; a++) Data_2[a] = 0;
        }
    }
} //millis
} //void fan_speed_AUTO

void servo()
{
    斷往左或往右轉
    if (digitalRead(sensor_right) == 0 && mid_trigger == false) {
        turn_left = false;
        turn_right = true;

    } else if (digitalRead(sensor_left) == 0 && mid_trigger == false) {
        turn_right = false;
    }
} //判

```

```

    turn_left = true;
} else if (digitalRead(sensor_right) == 0 && mid_trigger == true) {
    turn_right = true;
    turn_left = false;
    mid_trigger = false;
    //delay(10);
} else if (digitalRead(sensor_left) == 0 && mid_trigger == true) {
    turn_left = true;
    turn_right = false;
    mid_trigger = false;
    //delay(10);
}
servo_stop();
pos_pluses();
myservo.write(pos);
delay(10);
}

```

//判斷+-風速目前在 mode 幾

```

void judge_mode() {
    if (fan_mode == 5 && (int)analogfan_speed_show <= 39 &&
(int)analogfan_speed_show >= 1) {
        fan_mode = 0;
    } else if (fan_mode == 5 && ((int)analogfan_speed_show <= 78 &&
(int)analogfan_speed_show > 39)) {
        fan_mode = 1;
    } else if (fan_mode == 5 && ((int)analogfan_speed_show <= 99 &&
(int)analogfan_speed_show > 78)) {
        fan_mode = 2;
    }
}
}

```

```

void fan_speed_add_chack() { //加速時若速度大於臨界值
59 89 更改風扇模式 L39 -> C78 -> H99
/* 取低 39 中 78 高 99 之中間值得 59 89*/
if ((int)analogfan_speed_show <= 39 ) {

```

```

    Data_2[0] = 11;
} else if ((int)analogfan_speed_show <= 59 && (int)analogfan_speed_show >=
59 ) {
    Data_2[0] = 12;
} else if ((int)analogfan_speed_show <= 89 && (int)analogfan_speed_show >= 89 )
{
    Data_2[0] = 13;
}
}

```

```

void fan_speed_minus_chack() { //減速時若速度大於臨界
值 59 89 更改風扇模式 L39 -> C78 -> H99
/* 取低 39 中 78 高 99 之中間值得 59 89*/
if ((int)analogfan_speed_show <= 39 ) {
    Data_2[0] = 11;
} else if ((int)analogfan_speed_show <= 59 && (int)analogfan_speed_show >=
59 ) {
    Data_2[0] = 11;
} else if ((int)analogfan_speed_show <= 89 && (int)analogfan_speed_show >= 89 )
{
    Data_2[0] = 12;
}
}
}

```

```

void pos_pluses_app_btn() { //轉動
if (pos < 180 && turn_right_appbtn == true) {
    pos += 1;
} else if (pos > 0 && turn_left_appbtn == true) {
    pos -= 1;
}
myservo.write(pos);
delay(10);
}

```