

# 四旋翼巡航系統

林偉川  
指導老師  
e-mail :  
wayne@takming.edu.tw

呂碧蟬  
組長  
e-mail :  
mimi60723@gmail.com

林昱範  
組員  
e-mail :  
aa655156@gmail.com

劉宛嶺  
組員  
e-mail :  
s992237@gmail.com

劉子齊  
組員  
e-mail :  
j29721197@gmail.com

蔡宗穎  
組員  
e-mail :  
silent7335739@gmail.com

## 摘要

探勘地形、高山搜救，以往必須派遣專業人員徒步前往或者駕駛直升機，在危險的環境中人員的安全受到威脅。而現在日新月異的科技發展，原本是玩具的飛行器，演變成可以使用在救難救災上的四旋翼、多旋翼，不但可以偵測傳回數據，更取代了高危險的一些探勘工作。本研究目前使用飛控板結合四旋翼機架，透過 Arduino 開發環境來編寫飛控板之平衡程式碼後，藉由飛控板軟體測試，對平衡感應數據做校正，待四旋翼本身能自主平衡後，經由藍芽作為遙控方式將指令傳給飛控板，並於資料庫儲存飛行軌跡指令紀錄，使巡航能重複先前飛行路線，也結合樹莓派上 CCD 攝像頭做為定期拍攝飛行巡邏過程所拍攝的照片。

**關鍵詞：**四旋翼、PID 演算法、MultiWii 飛控板、樹莓派、CCD。

## Abstract

Patrolling terrain, mountain rescue are a certain of danger for persons or helicopter. As rapid growth of technology, the original toy aircraft, evolved to quadrotor or multicopter, could be used to rescue not only can detect the sent data backward, but replace some high-risky exploration work. This paper is proposed a quadrotor patrolling mechanism combined with an Arduino-based flight control board. Through this mechanism, the adjusting balance program of flight and collect the sensor data via Bluetooth is developed. The flight path is stored in the local database which is recorded the flight instruction, so that the patrolling path can repeat the previous flight path. The proposed quadrotor also combines with Raspberries pi with CCD camera for recording photographs in every regular patrol flight path.

**Keywords:** Quadrotor, PID algorithm, MultiWii flight-control board, Raspberry Pi, CCD.

## 1. 前言

隨著日新月異的科技進步，四旋翼[1]近幾年被廣泛開發運用範圍由最初的兒童飛行玩具，到愛好空拍玩家的四旋翼無人空拍機，最後至急難救災、水質檢驗、播灑農藥...等。飛控板的進步改善了穩定性，但大多數的四旋翼如果是公司本身所開發，會將所有功能合併寫入同一個電路板上，使用者並不能看到內部演算方式也不能自行擴充功能，而近期網路上另有開發支援 Arduino 的飛控板，但是缺點在於 Arduino 運算能力不足，所以我們採用樹莓派與飛控板做結合，除解決運算及擴充問題外，也可以在樹莓派的 Linux 系統上直接使用附加元件的軟體環境做開發。本研究利用自行組裝之四旋翼機架結合 MultiWii 飛控板，加入樹莓派搭配 CCD 模組，使用者可藉由手機 APP 透過藍芽來持續對四旋翼發送遙控、拍攝指令，並在本機端架設資料庫存取照片、飛行指令之數據於樹莓派端做資料傳輸，進而達到自主平衡、手機遙控、紀錄飛行軌跡、回復飛行動作、拍攝照片、查閱照片、影像串流[2]的功能，做為自動巡邏之用途。本論文第二節介紹相關研究，第三節為系統架構設計，第四節為系統實作與測試，第五節為結論。

## 2. 相關研究

### 2.1 美國 Intel 四旋翼結合穿戴式裝置

美國知名半導體公司 Intel，於穿戴式挑戰賽上發表了利用四旋翼結合穿戴式裝置[3]。操作者可將四旋翼彎曲佩戴至手腕上，需要時將

機身展開即可使用，其餘功能皆不亞於一般四旋翼，突破了旋翼機不易攜帶的印象。

## 2.2 TEDGlobal 四軸飛行器的運動能力

Raffaello D'Andrea 對於 TEDGlobal 四旋翼 [4] 的功能包含使水杯不會打翻的極穩定的平衡演算法、利用天花板上攝影機與筆記型電腦做室內定位計算達到自動回饋控制，及標定亮光貼紙的物品可搭配四旋翼重新做平衡，自動接球及丟球功能，在室內的四旋翼已經相當於一個會思考的機器人。

## 3. 系統架構

### 3.1 系統架構介紹

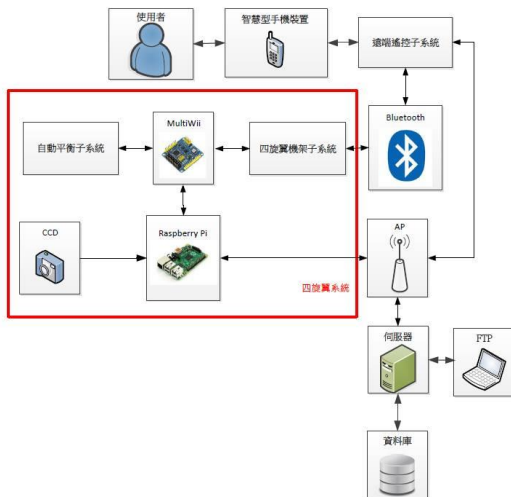


圖 1 整體系統架構圖

四旋翼巡航應用系統可分為四旋翼機架子系統、自動平衡子系統、遠端遙控子系統、資料庫。首先須利用 MultiWiiConf 軟體測好 PID 自動平衡之參數與操控指令寫入飛控板，實際操控時，我們使用手機 App 藍芽對 MultiWii 飛控板下達欲操控指令之參數，當 MultiWii 飛控板藍芽接收後，對馬達輸出做出相應改變，並將演算法參數及操控指令透過樹莓派 WiFi 存入資料庫，樹莓派 CCD 透過網路將影像傳至筆電，以時間為名稱定期存入資料庫。整體系統架構如圖 1 所示。

### 3.2 四旋翼機架子系統

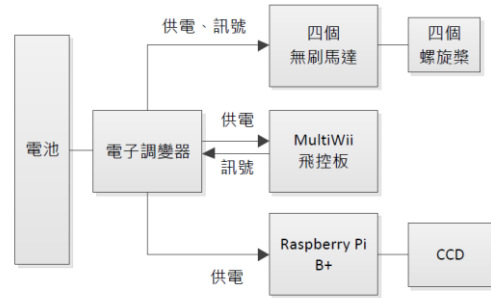


圖 2 四旋翼機架子系統架構圖

四旋翼機架子系統當中有五樣主要構件，分別為電池、MultiWii 飛控板、電子調變器、無刷馬達、螺旋槳，電子調變器為主要控制電壓及馬達轉速的元件。降壓提供給 MultiWii 飛控板及樹莓派 B+ 使用，其餘 CCD 模組是作為附加功能，另外，由於本研究所有元件除了需事先計算好互相供電方式 [5] 外，更須注意重量問題。四旋翼機架子系統架構如圖 2 所示。

### 3.3 自動平衡子系統



圖 3 自動平衡子系統運作流程圖

當機體偏移時，MultiWii 飛控板會感測到相關傾斜參數值，並顯示於 MultiWiiConf 軟體中，同時也以此軟體做 PID 調校，當 PID 調校完成時，MultiWii 飛控板即直接可對各馬達的輸出量做控制。運作流程如圖 3 所示。

#### 3.3.1 自動平衡子系統 PID 控制器公式

PID 控制器 [6] 顧名思義是由 Proportional (比例)、Integral (積分)、Derivative (微分) 三個元素所構成，透過  $K_p$  (比例增益)、 $K_i$  (積分增益)、 $K_d$  (微分增益) 參數，可構成一個線性且動態不隨時間變化的系統。PID 的運作方式為將一個輸入值與參考值做比較，比較後的結果差反饋至計算下一次的輸入值，以達到修正後之值得以保持於參考值間。其運算公式及說明如公式 1。

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

.....公式 1

$K_p$ ：比例增益，為調適參數

$K_i$ ：積分增益，為調適參數

$K_d$ ：微分增益，為調適參數

$e$ ：誤差 = 設定值 (SP) - 回授值 (PV)

$t$ ：目前時間

$\tau$ ：積分變數，數值從 0 到目前時間  $t$

比例控制項  $P_{out} = K_p e(t)$

誤差值與正值常數之乘積，只於輸出和誤差成正比時成立。如果比例增益太大會造成系統不穩定，反之如果太小會造成修正不足。

積分控制項  $I_{out} = K_i \int_0^t e(\tau) d\tau$

先前誤差值總和與正值常數之乘積，於先前平均誤差值計算出輸出結果誤差，結果誤差值會逐漸降低，最後趨近於設定值。如果積分增益越大，誤差校正速度也越快。

微分控制項  $D_{out} = K_d \frac{d}{dt} e(t)$

誤差值的一階導數與正值常數之乘積，可預測未來誤差提升速度與穩定性，如果微分增益越大反應速度越快速。

### 3.4 遠端遙控子系統



圖 4 遠端遙控子系統運作流程圖

使用者於手機 App 介面下達操控及取出照片指令，操控指令透過藍芽將數據傳給 MultiWii 飛控板藍芽接收，藉由 MultiWii 飛控板下指令讓馬達各值做出相對應的改變，使四旋翼做出相對應的飛行動作，再由樹莓派 B+ WiFi 將指令存入資料庫，而取出照片指令則由 WiFi 透過樹莓派 B+ 至資料庫，取出對應資料庫之照片，筆電再由 WiFi 將照片傳回手機 App，即可在手機上瀏覽及搜尋照片。運作流程如圖 4 所示。

### 3.5 資料庫 ERD

本系統之資料庫設計的 ERD，其內部表格為：飛行指令、動作、任務及影像。如圖 5 所示。

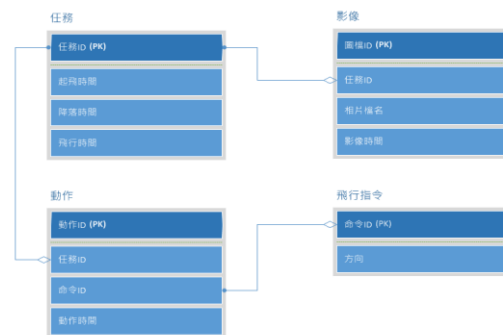


圖 5 資料庫 ERD

## 4. 系統實作與測試

### 4.1 硬體實作

#### 4.1.1 四旋翼機架子系統

中間部分為 MultiWii 飛控板、飛控接收器、藍芽、樹莓派、WiFi、CCD、電子調變器、電池、開關所疊加而成，四周為所需馬達及螺旋槳。如圖 6。



圖 6 四旋翼組裝正面圖

四旋翼組裝系統機體側面圖。如圖 7。

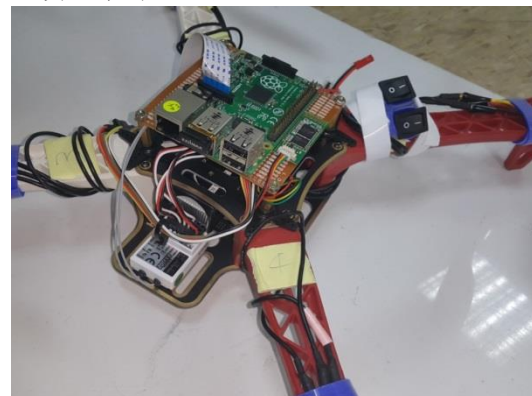


圖 7 四旋翼機體側面圖

由四旋翼側面可清楚看到中心所擺放之方式由低至高分別為：CCD、電子調變器、飛控藍芽接收器、電池、MultiWii 飛控板、樹莓派、WiFi。如圖 8。

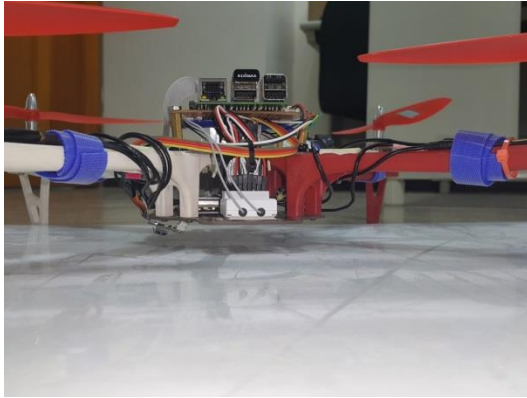


圖 8 四旋翼機體側面圖

CCD 固定於最下方以便拍照，如圖 9。

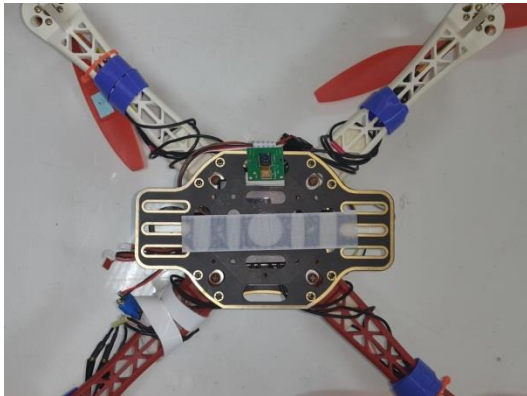


圖 9 四旋翼機體反面圖

#### 4.1.2 電量使用測試

耗電量估計之時間如表 1，由電池、馬達、飛控板、樹莓派所耗的電量來計算出使用時間之長度。

表 1 電量使用估計時間

硬體名稱	耗電量
電池	$2600\text{mah}=2.6\text{ah}=1\text{c}$ $8\text{c}=2.6\text{ah}\times 8=20.8\text{ah}$ $W=P\times t=230.88\text{wh}$
馬達	額定電流 $I_a$ $=\text{最大額定電流}\times 75\%$ $=20\text{a}\times 75\%=15\text{a}$ $P=VI=11.1\text{v}\times (15\text{a}\times 4)=666\text{w}$
飛控板	$P=VI=5\text{v}\times 3\text{a}=15\text{w}$
樹莓派	$P=VI=5\text{v}\times 3\text{a}=15\text{w}$
估計時間 $t$	$t=W/P$ $=230.88\text{wh} / (666\text{w}+15\text{w}+15\text{w})$ $=0.33\text{h}=19.9\text{min}$

耗電量實測之時間如表 2，將電池充飽電後測量實際可使用時間。

表 2 電量使用實測時間

次數	估計時間 (min)	實測時間 (min)	時間差 (min)
1	19.9	16.5	3.4
2	19.9	17.9	2
3	19.9	16.7	3.2
4	19.9	18.6	1.3
5	19.9	17.3	2.6
總次數	5	平均電量 時間	17.4

1	19.9	16.5	3.4
2	19.9	17.9	2
3	19.9	16.7	3.2
4	19.9	18.6	1.3
5	19.9	17.3	2.6
總次數	5	平均電量 時間	17.4

## 4.2 軟體實作

### 4.2.1 播放圖片

查詢按鈕可查詢開始到結束時間之間的影像檔案，播放按鈕可從頭撥放所查詢的影像，停止按鈕為停止撥放，停止按鈕的兩邊則有上一張下一張的功能鍵。如圖 10。

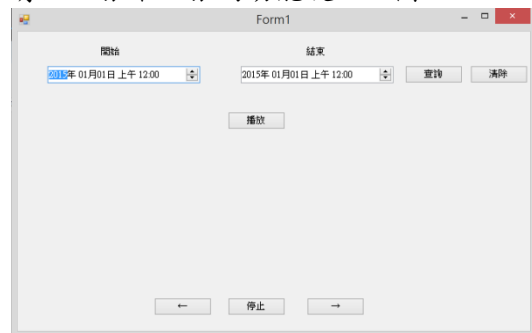


圖 10 影像串流設計視窗

檢視撥放畫面，圖片下放會顯示該張影像的檔名。如圖 11。



圖 11 檢視撥放畫面

按清除按鈕可將檢視結果清除。如圖 12。

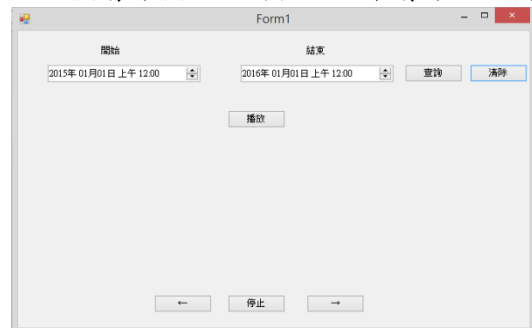


圖 12 清除結果畫面

### 4.2.2 自動讀取本機文檔

自動讀取本機文檔介面，三個新增分別會在指定的路徑內將指定檔名的文檔內容分別

對三個資料表做新增動作，如圖 13。



圖 13 自動讀取本機文檔介面

查詢可對三個資料表當前的情況做查詢動作，如圖 14 為未新增資料的情況。

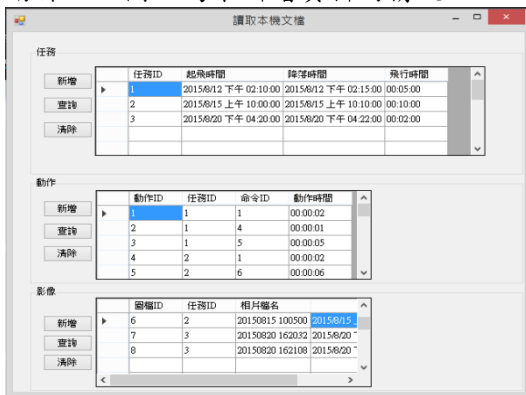


圖 14 查詢本機文檔

新增完畢後按鈕列下方會出現”完成”的提示字樣，如圖 15。

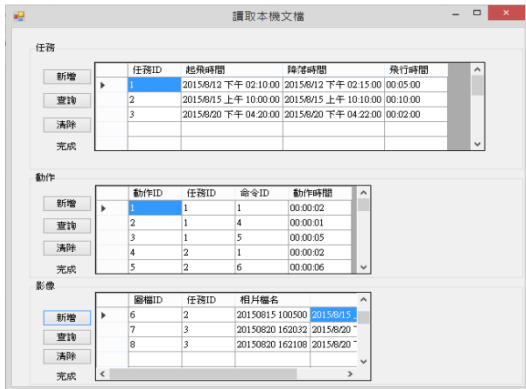


圖 15 新增資料介面

圖 16 為新增資料後的查詢結果。

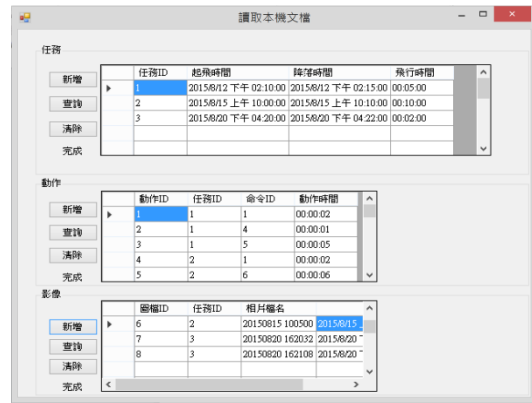


圖 16 新增資料後的查詢結果

清除鈕可將查詢結果清空，如圖 17。

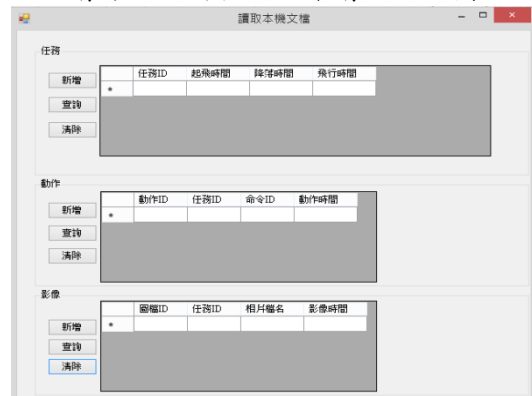


圖 17 清除查詢結果

#### 4.2.3 設定圖片位置

只有設定檔名為 image，沒有設定儲存路徑，因此都會在預設目錄，一般使用者的檔案位於【/home/username】，root 權限的使用者檔案位於【/root】，圖 18 為 root 權限下所拍的照片。

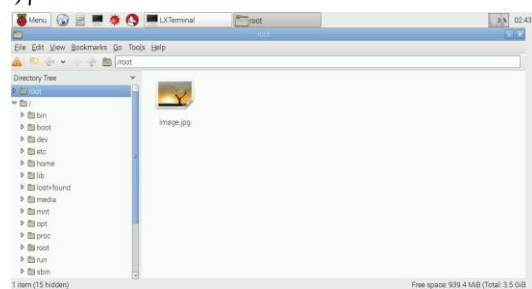


圖 18: 預設路徑下的照片

照片正常檢視之畫面，如圖 19。



圖 19: 開啟相片

#### 4.2.4 樹莓派與 WiFi 傳檔延遲時間

樹莓派透過WiFi傳檔至主機端延遲的時間測試如表3。我們由樹莓派端紀錄發送時間，主機端紀錄接收時間，相減所得到的時間做為延遲時間，以十次測試取出平均之延遲時間。

表 3：Raspberry Pi 與 WiFi 傳檔延遲時間

編號	傳送檔案時間 (樹莓派)	接收檔案時間 (主機)	延遲 時間 (s)
1	20:05'50.52"	20:05'51.35"	0.83
2	20:05'55.35"	20:05'55.97"	0.62
3	20:06'02.37"	20:06'03.11"	0.74
4	20:06'09.45"	20:06'10.73"	1.28
5	20:06'16.12"	20:06'16.98"	0.86
6	20:06'23.36"	20:06'24.50"	1.14
7	20:06'29.42"	20:06'30.69"	1.27
8	20:06'37.25"	20:06'38.07"	0.82
9	20:06'42.47"	20:06'43.45"	0.98
10	20:06'48.33"	20:06'49.17"	0.84
總數	10	平均延遲時間	0.94

### 4.3 整合測試

#### 4.3.1 起飛

室內試飛時，我們將四個方向分別綁上長度適中繩子，以調整其平衡值，如圖20。



圖20 綁繩子調整平衡及測試

平衡調整好之後，將繩子拆除，可在手上做更細微的校正，如圖21。



圖21 手上微調平衡及測試

上述校正皆完成後，即可帶到室外試飛，

起飛前轉速上升時的情況，如圖22。



圖22 四旋翼起飛前

四旋翼空中飛行圖，如圖23。



圖23 四旋翼飛行圖

高空拍攝圖，飛到四至五層樓高度時的情況，如圖24。



圖24 四旋翼高空拍攝圖

#### 4.3.2 降落

四旋翼降落時，轉速只需要維持在比飛行時的轉速降低一些，而且由於地面效應的關係，降落時也會多一層升力使其不會直接掉在地上，如圖25。



圖25 四旋翼降落時

降落後，螺旋槳逐漸停止轉動，如圖26。



圖26 四旋翼降落後

## 5. 結論

本研究所提四旋翼巡航應用系統，可輔助人類處理空中巡航及拍攝的功能，不僅運用樹莓派的擴充性，讓四旋翼不僅只有飛行功能且有定時、定點拍攝，讓使用者運用此巡航應用系統更加方便。藉由本研究所提的四旋翼，除了可以手機遙控飛行、串流影片拍攝照片外，也可以將飛行軌跡存至資料庫，做飛行軌跡紀錄回播動作，樹莓派上的CCD可以連續拍攝照片，以便做成串流影片播放做為巡邏之用，也可以使用手機App瀏覽所拍攝的照片，而樹莓派與飛控板MultiWii的結合，除可使四旋翼穩定飛行，也因為樹莓派運算能力強且其本身有Linux作業系統，所以在系統中可以使用開放性附加元件軟體開發環境做編輯，除了提升便利性，也能運作得更順暢。手機App的開發未來也將取代了原有的遙控器，讓使用者能搭配附加元件在手機APP板面上做改變。藉由本研究驗證了四旋翼的開發性不再是侷限於遙控飛行空拍功能，更可以運用於巡航的應用。

## 參考文獻

[1] 四旋翼:

<https://zh.wikipedia.org/wiki/%E5%9B%9B%E8%BD%B4%E9%A3%9E%E8%A1%8C%E5%99%A8>

[2] 影像串流

<http://ruten-proteus.blogspot.tw/2014/02/RasPi-Camera-Guide.html>

[3] 美國 Intel 四旋翼結合穿戴式裝置

<http://www.techbang.com/posts/20219-nixie-its-a-fantastic-fusion-when-the-remote-controlled-aircraft-into-a-wearable-device>

[4] TEDGlobal 四軸飛行器的運動能力:

[https://www.ted.com/talks/raffaello\\_d\\_andrea\\_the\\_astounding\\_athletic\\_power\\_of\\_quadcopters?language=zh-tw](https://www.ted.com/talks/raffaello_d_andrea_the_astounding_athletic_power_of_quadcopters?language=zh-tw)

[5] 供電方式:

[http://flyraspberrypi.blogspot.tw/2013/02/raspberrypi\\_23.html](http://flyraspberrypi.blogspot.tw/2013/02/raspberrypi_23.html)

[6] PID 控制器公式:

<https://zh.wikipedia.org/wiki/PID%E6%8E%A7%E5%88%B6%E5%99%A8>