

第十七屆旺宏科學獎

創意說明書

參賽編號：SA17-268

作品名稱：自製無人機監測感測器模組之應用

姓名：李映漾

關鍵字：無人機、溫溼度、PM2.5

目 次

摘要.....	1
壹、研究動機.....	2
貳、研究目的.....	2
參、研究設備及器材.....	2
肆、研究過程與方法.....	3
伍、結果與討論.....	15
陸、結論.....	28
柒、引注資料.....	29

摘要

本研究分成兩部分，研究一利用無人機可自主起降並執行任務，高度機動性的特點，自行發展一套即時監測系統，選擇市面上便宜且成熟的飛行載具，找出簡易的 PID 調整模式，使無人機在不同場域或飛一段時間後，仍能穩定飛行。再搭配各種感測器如溫溼度，MQ3 酒精感測器、PM2.5 空氣懸浮粒子偵測…等，經重量評估後測試運行，接著進行重心計算與體積修正，自製零件基座固定，再撰寫程式，將感測器資料即時透過無線訊號傳輸到控制端螢幕，除了能遠端即時監測數據外，並能降低零件成本，提高模組操作上的彈性。

在研究二中，首先透過撰寫程式顯示低電量安全性警告，解決無人機電力不足導致墜機的情況，降低使用者操作時災害發生風險。設計實驗，利用迴歸分析找出電池消耗的主因為重量。並透過實測探討，飛行中機翼旋轉產生空氣流動不會造成數據失真。最後在軟體方面，優化程式可同時傳遞不同感測器的資料，提高使用者操作效率與修改的便利性。

實驗顯示本研究自製的無人機的即時監測系統，能精準且迅速的取得各種監測數據，對於高空、災害現場或人員難以抵達等狀況將有助益。此外，本系統具備高機動性、改裝彈性與容易移植的優點，若將來運用無人機做為移動式監測站，將能有效補足現行定點監測站及人員稽核的不足，在未來應用更是無可限量。

壹、 研究動機

無人機越來越普及，加上相關零件取得容易，價格降低，應用情境也越來越多，從最開始的高空攝影和錄影，到農場農作物檢測與輔助耕作，再到後來如無人機快遞、機場驅趕鳥類，室內救援、輸送人道主義物資等，也越來越多人研究如何飛得更穩、更高、更快。

市面上已有的多軸遠端物聯網應用機，最便宜的售價都高於三萬五千元，在相關論壇也發現各家的飛控板為了減輕重量，都將體積盡量縮小，但還是有很多接腳用不到，我們想要開發一種模式，利用這些開放的接腳與各種傳感器連接，進行數據交換或控制，來作生活中實務上的應用，例如火災現場溫度監測，高空的PM2.5監測，下水道瓦斯監測....等，發現可能遇到的問題，嘗試用來保護相關工作人員的安全。

貳、 研究目的

根據上述研究動機，經討論後訂定以下研究目的：

- 一、 設計價格相對便宜的小型無人機監測模式，相關零件以取得容易為主。
- 二、 測試各項零件並規畫彈性組裝模式，並避免線路互相干擾影響重心。
- 三、 研究在不同場域中都能實作的監測系統，並分析各模組可能碰到之困難與解決。
- 四、 優化系統飛行狀態與安全性，並探討可行之應用模式。

參、 研究設備及器材

表 1 研究設備及器材

名稱	規格	數量	備註
四軸飛行器(自組)	自行購置零件	1	飛控板 Apm2.8
四軸飛行器(現成)	各家	3	國內各廠牌
Mission planner 飛行軟體	1.3.9	1	免費
Arduino 微控制器	UNO v3.0	3	
電子天平	電子式	1	
OLED	0.96 吋，I2C 介面	3	
3D 印表機	SKYMAKER R1	1	
溫溼度監測	DHT11	3	

酒精監測	MQ-3 感測	3	可測汽油，甲烷、苯等
PM2.5 懸浮微粒監測	GP2Y1014AU	3	

肆、研究過程及方法

研究一

挑戰利用無人機發展出一套不受時空限制的即時監測系統，首先選擇適合搭載感測器的無人機載具，嘗試設計實驗一套簡易的 PID 調整模式，讓無人機能在各種環境下能穩定飛行，再經測試無人機最大承重後，擬定實驗用感測器並分別實施前測，透過體積計算設計元件基座，最後撰寫程式整合功能模組。

一、飛行器選擇及設計：

在飛行器的選擇上，我們先參考相關網路文獻以及市面上已經上市產品，分別購買並將其實作並評估：

表 2 本研究嘗試之各種飛行器比較表

比較項目 \ 品名	自行購買組裝	奶油蒼蠅 2.0	LDSSLITE250 無人機
價錢	6000~8000	2350	4750
飛控板	ArduPilot Mega 2.0	Arduino Nano	Arduino Pro Micro
控制器	Atmega2560	ATmega328	ATmega32u4
飛控軟體	APM Planner	arduino	arduino
傳輸通訊協定	Wifi, zbee, 藍芽...	藍芽	nRF24L01
飛行控制距離	500~1000m 或以上	10m~15m	180m~240m
操作組裝圖示			

在實際組裝試過程中，先考慮自行組裝，評估的結果發現技術難度與零件整合很難推廣到一般人可以使用。由於本研究旨在利用四軸飛行器傳遞資料而非製作飛行器，所以轉而考慮市面上現有產品，其都具備操作簡單，零件購買容易，更重要的是可以自己撰寫程

式，實際操作後並與本研究所需做一需求評估，最後選擇以 nRF24L01 (2.4G)通訊協定的無人機產品來做研究。

二、系統發展規劃：

經過腦力激盪，我們初步規劃之研究流程如下，並一步一步的完成前端感測測試、行動傳遞與資料呈現，不斷修改調整，直到能正常運作：

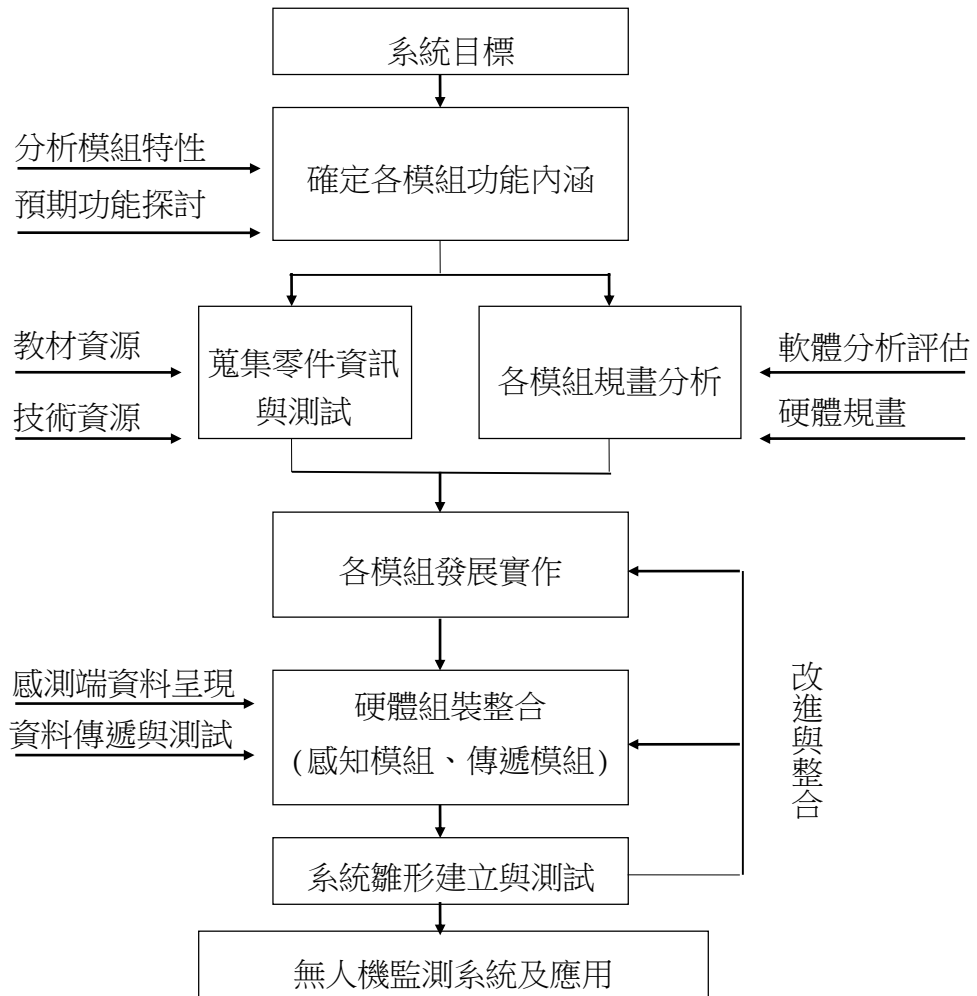


圖 1、本遠端監測系統之建構流程圖

三、四軸飛行器的預期功能，並找出系統在不同情況下的最佳 PID 值：

由於飛行本身原理是螺旋槳與空氣發生相對運動，產生了向上的升力，當升力大於四軸的重力時四軸就可以起飛了，實際上由於電機和螺旋槳本身的差異，造成我們無法做到四個電機產生相同的升力，這也造成很多人飛行小四軸幾次之後，會發現越來越難控制或者不靈敏，就不想再飛了。

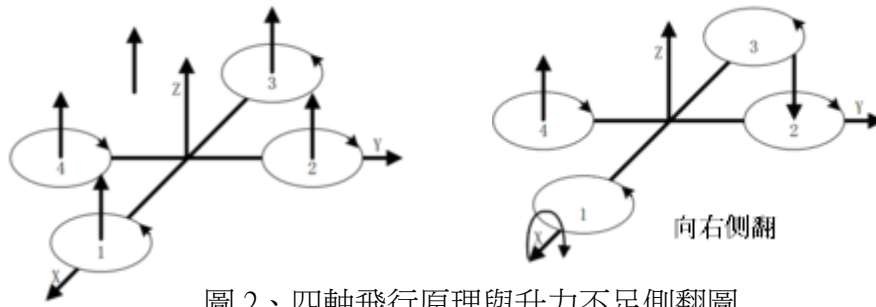


圖 2、四軸飛行原理與升力不足側翻圖

本組預期系統只需控制四軸飛行器的方向和高度，這時就需要一個反饋系統來自動完成飛機的穩定性，於是就須對 PID 控制系統研究修改：

- 比例(P)控制器：其控制器的輸出與輸入誤差信號成比例關係。當僅有比例控制時系統輸出存在穩態誤差。
- 積分(I)控制器：控制器的輸出與輸入誤差信號的積分成正比。一個自動控制系統如果在進入穩態後存在穩態誤差，則稱這個控制系統是有穩態誤差（System with Steady-state Error）。為了消除穩態誤差，控制器中必須引入「積分項」。因此，比例+積分(PI)控制器，可使系統在進入穩態後無穩態誤差。
- 微分(D)控制器：微分調節就是偏差值的變化率。能夠實現系統的超前控制。如果輸入偏差值線性變化，則在調節器輸出側疊加一個恆定的調節量。

綜合來說：PID 是以它的三種糾正算法而命名。受控變數是三種算法（比例、積分、微分）相加後的結果，即為其輸出，其輸入為誤差值（設定值減去測量值後的結果）或是由誤差值衍生的信號。若定義 $u(t)$ 為控制輸出，PID 算法可以用下式表示：

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

其中 k_p ：比例增益 k_i ：積分增益 k_d ：微分增益 t ：目前時間

e ：誤差=設定值(SP)- 回授值(PV) τ ：積分變數，數值從 0 到目前時間 t

本組希望透過以下兩個實驗設計找出系統在因應不同情況時，調整 PID 中各數值，測試其對飛行狀態的影響，以方便一般使用者未來應用。

(一) 調整 PID 值，測試無人機達平衡所需的時間的差異

將無人機以棉繩固定在四根鐵柱上使其懸空，並在其中一邊的兩根鐵柱上取適當距離再綁上一條棉繩，使下壓無人機時可以固定傾斜 50° ，先啟動無人機使棉繩繃緊，呈穩定

狀態就不再催動遙控器上的馬達，接著將其中一側向下壓致傾斜基準線，放開並同時計時其恢復平衡所需時間。先測試無人機廠商設定 PID 初始值，再分別調整 P、I、D 值與其中的 Yaw、Pitch、Roll，增加各自初始值之十分之一，分別以碼表計時自達平衡所需的時間，每組實驗重複進行五次紀錄測得秒數再取其平均值，觀察數據的差異並分析實驗結果。

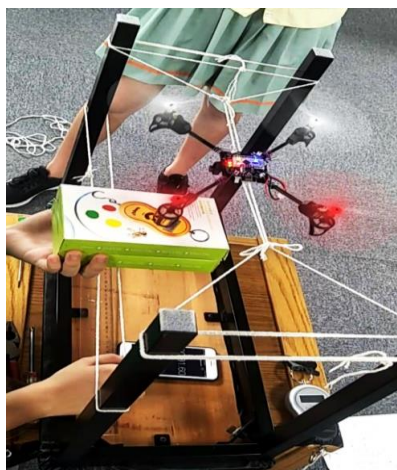


圖 3、遙控無人機呈平衡狀態



圖 4、下壓傾斜五十度



圖 5、紀錄電子秤上的數據

(二)調整 PID 值，測試其飛行時垂直與水平方向上拉力的差異

將棉繩一端綁在無人機的重心上，另一端則綁在電子彈簧秤上，觀測數據為無人機垂直方向的升力與水平方向的拉力，將電子彈簧秤的分別固定在垂直方向與水平方向，遙控無人機使其拉緊棉繩，分別調整 P、I、D 值與其中的 Yaw、Pitch、Roll，增加各自初始值之十分之一，以錄影機紀錄後找出最大值，每組實驗重複進行三次紀錄測得秒數再取其平均值，觀察數據的差異並分析實驗結果。

四、擬定感測器功能模組

接下來找出系統最大承重，我們認為扣除機身與電池外，應該包含：

$$\text{承載總重量} = \text{感測器重量} + \text{固定(或黏著)物重量} + \text{電線重量}$$

我們利用電子天平、砝碼、以及紙黏土，利用兩台飛機交互測試，以飛機能穩定上升到兩公尺高度，逐步增加其重量，測量其最大值，結果發現超過 95 公克之後，飛行的操控就稍微不好控制，但本組就先設定以 95 公克為承載總重量的最大值

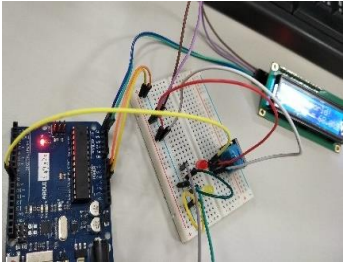
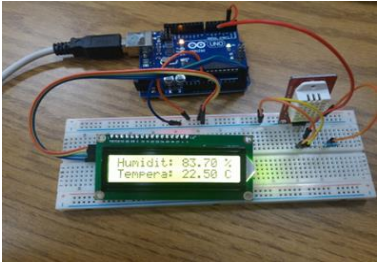


圖 6、利用砝碼，紙黏土找出最大載重測試圖(最右圖為 95 公克)

市面上感測器琳瑯滿目，但由於四軸飛行器體積限制，不適合太大的感測器，在實用性的考量下本組訂定溫溼度監測、環境氣體偵測(酒精)、空氣懸浮粒子偵測(pm2.5)等感測器作為本次研究的觀測項目，以下分析一些模組特性與原理及預期功能探討：

(一) 溫溼度感應器：溫濕度傳感器是含有已校準數字信號輸出的溫濕度復合傳感器。市面上常見有 DHT11，DHT22 等，其規格比較如下表 3，由於本建構模式對於相關精準度與靈敏度需求不大，所以選擇 DHT11 作為本實驗器材。

表 3、DHT11，DHT22 規格比較表

比較	DHT11	DHT22
		
測量溫度範圍	0°C ~ 50°C	-40°C ~ 80°C
測量濕度範圍	20%RH ~ 90%RH	0%RH ~ 99.9%RH
溫度測量精度	±2°C	±1°C
濕度測量精度	±5%RH	±2%RH

(二) 環境氣體偵測(酒精)：MQ3 酒精傳感器用於氣體洩漏檢測（家庭和工業），可以檢測酒精，苯、甲烷、己烷、液化石油氣、一氧化碳，且對其感測速度與回應非常快速。所使用的氣敏材料是電導率較低的二氧化錫 (SnO₂)。當感測器所處環境中存在酒精蒸

汽時，其電導率隨空氣中酒精氣體濃度的增加而增大。

除此之外 MQ-3 對苯的感測靈敏度也很高，而在生活中苯是屬於化工原料或裝潢材料，揮發後也很容易造成空氣汙染，所以系統也可以具備相關偵測功能。



圖 7、MQ3 酒精測試器顯示

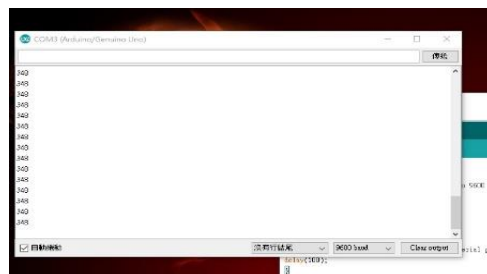


圖 8、MQ3 感測資料前測

(三) 空氣懸浮粒子偵測(pm2.5)：我們使用夏普 SHARP 灰塵感測器 GP2Y1014AU 用來感測空氣中的塵埃粒子，其內部對角安裝紅外線發光二極體和光電晶體，能夠探測空氣中塵埃反射光，靈敏度高，通常應用在空氣淨化系統中。

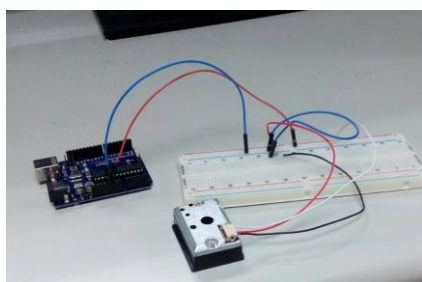


圖 9、空氣懸浮粒子偵測實體圖

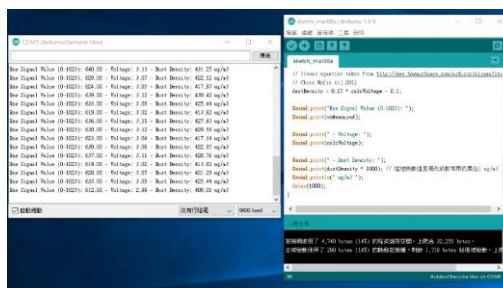


圖 10、空氣懸浮粒子偵測結果顯示

在 Serial 視窗中 Raw Signal Value 原始信號值，Voltage 是電壓，Dust Density 表示粉塵密度，最後一項 UG/M3 表示微克/立方公尺，用來表示空氣中懸浮粒子含量。

(四) 資料通訊 2.4G(nRF24L01 無線收發器模組)：序列周邊介面 (SERIAL PERIPHERAL INTERFACE BUS)是一種用於短程通訊的同步串行通訊介面規範，使用全雙工模式通訊，包含一個 MASTER 和一個或多個 SLAVE。常見的模組有 8 針與 10 針兩種，差別在於 10 針的模組多了一組電源和接地腳，實際的腳位可能會因製造商而異。

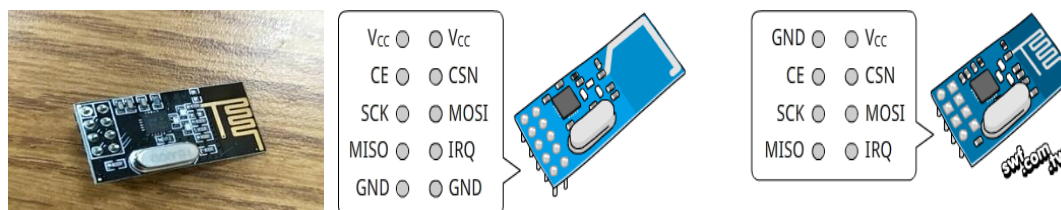


圖 11、nRF24L01 無線收發器模組實體及接腳圖

本研究所用 nrf24l01 晶片使用 2.4g 的頻道，可以設定 channel 與傳輸速度，以降低干擾的發生，下圖為利用 NRF2401 傳送文字陣列測試情形。



圖 12、NRF2401 連線測試圖

(五) 飛控主板(Arduino Pro Micro)：本研究所使用的飛控主板暨相關程式可以於 arduino 環境下編譯，板子選擇 Leonardo /GenuinoMicro 即可。其接腳定義如下圖：

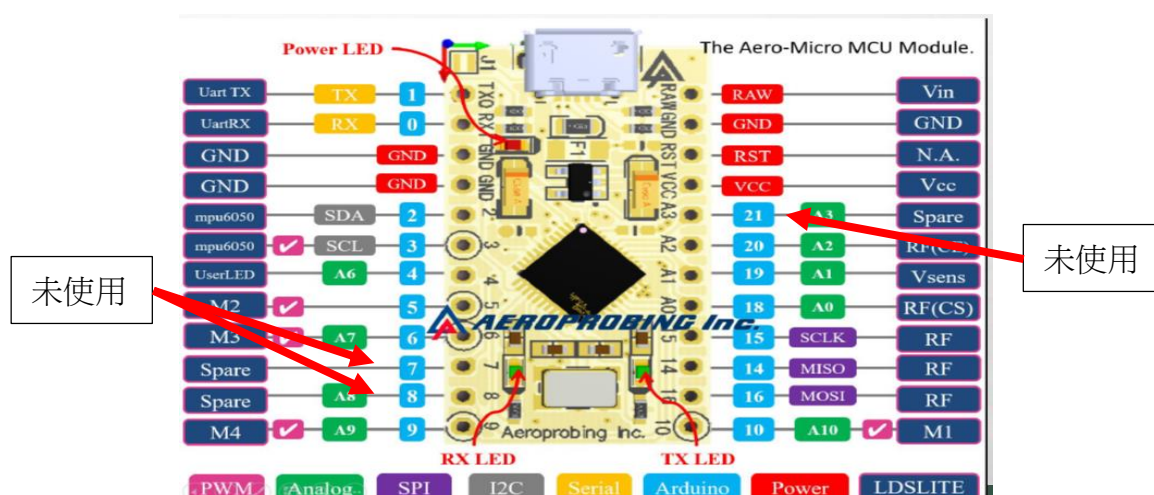


圖 13、飛控主板的接腳圖

根據上課公開文件資料，飛控主板控制器是 ATmega32u4，工作頻率：16 MHz，工作電壓：5V，數位 I/O 接腳：20 個，硬體 PWM 通道：5 個，類比 I/O：9 個。

根據上圖，接腳 7、8、21 在其系統中並未使用的，在感測器重量未超重的以及電壓允許的前提下，我們嘗試將感測器線路接到上述接腳，理論上其最多同時可以外接 3 組感測器。

(六) 設計元件基座：如何將感測器固定在無人機上，而且不能影響其飛行重心，就必須要考慮到現有位置的配置，我們分別以 3D 列印軟體設計元件基座如圖：

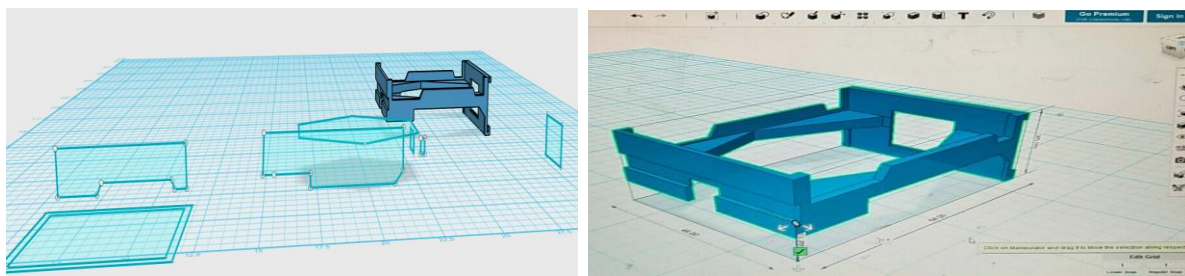


圖 14、以 123D 所設計的第一代元件基座設計草圖

在本次實驗中，由於空氣懸浮粒子偵測體積最大，所以本組配合其造型，需先固定空氣懸浮粒子偵測器一邊的嵌入固定方式。而其它 dht11、環境氣體偵測可以用鎖螺絲等固定方式固定，我們先行量測元件重量如下圖：



圖 15、元件基座與各感測器重量

由上圖得知即使所有元件相加，總重量約 42.3 公克，完全在無人機負重範圍內 (95 公克)。而若只是搭載 1 個感測器，連同基座(18 克)總重量甚至不到 30 克。都在系統可接受範圍，在空間配置上，由於 nRF24L01 無線收發器模組已經占據固定一側，選定相鄰的一邊固定空氣懸浮粒子偵測器，其他方向則各取中點位置設計孔洞位置。

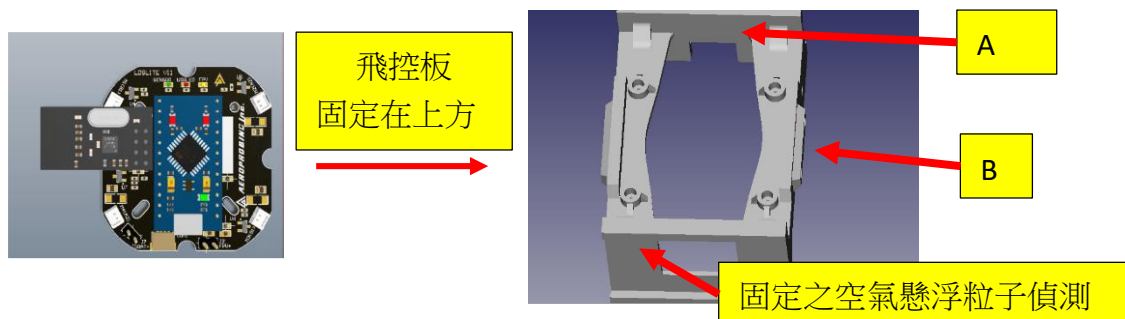


圖 16、各感測器初步配置圖

在位置安排上，由於小型飛機並不像昂貴機種，可透過遙控器人工補償重心偏移，而且元件重量相對很輕，為達重量平衡，我們將透過體積計算修正重心位置。

此外，我們設定 A 安裝位置就是與空氣懸浮粒子偵測器做平衡。B 安裝位置與 nRF24L01 來做平衡。依此類推，若我們僅需安裝一種感測器，則僅需裝在 B 處即可。經本組多次試飛結果，在方向操控與平衡控制上並無問題。

(七)數據資料呈現：本研究對於個別元件先在 arduino 上面先行測試，並將感測資料顯示在 led 上後，透過無線傳輸，將相關資料傳遞到遙控器端 oled 上即時顯示。最後，嘗試運用 visual C++撰寫相關程式，將資料擷取出來到電腦以供未來運用。

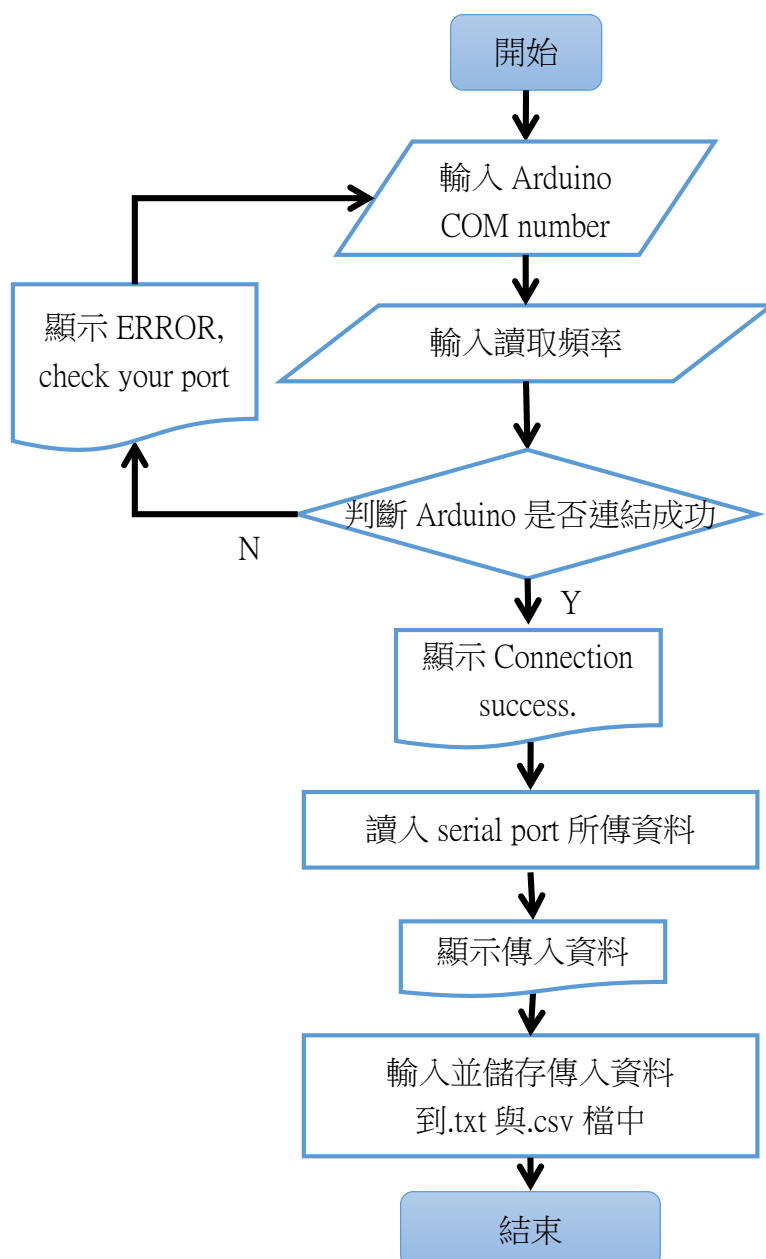


圖 17、COM 埠資料讀取程式撰寫流程規劃

透過流程規劃，希望所撰寫的程式可以在不需 Arduino 的情況下，讀取 com 埠所接收的資料，產生.txt 與.csv 相關檔案，以供後續研究使用。

研究二

根據研究一結果，經多次實際操作及與學者專家討論後，首先須解決無人機電力不足時會操控困難、甚至墜機，所以分析重量或感測器的電力消耗何者為電池消耗的主因，嘗試進行低電量安全性警告；接下來設計實驗，探討飛行中機翼旋轉產生空氣流動，是否造成測試數據失真；最後在軟體功能方面，優化程式，整合各感測器於單一程式中，提高使用者修改的便利性，並測試傳遞不同感測器資料的可行性。

一、低電量警告與顯示：

一旦無人機墜機，輕則摔個粉身碎骨，重則可能會危及地面的人身安全，操作無人機監測模組的過程中，也因電力不足有造成墜機情形，因此想透過撰寫程式讓電池電量資訊即時回傳到控制端，並顯示低電量警告，使用者即可隨時注意電量資訊避免意外發生。



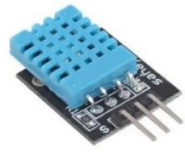


圖 18、電力資訊警告程式撰寫

依據上課圖 21，接腳電壓值 = $V_{BAT} * 1k / (10k + 1k)$ ，將其值逆推回去即可得到電池電壓，再透過 MCU 的 ADC 模組將類比的電壓值對應轉換成 10 位元的數位資料 0~1023 之間，理論上應該可以得到電壓值。此外，還需考慮除了傳遞電壓值外，感測器值也需同時傳遞。

二、找出無人機搭載感測器消耗電量的主因：

搭載不同感測器後，發現無人機模組飛行時間變短，到底其變短的原因主要為何：透過實驗記錄飛行時間長短，並輔以其規格表作回歸分析，以找出感測器耗電的原因。

表 4、各感測器與馬達規格表

	溫溼度監測	酒精監測	PM2.5 懸浮微粒監測	有刷直流馬達
				
感測器名稱	DHT11	MQ-3	GP2Y1014AU	8250 空心杯
消耗電流	0.5 mA	1.5 mA	11mA	200~400 mA
重量	2.6 g	6.5 g	15.2 g	

(一) 測試並記錄不同感測器飛行時間：

選取兩組全新電池，以遙控器控制無人機固定飛行高度與範圍為 210cm * 160cm * 150 cm (如下圖 21)，每組分別測試兩次記錄飛行時間後取算術平均數。每組飛完之後電池需重新充飽電後再行測試。



圖 19、固定無人機飛行高度與範圍搭載 MQ3、DHT11 感測器，紀錄飛行時間

(二) 分析感測器重量、電流量與飛行時間的關係：

根據規格表，觀察感測器消耗電流與飛行時間的關係，此外，由於馬達的消耗電流約是感測器的 36~400 倍，我們假設感測器重量會造成馬達負擔，也因此需要更大流量。因此，根據感測器重量與電流量，同時與飛行時間做數據分析。

三、 探討機翼旋轉是否造成數據失真與解決方法

夏天吹電風扇時令人感覺到涼爽，聯想到自製無人機監測模組是否也因機翼高速旋轉造成的空氣流動，使感測器數據失真，以溫溼度感應器為例設計實驗，分別在室外及室內各取 5 處測試，觀察在大範圍空間中與小範圍的點熱源數據是否受機翼旋轉影響。

- (一) 機翼旋轉是否造成室內空間、室外 30 公尺高度數據失真: 在室內實驗時將無人機用手機夾固定住讓溫溼度感測器資料回傳並記錄，觀察其與標準溫度計的誤差。在室外開放環境則將飛機飛到離地面高度約 35 公尺，距本校頂樓陽台約 1~3 公尺處，觀察其與頂樓陽台處標準溫度計的誤差。



圖 20、室內環境與室外開放環境探討機翼旋轉是否造成數據失真

- (二) 距離溫度梯度：以電暖氣設備製造出點熱源，將無人機以手機夾固定，再置於在距離熱源 30、50、100、150、200 公分處以無人機監測模組作測試並記錄。並與標準溫度計作比較，觀察其誤差情形，利用距離梯度來估算點熱源溫度

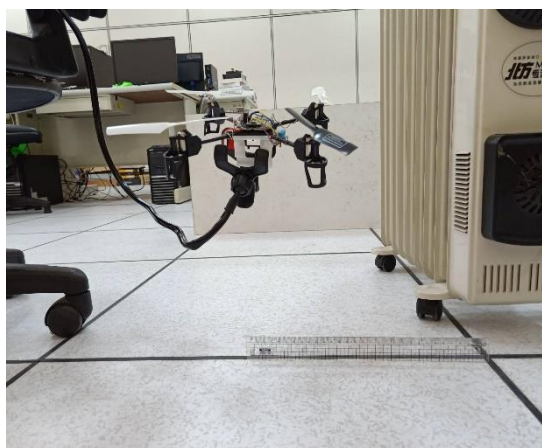


圖 21、距離熱源 30、100 公分處測量溫度估算點熱源實驗圖

四、測試傳遞多筆資料的可行性：

當個別感測器皆能經由無線通訊模組回傳數據，想嘗試即時回傳不只一筆資料，除修改相關程式外，並根據前述相關實驗結果研究其限制。

伍、研究結果與討論

研究一

一、 系統功能圖：經過前面系統發展規劃，以及不斷嘗試後，本組所發展之架構如下：

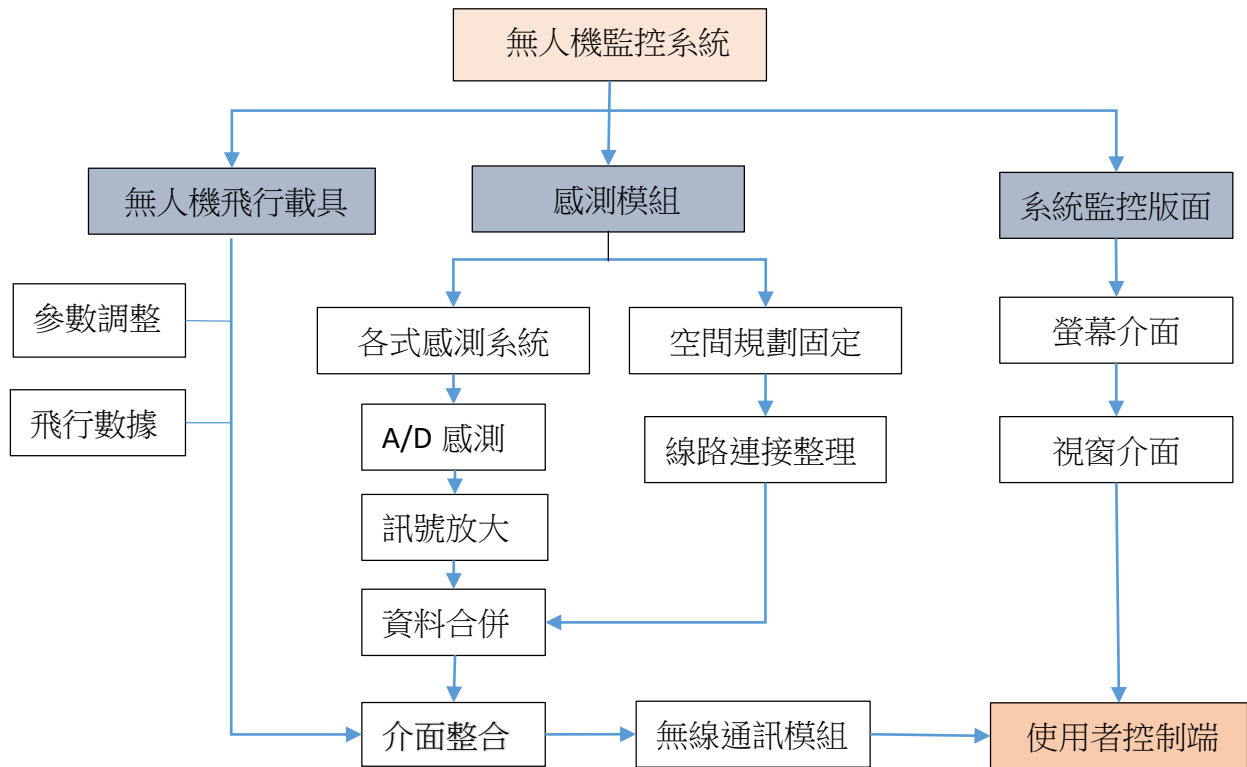


圖 22、無人機監測系統架構圖

如上圖本系統可分為三大部分:無人飛行載具、監測系統、整合監控介面。系統須整合飛行載具的控制、感測模組的各項資訊蒐集、監測資訊即時呈現等功能。為求控制成本、重量與避免干擾，所有數據皆透過相同頻段的無線傳輸方式回傳至使用者介面進行資料整合。

二、 無人機飛行載具控制調整：

PID 調整： PID 的調整是一種不斷隨機器與環境改變的過程，一般的使用者需要的是在短時間內回到平穩的飛行狀態，因此我們藉由以下兩個實驗，調整 $K_{pout}[3]$ 、 $K_{iout}[3]$ 、 $K_{dout}[3]$ 等值，歸納一種簡易的 PID 調整模式。系統初始值如下表：

表 5：程式內部 PID 預設值

	Yaw	Pitch	Roll
P 比例控制器	0.2	1.0	1.25
I 積分控制器	0.001	0.02	0.009
D 微分控制器	0.19	0.03	0.025

(一) 調整 PID 值，測試無人機達平衡所需的時間的差異實驗:

先測試無人機廠商設定 PID 初始值，再分別調整 P、I、D 值與其中的 Yaw、Pitch、Roll，增加各自初始值之十分之一，分別以碼表計時自達平衡所需的秒數，每組實驗重複進行五次紀錄測得秒數再取其平均值(如表 5)，分析實驗結果發現:

(一) 同時增加 Yaw、Pitch、Roll 數值:比例(P)控制器和積分(I)控制器能縮短系統達平衡所需反應時間，積分(D)控制器值則增加系統達平衡所需反應時間。

(二) 單獨增加 PID 值中 Yaw、Pitch、Roll 各數值:增加 P 的 Pitch 值，能使無人機失去平衡後在最短時間內重新達飛行穩定。

表 6：調整 PID 值分別測試無人機達平衡所需的時間

	P			I			D			平均值
	Yaw	Pitch	Roll	Yaw	Pitch	Roll	Yaw	Pitch	Roll	
初始值	0.2	1.25	1.25	0.001	0.009	0.009	0.1	0.024	0.024	1.868
同時調整 P 值	0.22	1.375	1.375	0.001	0.009	0.009	0.1	0.024	0.024	1.546
同時調整 I 值	0.2	1.25	1.25	0.0011	0.0099	0.0099	0.1	0.024	0.024	1.66
同時調整 D 值	0.2	1.25	1.25	0.001	0.009	0.009	0.11	0.0264	0.0264	1.938
調整 P - Yaw	0.22	1.25	1.25	0.001	0.009	0.009	0.1	0.024	0.024	2.104
調整 P - pitch	0.2	1.375	1.25	0.001	0.009	0.009	0.1	0.024	0.024	1.224
調整 P - roll	0.2	1.25	1.375	0.001	0.009	0.009	0.1	0.024	0.024	1.436
調整 I - Yaw	0.2	1.25	1.25	0.0011	0.009	0.009	0.1	0.024	0.024	1.932
調整 I - pitch	0.2	1.25	1.25	0.001	0.0099	0.009	0.1	0.024	0.024	1.72
調整 I - roll	0.2	1.25	1.25	0.001	0.009	0.0099	0.1	0.024	0.024	2.004
調整 D - Yaw	0.2	1.25	1.25	0.001	0.009	0.009	0.11	0.024	0.024	1.876
調整 D - pitch	0.2	1.25	1.25	0.001	0.009	0.009	0.1	0.0264	0.024	1.872
調整 D - roll	0.2	1.25	1.25	0.001	0.009	0.009	0.1	0.024	0.0264	1.958

(二) 調整 PID 值，測試其飛行時在垂直與水平方向拉力的差異實驗:

先測試無人機廠商設定 PID 初始值，再分別調整 P、I、D 值增加各自初始值之十分之一，以錄影機紀錄電子彈簧秤上的數據，用影片回放找出最大值，每組實驗重複進行三次紀錄測得公斤數再取其平均值，分析實驗結果發現:調整 P 值能稍微增加無人機飛行時在垂直方向的升力與水平方向的拉力，但是影響並不大，由此實驗可知調整 PID 值在無人機垂直方向的升力與水平方向上的拉力並無明顯改變。

表 7：調整 PID 值，測試其飛行時垂直與水平方向上拉力

	P			I			D			平均值(g)	
	Yaw	pitch	roll	Yaw	pitch	roll	Yaw	pitch	roll	垂直拉力	水平拉力
初始值	0.2	1.25	1.25	0.001	0.009	0.009	0.1	0.024	0.024	20	25
調整 P 值	0.22	1.375	1.375	0.001	0.009	0.009	0.1	0.024	0.024	20.08	25.09
調整 I 值	0.2	1.25	1.25	0.0011	0.0099	0.0099	0.1	0.024	0.024	20.06	25.03
調整 D 值	0.2	1.25	1.25	0.001	0.009	0.009	0.11	0.0264	0.0264	20.01	25.02
調整 P - Yaw	0.22	1.25	1.25	0.001	0.009	0.009	0.1	0.024	0.024	20.06	25.03
調整 P - pitch	0.2	1.375	1.25	0.001	0.009	0.009	0.1	0.024	0.024	20.02	25.01
調整 P - roll	0.2	1.25	1.375	0.001	0.009	0.009	0.1	0.024	0.024	20.07	25.05
調整 I - Yaw	0.2	1.25	1.25	0.0011	0.009	0.009	0.1	0.024	0.024	20.01	25.03
調整 I - pitch	0.2	1.25	1.25	0.001	0.0099	0.009	0.1	0.024	0.024	20.04	25.06
調整 I - roll	0.2	1.25	1.25	0.001	0.009	0.0099	0.1	0.024	0.024	20.03	25.02
調整 D - Yaw	0.2	1.25	1.25	0.001	0.009	0.009	0.11	0.024	0.024	20.05	25.02
調整 D - pitch	0.2	1.25	1.25	0.001	0.009	0.009	0.1	0.0264	0.024	20.06	25.07
調整 D - roll	0.2	1.25	1.25	0.001	0.009	0.009	0.1	0.024	0.0264	20.04	25.05

(三) 觀察其飛行主程式，其飛行控制流程圖與程式碼對照如下圖，而相關程式碼只要去上課皆可取得，應是屬於開放程式碼。

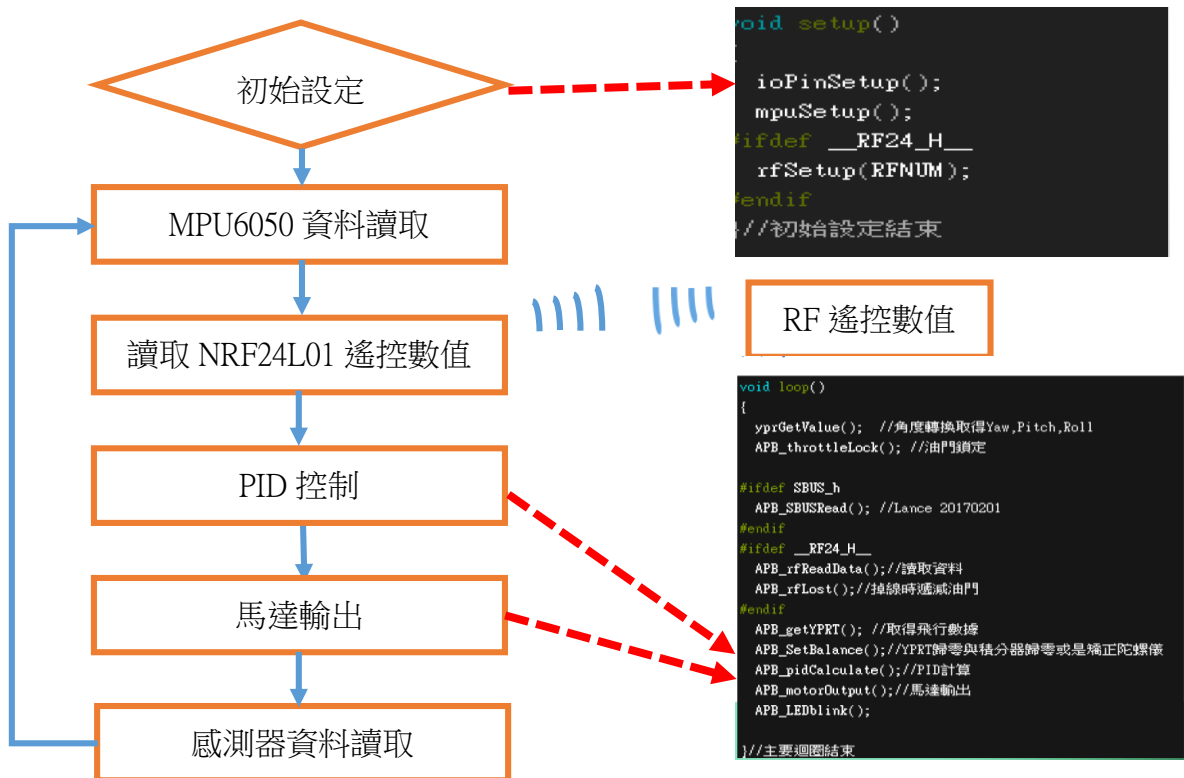


圖 23、無人機飛行控制流程

我們將飛控流程與程式碼逐一對照，發現主要迴圈結束處 APB_LEDblink其只是讓主機上 LED 燈亮，接下來就是重跑流程，我們可以在其後增加各式感測器等相關程式，自行增加測試功能。測試相關數據再透過後續功能傳遞。

(四)利用 nrf2401 傳遞數據資料：

在實驗階段我們參考網路資料，先行撰寫 send.ino 與 receive.ino 這兩個程式中，由於我們發射端與接收端都使用 nRF24L01 做無線通訊，所以無線頻率與基本組態設定都一樣！預設的情況下，nRF24L01+ 六個 data pipe 的自動確認 (Auto Acknowledgment) 功能都是開啟的，所以在發射端只要負責將數據準備好傳送 (write) 出去，接收端則是等待數據過來後進行計算，相關技術規格也跟原廠確認是全雙工雙向通訊非單向通訊。非常具有彈性。

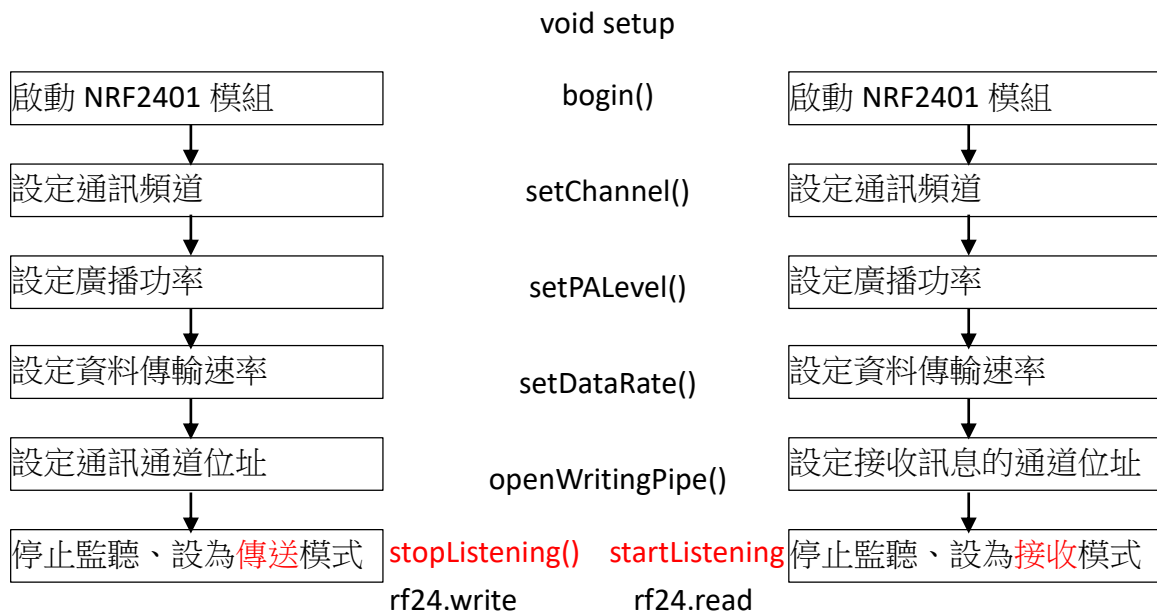


圖 24、NRF2401 基本運作原理圖

經過分別測試，我們發現使用字串陣列、整數陣列都可以達到傳送資料的結果，使用者可依自己喜好撰寫，網路也相當多的參考範例。本系統則使用整數陣列傳遞資料，在發送端，我們先發送資料，根據發送的資料量，決定陣列的大小，接著在接收端，一樣使用整數陣列做接收資料，且會根據存放的陣列位置，寫入相對應的邏輯判斷，讀出我們所需的資料值。

利用整數陣列接收資料。

```

int senddat0=dat[0];
int senddat1=dat[1];
int senddat2=dat[2];
int senddat3=dat[3];
msg[0]=senddat0;
msg[1]=senddat2;
rf24.write(&msg,sizeof(msg));

```

利用整數陣列發送資料。

```

if (rf24.available(&pipe)) {
  int msg[2];
  int readdat[2];
  rf24.read(&msg,sizeof(msg));
  for(int i =0 ; i < 2; i++){
    readdat[i]=msg[i];
    if(i%2==0){
      Serial.print("Current humidity = ");
      Serial.print(readdat[i]); // 顯示訊息內容
      Serial.println('%');
    }
    else if(i%2==1){
      Serial.print("Current temperature = ");
      Serial.print(readdat[i]); // 顯示訊息內容
      Serial.println('C');
    }
  }
}

```

圖 25、傳送資料邏輯程式

三、整合感知模組方面：

我們先將溫溼度等各式感應器程式撰寫並編譯後，先在地面觀察並除錯，之後裝上機翼並將其起飛到遠端，控制端所得到的資料顯示如下圖：

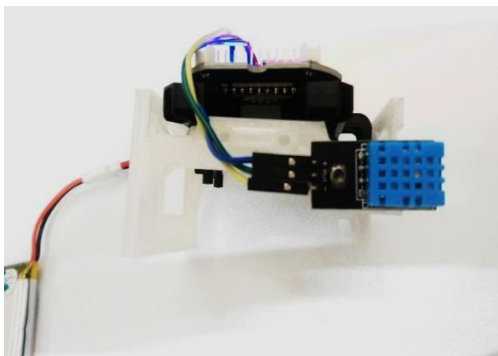


圖 26、將溫溼度感應器固定在承座

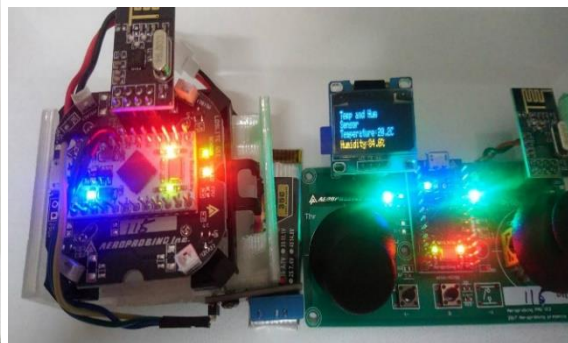


圖 27、無人機地面測試



圖 28、機翼可拆下重複利用

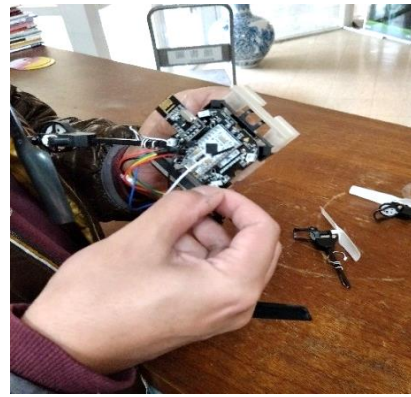


圖 29、所設計零件承座也可以更換



圖 30、重新編譯，改成空氣懸浮粒子偵測畫面



在使用彈性上，機翼與機翼架、螺旋槳如果損壞可以隨時更換，相當方便，而若經費不允許只有一台，在更換感測器也很方便，不同的感測元件可以視重心放在相對位置來做固定，經實測飛行完全正常。

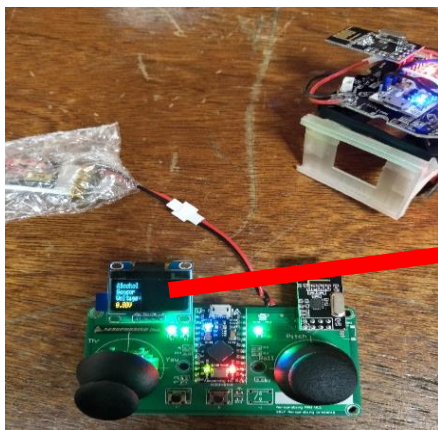


圖 31、酒精濃度感測畫面



三、設計元件基座的考量：不同感測器形狀、重量不一，所以要考量整體重量平衡，我們分別針對不同的感測器設計凹槽或螺絲孔，方便固定，由於空氣懸浮粒子偵測器體積最大且不能鑽孔，我們先將其固定在一側，初步設計如下圖：

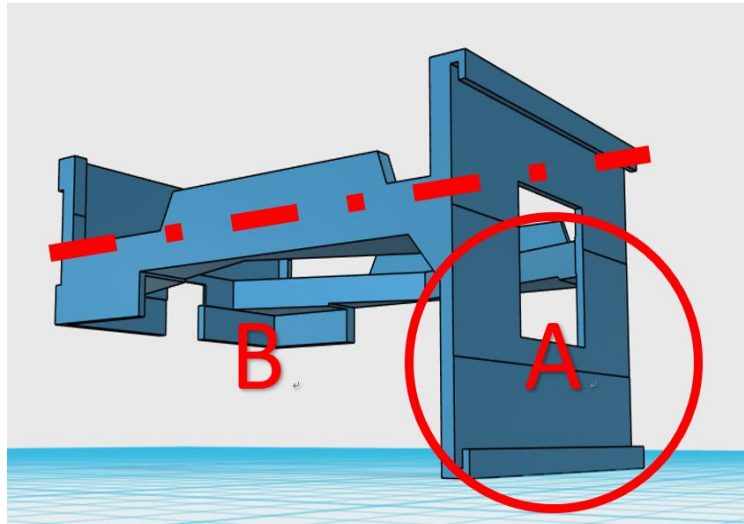


圖 32、無人機底座 A 區體積變大

如上圖在右側增加固定架(A)的設計，在紅線下方 A 測體積增加，容易導致重心偏移，飛行不穩定，經過研究我們發現，從紅色剖面來看，在右圖 A 區中，其體積增加了： $46 \times (18 \times 1 + 3 \times 1 + 2 \times 1) = 1058$ 立方毫米。

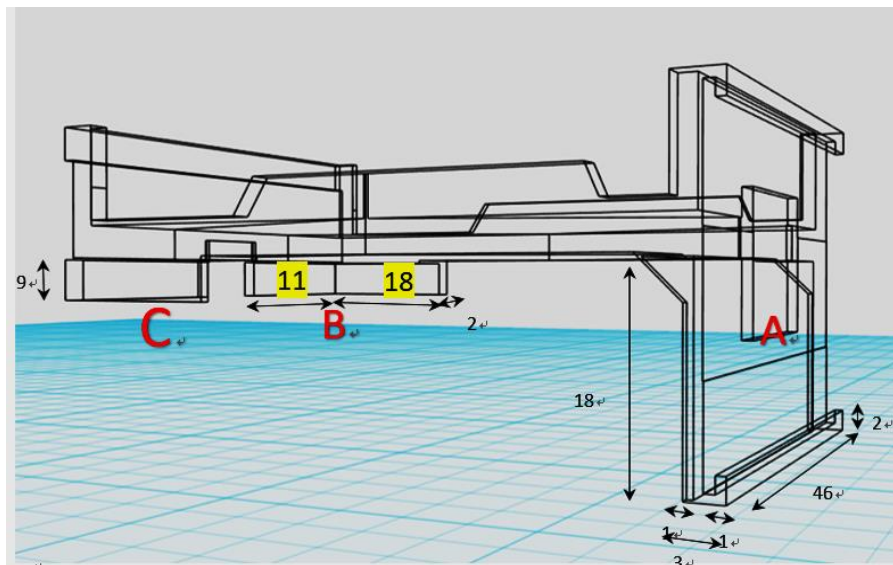


圖 33、無人機 B、C 區體積相加需與 A 區體積相等

在飛行的時候為使重量平衡，我們須在左側增加相對應的體積，以使元件基座重力平衡，此外，由於需考慮對稱性(上圖 B、C 應該一致)，所以在 B 或 C 區需各增加 529 立方毫米，我們將其逆推回去計算出 $529 / 2$ (寬) / 9 (高) = 29.388，也就是上圖 B 區兩者的值加起來應為 29.4(上圖 B 為 $18+11=29$)左右，即可達到平衡的效果。

此外，我們還製作 10 與 19 的搭配，9 與 21 搭配，經多次試飛結果確認能平衡控制，在方向操作上順暢無礙。

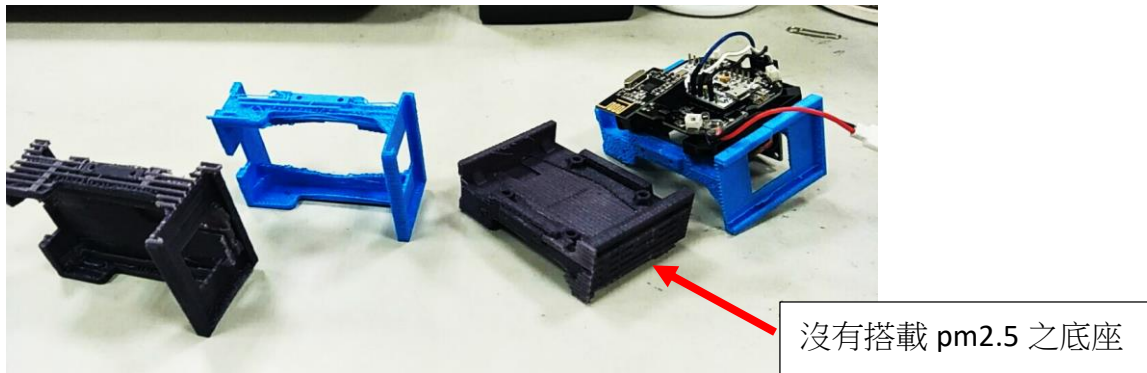


圖 34、各式測試基座，無人機 B+C=A 區體積相加

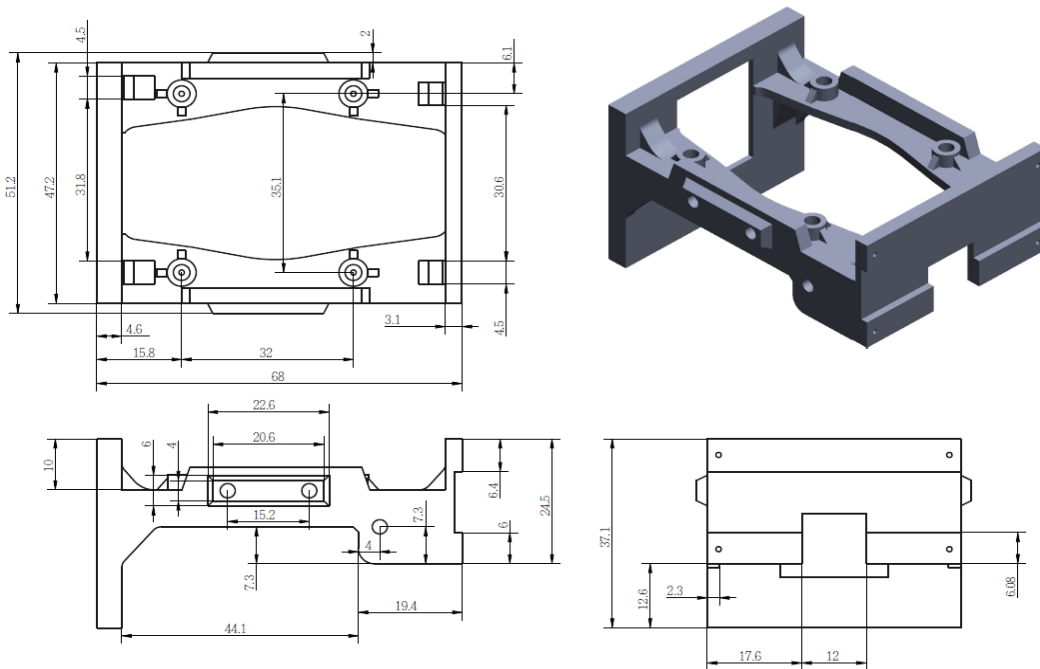


圖 35、完成無人機底座之工作圖

四、資料讀取：本組所設計的執行檔，使用者須告知使用哪一個 com 埠，這可以從電腦管理=>裝置管理員=>連接埠(COM 和 LPT)而得知，然後再輸入監看的時間間隔(單位毫秒)，程式利用 readSerialPort 這個類別，將序列資料讀取到字串陣列 dataFrom () 後，可以用 cout 輸出到畫面後即可在 dos 視窗下看到即時的資訊如下圖：

```

C:\Temp\20180308\Arduino Sensor read\SensorSaveFile
Enter your Arduino COM number : 3
Enter reaad speed(ms) :

```

圖 36、資料讀取之輸入介面

```

C:\Temp\20180308\Arduino Sensor read\SensorSaveFile.exe
Enter your Arduino COM number : 3
Enter reaad speed(ms) : 1
Connection success.
Humidity 79.00 % Temperature 22.32 C
Humidity 79.00 % Temperature 22.32 C
Humidity 79.00 % Temperature 22.32 C
Humidity 79.00 % Temperature 22.32 C

```

圖 37、在 DOS 模式中呈現感測資料

我們考慮方便保存資料及後續應用，我們希望同時存成 txt 或 csv 檔，方便將來使用，我們建立類別 fstream fp, fpe;做好相關宣告並開檔後，利用以下語法即可同時輸出到三個地方，執行結果如下圖：

```
fp << "Voltage\t" << dataFrom[1] << endl;//寫入到文字檔
```

```
fpe << "Voltage," << dataFrom[1] << endl; //存成 csv 可用 excel 開啟
```

```
cout << "Voltage \t" << dataFrom[1] << " V" << endl;//在 DOS 模式中出現
```



圖 38、系統擷取資料產生之文字

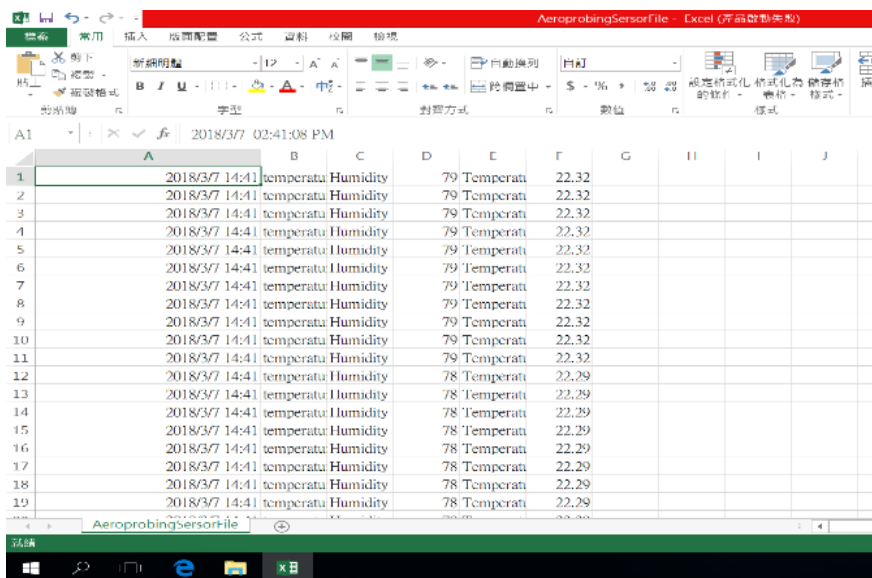


圖 39、系統同時產生之 CSV 檔

程式利用 ios::app 設定為從檔案結尾寫入(輸出)資料(如上圖 39)，若想清空資料或從頭開始，可將原文字檔刪除，系統會自動重新產生。而這些資料未來要建立圖表或透過網路傳送到其他資料庫，也非常方便。

研究二

一、低電量警告與顯示

首先撰寫程式讀取類比接腳電壓資訊，再逆推回去計算電池電量，這部分利用上課所得的 $VCC=3v$ ， $ADC=10bit$ 等資料，透過 $V=analogRead(\text{接腳位置}) \times 3.3 \times 11 / 1024$ 即可得到電壓值，經測試電壓低於 **3.4V**，飛行器隨時就有墜機可能。

接下來要克服如何同時傳遞感測器資料與電壓資料，經不斷測試結果，系統可同時回傳到控制端並顯示低電量警告如下圖，使用者即可隨時注意電量資訊。

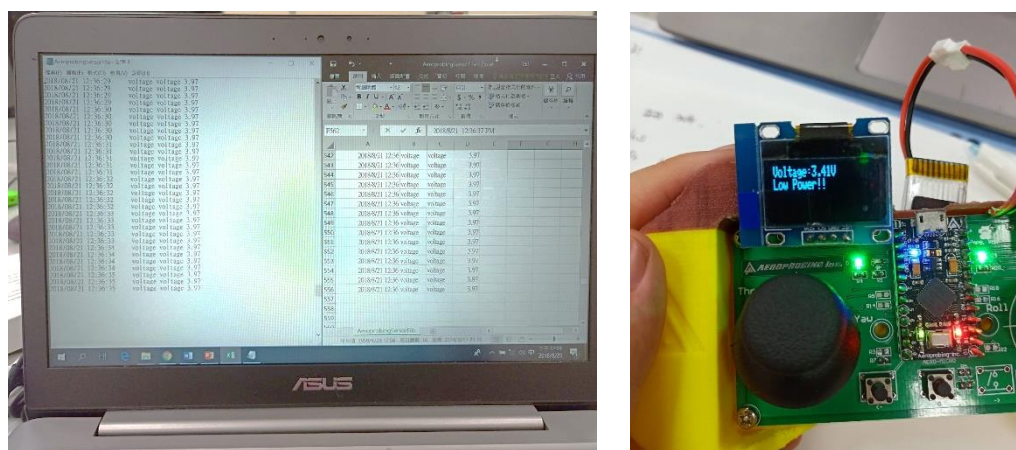


圖 40、透過撰寫程式使用者控制端顯示低電量警告

二、無人機搭載感測器消耗電量

(一) 測試搭載不同感測器的系統飛行時間

先比較搭載不同感測器的無人機監測系統飛行時間差異，隨機選取四組電池，並控制遙控器固定無人機的飛行高度與範圍，每組分別測試兩次記錄飛行時間，結果如下表：

表 8、比較不同感測器對飛行時間的影響

	電池(10.3)		電池(1.5)		電池(4.6)		電池(7.9)	
	test1	test2	test1	test2	test1	test2	test1	test2
DHT11	07.00.44	06.10.03	06.02.88	05.44.54	05.26.92	04.32.74	05.44.45	06.12.53
MQ3	05.42.07	05.03.04	05.43.23	04.23.16	04.27.99	05.24.03	05.58.08	05.37.86
pm2.5	05.00.44	04.11.21	05.12.23	04.10.24	04.08.99	05.34.27	05.44.42	04.57.64
空機	07.21.04	06.43.10	04.02.31	05.52.08	05.32.88	04.32.56	05.28.44	07.33.18

取算術平均數後做橫條圖分析，飛行時間雖然因不同電池組合而有差異，但相較之下無人機飛行時間由短到長皆依序為：搭載 pm2.5、MQ3、DHT11、空機。

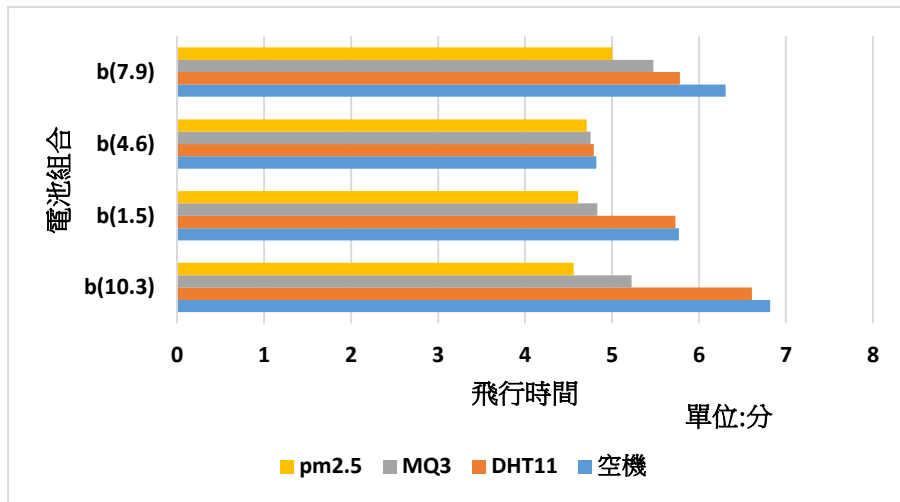


圖 41、比較不同感測器與飛行時間比較圖

(二) 感測器重量-飛行時間與消耗電流-飛行時間的迴歸分析如下：

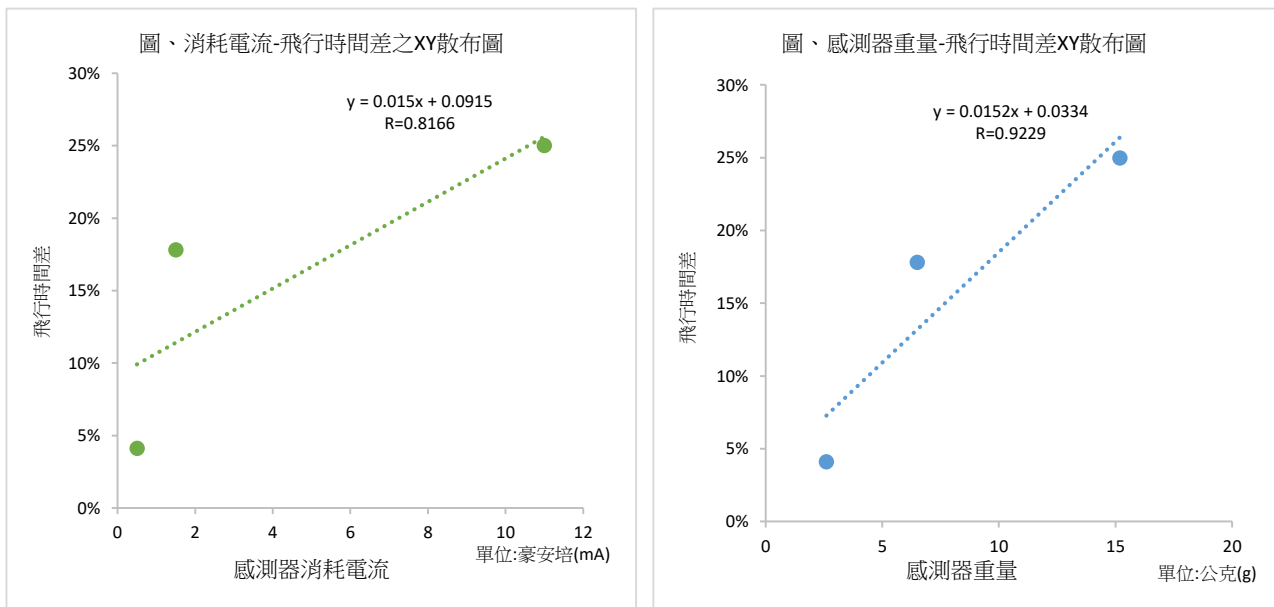


圖 42、比較感測器與飛行時間迴歸比較圖

觀察圖表並計算其相關係數，感測器消耗電流-飛行時間的相關係數為 **0.8166**，感測器重量-飛行時間的相關係數為 **0.9229**，證實感測器重量越重，馬達需消耗更多的電量，無人機監測模組的飛行時間縮短。

三、探討機翼旋轉是否造成數據失真與解決方法

(一) 機翼旋轉是否造成室內空間、室外 30 公尺高度數據失真:室內封閉環境中、室外開放環境其測試結果如下表：

表 9、探討室內封閉環境中機翼旋轉是否造成數據失真

	test1		test2		test3		test4		test5	
	溫度°C	濕度%	溫度	濕度	溫度	濕度	溫度	濕度	溫度	濕度
溫濕度計	21.8	58	26.9	56	29.3	60.1	28.3	57.2	23.3	61.3
DHT11	21.8	58	26.9	56	29.3	60.1	28.3	57.2	23.3	61.3
機翼全速旋轉	21.8	58	26.9	56	29.2	60.1	28.3	57.2	23.3	61.3
誤差值	0	0	0	0	0.1	0	0	0	0	0

表 10、探討室外開放環境中機翼旋轉是否造成數據失真

	Test1		Test2		Test3		Test4		Test5	
	溫度°C	濕度%	溫度	濕度	溫度	濕度	溫度	濕度	溫度	濕度
溫濕度計	34.8	58.3	36.7	56	32.6	57.2	34.6	57.4	36.3	61.7
DHT11	34.8	58.3	36.7	56	32.6	57.2	34.6	57.4	36.3	61.7
機翼全速旋轉	34.8	58.3	36.7	56	32.6	57.2	34.5	57.4	36.3	61.7
誤差值	0	0	0	0	0	0	0.1	0	0	0

根據表 9、表 10 溫度誤差後發現室內與室外誤差值均非常小幾乎不到 2%，濕度則幾乎接近零誤差，證實無人機監測模組機翼旋轉不會造成數據失真。推估電風扇會讓人感覺涼爽，是因為其把皮膚表面的熱空氣帶走、讓汗水容易蒸發，此過程需要吸熱所以會帶走大量體熱，因此人們會覺得涼爽，但周圍溫度並不會因此改變。

(二) 距離溫濕度梯度測量

以電暖器模擬測量距離溫濕度梯度結果如下表:

表 11、測量點熱源距離溫濕度梯度

	30cm		50cm		100cm		150cm		200cm	
	溫度°C	濕度%	溫度°C	濕度%	溫度°C	濕度%	溫度°C	濕度%	溫度°C	濕度%
標準溫度計	34.1	53.2	34	53.2	33.5	54.3	33.4	56.4	33.1	55.3
機翼全速旋轉	34.1	53.2	34	53.2	33.5	54.3	33.4	56.4	33.1	55.3
誤差值	0	0	0	0	0	0	0	0	0	0

根據表 11，計算無人機機翼全速旋轉與標準溫濕度計所得數據，距離兩公尺內誤差值為零，因此使用者可以無人機直接做點熱源測量，不需另外以梯度計算校正。

四、測試傳遞多筆資料的可行性：當各項單一感測器資料都能正確傳送與接收後，我們發現在機翼端的感測器主要都是依照載入函示庫 => 初始化 => loop 主程式中開始運作 => 紀錄到&senddata 位址，再傳送資料出去。所以我們如果想要傳遞兩筆資料，只要將自訂的感測器依上述流程初始化後，將所讀到的資料寫入 sendData.Data[0]，sendData.Data[1]等陣列即可。在接收端的程式方面，只需增加相關條件判斷迴圈，並根據所接收字串陣列 dataform[0]的值來判斷應呈現的資料，如下圖中我們希望同時傳遞溫度資料及電池電壓，即可在文字檔及 excel 檔中出現使用者自訂感測器數據資料。而由於是自訂資料，在單位上就不予呈現，以數據為主。

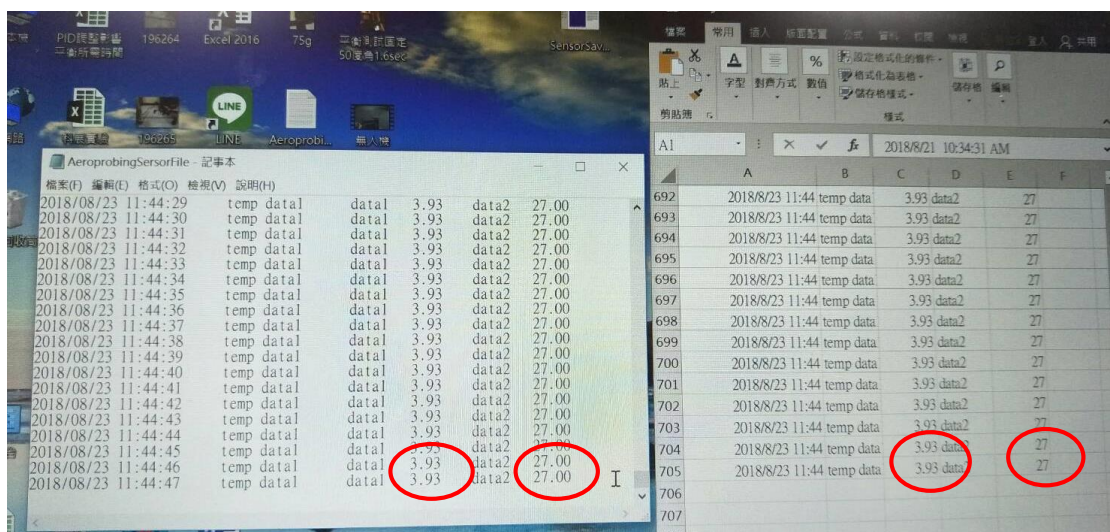


圖 43、同時傳遞電量、溫度兩筆資料結果呈現

經不斷測試結果顯示，自訂傳感器的主要限制是：總重量不能超過最大承重，機身剩下的接腳數只剩 3 個。此外，電池供電品質也會影響參數傳遞的成敗。

陸、結論

一、 本組發展以無人機搭載感測器的優勢有：

- (一) 降低成本：一般來說，固定式空氣感測器之裝置除了資料傳輸考量外，還需考量供電及裝置地點的難易度，其精準度會依裝置密度而有所不同，若是需要密集裝置則感測成本將大幅提高。反之，使用無人機搭載感測器並能即時回傳感測資訊，則能大幅降低裝置成本，符合經濟效益。
- (二) 三維度空氣感測：除高空外，無人機的運動靈活性也可適用在很多情況，例如災害現場、河岸湖泊、隧道...等，具備在不同維度下可工作的特性及優勢。
- (三) 簡易 PID 調校方法：當使用者在戶外發現有風干擾使無人機不好操控，在風速低於 3.6 km/h 情況下，可優先透過調整 PID 值中 P 的 Pitch，即可縮短系統平衡反應所需時間。
- (四) 電量偵測與警告：無人機飛行過程中，可能因電力不足導致墜機情形，本系統利用無線訊號將飛控板接腳的電壓資訊回傳，並在低電量時顯示警告，降低使用者操作上的風險與災害發生機率。

二、 本監測模組的彈性：

- (一) 改變感測器的彈性：本感測器模組在未來都可以彈性變化，只要重量或體積允許(不超過 95 公克)，都可以改裝成自己想要的感測器，例如紅外線，超音波，甚至災難現場瓦斯偵測，或者河川的 pH 值偵測等，而且市面上大部分感測器重量都低於 10 公克，在應用上具備相當高的改變彈性。
- (二) 機翼與各零件組裝彈性：機翼耗損是無人機無法避免的事實，當發生墜機，機翼折斷無法飛行時，無論是從別台拆機翼過來，或者是從本台拆感測器到別台組裝都可以，相當具有彈性。
- (三) 可以同時搭載、回傳不同感測器數據：透過修改相關程式，讓自製的無人機即時監測系統可以同時傳遞不同感測器資料，提高效率，讓原本需飛兩趟的任務一次搞定。

三、 容易移植與應用：

- (一) 本監測模組在技術面上應用學校所教 **Arduino** 與 **Visual C++** 程式設計的觀念，結合市售成熟便宜的無人機所開放的腳位來開發。在硬體的維護與技術的移植都相當方便。
- (二) 所蒐集到的數據，已經轉成文字檔及 **csv** 檔，無論未來要傳到網路或資料庫，都很容易分析利用。

柒、引注資料及其他

- 一、 賈澤民，詹為淵，徐子圭，江志煌, 2013。連翼構型飛翼船研製開發，航太年會及學術論文研討會。
- 二、 用 **Arduino** 製作四軸飛行器（二）飛行原理。資工人日常。民 103 年 3 月 23 日，取自：<http://csenichijou.blogspot.tw/2014/03/Quadcopter-2.html>
- 三、 **Infinityhobby**。你也能看見台灣-無人飛機空拍入門(第零篇)。Mobile01。民103年4月19日，取自：<http://www.mobile01.com/topicdetail.php?f=368&t=3853906&p=1>
- 四、 柯博文(民 103)。 **Arduino** 互動設計專題與實戰－深入 **Arduino** 的全方位指南。台北，基峰出版社。
- 五、 邱聰倚、姚家琦、劉耀鴻(民 103)。 **Autodesk 123D Design** 與 **Tinkercad 3D** 設計速繪美學。台北，基峰出版社。
- 六、 **Arduino WIFI(ESP8266)** 初次上手應用。民 104 年 11 月 27 日，取自 <http://lolwarden.pixnet.net/blog/post/82031214>。
- 七、 賈澤民，無人飛航載具監測環境污染之功能，航測及遙測學刊 第二十一卷 第 3 期 第 163-182 頁，民國 105 年 11 月