

第 15 組

機械工程實務 期末報告



指導教授：林沛群、陳湘鳳、楊馥菱、

詹魁元、蔡曜陽、鄭榮和

助教：陳牧心、簡士為

學生：B06502016 陳佳好

B06502110 簡揚開

B06502128 鄭丞佑

B06502157 陳昱睿

B06502161 陳羿廷

教授	詹魁元	楊馥菱	鄭榮和	林沛群	蔡曜陽	陳湘鳳
評分						

1 緒論

1.1 核心概念

在上學期機械設計原理的專題中，我們盡可能地使用生活上可以用到的器材，並且在期末呈現中也有不差的成果，因此我們這組這次實務的核心概念將延續上學期的概念，那就是以最小金錢和時間上的成本，完成機械工程實務所要求的目標。由於上學期我們在時間的進度安排上不是很理想，這次我們在一開始就訂定好分工、目標以及規定完成的時間，以達到最有效率的進行方式。由於我們這組的組員在實作部分都沒有相對多的經驗，為了減少彼此的負擔，也避免超過我們能力負荷的要求，因此不管在車體還是程式設計上，我們希望能夠越簡化越好。因此，我們氣動車的總體核心概念是低成本、符合需求、高效率與最簡化。

1.1.1 文獻探討與資料收集

車體組與控制組主要參考學長姐們的報告有些初步的了解後，及發揮自己的創意與想像力，並力求達到期中與期末測試的目標要求。

風扇組首先則閱讀流力課本機翼升阻力部分的內容與查閱扇葉相關的名詞定義，例如 NASA 與 WIKI…等網站介紹翼型的部分，接著探索相關可用的網站與軟體，例如 Airfoil Tools、QBlade、Fusion 360、Autocad 與 UIUCPropellerDataSite…等等，以利後續進行風扇的分析與繪製。

1.1.2 設計過程

在第一次的開會過程，我們除了進行分工與基本規範的訂定之外，也一起詳細閱讀了期中與期末測試的實際規範並進行腦力激盪，從車體的形式、動力裝置、轉向機構與控制系統…等等，逐一分享每個人的想法，並互相討論其可行性。整體在設計上有個大方向後，再經由分工後每組不同的組員各自參考學長姊的報告、網路資料與資源…等等，並進行小組內部的討論，在第二次的開會過程中，再向大家報告一週內所參考得到的訊息，並互相提問討論。最後在確定每組的設計方向後，開始交由各組進行後續的設計、分析與製造。

總體來說，本次實務課整體的設計流程如下圖(一)：



圖一：設計流程圖(XMind:Zen)

1.1.3 團隊與分工方式

組員為上學期機械設計原理同組組員，除了考量到上學期期末專題成果表現不錯之外，經過了上學期一學期的合作與討論，也讓我們在不同組員間的默契、協調與互相討論都更為深刻與了解，並對彼此的分工與能力互相信任，建立一個比上學期更有效溝通的團隊。

我們先一起討論將這次的實務課程專題分為車體組、風扇組與控制組，與其各自須負責的項目與內容，並依據大家個人的喜好與相關能力作為考量，輪流選擇最想加入的組別。最後分工的結果為(期中測試前)：陳昱睿與陳羿廷為車體組，陳佳好與簡揚開為風扇組，鄭丞佑為控制組。期末測試由於控制的通作量較大，因此微調為：陳昱睿與陳羿廷為車體組，陳佳好為風扇組，簡揚開與鄭丞佑為控制組。

在期中測試後，我們與另一組(第 10 組)首先召開兩方的全組會議，一起研究期末的測試要求與規則，以利對期中的車子進行優化與策略規劃。在第二次開會中，優先集結兩組的控制人員進行更詳細的策略討論，在確定整體期末測試的方向後，與各組組內成員先進行討論如何修改車體以符合最終期待。接著兩組各自先完成期末賽道的控速、循跡與超車測試，並進一步檢測與調整超音波感測器(跟車)的參數與置放位置。最後兩車進行合體，並在最後階段不斷測試與修改程式(控制的方式與參數)，最終成功完成期末測試的標準。

1.2 設計特色

特色	
底盤	設計單純 組裝容易 最大空間 結構穩定
車輪	容易轉動 最少設計(3 輪)
風罩	符合安全 結構穩定
風扇	符合推力
控制	簡化架構

1.3 優缺點

在底盤設計的部分，在特色的其中一項為最大化空間，但此與設計單純與組裝容易可能有點矛盾，由於最簡化了前輪的設計過程，反而忽略了前輪附近的空間，造成車體前身的空見配置不甚理想，是還可以再改善的地方。此外，在車輪的部分我們選擇了最少設計，以 3 輪為結構並省去了阿克曼機構設計的複

雜，但這使我們在期中測試時吃進了苦頭，由於在第一停靠區前有多段膠帶重和的地方，並且在中央循跡線的位置，使得我們必須加大風扇轉速使前輪通過，耗了我們不少的精力，這是在我們期初設計時沒有想到的問題。

在控制的部分我們選擇的特色是最簡化架構，優點除了容易設計、修改之外，也減少了系統延遲跟不穩的問題，但缺點是程控的方式較死，需要不斷調整參數以獲得改善，並且較吃其它硬體的精準度(例如計步器)，但整體而言要達到期中與期末測試的需求還是很足夠的。更詳細實體設計的優缺點將會在之後的章節逐步討論。



目錄

1	緒論	2
1.1	核心概念.....	2
1.1.1	文獻探討與資料收集.....	2
1.1.2	設計過程.....	2
1.1.3	團隊與分工方式.....	3
1.2	設計特色.....	3
1.3	優缺點.....	3
2	目錄.....	5
3	功能需求	8
3.1	氣動車功能需求與規範.....	8
3.2	系統功能與量化指標.....	10
3.2.1	風扇動力系統.....	10
3.2.2	車體系統.....	11
3.2.3	轉向與輪組系統.....	12
3.2.4	自動控制系統.....	12
3.2.4.1	循跡.....	12
3.2.4.2	煞車.....	13
3.2.4.3	計步.....	13
3.2.4.4	控制速度與加速度.....	13
3.2.4.5	跟車.....	14
3.2.4.6	超車.....	14
3.3	相關性與相依性.....	14
3.3.1	相關性.....	14
3.3.2	相依性.....	15
3.3.3	設計優先順序.....	15
3.4	期末超車策略.....	16
4	設計概念與布置.....	18
4.1	車體設計概念.....	18
4.1.1	底盤規格與設計.....	20
4.1.2	轉向機構.....	24
4.1.3	車輪與煞車.....	26
4.1.4	馬達架.....	31

4.1.5 風扇罩.....	34
4.2 風扇.....	36
4.2.1 理論應用.....	39
4.2.1.1 Control volume analysis	39
4.2.1.2 Actuator disk theory.....	40
4.2.1.3 Blade element (momentum) Them.	42
4.2.2 第一代風扇與參數設計.....	44
4.2.3 設計流程與繪製方法.....	52
4.2.3.1 Q-blade 詳細設計流程.....	53
4.2.3.2 3D 繪圖詳細繪製流程.....	53
4.2.4 各代風扇的比較與改善.....	54
4.2.5 理論推力計算.....	60
4.3 控制程式與硬體設計.....	61
4.3.1 系統電路.....	61
4.3.2 機電元件.....	62
4.3.3 循跡.....	65
4.3.4 程式架構(期中控速).....	66
4.3.5 程式架構(期末超車).....	69
4.3.5.1 控制邏輯.....	69
4.4 期中技術規格.....	73
4.5 期末技術規格.....	74
4.6 期末 BOM 表.....	75
5. 設計分析與驗證.....	75
5.1 分析.....	75
5.1.1 風扇模擬分析.....	75
5.1.2 車體底板應力分析.....	77
5.2 繪製工程圖(組合圖).....	79
5.2.1 車體.....	82
5.2.2 車輪組.....	85
5.2.3 風扇架與風扇罩.....	91
5.3 製造.....	96
5.3.1 扇葉.....	96
5.3.2 車體.....	97
5.4 驗證.....	98
5.4.1 車體.....	98
5.4.2 扇葉.....	101
5.5 實驗.....	102

5.5.1	風洞螺旋槳推力實驗.....	102
5.5.2	風洞風阻實驗.....	105
5.5.3	風扇風速與轉速測試.....	108
5.5.4	車體最大靜摩擦力實驗.....	109
5.5.5	車體側滑之最大靜摩擦力實驗.....	110
5.5.6	紅外線感測器實驗.....	111
5.6	討論.....	111
6	工作進度與分工.....	112
6.1	基本規範.....	112
6.2	工作分工.....	113
6.3	甘特圖與完成圖.....	113
6.4	期中後未來工作與期末檢驗.....	114
6.4.1	車體組.....	114
6.4.2	風扇組.....	114
6.4.3	控制組.....	115
7	心得與專業建議.....	115
8	參考資料.....	118
9.	附錄(程式碼).....	118
9.1	期中測試.....	118
9.2	期末測試.....	130

3 功能需求

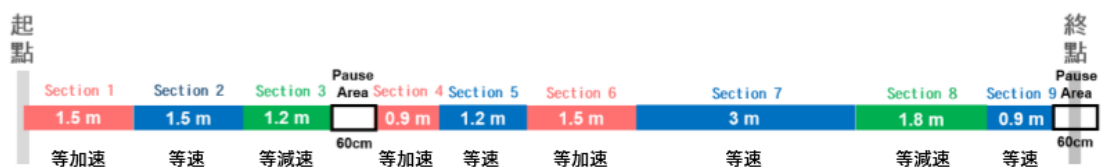
3.1 氣動車功能需求與規範

設計規範：

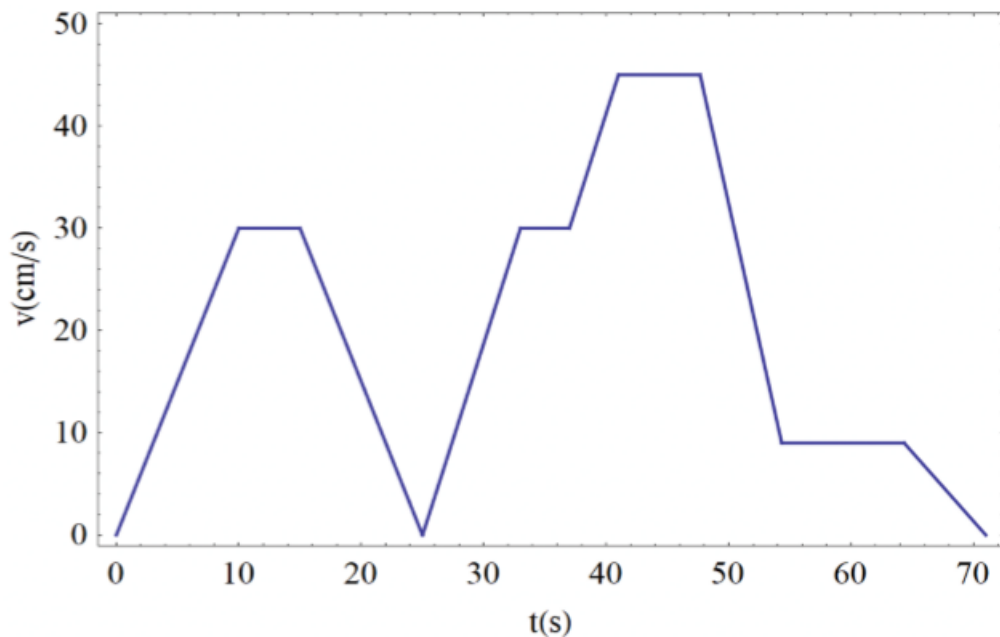
- 以螺旋槳產生之風力來讓車輛達到驅動與制動之所有的動力，螺旋槳需自行設計與製造，螺旋槳需有保護裝置，避免因葉片高速旋轉時脫落崩壞導致人員受傷。除螺旋槳機組外不得有其他動力來源。
- 車輛之煞車方式不限，唯不得由其他車輛提供。
- 整體機電系統需自行配置組裝，可採購市售機電零組件，電池與馬達數量及規格均不限制。
- 成本管控為工程設計的重要一環，本次競賽最終設計成品的材料總成本不得超過新台幣 3000 元。凡規格品元件均需保留原始型號及發票，凡自製品均需有資料佐證，自製元件之材料也須列出，嚴禁元件委外加工。
- 車體大小不得大於 A4 尺寸 (21.0cmX29.7cm)。
- 配重 250ml 的未拆封鋁箔包飲料作為車手重量，其位置與角度可任意擺放，惟需於車內。

期中測試規則：

- I. 賽道全長 14.4m，寬 60cm。測試時間 5 分鐘，車輛全自主循跡，以風力驅動。
- II. 測試過程中車輛出線或以外物碰車，須返回起始點重新開始，期間不停錶。
- III. 不得追跑道兩側線，請以中線為循跡線。
- IV. 前半段賽道未達各段速度標準需返回起始點重新開始。後半段賽道未達各段速度標準不需返回起始點，可繼續跑完後面賽段。
- V. 前半段賽道跑完並進入第一停止區後，可將車輛調整至適當位置後再繼續走後半段賽道，但過程不得進行更換、調整、拆卸、添加任何軟硬體設備等其它非移動車體之行為，否則必須回到起始點重新測試。
- VI. 賽道除了起始點、終點與停止區以外，其餘各處以每 30 公分為間隔設置一速度感測器，並依據速度感測器量測之速度是否符合設定要求進行計分，單趟總分為各段所得分數和。
- VII. 完成賽道後可返回起始點重新測試。取單趟最高分者為期中測試分數。



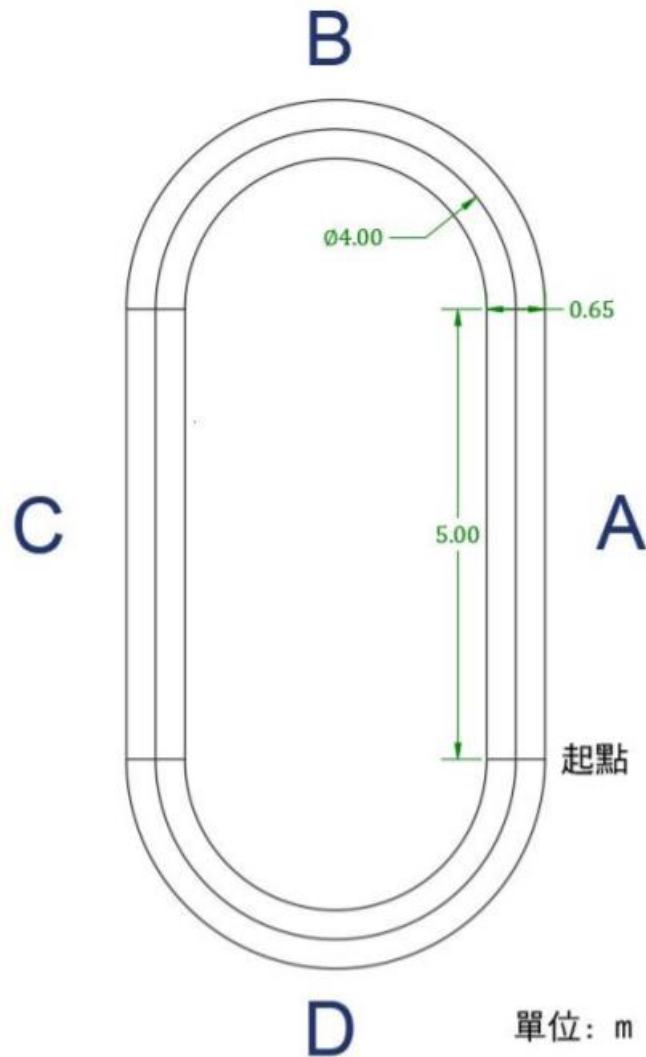
圖二：期中賽道各區段之加速、等速、減速規範



圖三：期中賽道之速度規範（速度對時間分布）

期末測試規則：

- I. 賽道全長約 22.57m，寬 65cm，如下圖(四)所示，分為四個區段(AC 直線段，BD 彎道段)。測試時間 10 分鐘，車輛全自主循跡(或跟車)，以風力驅動。
- II. 驗收任務為兩輛車皆以逆時針方向跑完賽道三圈，後車跟隨著前車，並在過程中完成兩次超車(前後車互換)。
- III. 依據四個項目(循跡、跟車、超車與平均速度)的完成度或優劣進行評分。
- IV. 除風扇產生之風力外，不得以其它動力驅動車輛。
- V. 以後車行走距離除以行走時間作為平均速度。
- VI. 驗收過程中任一車輛投影面積完全沒有在軌道範圍內或以外物碰車，須返回起點重新開始，期間不停錶。
- VII. 若前車領先後車超過半圈，兩車須回起始點重新開始。
- VIII. 兩車至少皆須跑完第一圈賽道才有分數。
- IX. 請以中線為循跡線，不得追跑道兩側線。
- X. 驗收過程中若欲進行更換、調整、拆卸、添加任何軟、硬體設備等行為，必須回到起始點重新測試。
- XI. 完成賽道後可返回起始點重新測試。取單次最高分者為期末驗收分數。



圖四：期末賽道圖

3.2 系統功能與量化指標

本次實務課所需要的氣動車主要功能有五個：氣動、轉向(含超車)、循跡、煞車與跟車(期末)。

首先是氣動功能，由無刷馬達輸出動力，轉動風扇以推動車輛；轉向裝置由伺服馬達驅動前輪機構，使得車體轉向；循跡以紅外線感測器感測地面黑線，輸出訊號至微型控制器以調整前輪轉向機構；煞車裝置則利用安裝在車底板後方的煞車柱，以伺服馬達驅動向地板增加摩擦力，使得車子減速或停止移動；跟車則以超音波感測器讀取前車距離，輸出訊號至微型控制器以調整風扇轉速。

3.2.1 風扇動力系統

由風扇高速旋轉時所產生的推力提供，因期中測試車體需有一定的加速度，由上圖(三)可得知最大加速度為 3.75cm/s^2 ，故希望風扇推力要大、車體摩擦力要小。

- 風扇：所有的動力來源，也是這次實務課的重點。根據學長姐們的報告顯示，風扇的轉速可能高達 10000rpm 左右(但這學期期中後實際沒那麼高)，在高轉速的情況下，為避免風扇斷裂十分危險，在設計時必須考慮扇葉強度的問題。綜合以上，我們希望風扇的功能與設計方向有三：
 1. 足夠推力：風扇需克服輪子與測試場地間的摩擦力與車體的空氣阻力，因此足夠的推力為風扇設計的主要目標。
 2. 推力穩定：氣動車需依循地上的軌跡行進，因感測系統與訊號具有有限性，若風扇所產生的流場太不穩定，可能會影響循跡的效果，也可能會不符合期中各路段的速度規定。
 3. 盡可能低轉速：產生足夠且穩定的推力後，接著需考慮馬達與風扇的整體效率，過高的馬達轉速將導致噪音與震動，因此我們希望在能符合測試的要求內，轉速能越低越好。
- 馬達：提供風扇轉動動力，需要價格合理、扭矩足夠且轉速高。
- 風罩：遮擋在風扇外側，避免高速轉動的葉片斷裂噴出。

量化指標：

- ✓ 風扇不同轉速下的推力
- ✓ 風扇的導流效果
- ✓ 車體重量
- ✓ 車體摩擦力與阻力係數
- ✓ 風扇在特定轉速下的應力分析

3.2.2 車體系統

- 車體：車體需要乘載一定重量、在承載一定風扇推力下不致變形，在轉向時不會因為配重不當而過度搖擺甚至翻覆，以及要有足夠的空間以及適當的收納規劃。

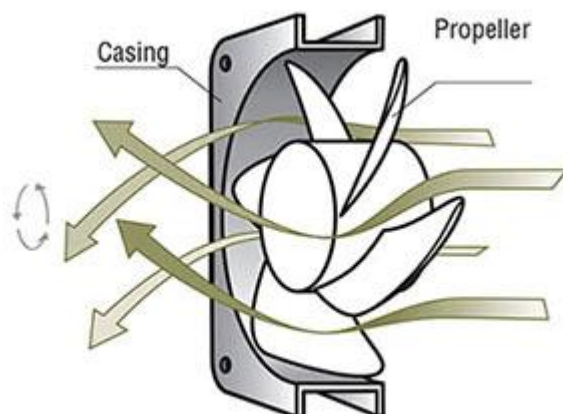
量化指標：

- ✓ 重心位置
- ✓ 應力分析
- ✓ 傾斜狀況

- 風扇罩：對於本次工程實務之課程要求而言，風罩的主要功能為增加安全性，避免風扇意外飛出時會直接打傷人；另外，根據過去學長姊的經驗，良好的風罩設計應同時考慮空氣動力學，如下圖(五)，氣流在經過風扇後會受扇葉的加速而壓縮，若風罩的形狀設計得宜，則可提高推進效率。

量化指標：

- ✓ 風扇罩個別能承受不致斷裂的力
- ✓ 風罩的導流效果



圖五：風扇氣流方向^[1]

3.2.3 轉向與輪組系統

- 轉向：轉向的存在讓車子能隨著循跡的路線行進，藉由控制完成，並配合路線進行適當角度的轉向，同時考量在行進時對車輪進行旋轉帶來的打滑以及不穩定的可能，需有完善規劃來配合控制達到循跡此一目的之完成。
- 輪胎：輪胎需要支持車體的重量，並且維持車子一定的直進性。

量化指標：

- ✓ 胎皮與地面接觸的摩擦力
- ✓ 軸承連接的配合程度
- ✓ 負載後的傾斜角度

3.2.4 自動控制系統

車子每個零件的精確度都與控制息息相關，尤其是當要控制速度與加速度時。由於控制速度是最為關鍵的分數來源，所以我們將會把它作為第一考量，而後續將會是循跡、煞車、計步以及轉向，期中後的順序則再加上跟車。我們這麼安排是因為如果循跡的精確度很高，車子走直線的穩定度會大大提升，對於控制的速度會有極大的幫助。其他的部分也都會影響控制速度的難易度，所以雖然分出了先後順序的考量，但我們認為全部都該做好才能完成一個好的控制系統。

3.2.4.1 循跡

在這次期中測試中，我們將無法以人為操控的方式對車子做任何方向的改變，而是要讓它跟著地板上寬約 2 公分的黑線直線前進，期末則還要考慮彎道的曲率半徑。

經過期中測試後，我們發現循跡是我們這台車最重要的需求，沒有循跡其他事情都不用想。我們的目標基本上跟期中一樣，讓它跟著地板上寬約 2 公分，的

黑線直線前進。困難的地方是我們有直徑為 4m 的兩個半圓，以及加上超車要循到隔壁的線。

量化指標：

- ✓ 車子速度對於轉向的影響
- ✓ 感測器對於黑線的敏感度
- ✓ 取得訊號後的處理
- ✓ 處理後的誤差及轉向控制
- ✓ (轉出去的角度以及時機)

3.2.4.2 煞車

在這次的期中測試，我們的評分標準是控制速度以及加速度，所以要有有效的方式來減速。通常減速都會用煞車來減速(單靠風扇的停止運轉可能不夠)，而機構組經過討論後決議將煞車磨擦地板來進行減速，而煞車的系統我們將採用伺服馬達來控制。在期末測試中，並沒有嚴格要求控速的部分，加上我們跟車的系統穩定，前方的車也還有向後的風扇阻力，並考量程式設計的架構，因此沒有使用煞車系統進行減速，跟車時並不會與前車發生碰撞。

量化指標：

- ✓ 減速的效果
- ✓ 煞車的穩定性
- ✓ 點煞的需要性

3.2.4.3 計步

在這次的期中測試，我們將要在一定的距離內控制不同的速度與加速度，如果加速度與速度夠精準，可以用時間來測量，但現實中有太多不可控因素了。所以我們用 encoder 來計算車子行走的距離，並且計算其速度與加速度，確定車子到達不同的距離，可以送出不同的速度與加速度。

而在期末測試中，除了控速之外，最重要的是判斷直線與彎道的的位置，以及超車的位置，因此計步的功能尤其在計算距離上，格外重要。雖然期末不需要精準地控制速度，但是我們還是需要把速度控制在 40cm/s 以上，才能拿到滿分。我們也需要精準的位移量來幫助我們抓到好的時機來超車。

量化指標：

- ✓ 計步器的精確度
- ✓ 計算的精確度

3.2.4.4 控制速度與加速度

控制速度與加速度是這次最難的地方了。要達到目標結果需要穩定的風扇及煞車，並且運用控制板計算誤差，再來才能輸出優化後的結果。

量化指標：

- ✓ 風扇在不同電壓中的推力
- ✓ 風扇的穩定性
- ✓ 車體的穩定
- ✓ 煞車的效果及穩定性
- ✓ 控制板對於數據的运算速度及精確度

3.2.4.5 跟車

良好的跟車是兩台車的加速度相同，能以相同的加速度與速度進行加速。然而由於風扇、馬達及車體的不同，不太可能以相同的 duty 值跑出相同的結果，所以我們需要超音波測距模組的幫忙。由超音波測距模組測距後，我們可以設定要將風扇的 duty 升高或降低，以達到加速或減速的效果

量化目標：

- ✓ 超音波測距精準度
- ✓ 影響 duty 的數值
- ✓ 風扇穩定性

3.2.4.6 超車

這次的期末有一部份的分數來源是超車的部分，我們要在第二圈內部相鄰的兩個區域個超一次車。超車的部分又包含了兩台車的相對速度，所以兩台車的速度要控的穩定，才能穩定超車。由於我們沒有在側邊裝感測器，所以無法測得是否已經超車，只能靠找好的時機點回到中間的直線。

量化目標：

- ✓ 風扇的推力及穩定
- ✓ 轉出及轉進的時機點
- ✓ 轉出及轉進的速度差

3.3 相關性與相依性

雖然組內分工將整體分為車體、風扇與控制組，但各組許多的設計與條件都是環環相扣的，可能牽一髮就會動全身。因此，需要考慮各個需求之間的相關性與相依性，才能達成有效的協調與整合能力，也需考慮各個設計的優先順序，以利循序漸進地達到期中與期末測試的要求。

3.3.1 相關性

以下探討設計規範、穩定性、車體速度、訊號處理與其他功能間的相關性：

設計規範

直接影響了尺寸與成本，進而影響車體、轉向機構與風扇的大小與精度設計。

穩定性

在馬達高速旋轉下，如設計不良，將可能造成整台氣動車開始震動，因此需要有非常好的穩定性來保持車身各個部件能緊密接合，也與循跡、轉向機構的靈敏度有關。此外，安裝在車子上的電路硬體元件、鋁箔包…等等，也必須固定好，不能從車上震動掉落。

行進速度

- 與循跡的相關性：要維持一定的車速，循跡感測器要夠靈敏才能使車子迅速改變至所需的方向，若是循跡的反應過慢可能會導致車子直接衝出跑道外。此外，行進速度也會影響轉向機構的靈敏度，進而影響循跡的效果。在期末測試中我們的超車策略為在特定位置轉向一角度，並關掉循跡，在循到外側黑線後再開啟直線循跡，因此紅外線感測器的靈敏程度至關重要，需要找到符合感測器靈敏度但又同時不會太慢的車速。
- 與穩定的相關性：穩定性越高，車子的極限速度就能越快。
- 與跟車的相關性：在期末測試中，後車的車速主要與前車有關，因此不考慮其行進速度。但如果前車的行進速度過快的話，可能會導致後車跟車系統的不穩定，整體跟車速度受限於跟車PID系統控制的性質。因此我們希望能滿足期末測試的40cm/s的最低車速即可，降低跟車可能造成的不穩定性。
- 與超車的相關性：在期末測試中我們的超車策略為在特定位置轉向一角度，並關掉循跡，如車速過快，可能會造成前輪轉彎時打滑，進而影響超車的角度，而車速過慢則可能受後方車輛撞擊或跟不上前車。

訊號處理速度

若計算量過於龐大將導致運算時間過長，容易造成系統反應時間延遲，尤其在期末測試更為複雜的架構下。精簡的程式架構較能使各項能力更為順暢。

3.3.2 相依性

培林與輪軸的摩擦力、輪胎與地面的摩擦力與車子前進所受的阻力之間具有相依性。在設計時需要考量各個受力狀況，才能最佳化氣動效率。

3.3.3 設計優先順序

由於整台氣動車包含許多功能，我們需要考慮這些功能的重要優先順序，以及要利於組裝與測試，並且將在各個功能互相不抵觸的情況下，盡量地將所有功能進行最佳改良。

在期中測試前，我們這組考量的優先順序為設計規範(包括車體大小、預算上限)>安全性>風扇推力>穩定度>循跡>(煞車) 期末>超車>跟車。第一步

一定要先遵守課程的設計規範，否則再好的設計也毫無意義。確認所有的規範後，首先是先設計與製造出風扇與簡易的馬達座，以利風扇的測試與後續修改，在過程中因為馬達的高速旋轉，風扇如斷裂飛出，將有非常強的攻擊力，所以不管成果如何，組員與同學的生命安全為優先考量，故在任何測試階段都要使用風罩或其他可以罩住風扇的遮蔽物。再來是風扇推力，使車體確定能達到期中測試所需之加速度要求。接著是穩定度，改良車體的震動、轉向與行進時的穩定度，以利接下來在循跡與行進過程中予以優化。緊接著是循跡，確認車體能穩定地行走並達到所需的推力後，開始進行循跡的測試與調整。最後是煞車系統，也是期中測試最關鍵的功能之一，達到使車體減速與停止的效果，最終完成期中測試的所有要求。

在期中測試後剔除掉煞車的部分(在前述 3.2.4.2 中有所闡述)，並將超車與跟車接續補在其後，會將超車放在前的主要原因是，在各組還沒在期末賽道穩定達到控速、循跡前，我們沒有另一台車能夠進行跟車的試驗，在加上超車的位置與穩定性也會影響跟車的策略，因此優先考量超車的部分，也是期末測試的重點。最後考慮跟車的穩定性，其中的原因也是規定較為寬鬆，但跟車的穩定性也會反過來影響超車是否能成功，因此也是非常重要集大成的功能，由於期末測試的要求為平均距離小於 50cm，因此我們首先將 40cm 做為保守的參考值。

總結來說，首要考慮的是無法更動的設計要求，接著以平穩運行的前提下，突破各種挑戰，最後實現各功能的重複性。

3.4 期末超車策略

創新發想

由於一開始我們並不知道超車可以循外側線，因此有了以下 2 種超車策略：

- [1] 超大與超小車：由於超車的規範為前後車的位置交換，加上並沒有規定超車一定要繞開，跑道的寬度夠寬且車體並沒有高度限制，因此我們想到其中一台車做的非常寬與高，另一台車則做的非常纖小、細長，使得小車能夠從大車的下方直接通過，就像火車過山洞一樣，這樣超車會變得非常簡易，等於是只要調整速度即可完成超車，優點除了不需要寫向外超車的程式、減少不確定性外，跟車也可以相對較近(超音波只要擺設斜的，能照到另一台車輛即可)，因為沒有碰撞的問題。要達到這樣的策略還必須考慮大車循跡的方式，而我們想到在前方設置一可垂釣的紅外線感測器，當小車要超車通過時，停止並將感測器升起。但缺點是要重新製作兩台完全全新的車很費時、費力，並且大車在彎道上可能會有較多的不穩定性，無法高速行駛，小車的設計與循跡時左右的搖擺也可能造成問題，加上第二次超車大車循跡的方式還需進一步設計，因此最終沒有選擇此方案。

- [2] 兩台車並行循中線：兩台車皆向外延伸紅外線循跡模組，只要高度可以錯開即可，優點是超車一樣只需要進行速度的控制即可達成前後車互換，但缺點是循跡的穩定性備受考驗，如過於晃動的話兩台車可能會經常擦撞，而延伸過長的感測器可能也有車體不平衡的問題，因此為了降低困難度，我們最終也沒有選擇此方案。

✚ 實際策略

第三種策略則為最傳統、平易近人的超車思維，即其中一台車向外繞開，以達到超車的效果，加上後來得知超車可循外線，此種方式是最安全、穩定的。在此方案下有幾種情況需要討論：

- [1] 前車讓後車 or 後車超前車？

前者的方式為前車先出去到外車道，並等速或減速，讓車道給後車，後車加速或等速超過前車；後者則為前車相對不變，後車加速至外車道並超越前車後在切回至中線，與平常開車時的超車較像。比較這兩種形式，前者（讓車）的優點有：超車較穩定（速度較慢）、可以順勢藉由後車超音波讀值為 0 進行加速，缺點有：超車可能較為緩慢、後車易與前車擦撞、前車變成後車後不易與前方車輛保持距離。後者（超車）的優點有：超車較為快速，較能符合速度需求，缺點有：超車的不穩定性較高（車速較快）。由於超車的穩定性是我們首要考量的因素，因此我們選擇“前車讓後車”的方式作為我們的超車策略。

- [2] 直線 or 彎道超車？

在直線段超車的好處在於轉彎的不確定較低，但缺點是沒有先天內外車道長短的差異，需利用較大的速度差進行超車，彎道超車則相反。我們最後選擇在直線段進行超車，因為轉彎處在不同的位置打同一角度的轉彎效果都不一樣，不確性過高，因此選擇較保守的直線超車方式。

- [3] 等速、減速 or 停止？

不管是前車讓後車或者後車超前車的方式，慢車都有要不要減速、等速或停止的問題，由於我們選擇的方式為前車讓後車，經過計算兩台車的行駛距離並沒有太大的差異，因此等速並不是非常好的選擇，為了使後車在超車時的穩定性提升，我們設計後車的車速不變，等速直線前進即可，因此前車勢必須進行減速，經過理想的距離計算，後車以等速前進，前車並不需要停止後車即可通過，此外停止後要在加速跟車比較困難，因此最後選擇的是減速的方式。

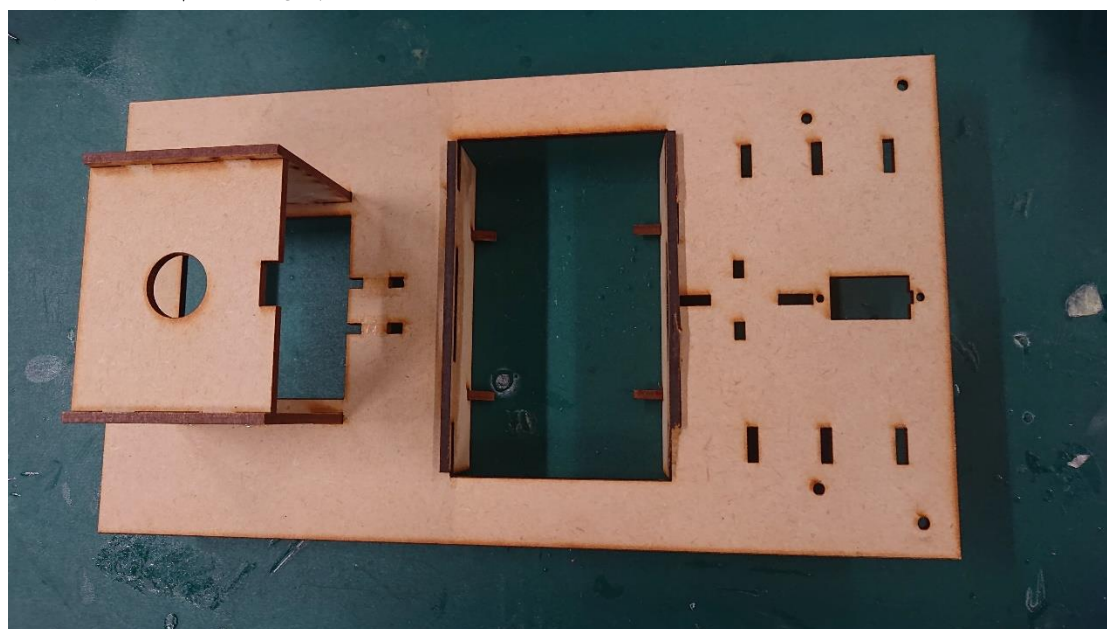
雖然在最後調整的過程中與一開始的構想有所差異，但整體來說還是依此構想作為超車的策略概念。總結來說，我們一、三圈為定速 45cm/s 的跟車（以 40cm 為基準）模式，第二圈則為兩次在直線段的前車讓後車，前車向外車道移動並減速，後車則維持 40cm/s 的速度直線前進。更詳細的超車細節策略將會在控制的設計概念再加以補充。

4 設計概念與布置

4.1 車體設計概念

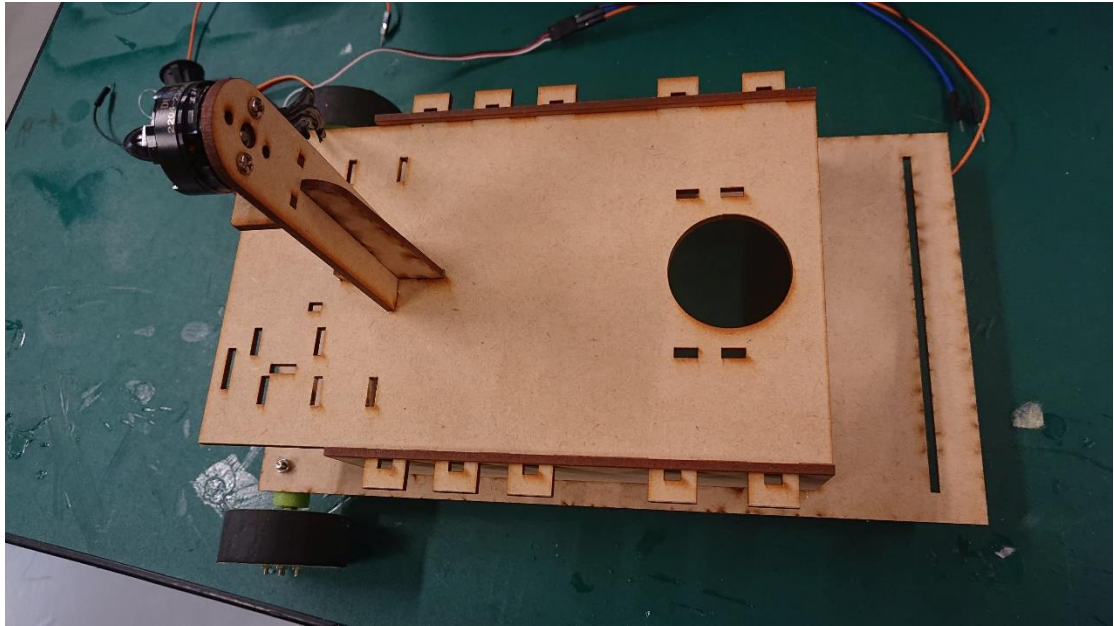
車體的結構主要可分為平行地面的底板以及垂直地面支架所組成。若假設風阻的來源皆由以行進方向為法向量的等效截面所貢獻，也就是車體的垂直部件，那對於平行地面的底板尺寸對於風阻的影響則幾乎可以忽略。因此，底板的設計構想以盡量接近 A4 長方形尺寸為主，如此可增加底盤可供配置的空間，使得後續車子的其他部件如輪子、馬達、風扇等等的安裝更具自由度。而車體垂直部件的部分（不包含風扇支架），則是以能遮蔽並收納各個電子元件為前提之下，越簡潔越好。另外，在結構穩定度方面，為了避免車體因負載造成的彎曲，各個方向的支架設計也在考量之中。

以下附三張車體演變圖，並說明之：



圖六：初代底板

初代底板尚未有第二層的設計，前輪的轉向機構如上圖(六)左方高突之密集板所構成，中間之空位則是設計承載鋁箔包之處，最右側之處則是用來搭載風扇的。



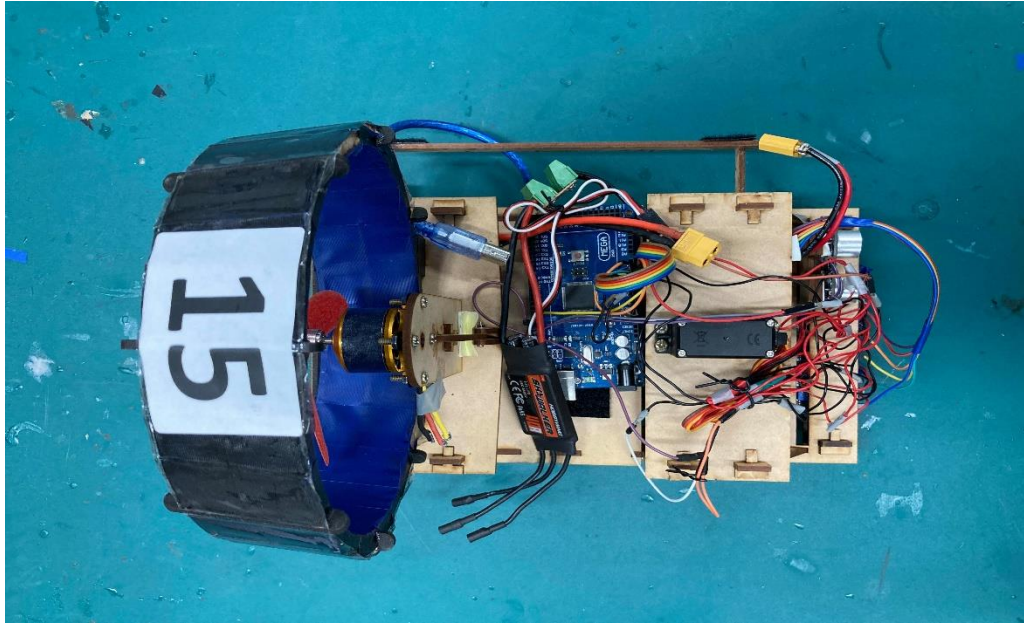
圖七：第二代底板

二代底板之設計則是一改初代底板之簡陋，多了第二層的架構來放置電路板與電池和線路等，分離其與輪胎和風扇之間的擁擠與雜亂，而結構的连接上採用卡榫增加整體的穩定與堅固程度。



圖八：第三代底板(期中測試版本)

相對於二代底板的設計，於上層底板的中间挖出一空間來放置控制所需之電池與線路，而與二代底板的差別在於，藉由將線路與其他相對大體積之零件放入車體中央，減少其對於上層風扇運作時，所間接產生之流體影響，減少渦漩與阻力的部分，同時將整體重心又再度向下，提升整體車子的穩定度。



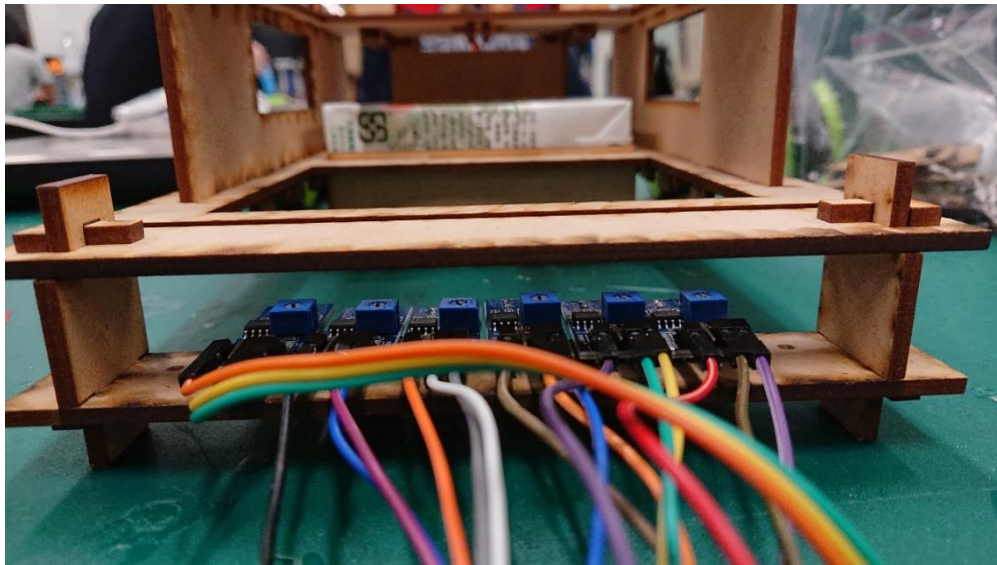
圖九：第四代底板(期末測試版本)

相對於第三代底板，我們選擇將上板拆成前中後三塊來進行安裝，同時降低中間的底板的高度，增加了放置 mega 板的空間，同時在下方安置了行充，減少了上方的風阻，並將其 usb 線收納其內，增加拆卸上的方便與安裝的便利性。

4.1.1 底盤規格與設計

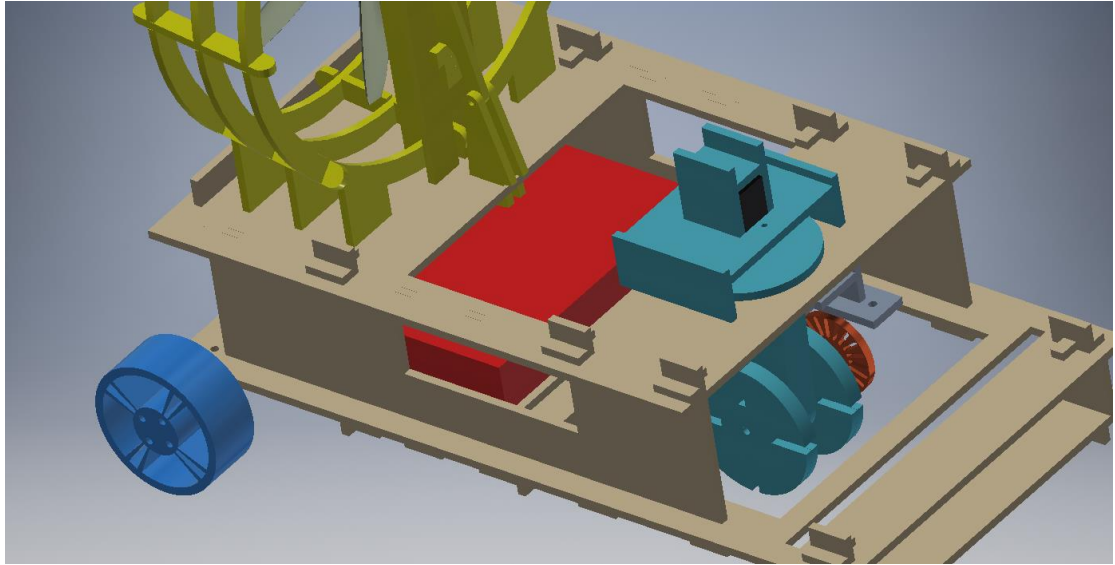
車體主要由五(期末→八)片密集板構成，分別是底板、上板(上前、中、後板)、後板以及二片側板。

底板：



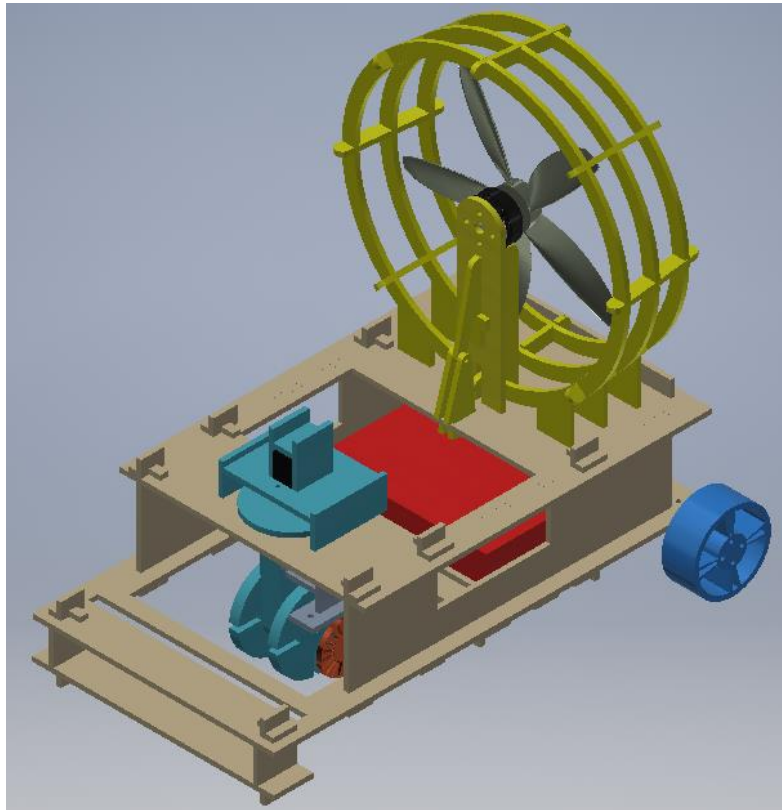
圖十：放置紅外線感測器後之車體圖

前端以密集板向下連接一片放置紅外線感測器的板子，這樣的設計可以調控高度，在初步控制的過程較為方便調整，同時線路除錯時方便檢查，能快速找出問題所在。



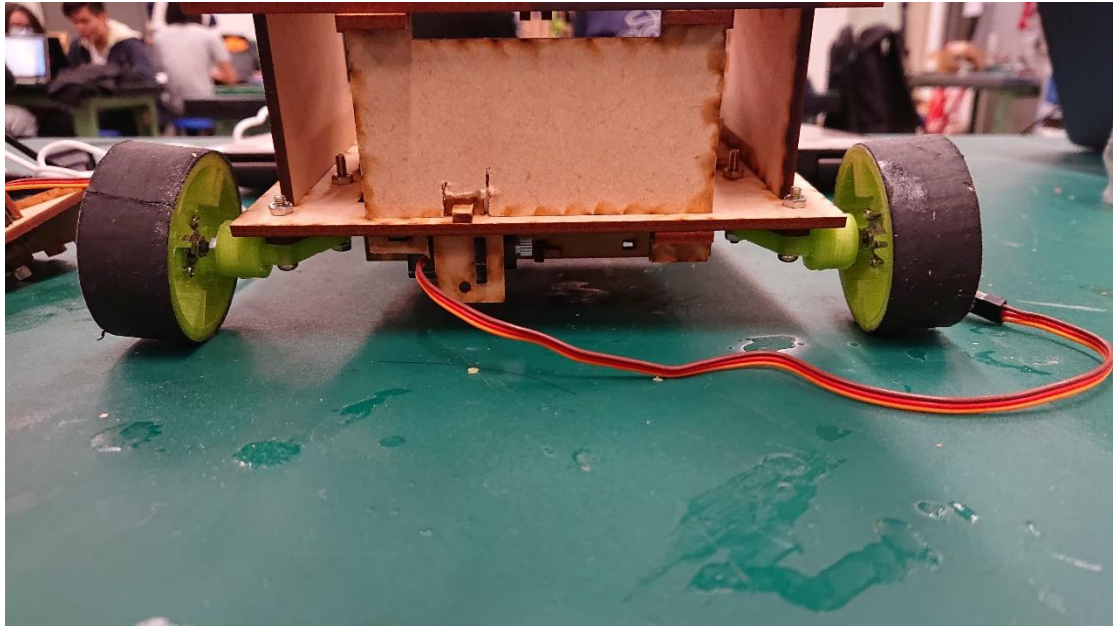
圖十一：放置紅外線感測器前之車體圖

向後一些的前輪，留作前輪轉動的空間，且由於前輪配有測速盤與其相關之電路板，因此需要相對較大的空間。



圖十二：前輪空間與測速盤之配置圖

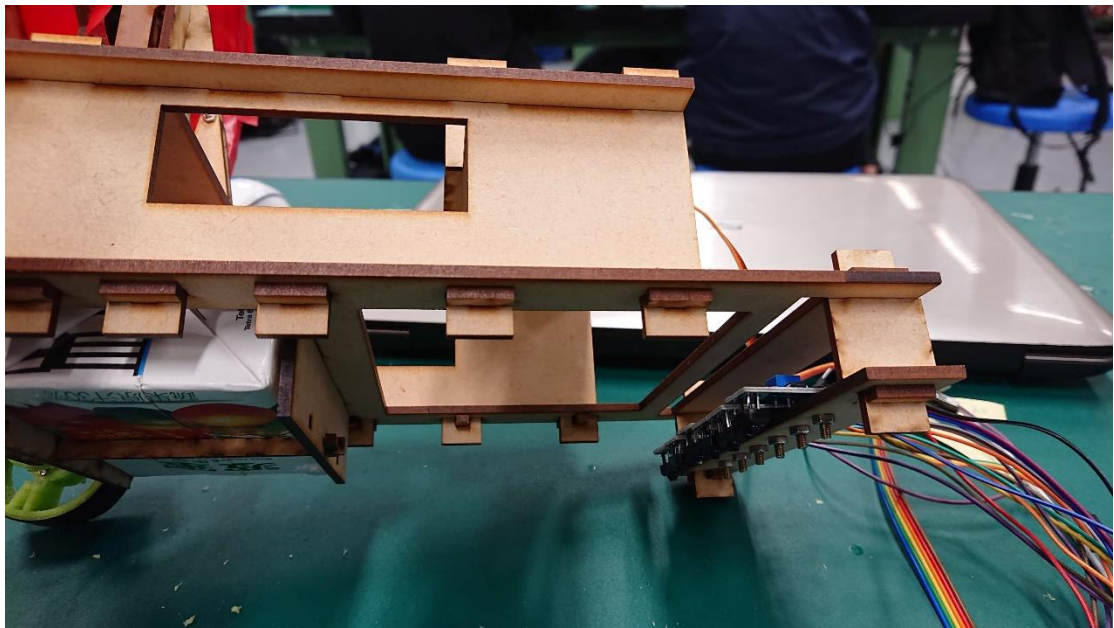
中間長方形向下延伸的設計，用來放置鋁箔包，圖中之紅色長方體為鋁箔包，長邊平行底板短邊，主要是預留足夠的空間放置前輪與後方的配置。



圖十三：車體後方視圖

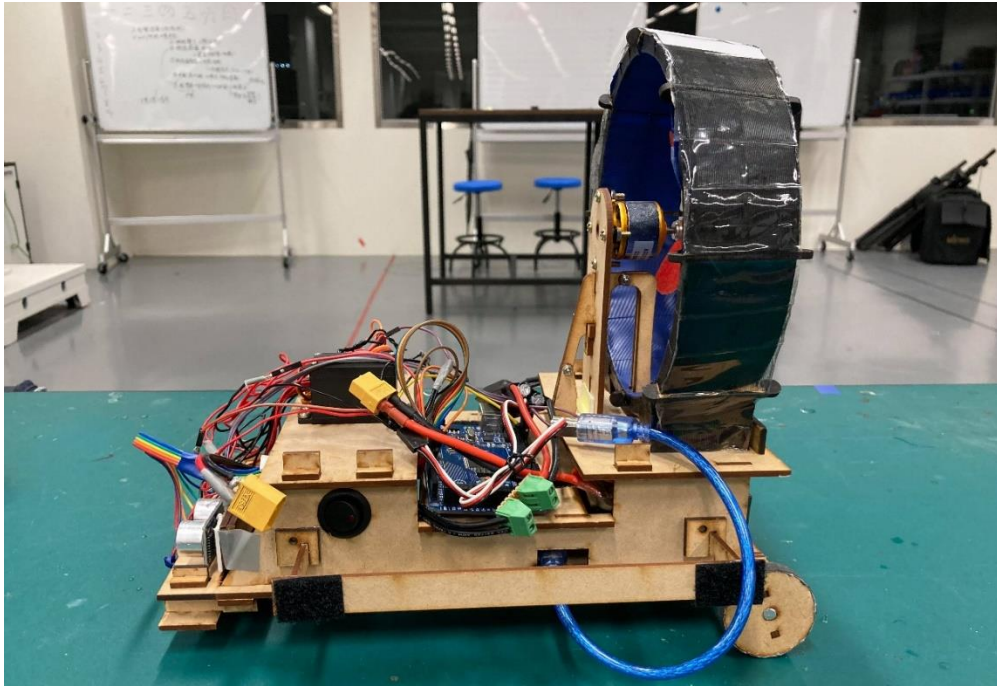
後方除了輪子之外，最後面卡榫的設計是為了緊緊地固定住煞車裝置，該線路來自於煞車裝置之伺服馬達。

側板：



圖十四：車體側板(期中)

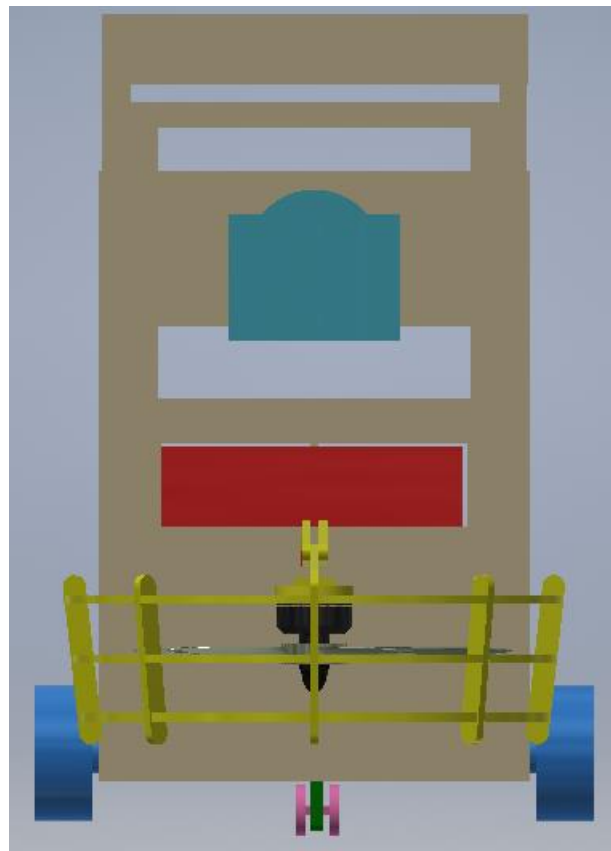
主要是防止車體在長邊方向上彎曲，並且向上墊高致使下方有剩餘空間收納其他物件。與上、底板的固定多增加了卡榫設計，同時側邊的開口允許杜邦線與電路線可伸出至風扇與煞車還有轉向機構處進行連接控制，且在需要調整電路板與內部重量配置時可以更方便調整與處理。



圖十五：車體側板最終板(期末)

後代車體側板的空間，由於前代側板在板件中挖空，在插線與安置零件上會很卡手，所以決定移除，然後另立一個全新的平面，來放置零件。

上板：



圖十六：上板(灰色)示意图

前面的圓孔用來連接前輪，並且固定馬達架使用。後方則是固定風扇架與風扇罩，並在最後面作一支撐板與底板連接。

整體車體結構與期中前相同，比較大的改變是由上圖(十六)可以看到，將上板的中間改為向下凹下的結構，為了最大化車體可以應用的空間，又不希望風扇的高度過高或物體擋到風扇的流體分布，因此選擇此方式，而底板與中間凹下得上板間的空間，剛好可以置放行動電源，而上部置放 mega 板也不會使得風遭受太多阻礙。

4.1.2 轉向機構

在設計轉向機構上的考量，主要分為轉彎幅度與穩定性和機構之簡易程度，在轉彎幅度上，儘管期中為一直線測試，並不包含大幅度轉向，但是考慮到日後其他障礙的可能，因此在這之上的顧慮，仍然納入設計當中，而穩定度的部分，主要避免車子本身在轉彎時產生之扭矩導致不穩定甚至翻車的結果，同時不穩定之晃動也會導致控制上的困難，特別是控制系統回饋時，若循跡時產生晃動，而反向拉回車頭時，會導致車身的晃動與速度的不穩定，更可能導致車子控制失敗而失控暴衝出跑道的可能，綜合以上所述，穩定度也被歸類在設計轉向機構上的首要考量，而機構的簡易程度除了能減少在控制上的負擔，同時也能節省車體空間並降低重量，同時減少桿件傳遞間能量之耗損，提升轉向時所傳到之實際扭矩，因此機構之簡易性也成為本次設計時的要點。

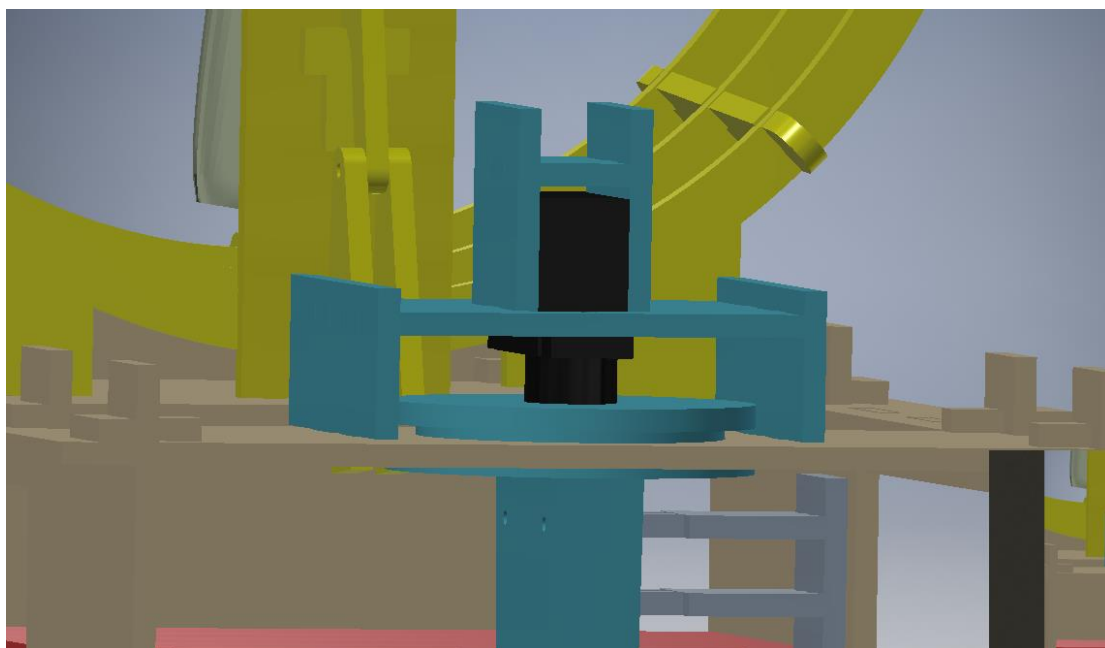
◆ 初代

我們最一開始的想法是選用三輪作為車子本身的考量，而轉向系統上我們採取的方法是前輪(單輪)使用四連桿機構轉向，同時後輪(雙輪)使用阿克曼連桿作為我們轉向上的實際設計，而此設計的主要目的是在於，考量到以往有大幅度轉彎與超車的選項，在設計上應選用轉彎幅度大，且簡易之機構，因此我們讓前輪使用四連桿機構，搭配伺服馬達進行轉彎上的控制，同時我們也在不少現代的汽車上發現了後輪轉向系統，透過後輪旋轉相對較小之角度即可幫助車子順利超車，同時扭矩對車體本身帶來之晃動也會隨之減少，對穩度性而言，是個很棒的設計，因此納入了初代的設計考量。

◆ 第二代

而在實際設計轉向機構並付諸實行後，我們發現到前輪原先之四連桿設計未能良好地傳動馬達轉動產生之扭矩，仔細檢查後發現是 3D 列印之桿件與桿件間之連接處在馬達旋轉時，會產小微小之變形，同時螺絲未能與桿件良好地貼好產生之微小晃動，由於本身連桿設計之轉向角度為左右各 45 度，導致雖然馬達在轉，但前輪仍維持原地不轉動的情形，因此在第二代的前輪轉向設計，改為直接使用伺服馬達連接前輪上方；此外，在控制系統上我們發現到同時調整前輪之轉向與後輪之轉向對於控制系統負擔過大，難度也較高，就雙轉向系統來說，除了本身對於轉彎所對應之機構轉向幅度變小，導致伺服馬達可能忽略訊號，同時我們發現前後輪的轉向並未同時啟動，而導致訊號回饋時難以調整參

數，且考慮到期中並未需要大幅度之轉彎，因此第二代的轉向系統僅留下前輪之轉向系統。

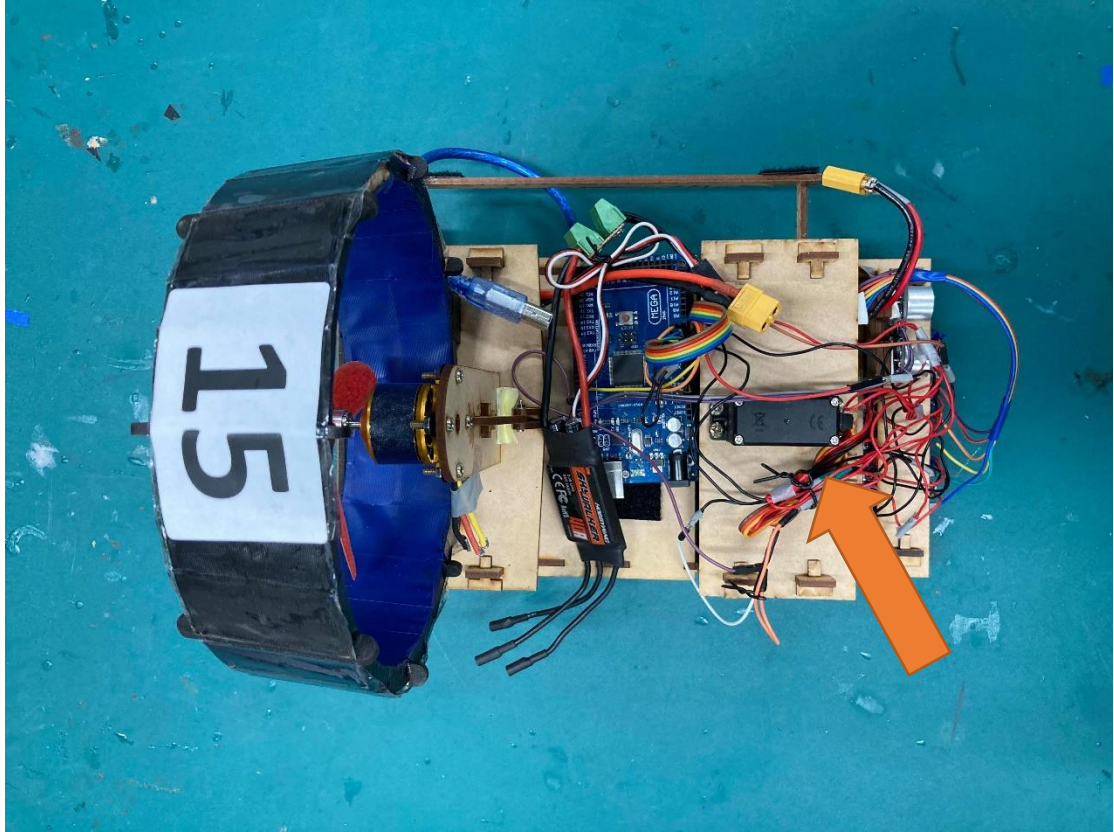


圖十七：前輪轉向系統伺服馬達之配置

藉由馬達轉軸直接連接到前輪垂直地面之方向，直接利用馬達自身的舵機帶動前輪的轉向，同時藉由前輪上方之中軸以類似自行車龍頭的方式夾持上板，避免前輪產生晃動的可能性，同時捨去了原先初代連桿桿件的部分，增加了車體空間，也避免掉桿件之間傳動時帶來的能量耗損。

第三代：

原本採用 MG90S 作為主要轉向之舵機，但是在期中測試之後，我們發現由於原本使用機種之扭矩太小，以至於轉彎控制的不太好，在循跡上，容易有不穩定的狀況，於是決定採用同樣的安裝方式，並配合上 MG996R 作為主要之舵機，相較於 MG90S，MG996R 擁有較大之扭矩，且擁有相對精準之角位，可以在循跡上有更好的表現。



圖十八：前輪轉向系統伺服馬達之配置(MG996R)

與先前一樣的轉向系統，利用 MG996R 直接連接於前輪上，改善不必要的力之耗損，也能相對簡單地控制轉動之角度。

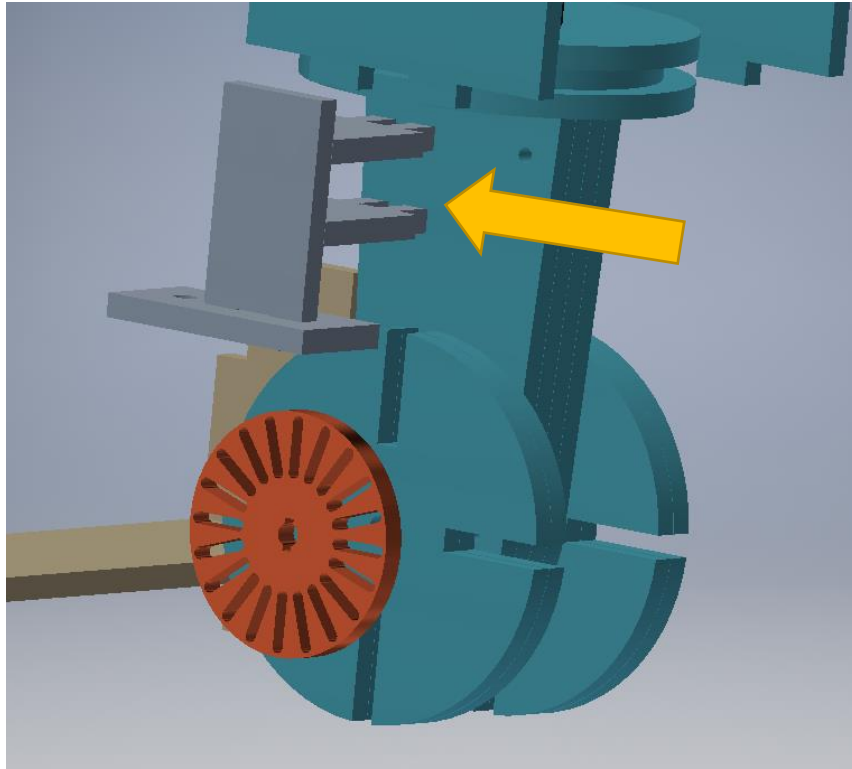
4.1.3 車輪與煞車

車輪的選用為了配合設計的底盤高度，加上市售輪胎可能有打滑的情況，於是決定自製輪胎。

- 前輪

前輪的設計初代為 3D 列印產出，為一類似於自行車輪胎的結構，並在車輪左右各放一微型培林，穿過一 3D 列印之車軸，作為前輪，同時為了轉向的部分，於上方設置一圓形平坦盤面與密集板接觸並發揮支撐作用，上方保留與最初四連桿相連接之螺孔，可連接四連桿於一伺服馬達 MG90S 進行轉向上的控制，為初代前輪。

後考量到 3D 列印在尺寸微小時之精度極差，導致零件彼此間配合的不穩定性，尤其是在組裝前，常須先使用銼刀進行打磨，後使用砂紙進行表面光度的處理，才能進行組裝，特別是培林與 3D 列印零件之間的配合，礙於 3D 列印餘料的存在，以及微小孔洞難以進行打磨，導致組裝不順，或是組裝後之情形未達理想，於是採用密集板的方式，透過與車軸中央放置一微型培林，然後左右各一小圓板作為輪胎，輪胎表面黏上網球拍所用之握把布。



圖十九：前輪與測速盤配置圖

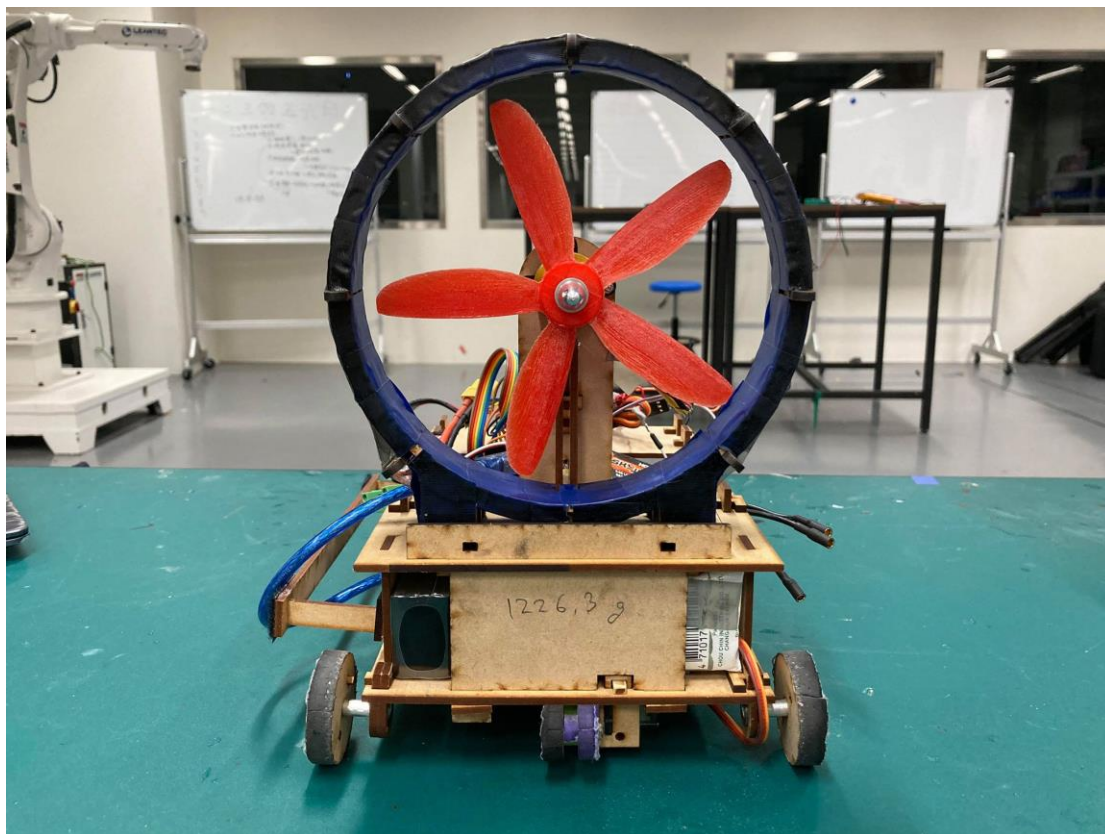
期中後的版本除了將 MG90S 伺服馬達改為 MG996R 之外，最大的不同在於我們將前輪的握把布輪胎皮給拆掉，一開始會黏握把布的原因是怕前輪在巡跡轉向的時候會打滑，但我們發現車輛得啟動摩擦力有點過大，馬達需要很高的轉速下才能推動車子啟動，為避免期末車子啟動不了，與馬達高轉速下的危險性與噪音，我們最終選擇將握把布拆掉試試看。由於期末測試場地還加上了彎道與超車的部分，有打滑的危險，但經過實際測試，並沒有打滑的情況，除此之外也使得整體摩擦力降低，有效降低馬達的轉速，並且前輪的伺服馬達也較易驅動車輪轉向，轉向機構的誤差較小，也較穩定。

● 後輪

輪胎以 3D 列印產出，並且搭配連軸器與輪胎以螺絲相接，如此可確保輪胎與輪軸的垂直度有一定的品質，連軸器與螺絲、螺帽加上墊片鎖緊，以螺絲作為輪軸，可以簡單地並且提供足夠的摩擦力使得輪軸與輪子同時轉動，螺絲尾端接上一個培林，放入固定座之中，再將固定座與底板相接。胎皮的選用則也是網球拍的握把布，與地面接觸的摩擦力以手壓輪子實測的經驗，還算足夠。

二代後輪：

由於期中前所採用之 3D 列印輪胎，除了尺寸精度的差異外，輪胎本身也因為輪軸太小，而有明顯之外傾角的存在，使得輪胎本身並非垂直於地面，於是決定在期末時選用尺寸較大之培林，同時使用較粗之鋼棒作為車軸，大幅減少彎曲之程度，且本次改換使用密集板而非 3D 列印作為輪胎，大大地減少成品之誤差，使得新的後輪能夠很好地與培林貼合。



圖二十：二代後輪圖

雖然採用較粗之車軸作為支撐，車輪本身之外傾角仍然存在，但相對於原先之情況，摩擦力較小，車輪本身足夠滑順，大幅減少啟動時所需之風扇動能，算是一大優點。

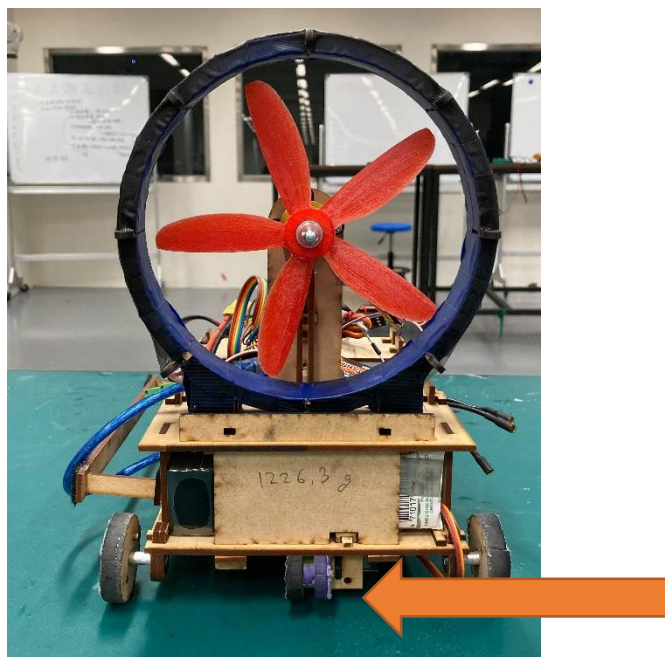
● 煞車

最初的煞車裝置考慮有以下三種：

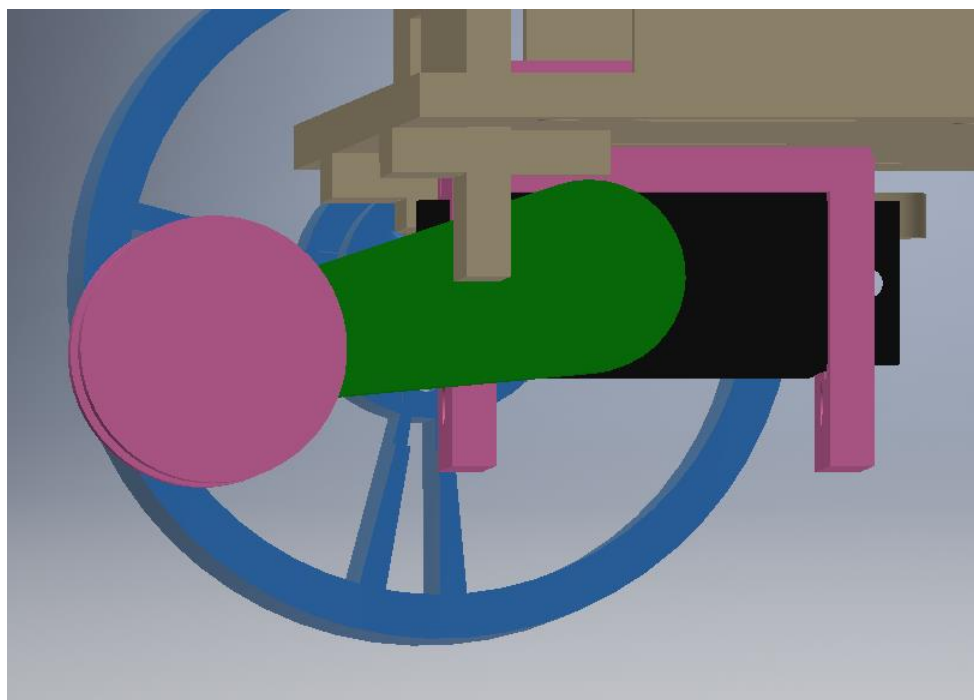
1. 摩擦輪胎表面：直接使用一摩擦塊藉由伺服馬達之配合來接觸輪胎表面。
優點：機構簡單且容易製造，煞車力道理論上也維持定值，在等減速的部分能滿足所需。
缺點：考慮到與輪胎表面接觸之材質的問題，過硬可能直接對輪胎磨損，而太軟又無法產生足夠煞停之力道，且煞車力道對於兩個後輪可能並非等值，從而產生扭矩或是速度差，對於穩定性構成威脅。
2. 摩擦車軸：藉由伺服馬達對車軸直接進行夾持來達到減速的方法。
優點：能避免對輪胎之磨耗，製造容易，產生穩定之減速力道。
缺點：由於車體的設計，只有前輪有夠長之車軸來讓煞車能夠進行發揮，而前輪又肩負轉向的功能，因此在此之上，對於該設計不予考量。
3. 摩擦地面：藉由一機構搭配伺服馬達來摩擦地面進行減速。
優點：機構簡單且製造容易，煞車力道比起前兩者更容易進行調整，同時力道也維持定值。

缺點：在地面磨擦的部分可能力道不足，且在非均勻地面時，可能產生不同力道，而且須對不同地面材質進行分析。

最後我們選擇了摩擦地面作為本次的制動系統，雖然有力道不足的問題，但我們實際測試，發現其實效果不錯，甚至於當摩擦之機構同時與地面還有車體本身接觸時，能夠立即煞停，而且藉由調整與改良，未來可以藉由改變不同接觸面積的大小，進行力道上的控制。



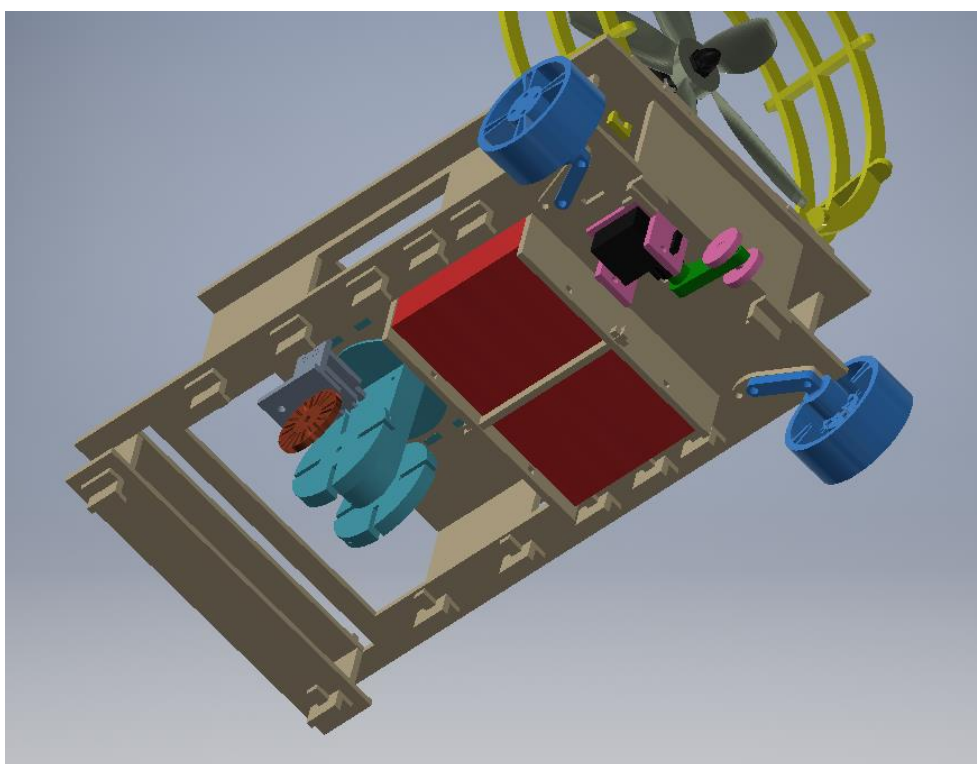
圖二十一：煞車裝置配備之伺服馬達與位置



圖二十二：煞車裝置圖

圖中黑色處為一伺服馬達，而綠色部分為 3D 列印的支架與聯軸器的桿件，粉色部分則是密集板製造用來摩擦地面的部分，透過馬達旋轉來改變摩擦力的大小，當旋轉至一定角度時，粉色區塊會與車體還有地面同時接觸，產生一極大阻力，幾乎能夠立即煞停車體。

煞車裝置直接使用 MG90S 裝上連軸器，加上一個自製的 3D 印製的支架，用來固定與地面摩擦的零件。支架中間留有一個槽，可以使摩擦零件固定在上面，但又保有徑向的自由度，在煞車啟動後，藉由馬達不同角度的控制，使得摩擦零件處在不同徑向方向，此時摩擦力因正向力隨著徑向方向改變而變化，可以漸進地調控煞車力道(圖二十三)。摩擦零件的組成由密集板製的圓版以及作為固定並連結馬達支架的方形長軸所組成，上面貼上網球握把布，增加接觸地面時的摩擦力。

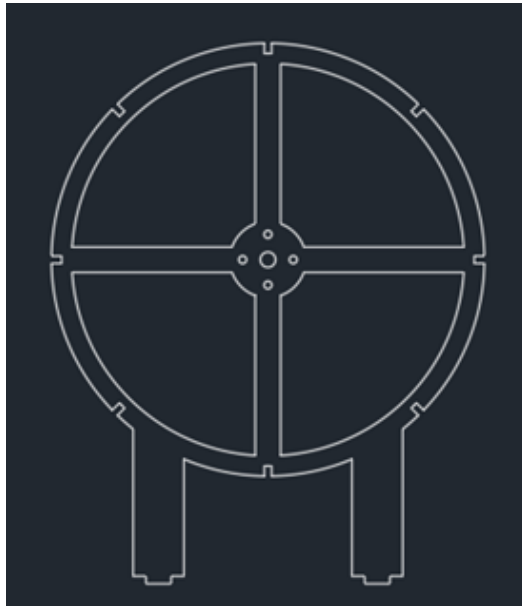


圖二十三：車體底部示意圖

藉由將煞車裝置與鋁箔包還有其他中量相對大之物體放置於車體中央，來維持平衡，當煞車啟動時，能避免對於車體本身產生一太大的扭矩，導致車體的左右不平衡，甚至是翻車的慘況。

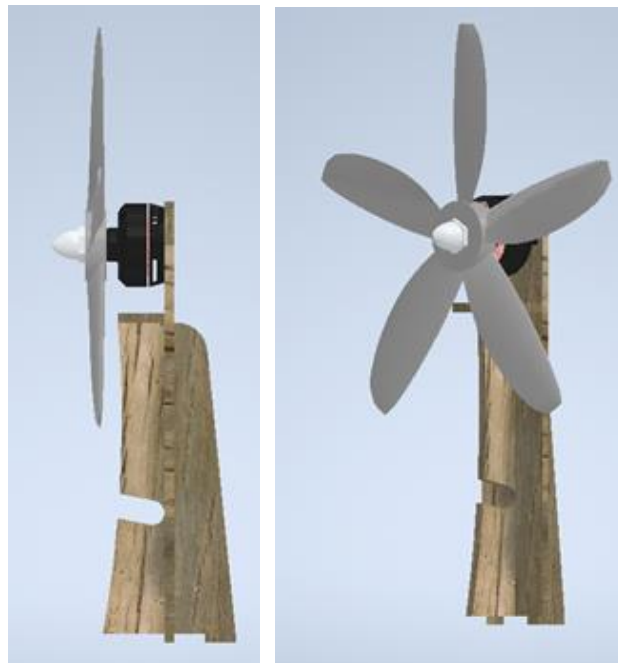
4.1.4 馬達架

◆ 第一代



圖二十四：以風罩固定馬達示意圖

一開始討論要如何固定馬達時想到了兩種方式：第一種為直接向下延伸固定在底座上、第二種為直接將馬達固定在風罩最後一層上，如上圖(二十四)。但我們覺得在氣流通過的區域內應該盡量減少機構的阻擋，以免氣流不順產生紊流而影響風扇表現，因此最後採用的是第一種設計，如下圖(二十五)：



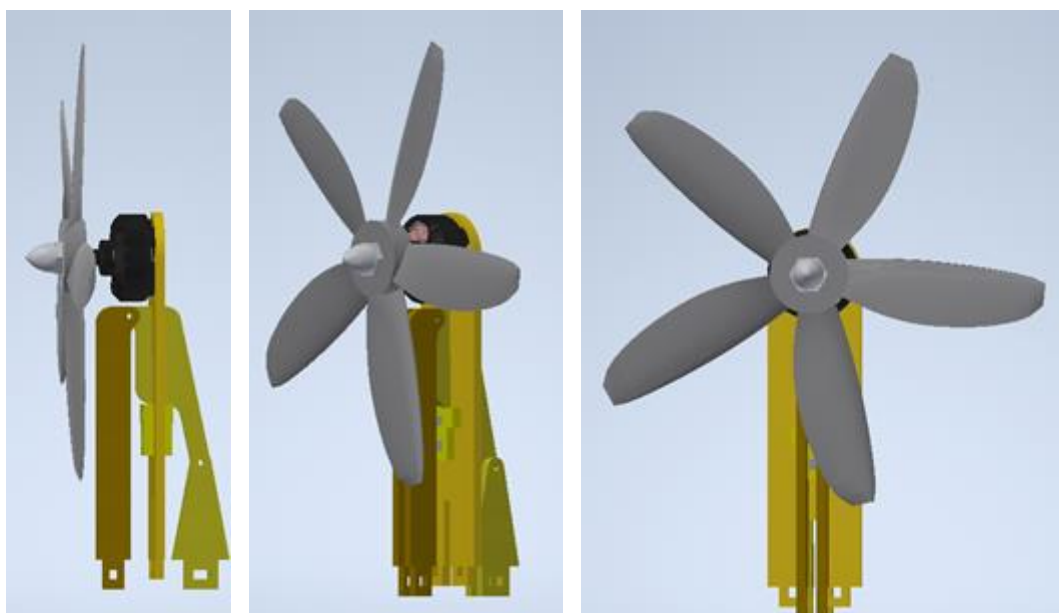
圖二十五：馬達架第一版

因為希望馬達與風扇不要對馬達架造成太大的向前力矩，因此將馬達架設計成直立式，然而如果希望風扇側面的位置恰在風罩正中間，馬達架就會和風罩的位置重疊，因此只好在馬達架前面挖了一個凹槽。

缺點：

首先這個具凹槽的結構從力學看來強度不太理想，外觀上也不太好看；其次這個一體成型的設計導致它在測試時震動非常嚴重，雖然震動原因也可能是風扇偏心所造成，但我們還是希望能讓馬達架有一點避震效果；最後則是和風罩一樣的問題，與車底板銜接的部分容易晃動與鬆脫。

◆ 第二代



圖二十六：馬達架第二版

第二個版本的馬達架，主要的改變是將一體成形的結構改成四連桿機構，因為我們期待馬達的震動能經由四連桿機構被吸收掉一些，以此達到避震的效果，而改成連桿時也順便解決了先前需要在結構上挖槽的問題；另外和車底板的連接也改成插銷以增強結合力。和風罩的組合圖如下圖(二十七)所示。



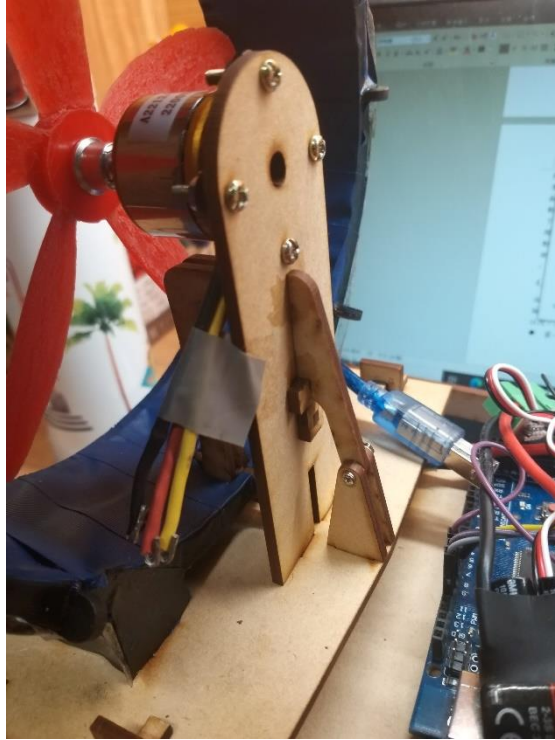
圖二十七：風罩與馬達架最新版組合圖

缺點：

避震效果仍不盡理想，這方面之後須持續調整。

◆ 第三代

期中測試過後原本想繼續研究如何把避震做得更好，但首先發現原來使用的連桿方式反而會讓機構接合處不穩而震動得更嚴重，如果要用這個方法避震應該要再加裝彈簧機構；但是我們在測試途中也同時發現其實只要把馬達主架和底板固定牢，震動的情況就會改善很多，並不像一開始擔心的那樣會讓整台車晃動的很厲害，因此最後決定直接把底板和馬達主架的固定方式由原本的直接向下插入固定、增加一個從底下穿過的插銷，並把四連桿之間固定用的螺絲與螺帽之間用墊片防止鬆脫；尤其在後來換新馬達後，整車行進與風扇運轉時的震動幅度和先前相比就大幅降低了。除此之外，只有將與無刷馬達的螺絲孔處有做尺寸上的修改。

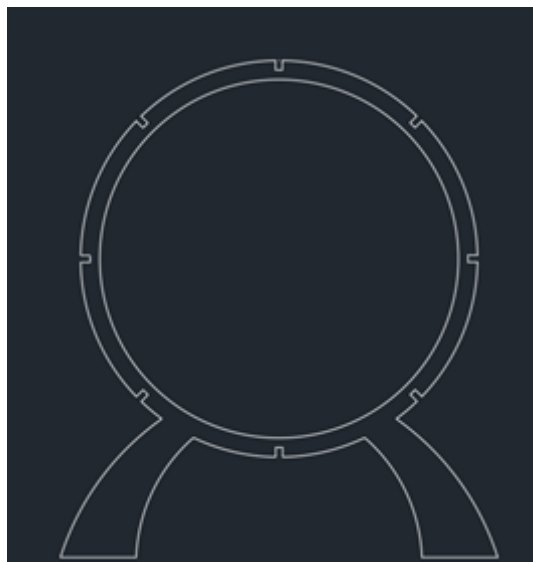


圖二十八：最終的馬達架樣式

4.1.5 風扇罩

由於尺寸限制為一張 A4 紙的投影面積，也就是寬度必須小於 21 公分，經過討論，本組將風扇的直徑大小定在 14 公分、並決定將扇罩的外徑控制在 18 公分以內。

◆ 第一代

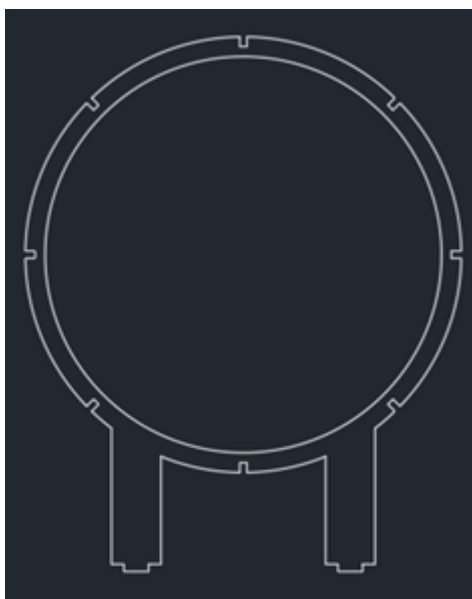


圖二十九：風罩第一版

由於上述考量，在設計第一版風罩時決定採取喇叭狀設計，分別使用 16cm、15.5cm、15cm 當作風罩內徑，讓入風口的直徑大於出風口，希望以此讓風扇得

到較好的效率；支腳則是考量到穩固性而設計成八字形。

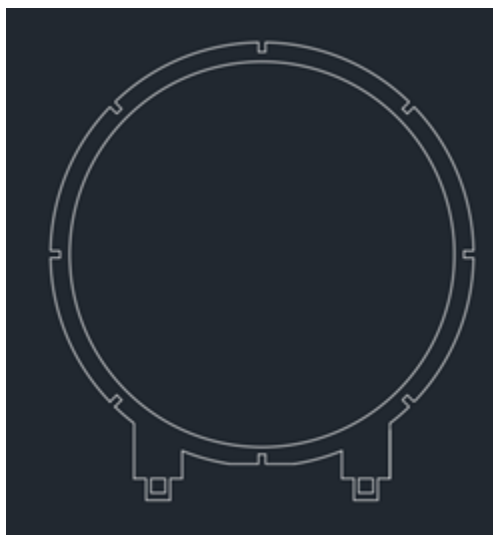
◆ 第二代



圖三十：風罩第二版

和底盤組討論過後，將支腳的部分改成直的，以控制車體尺寸與空間；另外我們後來發現如果風罩縮口率過大，反而會阻礙氣流流出、進而影響風扇效率，因此維持原來的口徑、但把三層風罩之間的距離拉開，降低風罩的錐度、同時也因包裹住風扇的幅度變大而增加一點安全性。

◆ 第三代



圖三十一：風罩第三版



圖三十二：風罩第三版組合圖

第三版的風罩是是期中與期末的最終版，主要是為了和車底板連接更順利而做了調整：為了改成連接車底板的第二層且不影響高度而縮短了支腳；不希望風扇太高是為了減少風扇轉動時造成的車體震盪與前輪增加摩擦力所帶來的影響。第二個調整處是改變和車底板連接的方式：由於使用第二版測試時風罩的晃動有點嚴重，於是改成卡榫和插銷的方式，讓風罩不會因為震動而鬆脫。

4.2 風扇

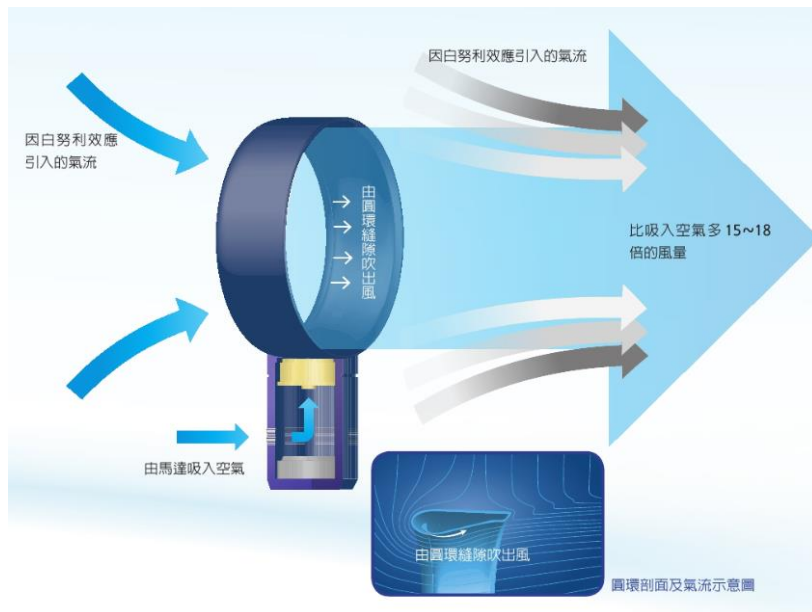
風扇是這次實務課程最重要的元件之一，是所有的動力來源，它是一個結合我們過去所學的課程：流體力學、風扇理論、量測原理與機工實驗…等多種專業知識結合的產物，需事先考慮許多參數設計，例如翼型、攻角、馬達轉速、推力、風扇尺寸、弦長與材料…等等的問題，並經由繁瑣的計算來求得所需要的參數，也是分析、實驗與理論最多的一個項目。在此，先不考慮可行性，我們比較了兩種較為常見的風扇形式：

1. 無扇葉風扇

由下圖(三十三)可觀察到，無扇葉馬達也不是完全沒有馬達與風扇，而是由下方的入風處，經由馬達吸入空氣，再將空氣推升到環狀管道內，此時藉由環內、外的壓力差，並應用伯努力定律，帶動後方周圍空氣往圓環前方流動。

優點：非常安全，沒有扇葉斷裂飛出的危險性、氣流較穩定

缺點：整體結構設計與製造困難、推力不夠、馬達需要非常高的轉速



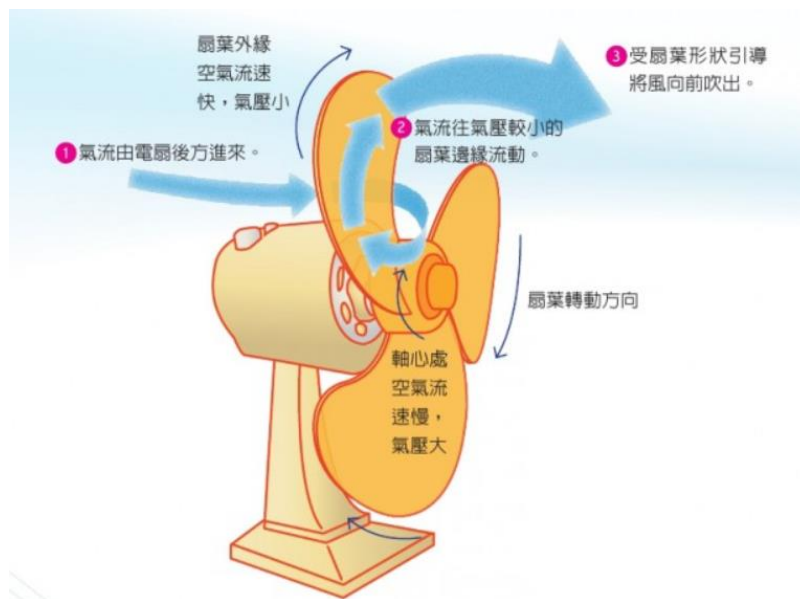
圖三十三：無扇葉馬達原理圖^[2]

2. 扇葉型風扇

設計時需設計半徑、攻角、翼型…等等參數，利用馬達轉動風扇，吸入空氣使其加速排出，並利用作用力與反作用力的原理，推進車體。

優點：推力較大、製造方法多元、成本較低

缺點：風扇容易變形、有較嚴重的振動問題、有安全性的疑慮



圖三十四：一般扇葉型風扇^[3]

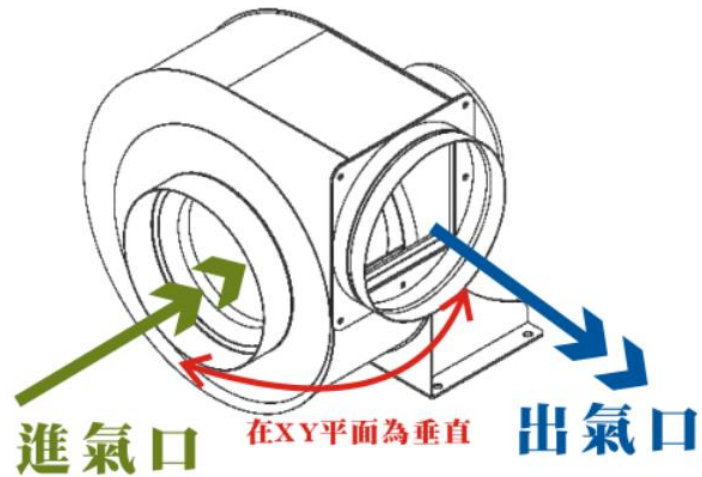
將上述兩種風扇類型進行需求比較並整理於下表(0~5分，數字越大越利於選擇)：

	推力	設計難易度	扇葉強度	出風穩定度	成本	總分
無扇葉	2	2	5	3	1	13
扇葉型	4	4	2	2	3	15

因此，我們選擇以扇葉型風扇做為這次風扇種類的選擇。然而，扇葉型風扇常見的種類又可細分為離心式與軸流式，所以我們接著比較兩種形式的優缺點：

➤ 離心式風扇

運轉時，氣流由葉輪中心被吸入，葉輪使氣流帶有離心慣性，沿葉輪方向放射流出，順著螺旋外殼被導出風扇。



圖三十五：離心式風扇^[4]

優點：產生的風力較強

缺點：不易製造、空間配置較困難

➤ 軸流式風扇

氣流垂直葉輪流動，如一般家用電風扇即是。



圖三十六：軸流式風扇^[5]

優點：風量大、安裝簡單、效率高、可高速運轉

缺點：噪音高、風壓小、無法做長距離的氣流輸送、葉輪較易毀損

將上述兩種風扇類型進行需求比較並整理於下表(0~5分，數字越大越利於選擇)：

	離心式		軸流式	
風力強度	過強	2	適中	4
製造難易度	複雜	1	簡易	3
佔用空間	空間配置較難規劃	2	空間佔用較小	4
得分	5		11	

綜合以上，我們將使用一般扇葉型軸流式風扇做為這次風扇的設計種類，並考量到加工的精密度與風扇本身的強韌性為主，再以軟體分析扇葉性能。

4.2.1 理論應用

除反覆利用實驗驗證最佳結果外，藉由理論計算適合的參數，也能提高效率、降低成本與節省許多時間。以下統整與簡介以下三種可用於風扇設計的理论：

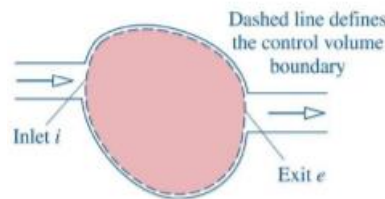
4.2.1.1 Control volume analysis

在此我們列出 2 項重要的結果：

1. 質量守恆

$$\left[\begin{array}{l} \text{Time rate of change} \\ \text{of mass contained within} \\ \text{the control volume at time } t \end{array} \right]_{CV} = \left[\begin{array}{l} \text{Time rate of flow} \\ \text{of mass in across} \\ \text{inlet at time } t \end{array} \right]_i - \left[\begin{array}{l} \text{Time rate of flow} \\ \text{of mass out across} \\ \text{the exit at time } t \end{array} \right]_e$$

$$\frac{dm_{CV}}{dt} = \dot{m}_i - \dot{m}_e$$

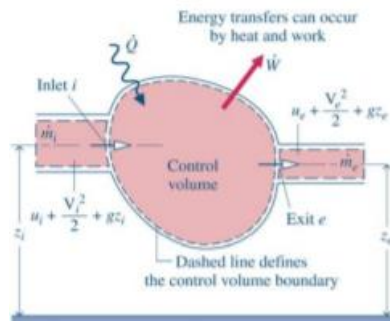


圖三十七：簡單系統的質量守恆^[6]

2. 能量守恆

$$\left[\begin{array}{l} \text{Time rate of change} \\ \text{of energy contained} \\ \text{within the control} \\ \text{volume at time } t \end{array} \right]_{CV} = \left[\begin{array}{l} \text{Net rate of energy} \\ \text{transferred in as} \\ \text{heat transfer} \\ \text{at time } t \end{array} \right]_Q - \left[\begin{array}{l} \text{Net rate of energy} \\ \text{transferred out as} \\ \text{heat work} \\ \text{at time } t \end{array} \right]_W + \left[\begin{array}{l} \text{Net rate of energy} \\ \text{transfer into the} \\ \text{control volume} \\ \text{accompanying} \\ \text{mass flow} \end{array} \right]$$

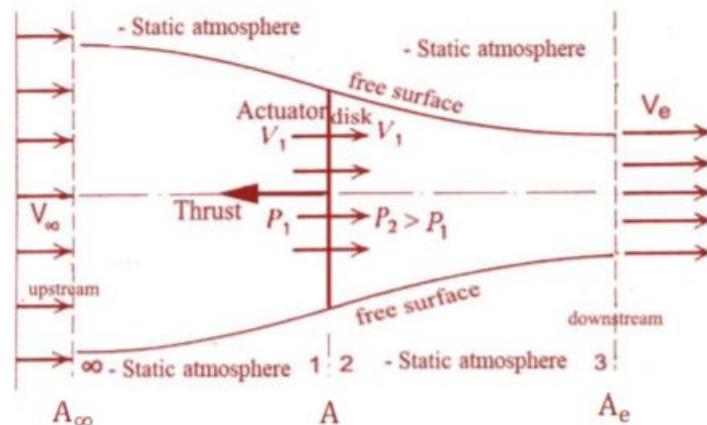
$$\frac{dE_{CV}}{dt} = \dot{Q} - \dot{W} + \dot{m}_i \left(u_i + \frac{V_i^2}{2} + gz_i \right) - \dot{m}_e \left(u_e + \frac{V_e^2}{2} + gz_e \right)$$



圖三十八：簡單系統的能量守恆^[6]

4. 2. 1. 2 Actuator disk theory

此種方法與上節的控制體積類似，但條件限制較多，例如要為不可壓縮、穩態、非黏性、無旋度流體…等等，但還是能應用在螺旋槳上，例如 Betz's limit 的證明。



圖三十九：Schematic of fluid flow through a disk actuator^[7]

根據上圖的定義與參數，可統整出以下三式：

$$\text{Continuity : } \rho V_\infty A_\infty = \rho V_1 A = \rho V_e A_e$$

$$\text{Balance of forces : } \rho V_e A_e V_e - (P_2 - P_1) A = \rho V_\infty A_\infty V_\infty$$

$$\text{Energy flux : } V_1 A (P_2 - P_1) = \rho \frac{V_e^2}{2} V_e A_e - \rho \frac{V_\infty^2}{2} V_\infty A_\infty$$

將上述第三式的 $(P_2 - P_1)$ 以第二式替代， A_∞ 與 A_e 則利用第一式替換成 A 的形式，整理可得以下結果：

$$\begin{aligned}
V_1 A (P_2 - P_1) &= \rho \frac{V_e^2}{2} V_e A_e - \rho \frac{V_\infty^2}{2} V_\infty A_\infty \\
\Rightarrow V_1 (\rho V_e A_e V_e - \rho V_\infty A_\infty V_\infty) &= \rho \frac{V_e^2}{2} V_e A_e - \rho \frac{V_\infty^2}{2} V_\infty A_\infty \\
\Rightarrow V_1 V_e A_e V_e - V_1 V_\infty A_\infty V_\infty &= \frac{V_e^2}{2} V_e A_e - \frac{V_\infty^2}{2} V_\infty A_\infty \\
\Rightarrow V_1 V_e \frac{V_1}{V_e} A V_e - V_1 V_\infty \frac{V_1}{V_\infty} A V_\infty &= \frac{V_e^2}{2} V_e \frac{V_1}{V_e} A - \frac{V_\infty^2}{2} V_\infty \frac{V_1}{V_\infty} A \\
\Rightarrow V_1^2 A (V_e - V_\infty) &= \frac{1}{2} V_1 A (V_e^2 - V_\infty^2) \\
\Rightarrow V_1 &= \frac{V_\infty + V_e}{2}
\end{aligned}$$

有了上式 V_1 的結果後，即可產生 Rankine-Froude Them.，並且定義：

$$\text{Power Coefficient} = C_p = \frac{\rho \frac{V_e^2}{2} V_e A_e - \rho \frac{V_\infty^2}{2} V_\infty A_\infty}{\rho \frac{V_e^2}{2} V_e A_e}, \frac{V_\infty}{V_e} = x$$

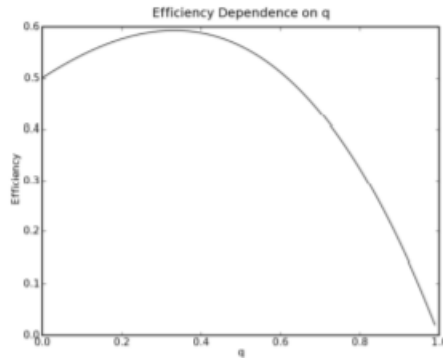
由連續方程式，我們有：

$$\begin{aligned}
A_e &= \frac{V_1}{V_e} A \Rightarrow A_e = \frac{V_\infty + V_e}{2} \frac{A}{V_e} = \frac{A V_\infty + V_e}{2} \frac{A}{V_e} = \frac{A}{2} (1 + x) \\
A_\infty &= \frac{V_1}{V_\infty} A \Rightarrow A_\infty = \frac{V_\infty + V_e}{2} \frac{A}{V_\infty} = \frac{A V_\infty + V_e}{2} \frac{A}{V_\infty} = \frac{A}{2} \left(1 + \frac{1}{x}\right)
\end{aligned}$$

將其帶入 C_p 的定義中可以得到：

$$\begin{aligned}
C_p &= \frac{\rho \frac{V_e^2}{2} V_e A_e - \rho \frac{V_\infty^2}{2} V_\infty A_\infty}{\rho \frac{V_e^2}{2} V_e A_e} \\
\Rightarrow C_p &= \frac{V_e^3 A_e - V_\infty^3 A_\infty}{V_e^3 A} = \frac{(V_e^2 - V_\infty^2) V_e A_e}{V_e^3 A} \\
\Rightarrow C_p &= \left(\frac{V_e^2}{V_e^2} - \frac{V_\infty^2}{V_e^2} \right) \frac{A_e}{A} = (1 - x^2) \frac{\frac{A}{2} (1 + x)}{A} \\
\Rightarrow C_p &= -\frac{1}{2} x^3 - \frac{1}{2} x^2 + \frac{1}{2} x + \frac{1}{2} \\
\Rightarrow \text{Solve } \frac{\partial C_p}{\partial x} &= -\frac{3}{2} x^2 - x + \frac{1}{2} = 0 \text{ to evaluate the extremum} \\
\Rightarrow x = -1 \text{ or } \frac{1}{3} &\Rightarrow C_{p,\max} = \max \left\{ C_p(-1), C_p\left(\frac{1}{3}\right) \right\} = \frac{32}{54} \approx 0.5926
\end{aligned}$$

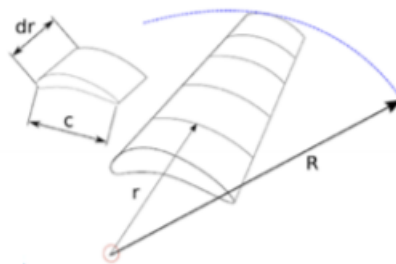
此 0.5926 的值即為 Betz's limit，為螺旋槳能夠提供的最大功率。



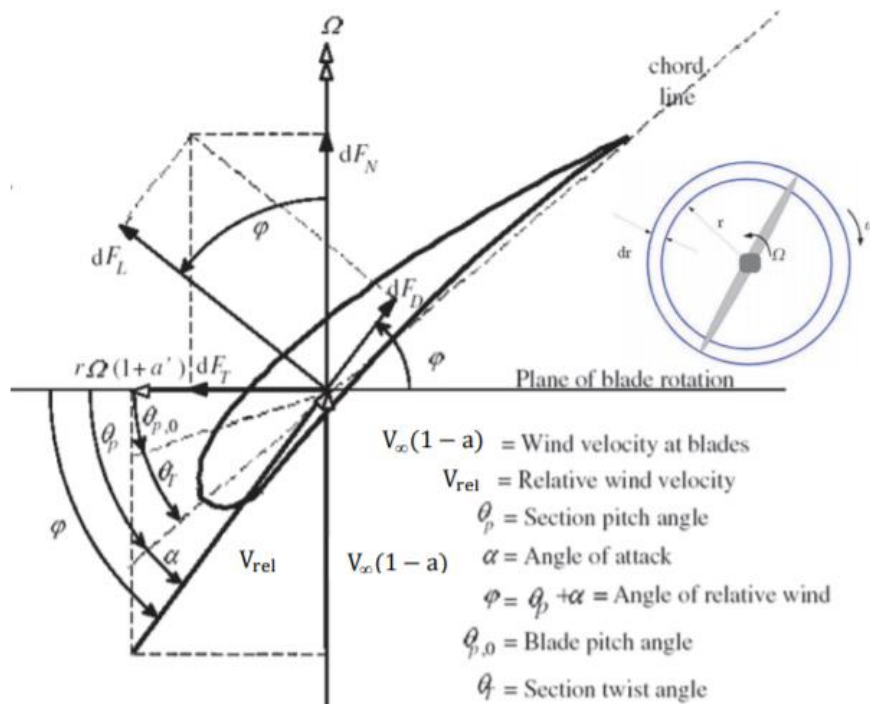
圖四十：Betz's limit on Wind Energy Efficiency^[8]

4. 2. 1. 3 Blade element (momentum) Them.

Blade element Them.的方法為將葉片切成十多個小段，使得每個小段的 rotational speed(ωr)、chord length(c)與 twist angle(γ)皆不同，分別計算各段的升、阻力後再相加。由於切成較小的原速進行分析，其精準度較高，普遍性也較高，不過要注意的是每小段間不能有空氣動力學的交互作用。



圖四十一：Blade element model^[9]



圖四十二：扇葉幾何分析參數^[9]

利用上圖(四十二)的參數，以下將推導以 Blade element Them. 的 C_p 表示式：

$$V_{rel} = \frac{V_{\infty}(1-a)}{\sin \varphi} \quad (\text{此處的 } V_{\infty} \text{ 就是上頁的 } V_1)$$

$$\tan \varphi = \frac{V_{\infty}(1-a)}{\Omega r(1+a')} = \frac{1-a}{(1+a')\lambda_r} \quad (\lambda(R) \equiv \frac{R\Omega}{V_{\infty}})$$

$$dF_L = C_L \frac{1}{2} \rho V_{rel}^2 c dr \quad (\text{詳見 Munson Ch9.3})$$

$$dF_D = C_D \frac{1}{2} \rho V_{rel}^2 c dr \quad (\text{詳見 Munson Ch9.4})$$

$$dL = dF_L \sin \varphi - dF_D \cos \varphi \quad (\text{角動量})$$

$$dT = dF_L \cos \varphi + dF_D \sin \varphi$$

假設螺旋槳葉片數為 B ，可將 dF_L 與 dF_D 改寫成：

$$dF_L = B \frac{1}{2} \rho V_{rel}^2 (C_L \sin \varphi - C_D \cos \varphi) c dr$$

$$dF_D = B \frac{1}{2} \rho V_{rel}^2 (C_L \cos \varphi + C_D \sin \varphi) c dr$$

將每單位的扭矩 dQ 以 rdL 表示：

$$dQ = B \frac{1}{2} \rho V_{rel}^2 (C_L \sin \varphi - C_D \cos \varphi) c r dr$$

Solidity ratio 的定義為：

$$\sigma = \frac{Bc}{2\pi r}$$

將單位扭矩與推力整理成：

$$dQ = \sigma \pi \rho \left(\frac{V_{\infty}(1-a)}{\sin \varphi} \right)^2 (C_L \sin \varphi - C_D \cos \varphi) r^2 dr$$

$$dT = \sigma \pi \rho \left(\frac{V_{\infty}(1-a)}{\sin \varphi} \right)^2 (C_L \sin \varphi - C_D \cos \varphi) r^2 dr$$

接著可以進一步推導 Blade element momentum Them.，由推力的定義與上述單位推力的公式，以及扭矩的定義與上述單位扭矩的公式並整理後(先將 C_D 假設為 0)可以得到：

$$\frac{a}{1-a} = (\sigma C_L) \frac{\cos \varphi}{4 \sin^2 \varphi}$$

$$\frac{a'}{1-a} = \frac{(\sigma C_L)}{4 \lambda_r \sin \varphi}$$

$$\frac{a'}{1+a'} = \frac{(\sigma C_L)}{4 \cos \varphi}$$

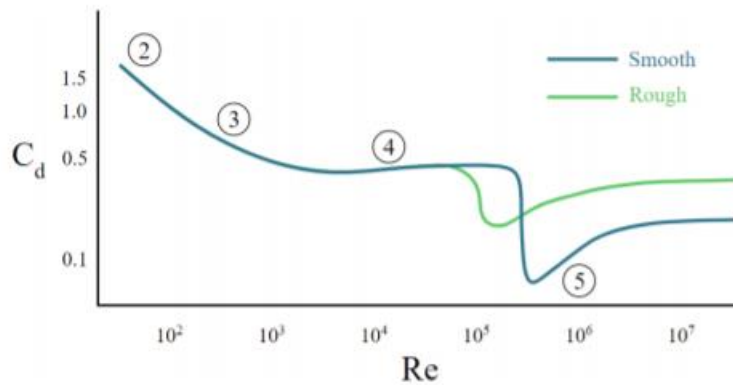
$$a = \{1 + [4 \sin^2 \varphi / (\sigma C_L) \cos \varphi]\}^{-1}$$

$$a' = \{[4 \cos \varphi / (\sigma C_L)] - 1\}^{-1}$$

經過相除與化簡 C_L 可推導出最後的結果：

$$C_p = \frac{P}{\frac{1}{2}\rho V_\infty^3 A} = \frac{\int_{r_h}^R \Omega dQ}{\frac{1}{2}\rho V_\infty^3 \pi R^2} = \frac{8}{\lambda^2} \int_{\lambda_h}^{\lambda} \lambda_r^3 a' (1-a) \left[1 - \left(\frac{C_D}{C_L}\right) \cot \varphi\right] d\lambda_r$$

因此理想上在設計風扇葉片的時候，要最大化的是 C_p ，而不只是升阻力比越大越好。



圖四十三： C_d 對 Re 作圖(量測講義)

4.2.2 第一代風扇與參數設計

1. 風扇數量

我們比較單個與多個(假定為兩個)風扇的優缺點並整理於下表：

風扇數量	單個	兩個
優點	只需一個馬達(成本較低) 風扇較不受大小限制 控制容易	可互相消除車輛轉動趨勢
缺點	風扇較大，較不利於轉動平衡 可能使車輛有轉動趨勢	多顆馬達(成本較高) 每顆馬達的振動可能會疊加 風扇尺寸限制較大 兩個風扇推力難以平衡

比較過後我們最終選擇單個風扇的方式，因為考量功能限制(大小與成本)的關係，單個風扇較有彈性，且多個風扇無疑增加控制的複雜度，加上多個風扇的推力難以平衡，在測量與校正上過於困難，因此排除多個風扇的想法。

2. 風扇位置

雖然大部分都是將風扇置於後方，但出於好奇，還是比較不同風扇位置(前、後)的優缺點，整理如下表：

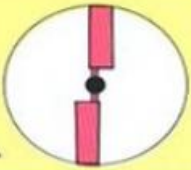





風扇位置	前方	後方
優點	吸入的流場較為穩定 造成的摩擦力較小 重心靠前，減少加速時因慣性造成的不穩定	推力的氣流不受其他元件影響 前輪負荷較輕
缺點	推力的氣流會受其他元件影響 前輪負荷較重 降低後輪與地面的接觸	造成的摩擦力較大 轉彎時較容易不穩

比較兩者後，我們這組最終選擇將風扇位置設於後方，原因出了考量上述優缺點外，我們的感測(循跡)裝置與轉向裝置以及期末的超音波感測器皆決定設置於車體前方，為避免車體的振動造成循跡與轉向機構與超音波的穩定性受影響，加上為了極大化空間使用率，我們排除了將風扇設置於前方的想法。

3. 風扇葉片數量

爬書文獻與學長姊們的報告可以發現，奇數的葉片較偶數的葉片較不會有共振的問題、較易達成平衡，有較好的穩定性也較不會斷裂(尤其在加工方式較無法有較好的精密度下)，而7片因其製作上的不易與時間成本問題，因此在此我們只比較過去較多組別使用的3片與5片於下表(0~5分，數字越大越利於選擇)並比較扇葉面積占比的性質：

	三葉	五葉
馬達負荷	轉動慣量較小 馬達負荷較小 4	轉動慣量較大 需要較大扭力 2
面積	面積小 阻力較小 4	葉片多 推力較大 4
葉片座	各葉片皆有較足夠的空間 4	葉片底部過於密集 可能干擾流場 2
得分	12	8

翼表面積對旋轉面積的比	轉速	轉軸扭力	性質與適用性
 <p>小</p>	 <p>快</p>	 <p>弱</p>	<p>翼面對旋轉面積小的風車，多屬於升力型。此型轉速快，扭轉力弱，適用於需高轉速的機器，例如發電機。</p>
 <p>大</p>	 <p>慢</p>	 <p>強</p>	<p>翼面對旋轉面積大的風車，多屬於抗力型。此型轉速慢，扭轉力強，適用於抽水泵和壓縮機等需大轉距者。</p>

圖四十四：扇葉數量比較^[10]

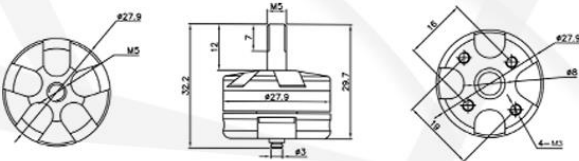
綜合以上，且參考學長姊們過去的結果，3片風扇的推力通常是足夠的，故先以3片風扇做為設計(但我們將此項列為優先可變動的變項，當推力不夠或振動過大時，將此列為第一先改變的順序，因為改變較容易且影響較顯著，不需要改變扇葉參數)。

4. 馬達選擇

在期中前由於過去對於馬達的了解與使用甚少，在未知情況下，我們根據學長姊們的經驗，選擇最低規格與成本，但又可以符合測試要求的馬達，所們選擇的是：MT2204 2300kv 自鎖式無刷馬達。其尺寸與規格如下：

MT2204 SPECIFICATIONS 产品参数

KV:2300
 MAX Thrust:440G Length:32.2MM
 NO.OF CELL:2-3S Weight:25G
 Framework:12N14P Shaft:3MM
 Propeller:5"-6" Diameter:27.9MM



The voltage 电压 (V)	Paddle size 桨尺寸	current 电流 (A)	thrust 推力 (G)	power 功率 (W)	efficiency 效率 (G/W)	speed 转速 (RPM)
8	Carbon Fibre Prop 6x3	6.4	240	51.2	4.7	11910
12	Carbon Fibre Prop 5x3	7.5	310	90.0	3.4	20100
	Carbon Fibre Prop 6x3	11.5	440	138.0	3.2	16300



圖四十五：2300kv 尺寸、實體圖與基本規格^[11]

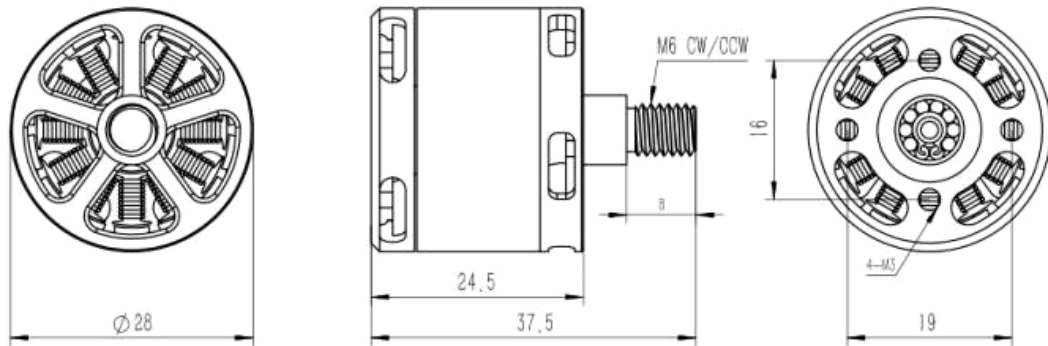
Specifications:

Model :	2204
RPM/V(KV) :	2300KV
Max. Thrust :	420g
Max.Current :	12A
Resistance :	0.112 ohm
Idle Current :	0.6A
Prop Shaft :	M5*12mm
Motor Weight :	27g
Motor Dimension :	Φ27.9*29.7mm

圖四十六：2300kv 規格表^[12]

但經過期中測試後，我們發現我們風扇的噪音明顯比起其它組別的還大聲，並且聽起來並不像是車體震動的聲音，因此我們選擇兩種方式進行優化：

- 將馬達換成較高扭矩的規格：在下方風扇設計的章節中，我們會提到最終我們使用的風扇為 5 片風扇，比起 3 片風扇，馬達需要的扭矩規格更高，此外，我們期中時的風洞實驗發現馬達實際的轉速(有裝風扇的情況)與理想轉速相差甚遠，使得要將風扇的 Duty 值(56%)開很大，才能達到起步的功能，很有可能是風扇的負載對於這個馬達來說太大了。因此我們將馬達從原來的 MT2204 2300kv 更換為 450 級 A2212 2200kv，並且價格也反而還比較便宜。但缺點是所使用的電變從原本的 15A 要升級至 40A 才足夠，整體來說還是需要較高的成品。
- 更改風扇參數以增加在相同轉速下的推力：此在下面的章節中會更進一步說明我們進行改變的流程。



MODEL	KV (rpm/V)	Voltage (V)	Prop	Load Current(A)	Power (W)	Pull (g)	Efficiency (g/W)	Lipo Cell	Weight (g)
A2212	930	11.1	1060	9.8	109	660	6.1	2-4S	52
	1000		1047	15.6	173	885	5.1		
	1400		9050	19.0	210	910	4.3		
	1800		8060	20.8	231	805	3.5		
	2200		6030	21.5	239	732	3.1		
	2450		6030	25.2	280	815	2.9	2-3S	

圖四十七：A2212 2200kv 尺寸、實體圖與規格表^[13]

由上圖(四十七)可以發現，最大推力從 420g 上升至 732g，但實際的推力還是需要進行實驗論證，這將在後面設計分析與驗證的地方進行討論。

5. 風扇大小

雖然風扇的尺寸越大會有較大的風力，但同時也增加了扭矩與振動、斷裂的可能，由於車體的尺寸限制(橫向 21cm)，因此風扇的尺寸不能太大，還要預留空間給風罩，加上據說實作中心的 3D 列印最多只能加工到直徑 16cm，在還未分析的情況下，加上參考學長姊們風扇的尺寸大小，先以直徑 14cm 做為設計參數。

另外一個尺寸限制是風扇與無刷馬達結合的距離，無刷馬達購買時提供子彈座

以夾持螺旋槳，由已選定的馬達上圖(四十五)可得到(經由實際測量後也是)，7mm 為適當的螺旋槳高度(也就是中心螺旋槳軸的厚度)，在此要注意的是，風扇葉片所延伸出的厚度也不能超過此限，否則可能會打到馬達。至於螺旋槳的中心軸徑，除了參考過去學長姊的報告外，也考慮了馬達尺寸與方不方便繪製，經過最後決定將扇葉長度定為 60mm，中心點至扇葉的距離為 10mm，中心內徑參考馬達規格並經由實際量測後可得到直徑為 5mm。

在期末測試時使用的新馬達有提供可換軸徑的槳夾頭，並且與馬達本身有一定的距離，因此風扇的厚度較不受限，而夾頭的直徑也剛好是 5mm，因此並不需要修改風扇的尺寸參數。

6. 基本參數

綜合以上幾點，在此先假定一些基本參數：

- 氣溫：假設實作中心室內溫度為 25°C
- 螺旋槳直徑：140mm
- 螺旋槳軸徑：23.324mm(剛好可以包覆扇葉至軸內的距離)
- 車速：假定為期中最高車速 45cm/s(期中測試的最高車速為 45cm/s，因此假設此值為評估的車速。期末測試我們這組的策略一、三圈的定速車速也是 45cm/s，因此繼續沿用此值進行計算。)
- 馬達轉速：本組使用 2300KV 無刷馬達，代表無負載情況下 1V 的電壓能產生 2300rpm，而在期中後則是 2200rpm(但加上負載後實際上會更低)。然而，考量到螺旋槳以及在最高轉速下可能斷裂的風險，故保守設定馬達轉速為 10000rpm。(在此先依學長姐的經驗保守估計，也較好計算。期中風洞實驗的結果平均轉速大約為 7000rpm，而期末則為 5000rpm 左右。雖然有點高估，但最高轉速可能仍接近 10000 轉，因此繼續沿用此值進行計算。)

此外，雷諾數($Re = Vl/\nu$)為設計時的重要參數：

25°C 空氣的 kinematic viscosity $\nu = 1.562 \times 10^{-5} \text{ m}^2/\text{s}$ ， l 為截面弦長(chord length)，馬達轉速 $\omega = 10000 \text{ rpm} = 1047.2 \text{ rad/s}$ ，而對截面而言的速度即為該半徑的切線速度與車速的向量總和：

$$V(r) = \sqrt{(1047.2r)^2 + 0.45^2}$$

根據去年第 4 組的做法，先取 0.75R 處代表整片扇葉的空氣動力性質，翼面弦長假設為 1.5cm：

$$Re = \frac{\sqrt{((1047.2 \times 0.75 \times 0.07)^2 + 0.45^2)} \times 0.015}{1.562 \times 10^{-5}} \approx 54043$$
，故接下來我們將以雷諾數大約等於

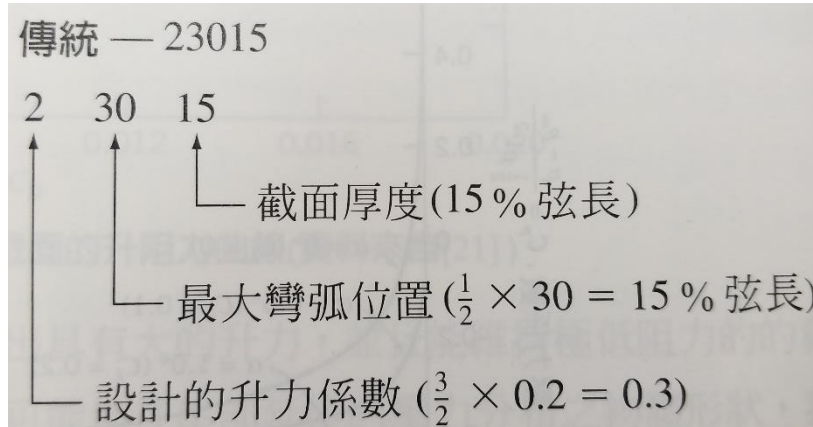
50000 作為分析基礎。

7. 截面翼型

我們這組在一開始翼型的選擇上主要使用 NACA 開發的翼型，主要原因是這些翼型在網路與軟體中通常較有機會有資料庫，對於後續的實測與尺寸調整較為容易，並利用 Q blade 免費軟體進行單片風扇的不同截面分析，其主要優點為有

內建好 NACA 翼型的資料庫，可以直接輸入翼型型號，就會自動匯入座標點產生整個機翼，進而進行分析。過程中可以針對不同雷諾數得到升力、阻力與升阻力比值對攻角圖，選定好翼型後，再經由輸入不同位置的弦長與攻角，Q blade 會自動最佳化形成最後的立體圖，可以直接輸出匯入繪圖軟體繪製。

NACA 翼型由美國國家航太諮詢委員會開發，主要分為 4 位數及 5 位數翼型。以 NACA23015 翼型為例，其各個數字代表的意義如下：



圖四十八：NACA23015 翼型的數字意義^[14]

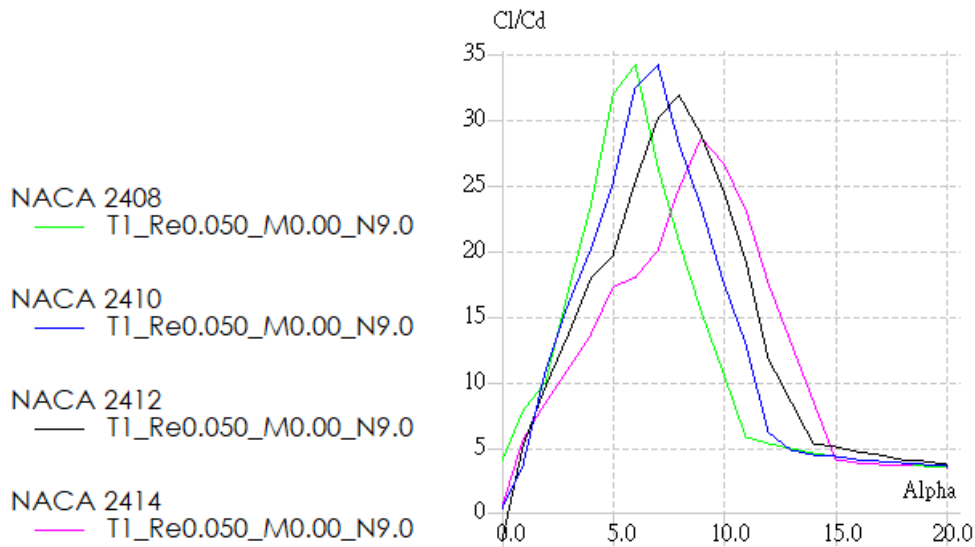
我們統整了一些學長姊常用以及分析的翼型：NACA16012、22112、23012、23018、23112、2408、2410、24112、2412、2414、2415、2418、4412、4414、4415、6412 與 6415，分批匯入 Q blade 進行分析。進入分析畫面後，將雷諾數設定為 50000，並將角度範圍選擇為 0 到 20 度之間，每隔 1 度進行一次分析，我們可以得到不同翼型下 C_l/C_d 對角度圖，雖然藉由下圖(四十九)可得知(在前個章節中有進行推導)，雖然 C_l 與 C_d 的比值只是螺旋槳分析的一部分，但仍可作為單個扇葉翼型的選擇依據。

$$C_P = \frac{8}{\lambda^2} \int_{\lambda_h}^{\lambda} Q \lambda_r^3 a' (1-a) \left[1 - \frac{c_D}{c_L} \tan \beta \right] d\lambda_r$$

單airfoil

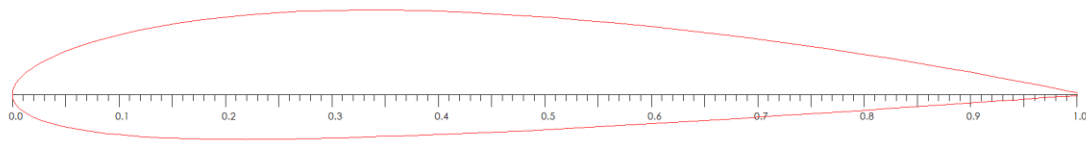
螺旋槳 ($V, \omega, \text{varying AoA, ...}$)

圖四十九：螺旋槳 power coefficient 公式圖



圖五十：不同翼型 Cl/Cd 對角度圖

將上述所有翼型進行篩選，選出升阻力比最優的 4 種翼型(依據高峰值)，根據分析結果，我們選擇使用黑線代表的 NACA2412 翼型，雖然他的峰值沒有 2408、2410 還來的高，但有較寬的高升阻力比範圍，且超過 10 度的攻角仍然有較好的升阻力比，其他截面除 2414 外則是急遽下降。



圖五十一：本阻所選擇的 NACA2412 翼型截面輪廓

8. 弦長

在弦長的部分，由於找不到完整的理論文獻，參考學長姊的方法，有做線性遞減、二次曲線...等等的方式，但在不確定哪種弦長函數有較好的效果下，不如使用現有利用市售螺旋槳的資料庫，所以我們使用學長姐所提供的網站 UIUCPropellerDataSite 的扇葉外形資料庫，根據網站所提供的弦長與半徑比例的數據做設計，我們選擇的風扇模板為 volume2/Vapor/140 mm X 45 mm，因為其尺寸(直徑 140mm)剛好與我們一開始所構想的風扇大小相同，風扇模板尺寸係數如下：

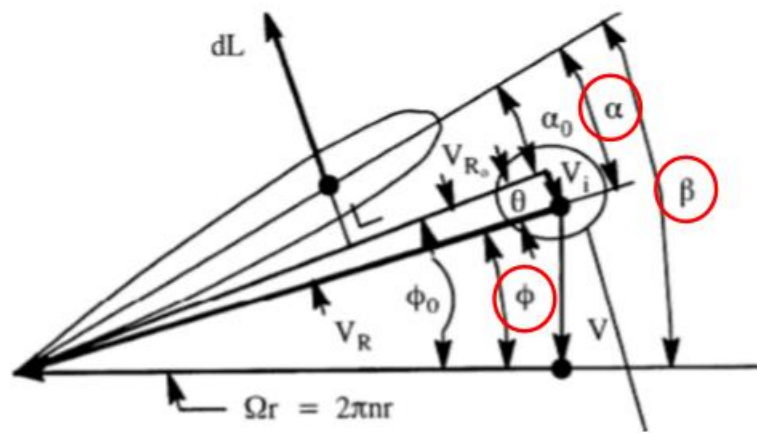
r/R	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
c/R	0.0919	0.115	0.1359	0.1543	0.1696	0.1808	0.1887	0.1927	0.193
r/R	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
c/R	0.1903	0.1852	0.1776	0.1671	0.1533	0.1354	0.1126	0.0775	0.0091

使用上述資料所提供的弦長分布數據，依照假設的半徑 14cm 等比例放大，並輸入到 Q blade 進行繪製。

9. 攻角與扭轉角

由上述 6. 基本參數所設之雷諾數，利用 Q blade 進行截面分析，可由上圖(五十)得到 NACA2412 的最佳攻角為 7.25° 。

在設計螺旋槳攻角時，由於由內而外扇葉與流體的相對速度不變，因此攻角的值不需改變。扭轉角的設計則與水平和鉛直速度有關，水平速度受馬達轉速以及後方形成的渦流影響，但由於渦流並不好評估，因此在此先忽略。鉛直速度則與軸向的風速有關，根據學長姐流場分析的報告平均風速約為 3m/s，取水平與垂直方向的 arctan 值，再加上攻角即可得到扭轉角，如下圖所示：



圖五十二：攻角 α 、flow angle 與扭轉角 β 的關係圖

由上圖(五十二)可得到，攻角 α 、flow angle 與扭轉角 β 的關係式為：

$$\beta = \alpha + \varphi = \alpha + \tan^{-1}\left(\frac{v}{\omega r}\right)$$

由上式可知，在風速(v)與馬達轉速(ω)為定值的情況下，半徑 r 越大，flow angle 越小，因此扇葉隨著半徑的增加，扭轉角會越小。將本組計算的扭轉角整理於下表：

r/R	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
beta	27.183	21.981	18.436	16.023	14.417	13.368	12.637	12.028	11.594
r/R	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
beta	11.152	10.701	10.182	9.63	9.095	8.639	8.162	6.756	0

4.2.3 設計流程與繪製方法

當假定好基本的參數設定後，接下來就是開始設計與繪製立體圖了，以下分別記錄設計與繪製風扇的詳細流程：

4. 2. 3. 1 Q-blade 詳細設計流程

A. Airfoil design

1. Import NACA 2412 airfoil.
2. Set up the parameters (thickness, camber , number of points..).

B. Airfoil analysis

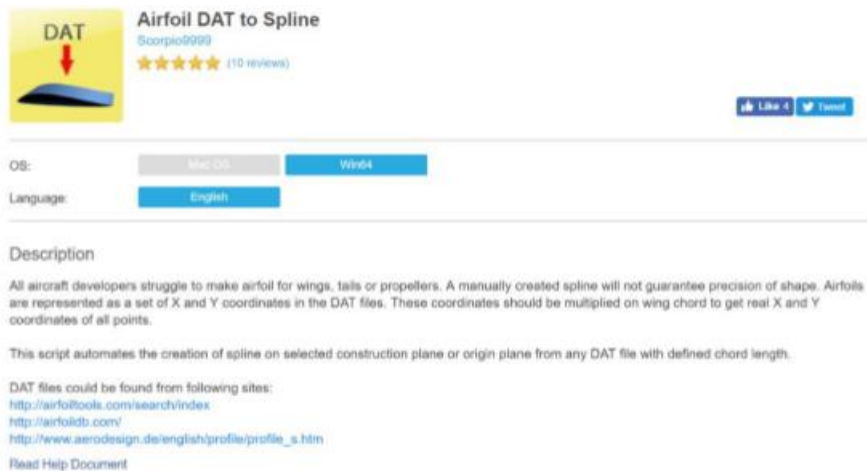
1. Click XFOIL analysis.
2. Click polar view.
3. Analysis/Define an Analysis.
4. Analysis parameters(Reynold number, start angle, end angle, angle increment Δ).
5. Click Analyze.
6. Find the attack angle α with Maximum C_L/C_D value ($\alpha=7.6^\circ$).
7. Click OpPoint View and check the operational points with C_L , C_D value.
8. Click 360°icon.
9. Click Extrapolate and adjust the parameters.
10. Click Save and Generate a circular foil.
11. Go back to airfoil design and check the circular foil.

C. Rotor Blade design

1. Click HAWT Rotor Blade Design.
2. New a Blade.
3. Setting number of segments of the blade.
4. Enter the position and Chord.
5. Set up the foil type.
6. Click Optimize. Use Betz optimization and attack angle optimization separately.
7. Check the nonzero Twist Value.
8. Set Hub radius.
9. Show Rotor and Thread at Centerline Max Thickness in Advanced Blade Design.
10. Click Blade/Rotor and select Export 3D Blade Geometry in STL file.

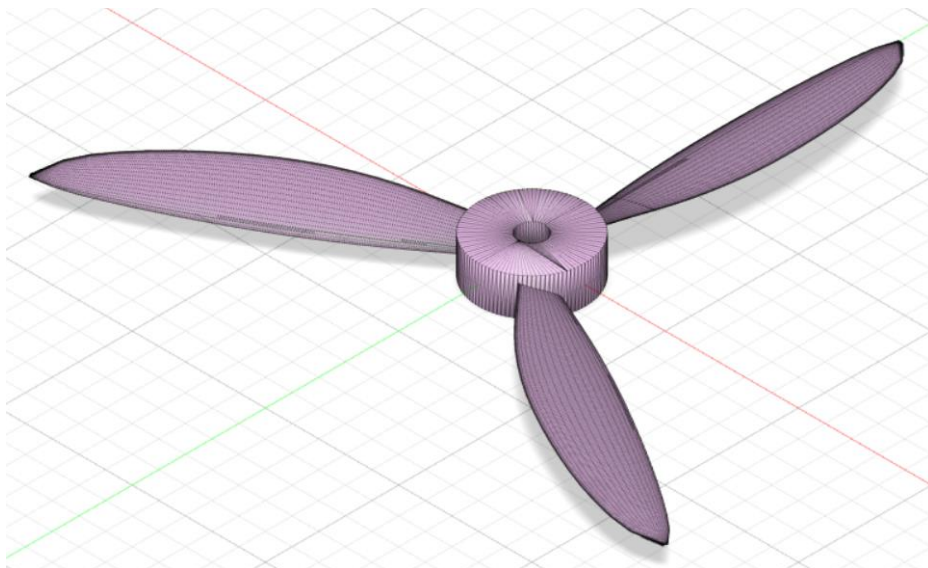
4. 2. 3. 2 3D 繪圖詳細繪製流程

我們使用 Autodesk Fusion 360 繪製螺旋槳的 3D 圖。首先在 Autodesk Appstore 上搜尋並下載 Airfoil DAT to Spline，其如下圖(五十三)所示：



圖五十三：Fusion 360 的外加套件

接著將上述 Q-Blade 所匯出的 STL 檔匯入 Fusion 360，將其繞 Z 軸複製出風扇葉片的數量，再加上風扇的軸心，及可完成風扇所有的 3D 繪圖。

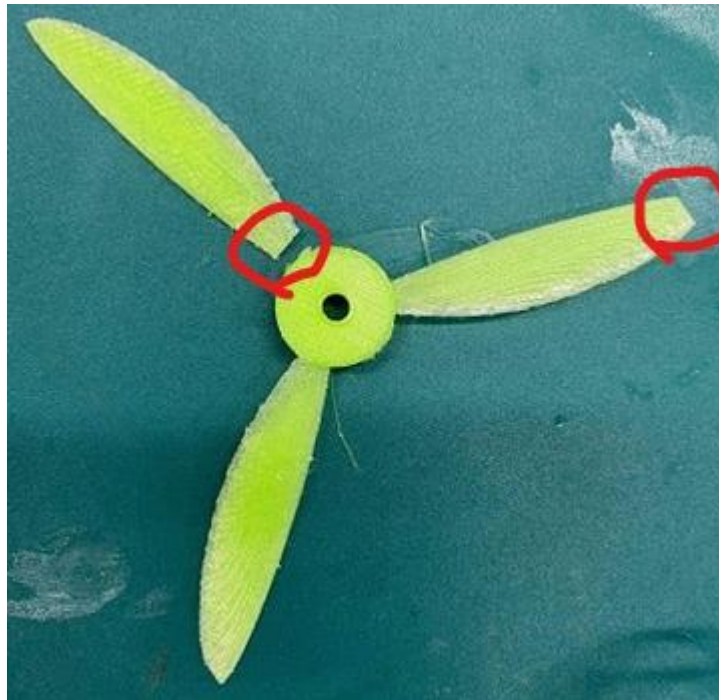


圖五十四：藉由上述流程所繪出的第一代風扇 3D 圖

4.2.4 各代風扇的比較與改善

◆ 第一代風扇設計與測試

將上述風扇的設計與繪製方法並以 3D 列印(Kingsssel)所製作出的成品，由於第一次進行除料(支撐材)的工作，還不習慣力道的掌控，故在除料過程中就將葉片給折斷了，因此沒有進一步進行測試。



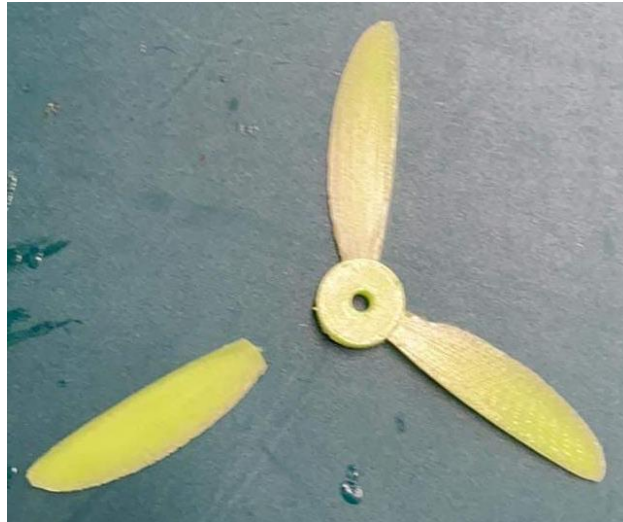
圖五十五：除料過程中斷裂的第一代風扇

由上圖(五十五)可發現，除了扇葉與中心銜接處可能不夠堅固外，有一片的風扇尾端也在除料過程中被剝下，故將在第二代風扇進行風扇結構上的加強。

◆ 第二代風扇(3片)與測試

由於第一代在與中心銜接處和尾端的地方不夠堅固，所以我們想到的辦法是直接將弦長統一加上 3mm，除了可以增加扇葉的強度之外，因為風扇面積的增加，也可以加強風扇的推力。

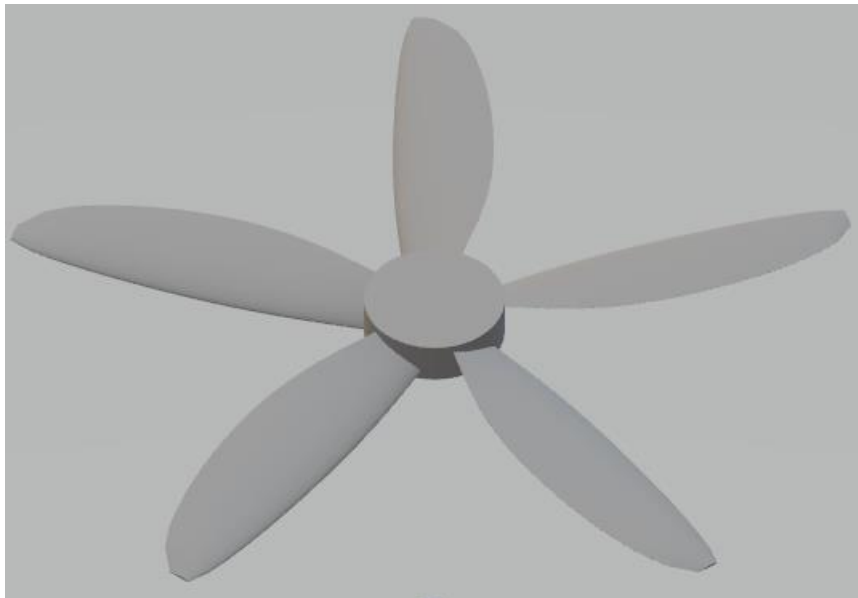
在實測的過程中，我們發現風扇的推力還是明顯不足，需要在非常高轉速(大於 10000rpm)的情況下才有足夠的推力，但這樣除了增加危險性外，也增加了車體震動的可能，在測試過程中由於控制組的疏忽，風扇在暴衝的情況下撞到支架而斷裂，這也讓我們警覺到風扇的危險性，因此為了在不需高速旋轉又有足夠的推力下，我們將原本 3 片的風扇，在所有參數設計不變的情況下改為 5 片(較簡易、快速的方法，如結果不甚理想，將進一步修改風扇參數)。



圖五十六：第二代風扇在高速旋轉下撞到支架斷裂

◆ 第三代風扇(5片)與測試

將原本第三代的扇葉由3片改為5片後，沒想到我們在除料的過程，風扇再一次被折斷了，但在上一代3片的結構下，在高轉速的時候風扇並沒有因此斷裂或變形，因此可能是3D列印機的誤差或者除料過程過於用力所致，但為了以防萬一，我們還是將風扇的結構再次進行加強，並聽從別組的經驗，將3D列印機由原本的Kingsssel改為23萬，希望藉此增加風扇的精度與穩固性。



圖五十七：第三代風扇 3D 圖

◆ 第四代風扇(3片)與測試

由於第三代的結構可能還是不夠穩固，所以我們將扇葉的弦長再整體加上3mm(比第一代多0.6cm)，並將扇葉數先改成原本的3片，試其效果，再經過測試後發現雖然推力在沒有其他硬體元件的情況下是足夠的，但轉速還是相對偏高，車體的震動有點大，為了改善車體震動的情況，我們希望馬達能夠在更低的轉速下達到足夠推力的要求，故我們將第四代風扇其他參數不變的情況下，

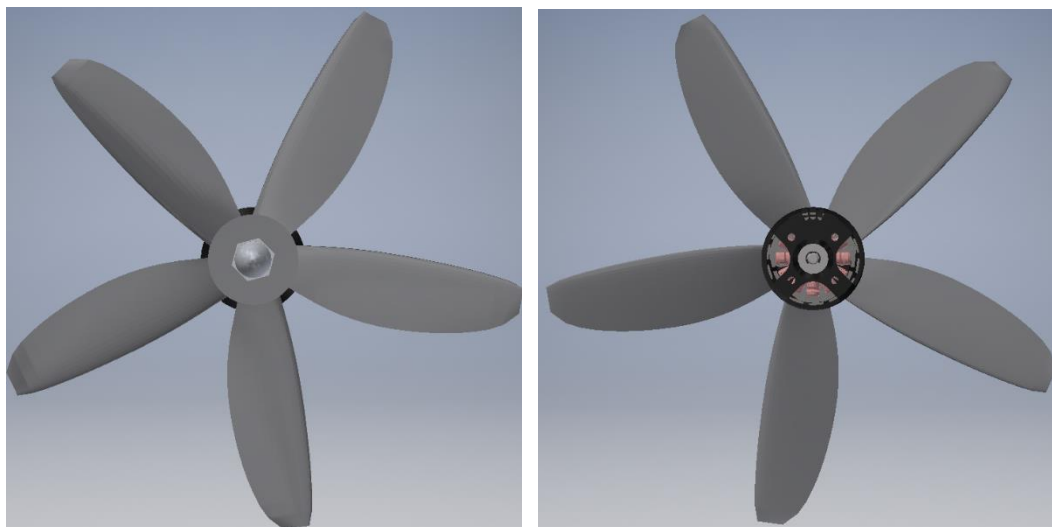
再改為 5 片風扇進行後續的測試。此外，我們在此(第 3 代→第 4 代)將 3D 列印機改為 23 萬後，我們發現在除料的容易程度和精度都有明顯的改善，因此繼續使用 23 萬印製下一代風扇。但 23 萬的缺點是，其列印時間大概是 Kingsssel 的 2~3 倍，且常常印到一半暫停，因此需要有組員一直堅守它，但為了有較好的成品，付出的時間成本還是在可接受的範圍內。



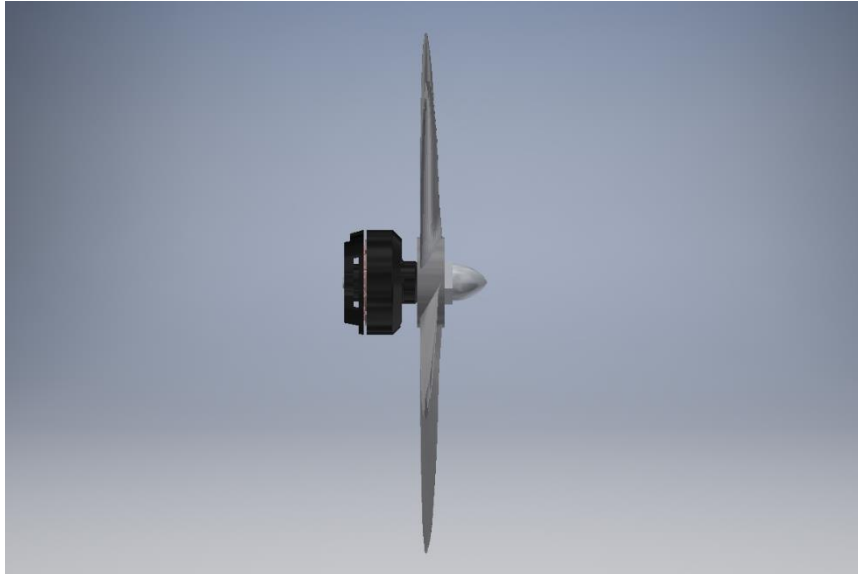
圖五十八：第四代風扇實體圖

◆ 第五代風扇(5片)與測試(期中測試版本)

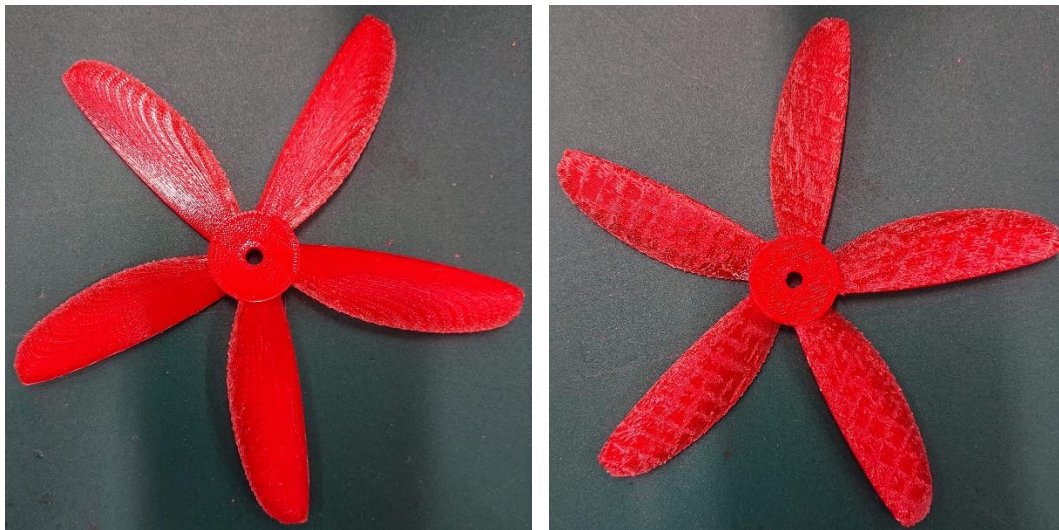
在將第四代風扇改為 5 片扇葉後我們發現，不僅在震動上改善了很多，推力在所有物件皆放置到車上也足夠期中測試的要求，故先就此進行之後的風洞實驗和期中測試。



圖五十九：第五代風扇與馬達的 3D(前視與後視)圖



圖六十：第五代風扇與馬達的 3D(側視)圖



圖六十一：第五代風扇期中測試的上視與下視實體圖

還需改善的地方：

雖然第 5 代風扇在推力和震動方面都比以前改善不少，但震動的情況還是沒有非常理想，雖然不確定震動的來源，但可以確定的是風扇轉速越低，震動的情況會大幅改善，故在期中後希望藉著改變不同的參數，在相同推力的下降轉速，以得到較好的車體穩定性，尤其在期末需要有較大幅度的轉彎和超車速度的條件。

◆ 第六代風扇(5 片)與測試(期末測試版本)

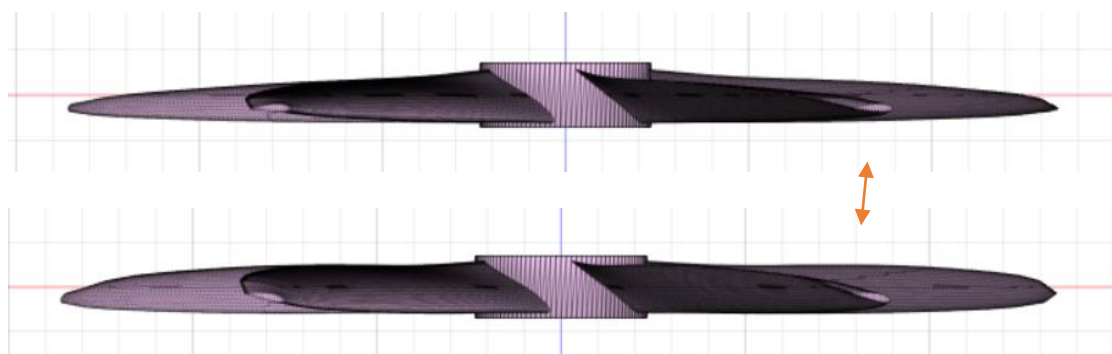
在經過期中測試後，我們發現我們風扇的噪音明顯比起其它組別的還大聲，並且聽起來並不像是車體震動的聲音，因此我們選擇兩種方式進行優化：

- 將馬達換成較高扭矩的規格：在第一代風扇時，我們已有加以闡述。
- 更改風扇參數以增加在相同轉速下的推力：在前面計算扭轉角的時候我們發現我們並沒有加上原始攻角，使得最外側的風扇角度為 0，但如果將所有位

置的角度都再加上 7.25° ，扇葉會因此超過中心軸的厚度，因此我們選擇的方法是利用線性增加的方式，在與中心軸接觸的位置攻角不變，線性遞增至尾端增加 7.25° ，理論上攻角的增加也能使風扇推力更大，可以有效降低馬達的轉速，經實際測試後發現啟動所需的風扇轉速確實降低了不少，推力大大的提升，並且風扇轉動的噪音也減少非常多，而實際實驗中所測量到的數據將在後續設計分析的章節加以驗證。雖然推力增加了，但在控制方面會稍微較不容易調控，因為增加一樣的風扇轉速，加速度變化的比較大，因此較不容易進行細調。



圖六十二：第六代風扇與馬達的 3D 圖



圖六十三：上為第 5 代，下為第 6 代(只有差在扭轉角，尾端較明顯)

以下整理了期中與期末測試時的風扇規格：

軸徑	23.324mm
中心軸徑	5mm
軸高度(厚度)	7mm
單葉片長度	6cm
風扇直徑	14cm
翼形	NACA-2412
馬達	MT2204 2300kv(期中) A2212 2200kv(期末)
3D 列印機	23 萬

弦長：

r(cm)	1	1.4	1.75	2.1	2.45	2.8	3.15	3.5	3.85
c(cm)	0.0919	0.115	0.1359	0.1543	0.1696	0.1808	0.1887	0.1927	0.193
r(cm)	4.2	4.55	4.9	5.25	5.6	5.95	6.3	6.65	7
c(cm)	0.1903	0.1852	0.1776	0.1671	0.1533	0.1354	0.1126	0.0775	0.0091

扭轉角(期中)：

r(cm)	1	1.4	1.75	2.1	2.45	2.8	3.15	3.5	3.85
beta	27.183	21.981	18.436	16.023	14.417	13.368	12.637	12.028	11.594
r(cm)	4.2	4.55	4.9	5.25	5.6	5.95	6.3	6.65	7
beta	11.152	10.701	10.182	9.63	9.095	8.639	8.162	6.756	0

扭轉角(期末)：

r(cm)	1	1.4	1.75	2.1	2.45	2.8	3.15	3.5	3.85
beta	27.183	22.464	19.342	17.352	16.169	15.543	15.235	15.049	15.038
r(cm)	4.2	4.55	4.9	5.25	5.6	5.95	6.3	6.65	7
beta	15.019	14.991	14.895	14.765	14.653	14.620	14.566	13.583	7.25

期中時風扇轉動的聲音連結：

<https://drive.google.com/file/d/11mw8OcN3MNRxJqjjoXD6WtOISDILFqOW/view?usp=sharing>

期末時風扇轉動(兩台車一起)的聲音連結：

<https://drive.google.com/file/d/11oPJ2CNJ2Cr5uMy63kzdPFyUTMbDEJAQ/view?usp=sharing>

由上面兩個影片(同一台手機錄製)可以發現，期中前的轉動噪音比起期末兩台車合起來的聲音都還要大上許多，只可惜沒有利用分貝計進行實際數值上的比較。

4.2.5 理論推力計算

根據 Blade Element Theory，推力 F_t 可由前面推導出的下式進行計算：

$$F_t = F_l \cos\phi - F_d \sin\phi = \frac{1}{2} \rho \int_{r_{hub}}^R N_B [(rw)^2 + V^2] c(r) (C_l \cos\phi - C_d \sin\phi) dr$$

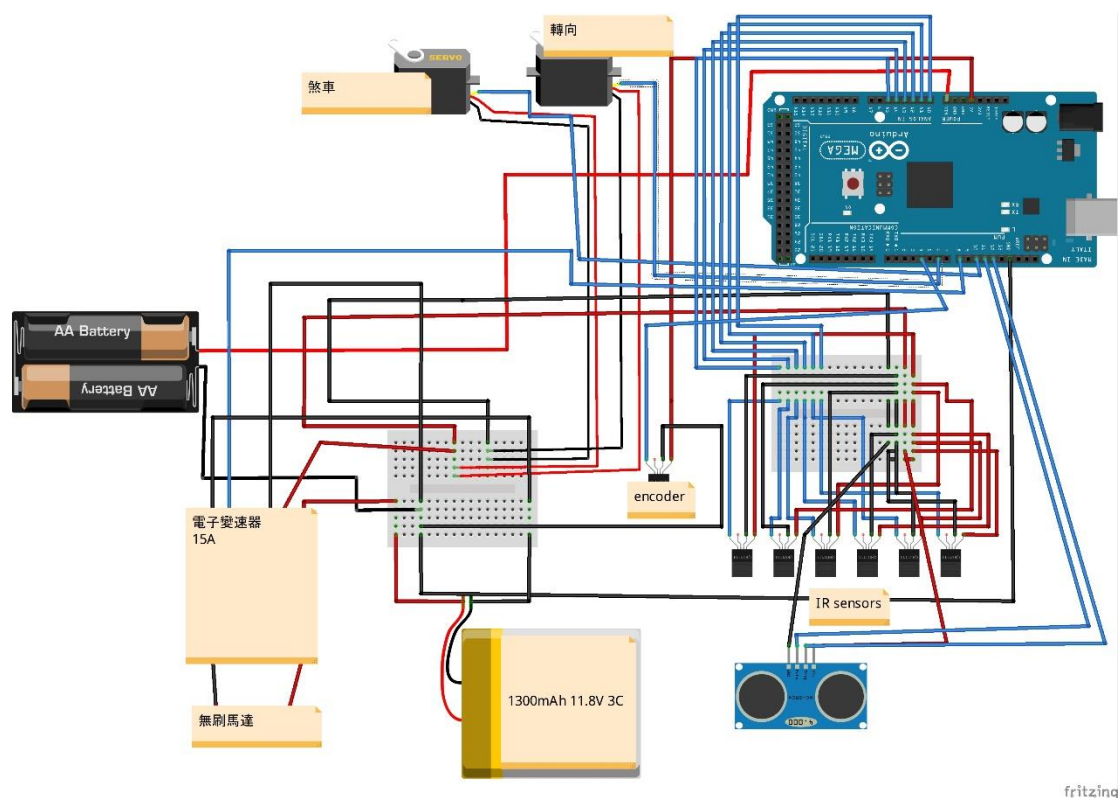
假設實作中心溫度為 25°C，空氣密度 $\rho = 1.184(\text{kg/m}^3)$ ，弦長由於沒有規律，

假設其為不受 r 值影響的平均值 $0.0162(m)$ ，扇葉軸半徑 $r_{hub}=0.01(m)$ ，扇葉數目 $N_b=5$ ，扇葉半徑 $R=0.07(m)$ ， $\phi = \tan^{-1}(u/rw)$ 。並且假設空氣動力性質 C_l 與 C_d 為 $0.75R$ 處的值，則期中測試時的理論推力經由上式並利用 Matlab 計算可得到靜止加速時的理論推力為 $6(N)$ 。期末測試時的理論推力則為 $6.73(N)$ 。但由於此值為取平均的理想值，於分析與驗證的部分會有理論值與實測結果更詳細的探討。

4.3 控制程式與硬體設計

這次的氣動車比起上學期機械設計原理的撞球車多了很多新配置的電子元件，在此將統整個系統電路與各個硬體的元件設計與思路。

4.3.1 系統電路



圖六十四：電路系統配置圖

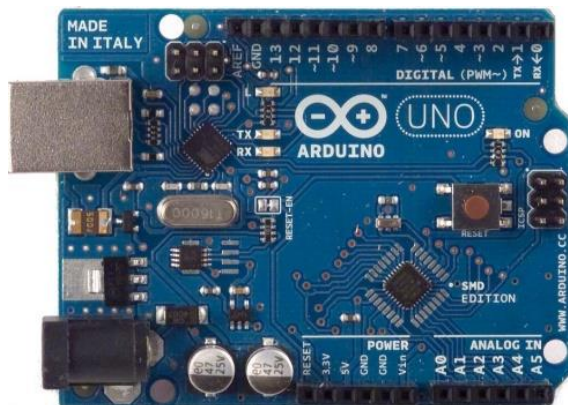
上圖為這次實務需求最為精簡的配置圖。在期中測試時，為了確保 arduino 不會將無刷馬達的電給吃掉，我們決定將以行動電源為兩個 arduino 板供電。至於鋰電池就給無刷馬達專用，讓它能長時間維持較大的電壓運轉。

期末測試的機電與期中的差不多，只是多了超音波測距模組，以及將原本的兩塊 arduino uno 整合成一塊 arduino mega。我們把 lipo 電池所供的電給風扇後提供給電子元件，只有 arduino 板與 encoder 是消耗隨身充提供的電。這是因為我們發現將 encoder 與伺服馬達的電路連接起來會導致 encoder 怪怪的，所以才將其分開。而期中測試所遇到的問題：無法同時啟動風扇與循跡，原本

期中後也無法啟動，但過了某天突然又可以一直到期末都沒問題，所以我也沒找到到底是什麼原因導致。

4.3.2 機電元件

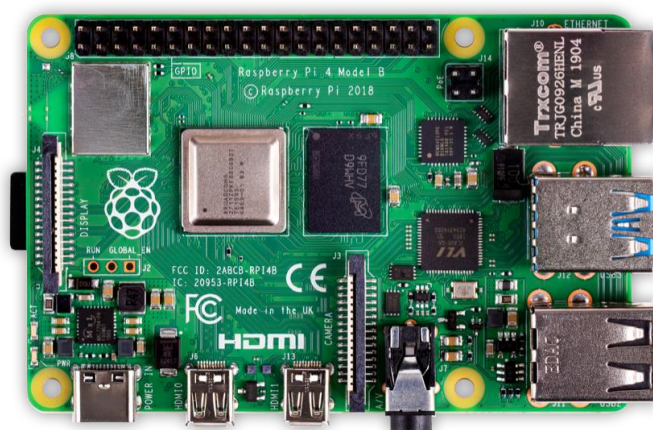
1. 控制板



圖六十五：Arduino Uno



圖六十六：Arduino Mega



圖六十七：Raspberry Pi

以下統整與比較了上述圖(六十五)~(六十七)不同控制板的特性：

	體積重量	處理速度	價格
Arduino uno	小且輕	普通	便宜
Arduino mega	較大較重	普通	稍貴
raspberry pi	小且輕	較快	很貴

控制板是我們控制車子的大腦，所以非常重要。我們原本要用一塊 arduino mega 來控制，但在歷經過循跡與風扇無法同時啟動的問題後，我們決定用兩塊 arduino uno 板，如此可以分散運算處理，但面積的占比會比較大。未來期末的車子如果要開發出影像辨識的功能，我相信 Rasberry Pi 將會是我們的選擇。

相信上述已經比較過 arduino uno，mega 與樹梅派的差別。我們期中時其實比較傾向於使用 mega 當作控制板，但是由於遇到不明問題，所以只能用兩塊 uno 代替。但經過之後的嘗試，我們的 mega 板又可以使用了，所以我們最後決定用 mega 板來完成我們期末的測試。而樹梅派的影像辨識，我們認為沒有必要把簡單的東西弄複雜，所以就決定放棄。

期末為了防止 mega 震動或晃動以致線脫落，因此使用魔鬼氈進行固定。

2. 鋰電池

參考過去學長姐所使用的電池規格，我們所使用的是 11.1V、3S、1300mAh 的鋰聚電池，如下圖：

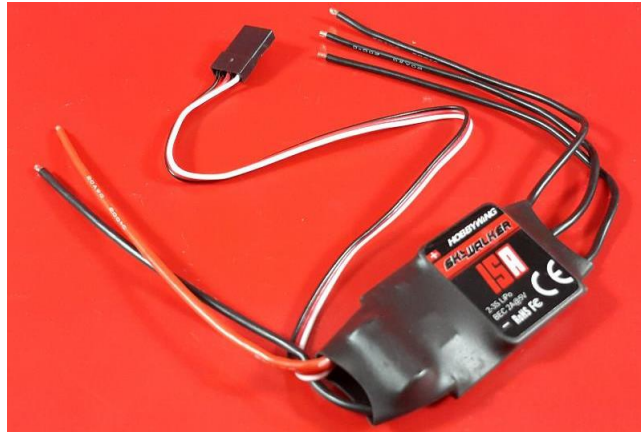


圖六十八：1300mAh 的鋰聚電池

3. 電子變速計(電變)

電子變速器連接鋰聚電池以及無刷馬達，其主要功能是控制板可以用小電壓來調整電池輸出的電壓至馬達。因為我們風扇組所選購的無刷馬達最大電流是 12A，所以我們買了最大電流為 15A 的電變。

期中測試後，由於我們換了顆無刷馬達，我們也順便將電變升級成最大電流為 40A 的電變。



圖六十九：最大電流為 15A 的電變



圖七十：最大電流為 40A 的電變^[15]

4. 紅外線循跡板

我們購買的循跡板是市面上販售的 TRCT5000，其原理是將紅外線發射出去，並由電晶體接收回傳電壓值。它有兩種訊號可以選擇：類比及數位：類比是將回傳 0 至 1024 的值給控制板，而數位則只會回傳 0 及 1 的訊號來判斷是否讀到黑線。需離待測物 3cm 以內的距離。



圖七十一：TRCT5000 循跡板

5. 超音波感測器

這是唯一期中沒有的東西，由於期末目標有跟車，故使用 HC-SR04 解決。我們比較了紅外線與超音波測距後，發現紅外線適合測超近的距離，而超音波適合中等的距離，故我們選用了超音波。

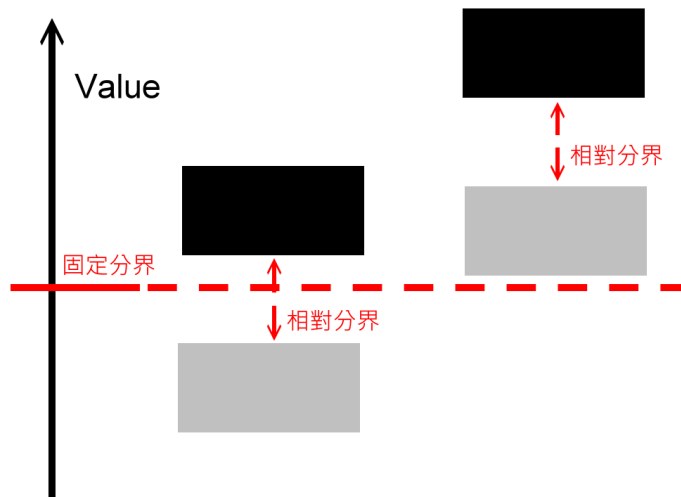


圖七十二：HC-SR04 超音波感測器

4.3.3 循跡

我們這次所使用的是 TCRT5000 的感測循跡模組，並且讀取其類比訊號，再轉成 0 與 1 的數位訊號。我們這麼做的好處有可以方便控制分界值，並以計算取代模組上的轉扭。

原先的模組數位訊號的分界是固定分界，將黑色與地面訊號分界，但這有可能會因為外界因素而導致誤差，像是高度的不同、環境光或是地板材質的不同都會影響訊號讀取結果。

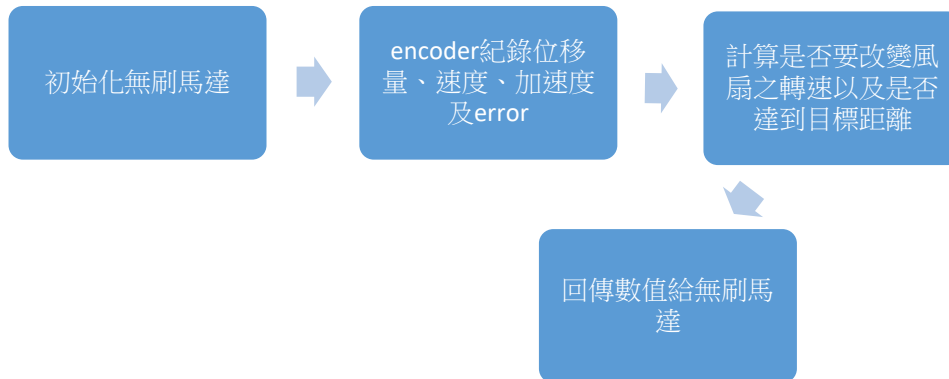


圖七十三：不同模組數位訊號的分界

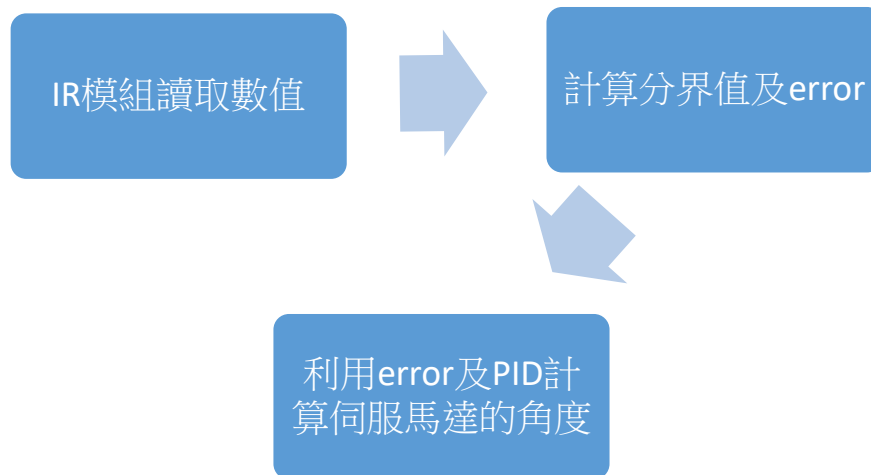
循跡模組還有一個很重要的參數，就是每個模組都會擁有不同的權重，來影響轉向。我們的車子因為轉向系統還較不穩定的關係，會無法快速從 90 度轉到大於 90 度的角度。因此，我們將左邊三顆的模組權重乘上 1.2 倍，以解決這個問題。

4.3.4 程式架構(期中控速)

因為有兩塊板子的緣故，我們的程式也分為兩個部分討論：



圖七十四：風扇馬達控制架構



圖七十五：循跡控制架構

1. 無刷馬達驅動程式

由於無刷馬達需要由電變控制，所以我們需要將電變初始化。初始化是要設定duty的最高與最低值，讓以後可以在這範圍內調整。

```
void brushlessset() {  
    brushless.attach(9);  
    brushless.write(180); //設定上界  
    delay(2000);  
    brushless.write(20); //設定下界  
    delay(2000);  
}
```

2. Encoder

由於行走距離是改變目標的重要因素，所以我們自己用 3D 列印製作了 encoder，光柵數為 20，直徑為 4.8cm。

```
encoder1 = digitalRead(encoderpin);
if (encodercount = 1) {
  starttime = millis();
}
if (encoder1 != encoder2) {
  if (encoder1 == 1) {
    if (pencoder1 != encoder1) {
      encodercount = encodercount + 1;
      presenttime = millis();
      pdisplacement = displacement;
      displacement = encodercount / 20 * 2 * 3.14 * wheelR; //cm
      Serial.print(encoder1);
      Serial.print(",");
      Serial.print(encoder2);
      Serial.print(",");
      Serial.println(encodercount);

      deltatime = (presenttime - pasttime - starttime) / 1000;

      pvelocity = velocity;
      velocity = deltadisplacement / deltatime;
      deltavelocity = pvelocity - velocity;
      verror = velocity - v;

      accel = deltavelocity / deltatime;
      aerror = accel - a;
    }
  }
}
encoder2 = encoder1;
```

3. 改變目標

經過 encoder 的計算後，我們要判斷我們在哪個地方、目標是什麼。所以下圖是在第一階段 a=3 時的 code，而當 encoder 讀到位移量是 150cm 後，程式會利用 switch case 來轉換階段。

```
switch (modee) {
  case 1: //a = 3 0-150
    output = routput[3];
    a = 3;
    if (aerror > -0.5 && aerror < 0) {
      if (output < 130) {
        output = output + 1;
      }
    } else if (aerror > -1 && aerror < -0.5) {
      if (output < 130) {
        output = output + 1.5;
      }
    } else if (aerror > 0) {
      if (output > 20) {
        output = output - 2.5;
        //brake
      }
    } else if (aerror < -1) {
      output = routput[6];
    }

    if (displacement > 150) {
      modee = 2;
      switchmode = 1
    }
  case 2: //v = 30 150-300
    if (switchmode = 1){
      output = routput[4];
      switchmode = 0;
    }
}
```

4. 計算 IR 讀值的分界值

由於有種種變因會導致 IR 讀取的數位訊號有誤，所以我們將類比訊號轉換成數位訊號，再來判斷黑線位置。

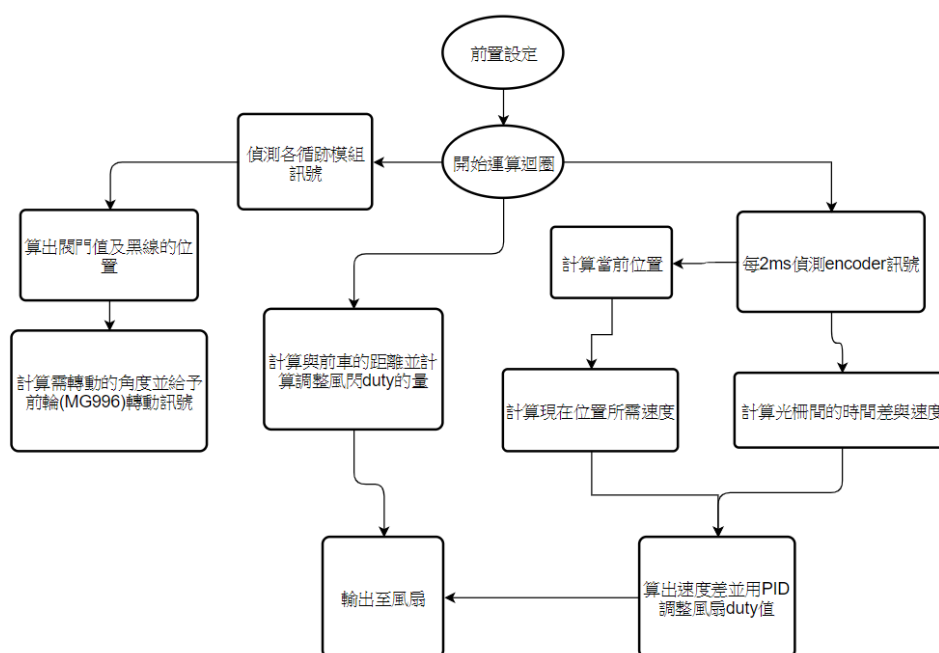
```
for (int i = 0; i < 6 ; i++) {
  IRread[i] = analogRead(IR[i]);
  //Serial.println( IRread[i]);
  gate = gate + IRread[i];
  Serial.println(IRread[i]);
}
gate = gate / 6 + 50;
for (int i = 0; i < 6 ; i++) {
  if (IRread[i] < gate) {
    IRfinal[i] = 1; //not black
  } else if (IRread[i] > gate) {
    IRfinal[i] = 0; //black
  } // Serial.println(i);
  /*Serial.println(":");*/
}
}
```

5. PID 計算角度誤差

最初構想是要用 K_p 、 K_d 以及 K_i 調控轉角，但因為還沒有機會測試合適的參數值，所以暫且只使用 P control， K_p 值為 1 來控制。

```
//PID計算角度誤差
intg += error;
der = error - previouserror;
pterm = Kp * error;
iterm = Ki * intg;
dterm = Kd * der;
output = (pterm + iterm + dterm) / (Kp + Ki + Kd);
//Serial.print(output);
previouserror = error;
ppreviouserror = previouserror;
```

4.3.5 程式架構(期末超車)



上圖為這次期末專題的整體程式架構。

4.3.5.1 控制邏輯

1. encoder

我們基本的架構與期中相同，因為期中控速的結果我們還能接受。我們有改變的是增加了 timer2 來偵測 encoder 現在的訊號。

```

MsTimer2::set(2, ticktock);
MsTimer2::start();
  
```

我詢問了很多同學的做法，大部分的人使用的是 interrupt 的程式碼，作用是當訊號值有改變時就會打斷正在執行的程式，來執行我們給他要執行的程式。這個方法看似可行，但是我們的 encoder 雜訊太多，會導致 interrupt 在錯誤的時候被叫出，導致計算的位移量與速度會有極大的誤差。而我最後所使用的就是利用 timer2 計算時間，讓他每兩毫秒就執行我們給他的程式，這程式就是來偵測 encoder 的類比訊號。我們也給予一個低的閥門與高的閥門，來避免偵測到不必要的雜訊。

2. 循跡與超車

由於期中的循跡程式執行的很順利，我們就沒有改變任何東西，就只是加了超車段的程式碼。由於兩隊討論後，我們的超車是要讓前車往外打並且減速，所以我就將超車的程式寫在循跡的城市之中，讓他可以蓋過循跡的結果。由於在超車時也要控制車子的速度，例如超車時前車要減速，超車後後車要加速追上前車，所以我也在超車的程式碼也了個 slow 的參數，而這個參數會告訴車子我

們當下所需要的目標速度，來控制我們車子的速度。由於版面關係只能截右轉的圖，若要看完整的程式碼請至附件一。

```
    }else if (x < 42900) {      //給超 26960+11280
      if (x < 36500) {
        overtake= 1;
        ang = 107;
      } else if (x < 37500 && overtake == 1) {
        ang = 100;
        if (IRfinal[5] == 0) {
          ang = 85;
        } else if (IRfinal[4] == 0) {
          ang = 88;
        } else if (IRfinal[3] == 0 ) {
          overtake = 0;
          //outside = 1;
        }
      } else if (x < 39200) {
        //outside normal
        slow = 2;
      } else if (x < 41800) {
        slow = 1;
        if (x < 39800) {
          overtake = 1;
          ang = 88;
        } else if (x < 41800 && overtake == 1) {
          ang = 93;
          if (IRfinal[0] == 0) {
            ang = 108;
          } else if (IRfinal[1] == 0) {
            ang = 106;
          } else if (IRfinal[2] == 0) {
            overtake = 0;
            //inside = 1;
          }
        }
      } else if (x < 42900) {
        if (velocity < 400) {
          boost1 = 1;
        }
      }
    } else if (x < 47300) { //回到候車
      slow = 6;
    } else if (x < 71300) {
```

我的邏輯是當我們到要轉出去的距離時，我們會將 overtake 開啟，裡面的 ang 會覆蓋掉循跡所算出要打的角度。我們會先往外打一個大角度並且行走 1000mm，再打一個小的角度慢慢靠向外面的線，一直到最右邊的紅外面模組偵測到黑線時，我們再將車子打回到與線平行，而當中間的模組偵測到黑線時，就關閉 overtake 不再覆蓋循跡，這樣就有種開關循跡的感覺，但是模組其實還是持續的在偵測黑線的位置。裡面的 slow 就是可以控制目標速度的參數，而 boost1 是當要加速時開啟，會讓風扇的 duty 值增加一個固定數值，造成起步的加速度提升的效果。

```

//速度需求
float track_desire_v(float x) {           //output desire x dot mm/s
  if (x < 71300) {                       //22646
    if (slow == 0) {                     //第二圈改400*****
      return 410;
    } else if (slow == 1) {
      return 380;
    } else if (slow == 2) {
      return 250;
    } else if (slow == 3) {
      return 430;
    } else if (slow == 5) {
      return 390;
    } else if (slow == 6) {
      return 540;
    } else if (slow == 7) {
      return 440;
    } else if (slow == 8) {
      return 0;
    }
  }
  else {
    return 0;
  }
}

```

3. 跟車

跟車是期末最大的挑戰之一，因為要在一定的範圍內跟好車，前後車的速度、家速度差不能太大，反應的時間也會影響跟車距離的表現。由於前車的速度與加速度跟我們相比非常的高，所以我們在很近的距離才關閉風扇也不會撞到。我們原本有思考要使用煞車讓我們跟車可以更接近，但是發現煞車會讓速度接近於零，且控制煞車的馬達非常不穩定，且煞車的摩擦力不大，所以我們就放棄了使用煞車。當超音波模組無法偵測到東西時，他會回傳 0 給 arduino，所以我們要在當前車時關掉 0，而當後車時開啟 0 的誤差。這是因為當我們是後車時卻偵測不到東西，代表前車的距離已經非常遠，要用夠快的速度來追才能追到前車。但是在轉彎的時候我們也有可能偵測不到東西，導致我們在轉彎處會有小小的暴衝，但是經過測試後似乎還能控制在範圍內不撞到。

```

//=====跟車=====
cardis = constrain(sonarf.ping_cm(), 0, 70);

if (x < 71300 && velocity < 520) { //在與前車距離 69980+250
  if (cardis != 0) { //若前方有東西
    if (cardis < 3) {
      diserror = -15;

    } else if (cardis < 6) {
      diserror = -6;

    } else if (cardis < 8) {
      diserror = 4;

    } else if (cardis < 15) {
      diserror = 5;

    } else if (cardis < 20) {
      diserror = 5;

    } else if (cardis < 25) {
      diserror = 6;

    } else if (cardis < 30) {
      diserror = 6;

    } else if (cardis < 35) {
      diserror = 7;

    } else {
      diserror = 8;

    }
  }
  } else if (cardis == 0 && x < 23000) { //候車追上
  diserror = 9;
} else if (cardis == 0 && x < 40000) { //前車控速 轉回時開啟
  diserror = 0;
} else if (cardis == 0 && x < 71300) { //後車追上
  diserror = 9;
} else {
  diserror = 0;
}
}

```

4. 最後的輸出

我們最後無刷的輸出會有一個底的值(32)加上速度的誤差以及跟車的誤差。這樣我們最大的值會是 50 左右，是我們不樂見的，因為在 38 左右我們的加速度已經足夠了。所以我將輸出限制在 20~40 的範圍裡，然後如果有需要 boost 時再加上五。最後當他跑到終點(71300mm 左右)時就將 duty 下修 20，這樣風扇就會不轉。

```

output = chooseduty(x) + fanpid(1, 0.3, verror, verror) + diserror ;

//=====write fan=====

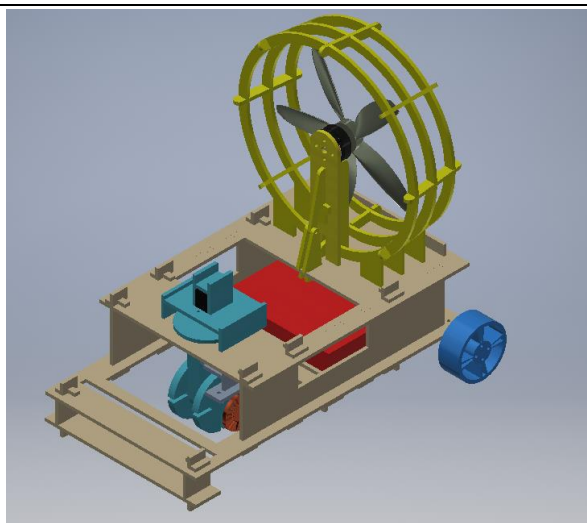
realoutput = constrain(output, 20, 40);

if (x < 71300 && boost1 == 0) {
  brushless.write(realoutput ); //一般情況
} else if (boost1 == 1) {
  brushless.write(realoutput + 5); //跟車前段加速
  if (velocity > 420) {
    boost1 = 0; //到達目標速度停止加速
  }
} else {
  brushless.write(20);
}
}

```


4.4 期中技術規格

外觀



尺寸及重量 長×寬×高：28cm × 18.5cm × 26.5cm(不含黑色紙板)

重量：1.127kgw(含車手)

車體 三輪車

全車採密集板材質(除後車輪與風扇為 3D 列印)

轉向機構 單輪馬達轉向

MG996 伺服馬達

循跡模組 紅外線感測器

6 × TCRT5000 循跡模組

煞車系統 摩擦地板減速

MG90s 伺服馬達

推進系統 3D 列印 5 葉風扇

MT2202 2300kV 無刷馬達

軟體控制



Arduino UNO × 2

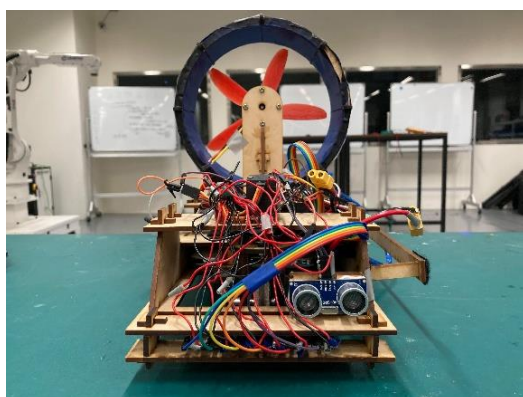
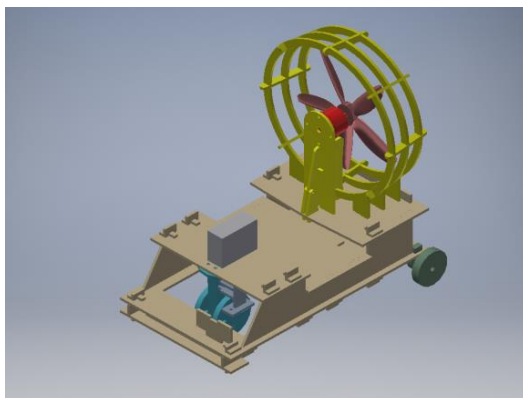
電力供應 11.1V 鋰聚電池供應無刷馬達

5V 行動電源供應伺服馬達與循跡模組

15A 天行者電變

4.5 期末技術規格

外觀



尺寸及重量 長×寬×高：28cm × 18.5cm × 26.5cm(不含黑色紙板)
重量：1.2263kgw(含車手)

車體 三輪車
全車採密集板材質(除車輪與風扇為3D列印)

轉向機構 單輪馬達轉向
MG996 伺服馬達

循跡模組 紅外線感測器
6 × TCRT5000 循跡模組

推進系統 3D 列印 5 葉風扇
A2212 2200kV 無刷馬達

跟車系統 超音波感測器
HCSR04 × 1

軟體控制 
Arduino Mega × 1

電力供應 11.1V 鋰聚電池供應無刷馬達
5V 行動電源供應伺服馬達、循跡模組與超音波感測器
40A 天行者電變

4.6 期末 BOM 表

編號	品項	規格	數量	單價	總額	備註
1	密集板	300*400*3(mm)	2	25	50	
2	培林	外12內6厚3(mm)	4	60	240	
3	伺服馬達	MG996R	1	190	190	
4	無刷馬達	XXD 2200KV	1	150	150	
5	電子變速器	SkyWalker 40A	1	280	280	
6	紅外線模組	TCRT5000	6	20	120	
7	測速器	AD10-speed2	1	60	60	
8	Arduino板	MEGA	1	800	800	學長二手
9	電池	Desire Power V8 鋰電池	1	417	417	11.1V 1300mAh
10	杜邦線		20	4	80	
11	黑紙	A4	1	11	11	
12	超音波感測器	HCSR04	1	60	60	
13	螺絲		21	0	0	實作中心提供
14	螺帽		21	0	0	實作中心提供
15	墊片		10	0	0	實作中心提供
16	前後輪金屬軸	直徑6mm	1	0	0	組員提供
17	按鈕開關		1	10	10	
18	行動電源	10000mAh	1	0	0	組員提供
19	魔鬼氈	5*15(cm)	1	16	16	
20	USB(TypeB)線	30cm	1	40	40	
21	風扇	直徑14cm	1	0	0	實作中心機台製作
22	輪胎皮	7.5mm厚(網球拍握把布)	1	0	0	組員提供
23	伺服馬達	MG90s	1	86	86	裝飾品
				總花費	2610	

總花費符合氣動車設計規範(<3000 元)。

5. 設計分析與驗證

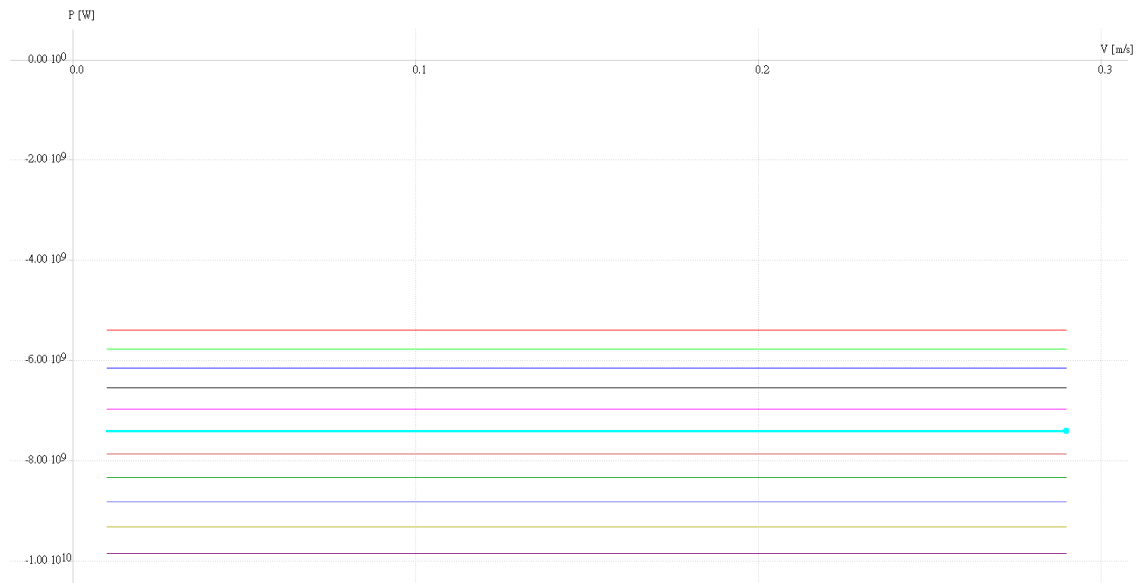
5.1 分析

以簡易的分析，讓我們知道一些物理的概念與方向，以利後續的設計與製造以及對成果的優化與驗證。

5.1.1 風扇模擬分析

Muiti parameter analysis:

在期末測試中，我們的風扇轉速設定約為 5000rpm、車子的最高行進速度約為 30cm/s，因此將參數設定為 $4500\text{rpm} < \omega < 5500\text{rpm}$; $1\text{cm/s} < v < 30\text{cm/s}$ 。



Structural Analysis:

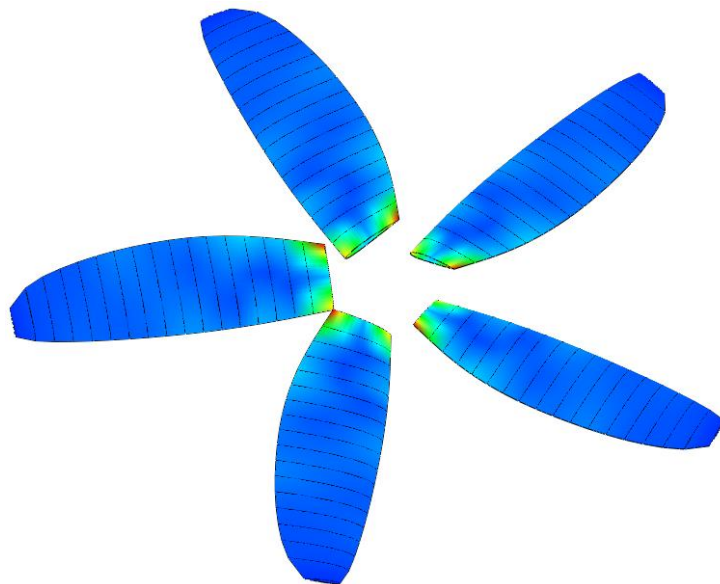
車速(風速)、功率與轉速的示意圖如上，不同顏色表示不同轉速下風速與功率的關係，由上圖可知轉速越高時所需功率也會越大。

目的：想了解葉片在運轉時的應力分布

需要器材：Q blade 軟體

測試方式：將以建立好的 Q blade 風扇直接進行應力分析，假設葉片平均受 10(N)的垂直力。

結果：

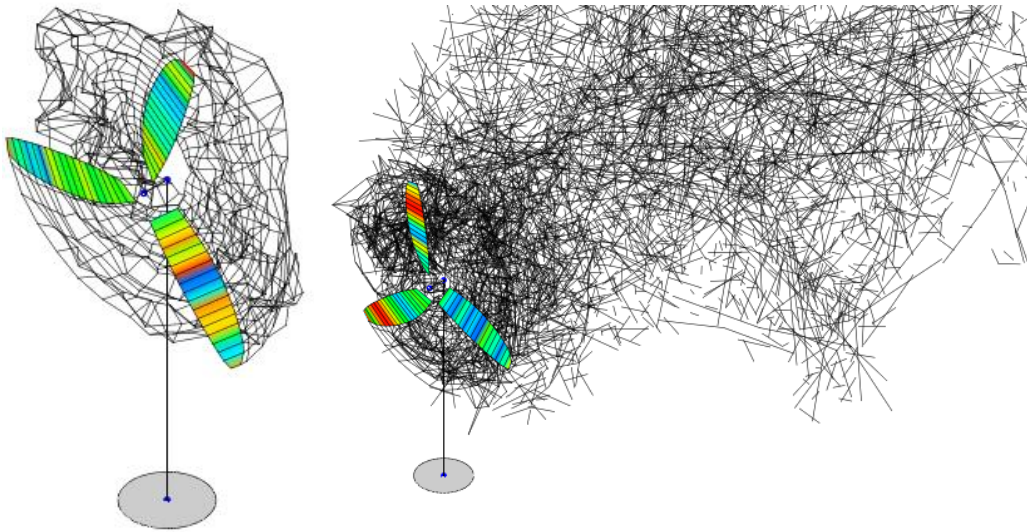


圖七十六：Qblade 應力分析

討論：

使用 Q blade 進行應力分析，可了解應力主要集中的地方，進而在設計與修改風扇時能夠將其考慮在內。由上圖可觀察到，應力主要還是集中在與中心連接

處，因此我們在設計與修改的過程中，主要考慮增加弦長，以增加銜接處的穩固性。

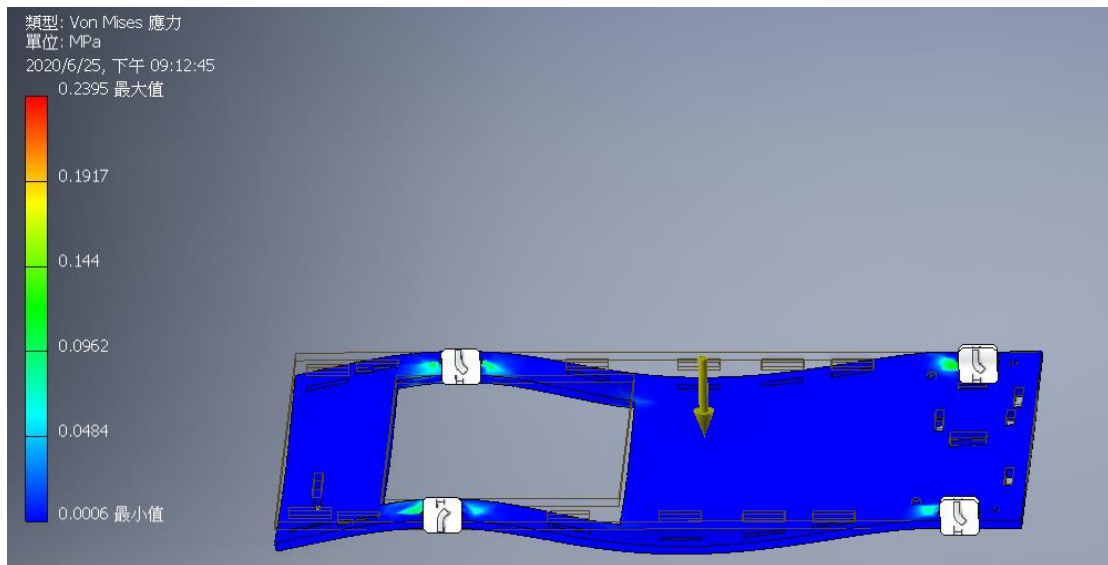


然而在轉動模擬的受力圖中我們可以看到實際的受力並非中心最大，而會因為風扇轉動時帶動氣流而造成中間部份壓力最大；此外我們也可以從氣流的示意圖（黑色網狀部分）看出，由於 5000rpm 的轉速與 30cm/s 的車速相較之下比值太大，而會造成紊流，也就是被風扇往後帶的空氣並不能形成一個順暢的螺旋狀氣流，因此也會造成扇葉需要使用更大的力量（對抗更大的阻力）才能帶動車體往前行進。

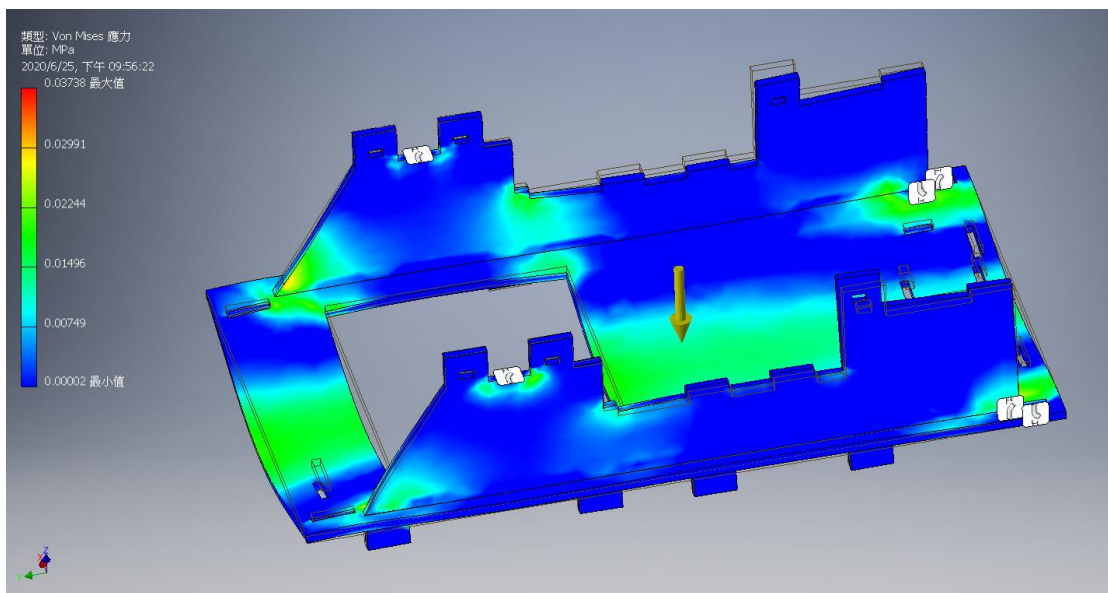
5.1.2 車體底板應力分析

軟體：Autodesk Inventor

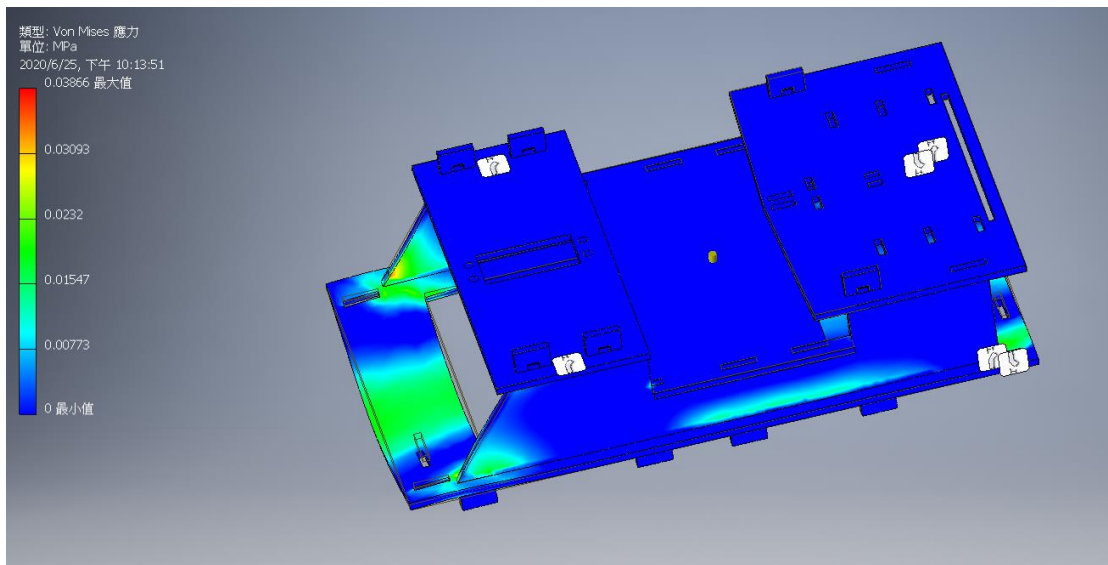
施加向上的正向力在三個車輪處，而重心置於車體中央靠後的位置，在單純看下板時，可以看到放置前輪處，由於板材太細，而有明顯之變形的情況。



圖七十七：車體下版應力分析



圖七十八：車體重心之受力應力分析圖

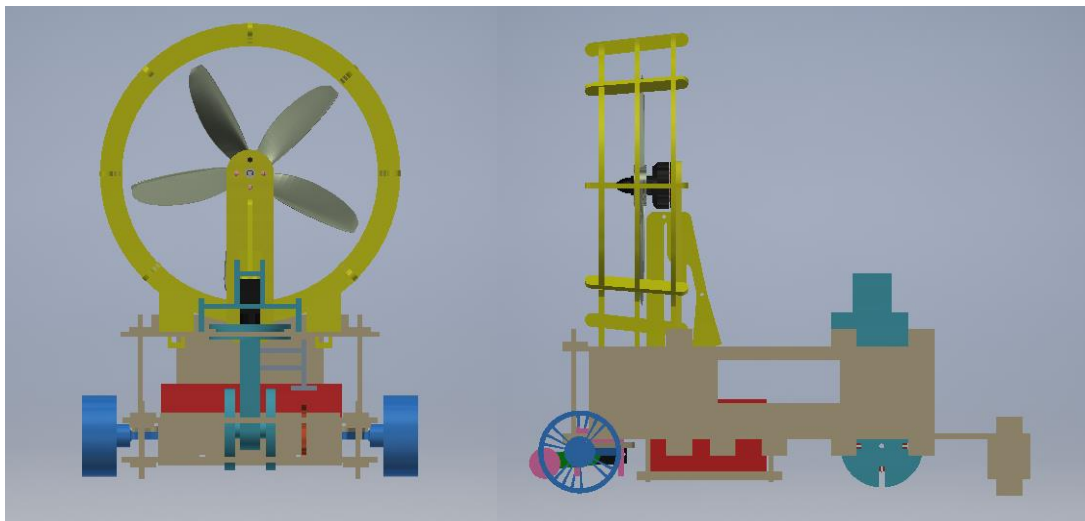


圖七十九：車體整體之應力分析圖

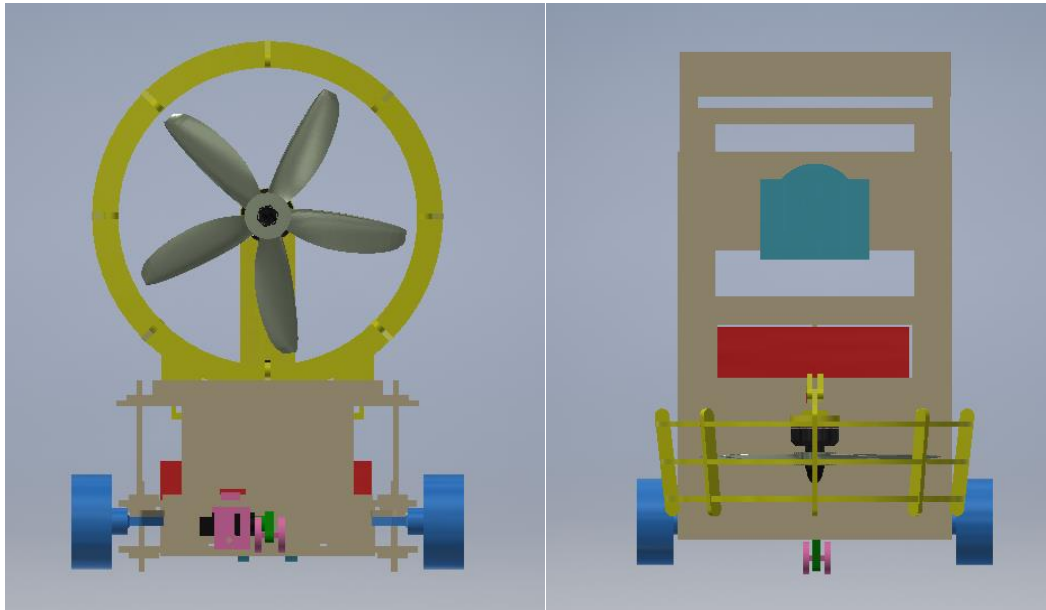
再加上側板的支撐材料，以及上板的嵌合，可以看到應力相對前幾張圖沒有那麼集中，而側板相對應力集中之處，也是位於邊角的部分，這點可以透過加入導角來改善，就整體來說，應力特別集中，材料相對脆弱的地方位於整體板材的中央，由於主要零件都分布在此，再加上無其他支撐材料，所以相對不良，但整體分析來說，其實還算可以，都還在材料的耐受範圍內，因此我們的期末車體就維持原樣，而未再更動。

5.2 繪製工程圖(組合圖)

期中車體組合圖：

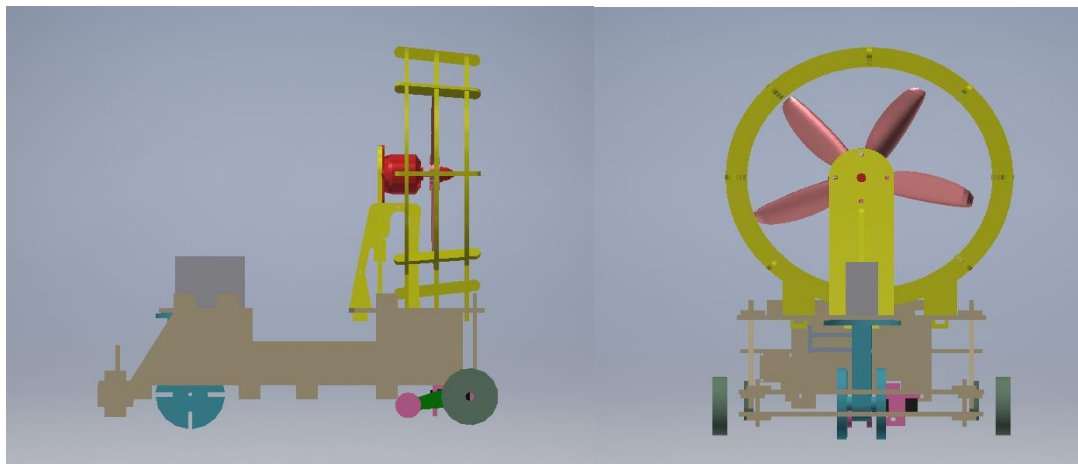


圖八十：氣動車立體圖前視圖與側視圖

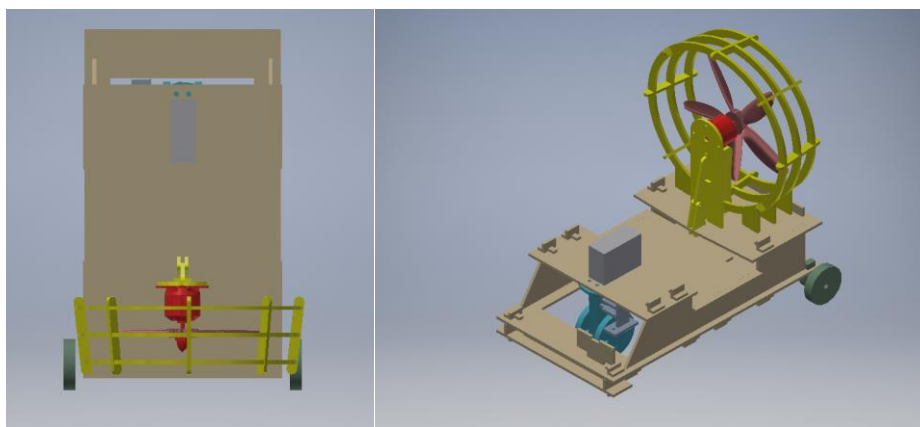


圖八十一：氣動車立體圖後視圖與上視圖

期末車體組合圖：



圖八十二：氣動車立體圖側視圖與前視圖



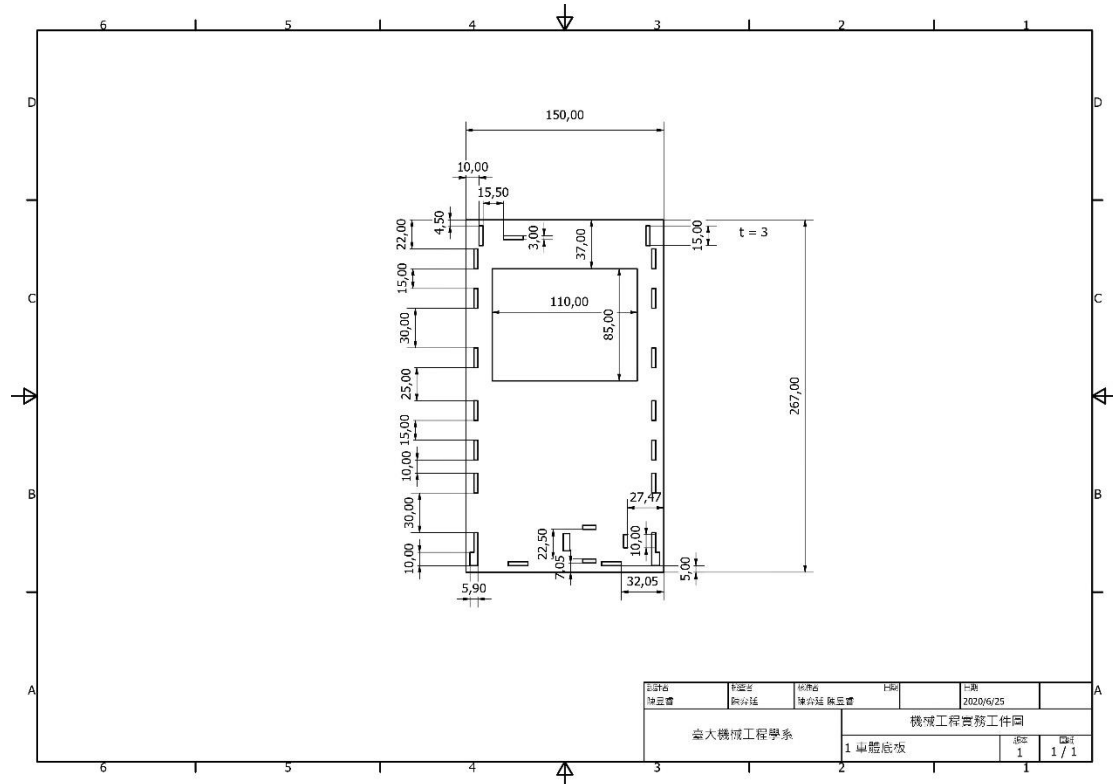
圖八十三：氣動車立體圖上視圖與視圖

以下整理了期末的主要車體零件：

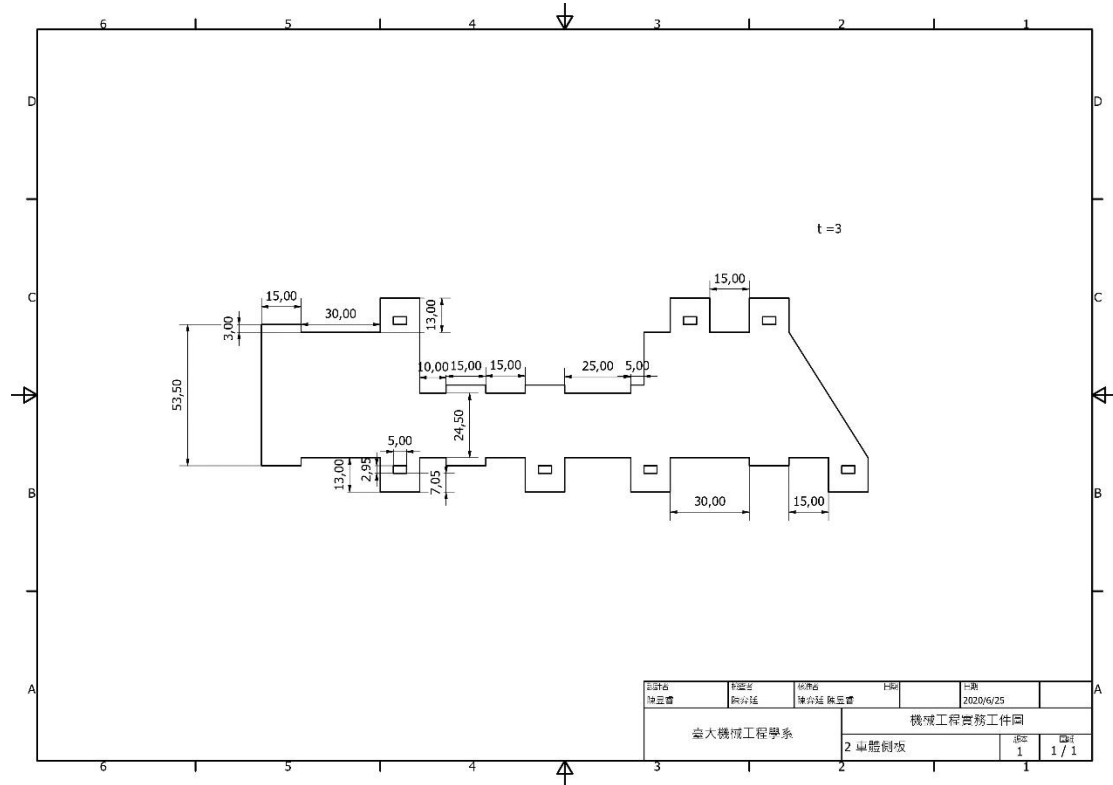
車體零件表

編號	零件名稱	數量
1	車體底板	1
2	車體側版	2
3	車體上板(前)	1
4	車體上板(中)	1
5	車體上板(後)	1
6	車體後版	1
7	風扇罩 1	1
8	風扇罩 2	1
9	風扇罩 3	1
10	車體卡準	18
11	風扇前支撐軸	2
12	風扇後支撐軸	1
13	風扇後固定座卡準	1
14	風扇罩卡準	12
15	風扇主軸前固定座	1
16	風扇葉片	1
17	風扇-上板固定卡準	2
18	風扇側支撐軸	2
19	無刷馬達	1
20	電變	1
21	鋰電池	1
22	傳輸線	1
23	行充	1
24	MEGA 板	1
25	TCRT7000 循跡感測器	6
26	超音波感測器	1

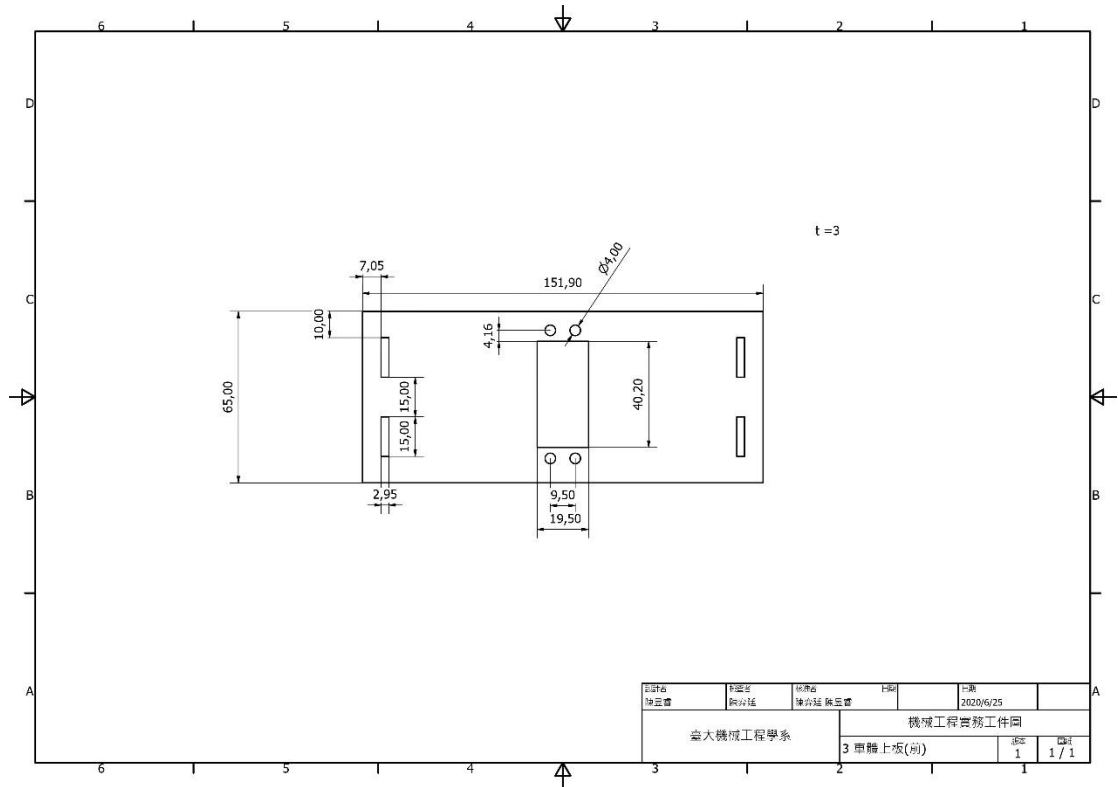
5.2.1 車體



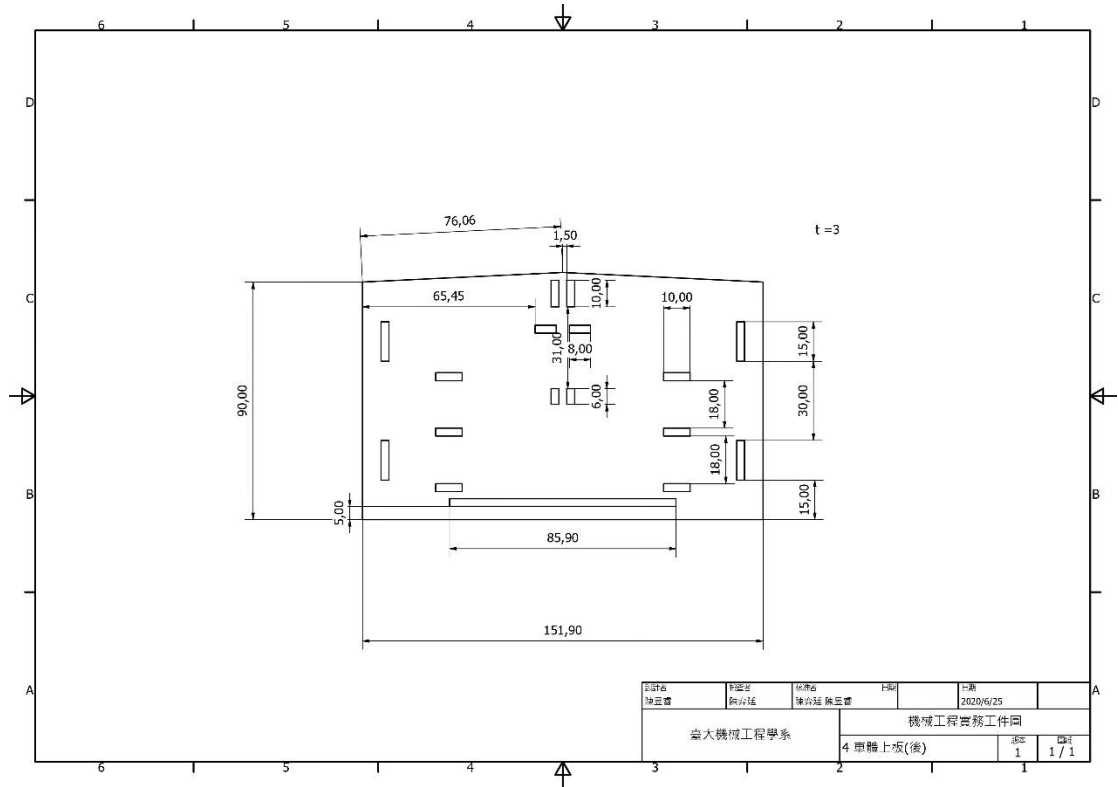
車體底板 2D 工程圖



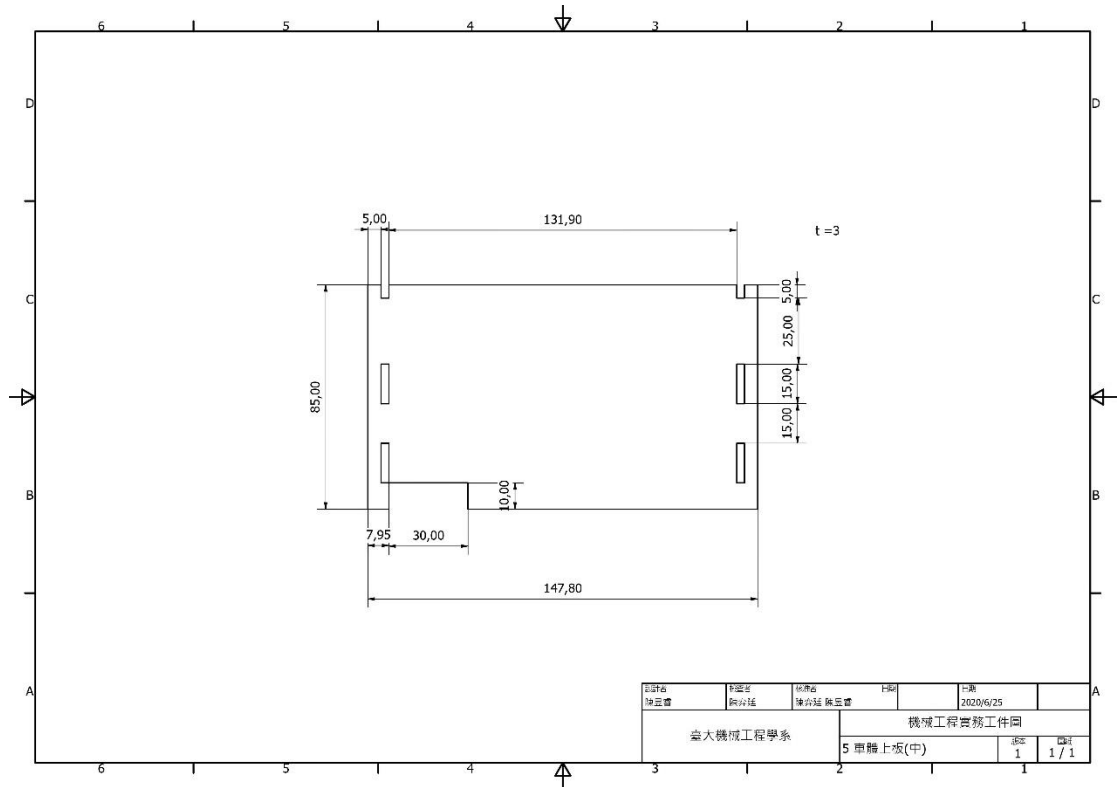
車體側板 2D 工程圖



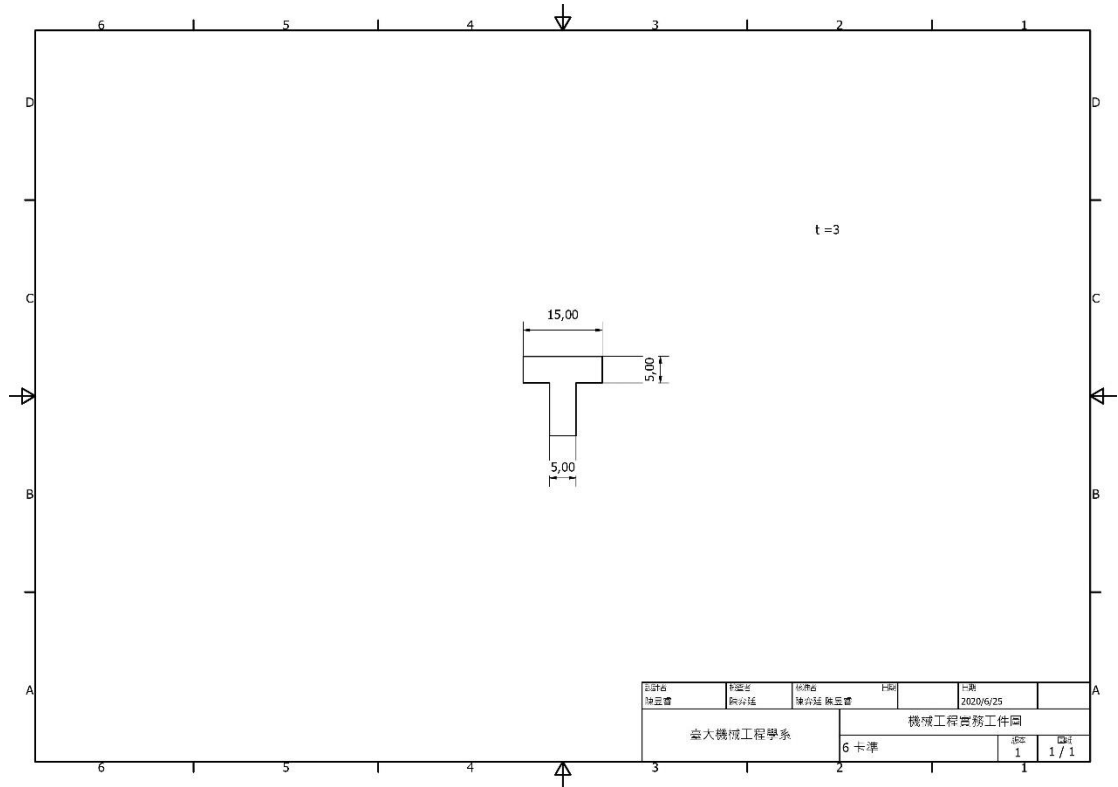
車體上前板 2D 工程圖



車體上後板 2D 工程圖

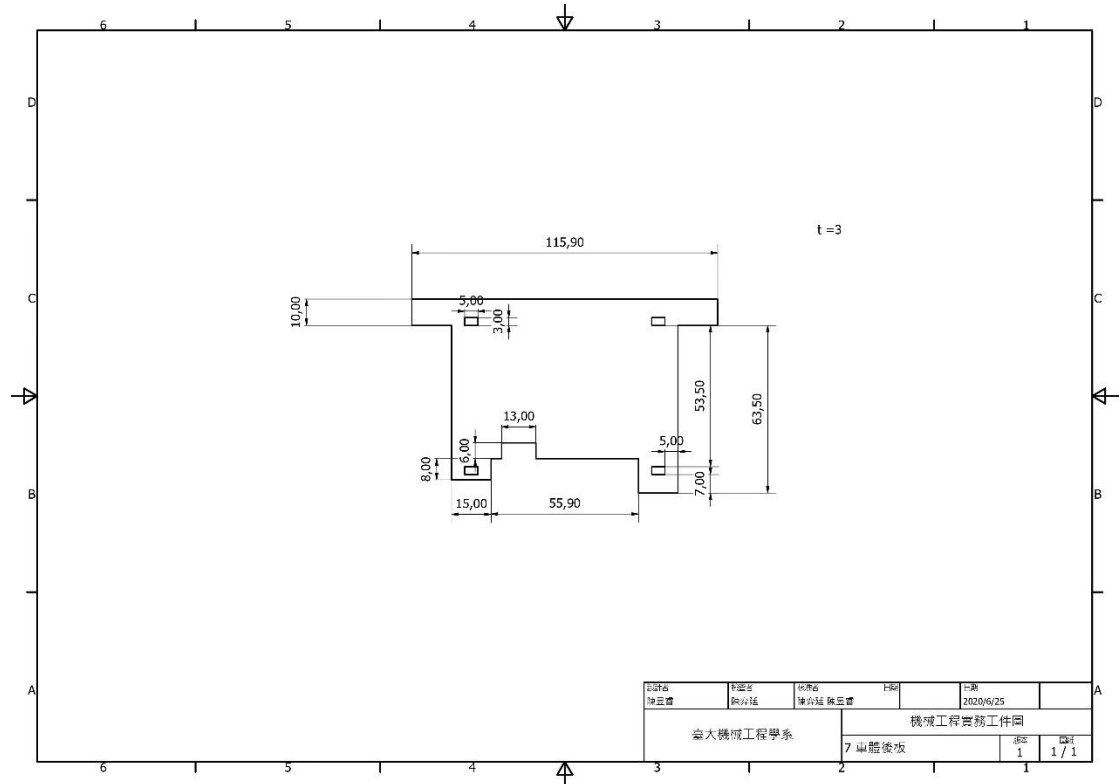


車體上中板 2D 工程圖

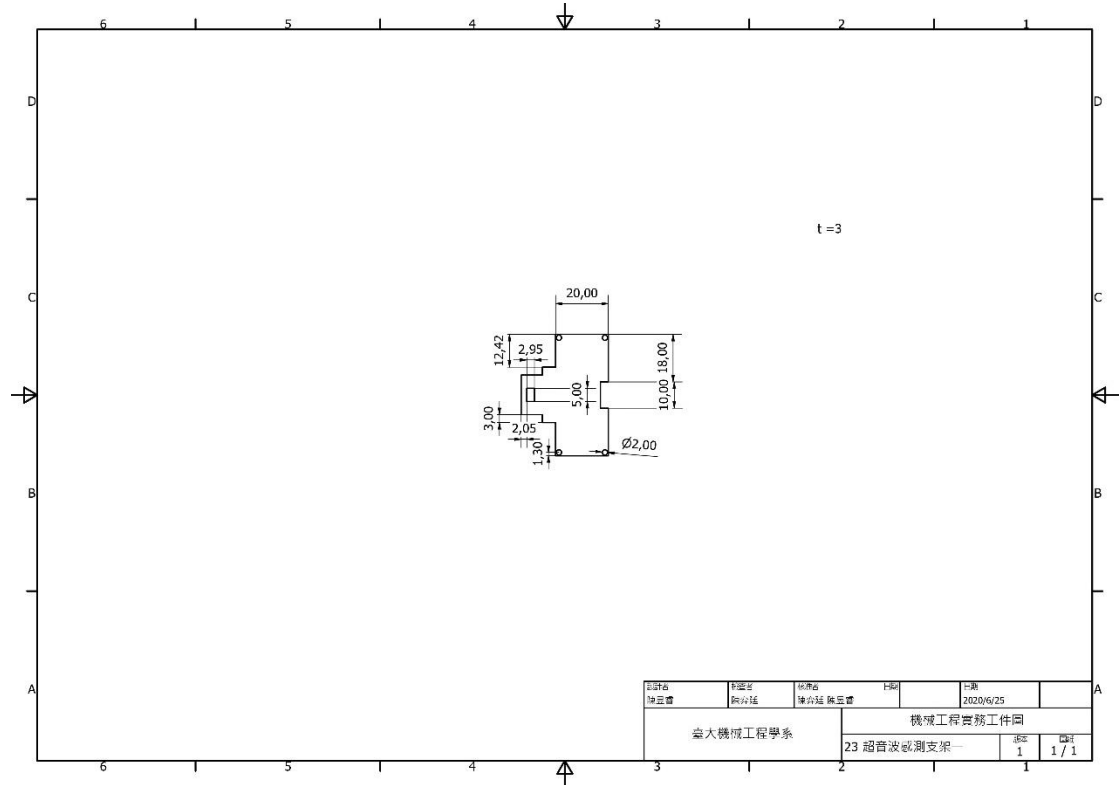


車體卡準 2D 工程圖

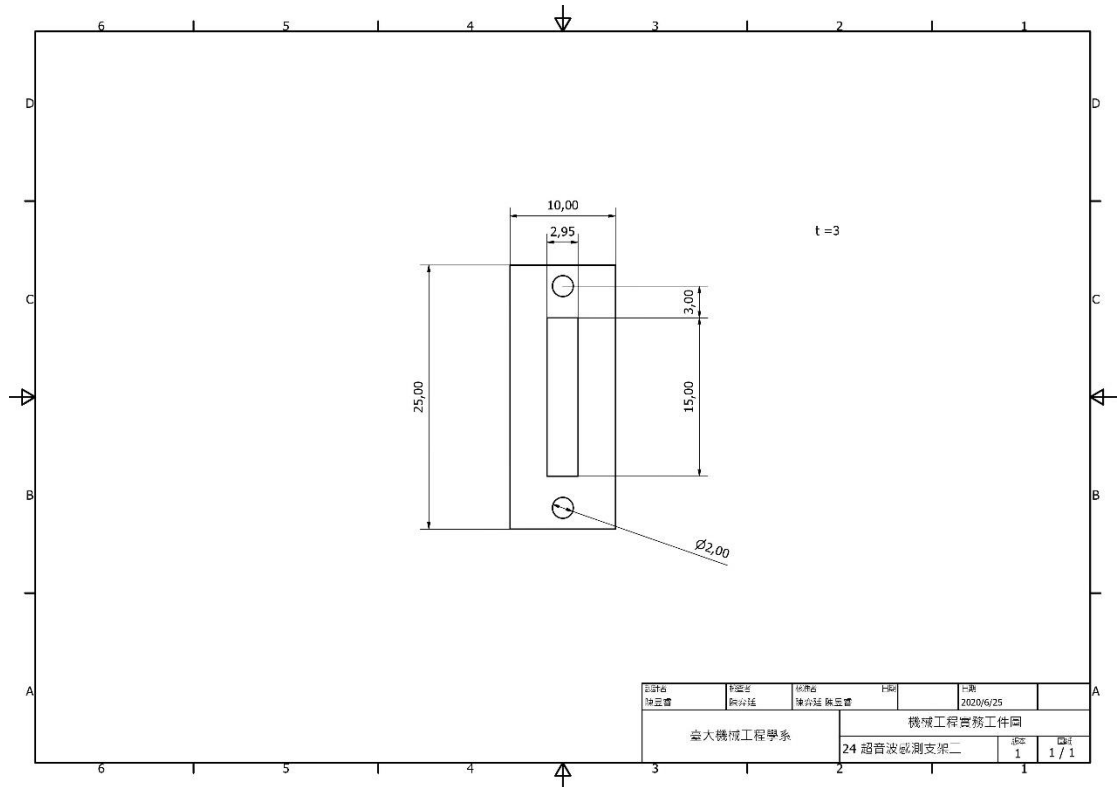
5.2.2 車輪組



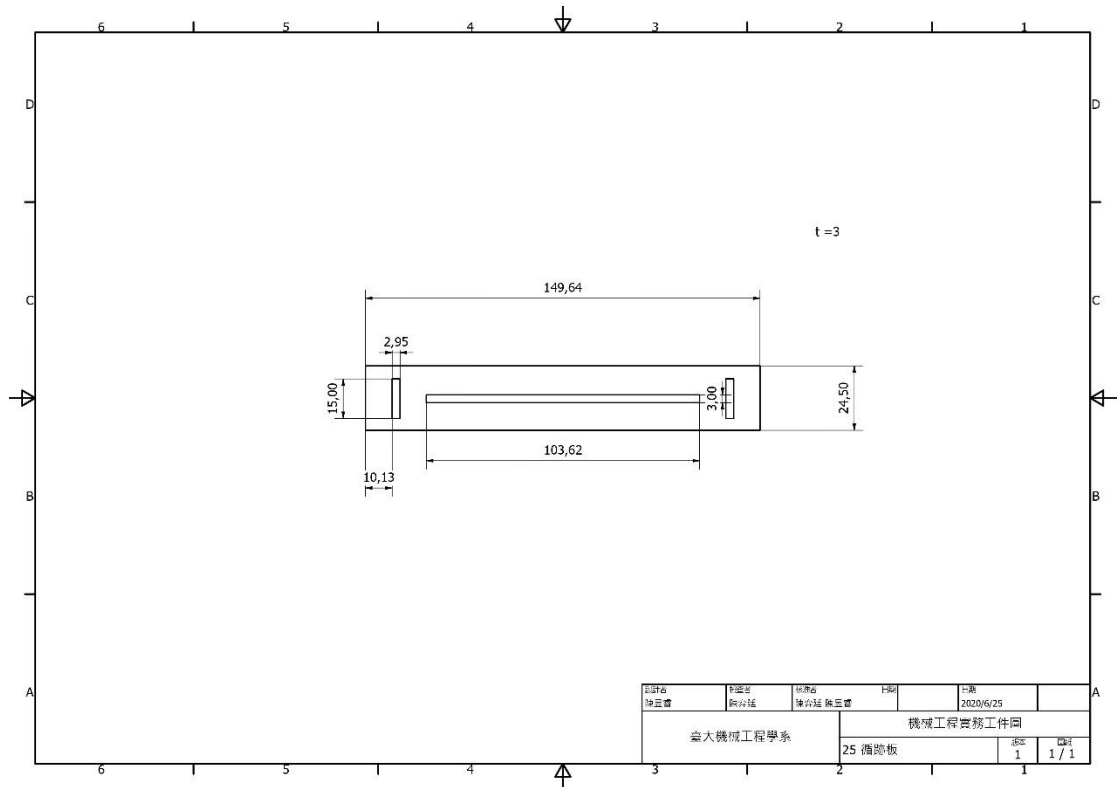
車體後板 2D 工程圖



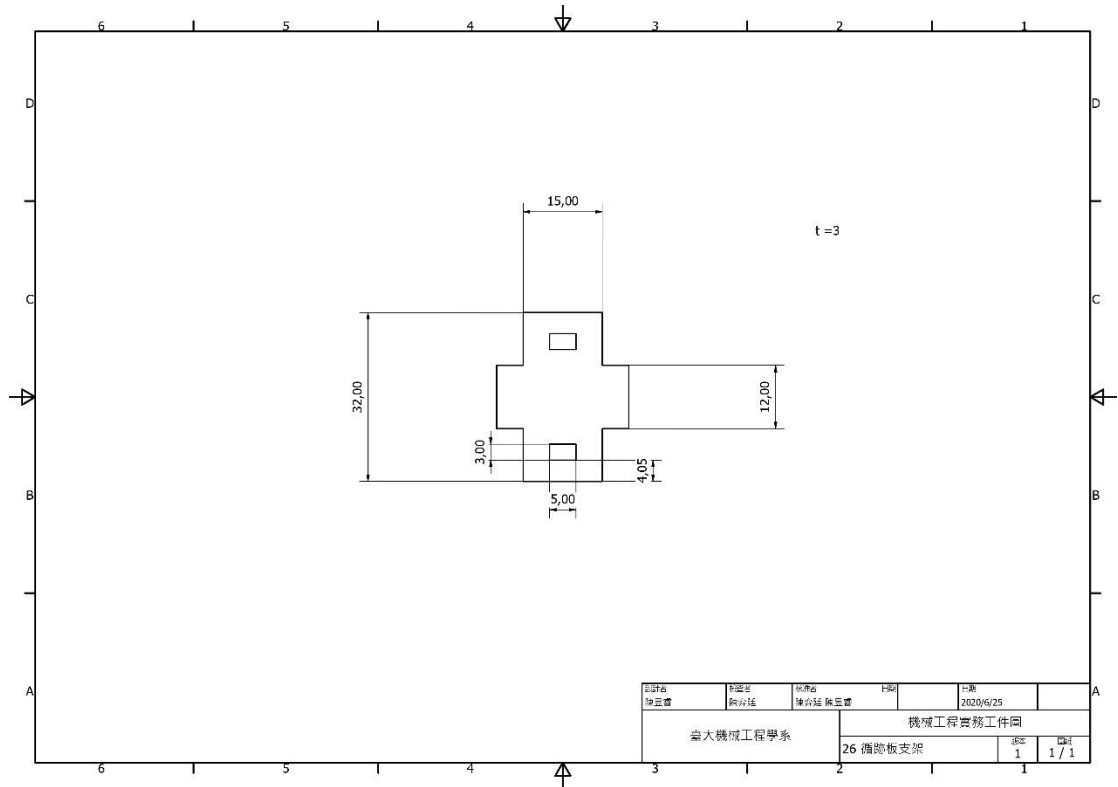
超音波測距卡準 2D 工程圖



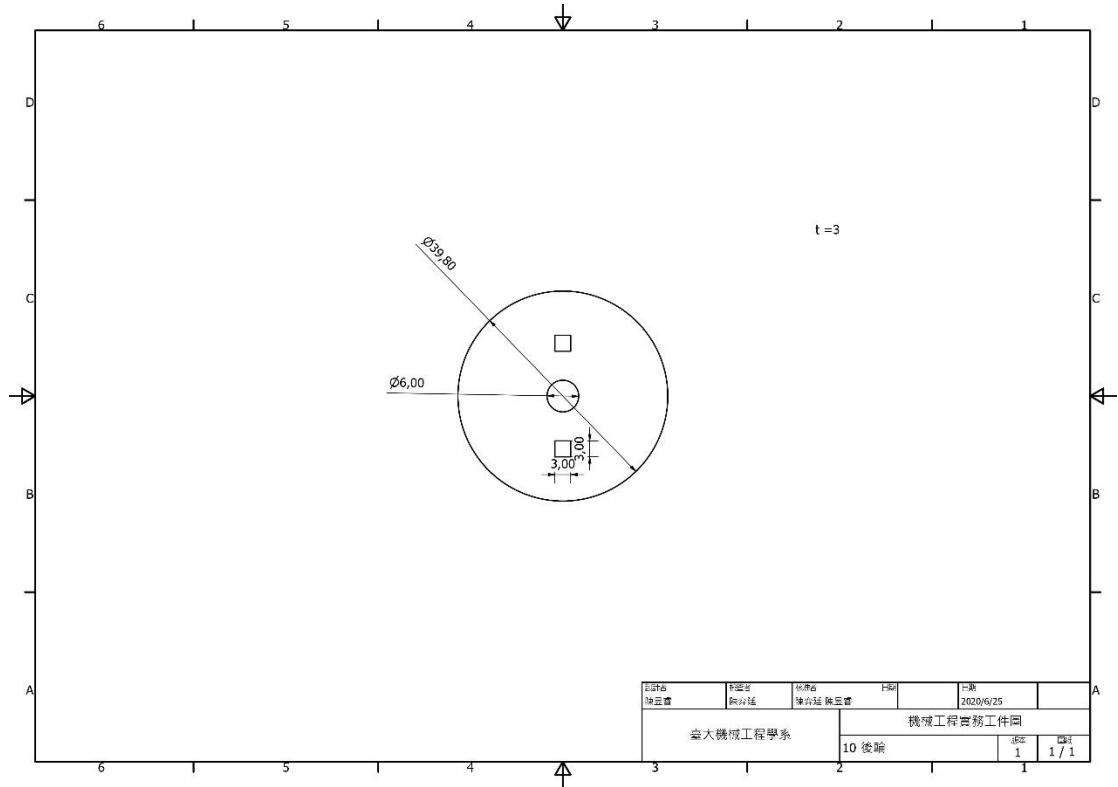
超音波卡準固定座 2D 工程圖



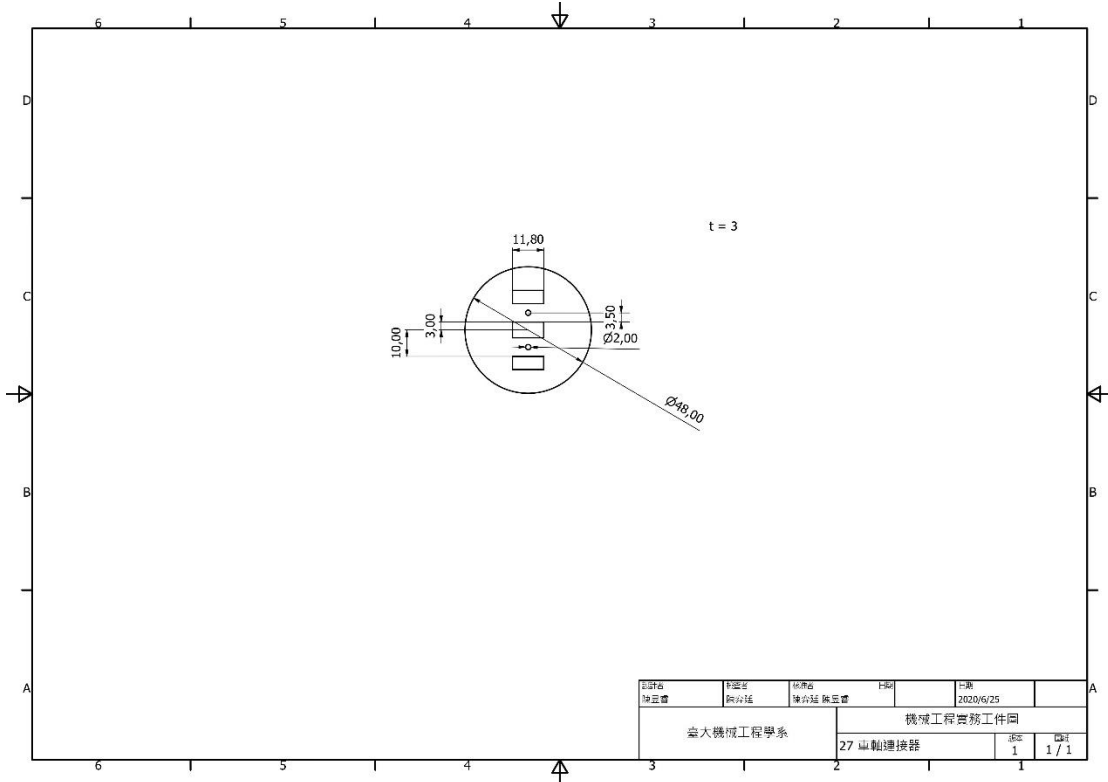
紅外線循跡卡準 2D 工程圖



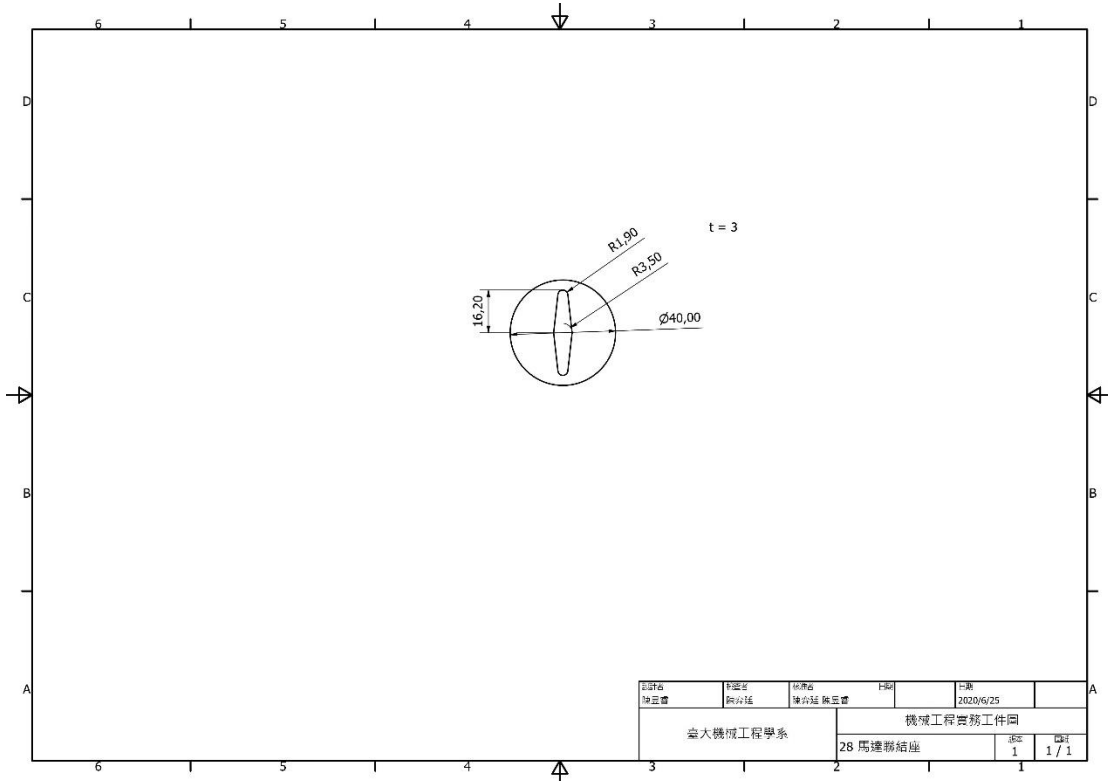
紅外線循跡側邊固定座 2D 工程圖



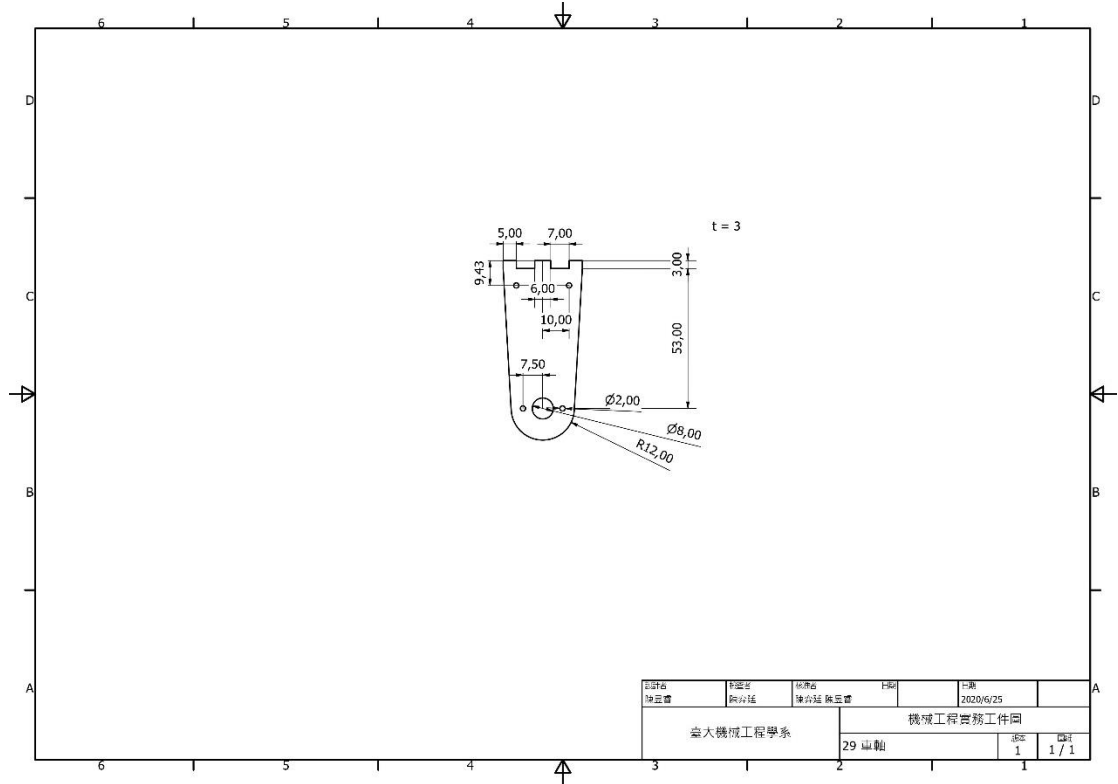
輪輪胎 2D 工程圖



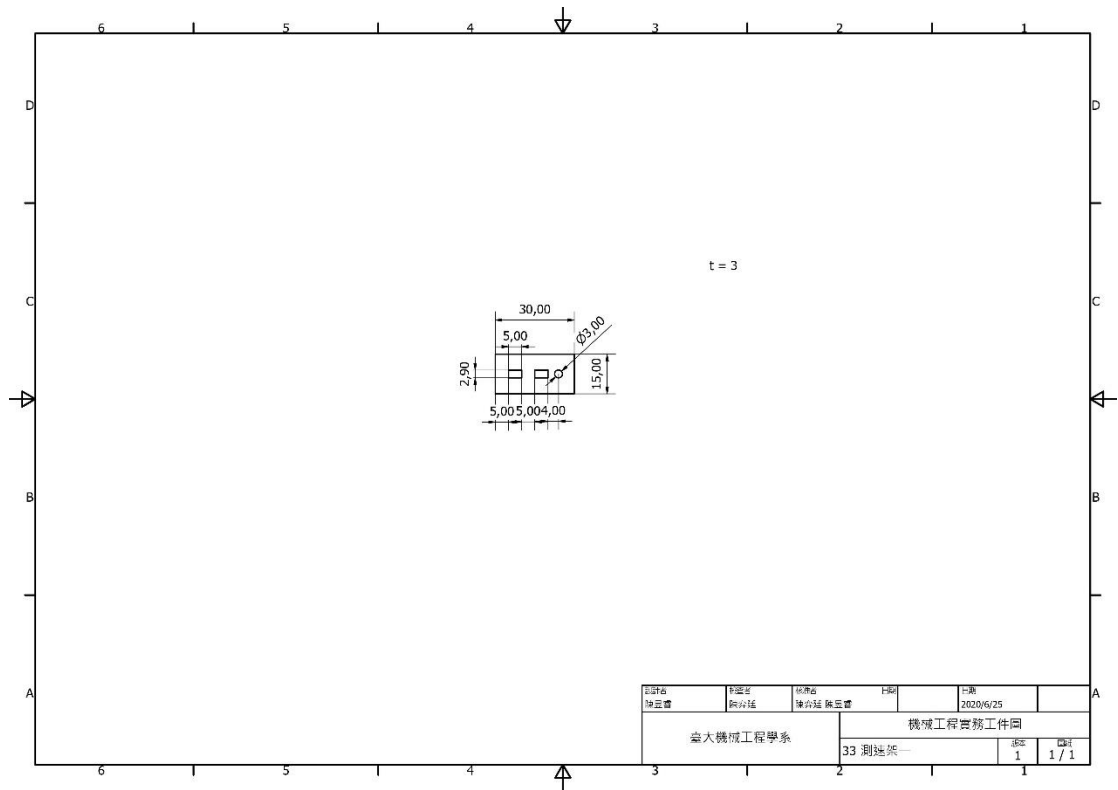
前輪-舵機固定板 2D 工程圖



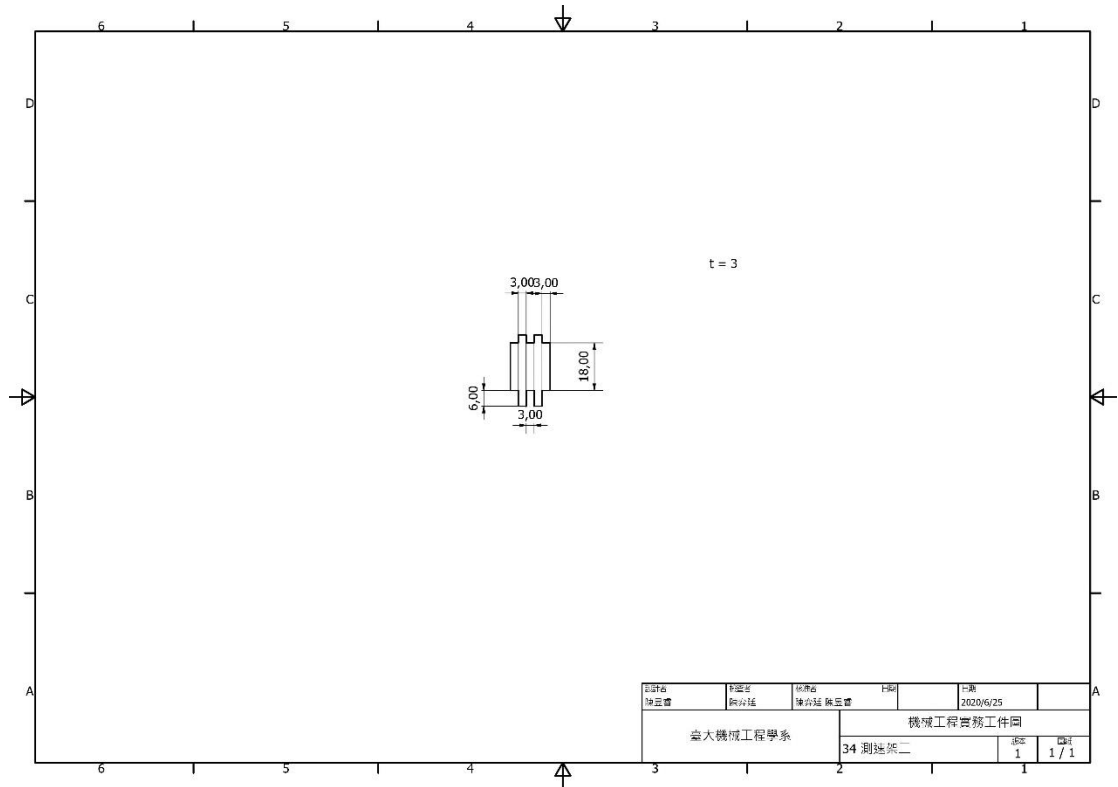
前輪-舵機連接板 2D 工程圖



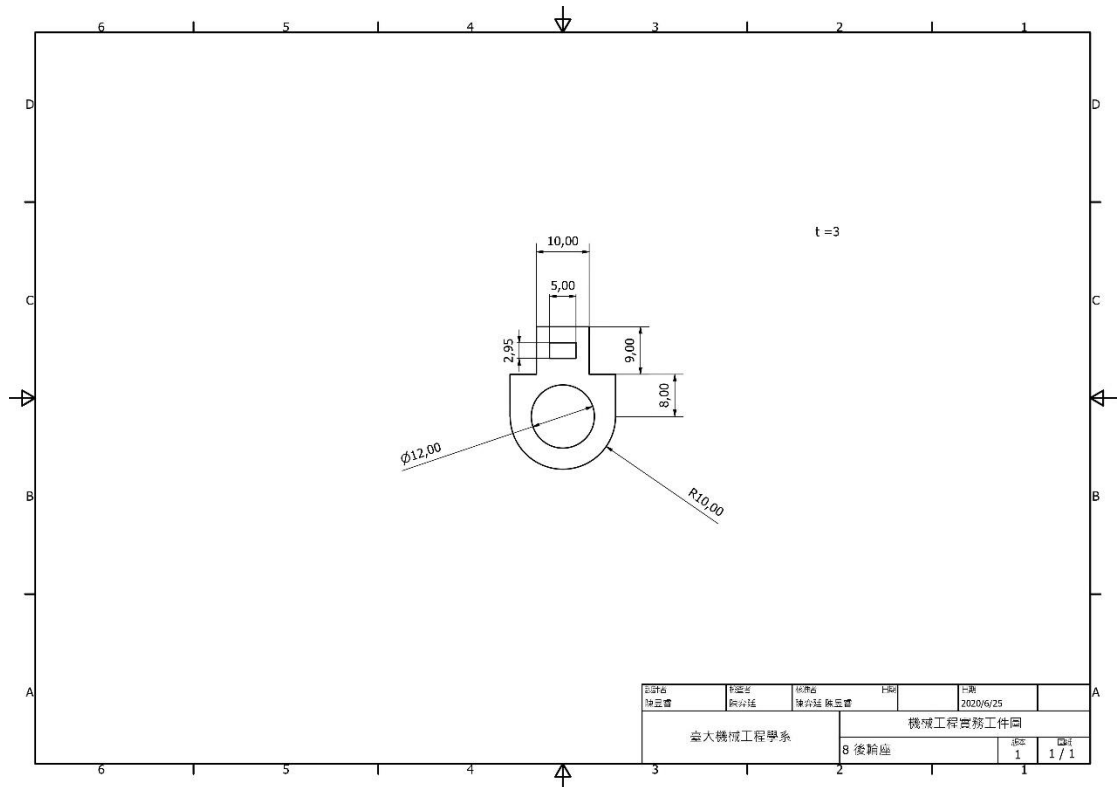
前輪車軸 2D 工程圖



測速計固定座 2D 工程圖

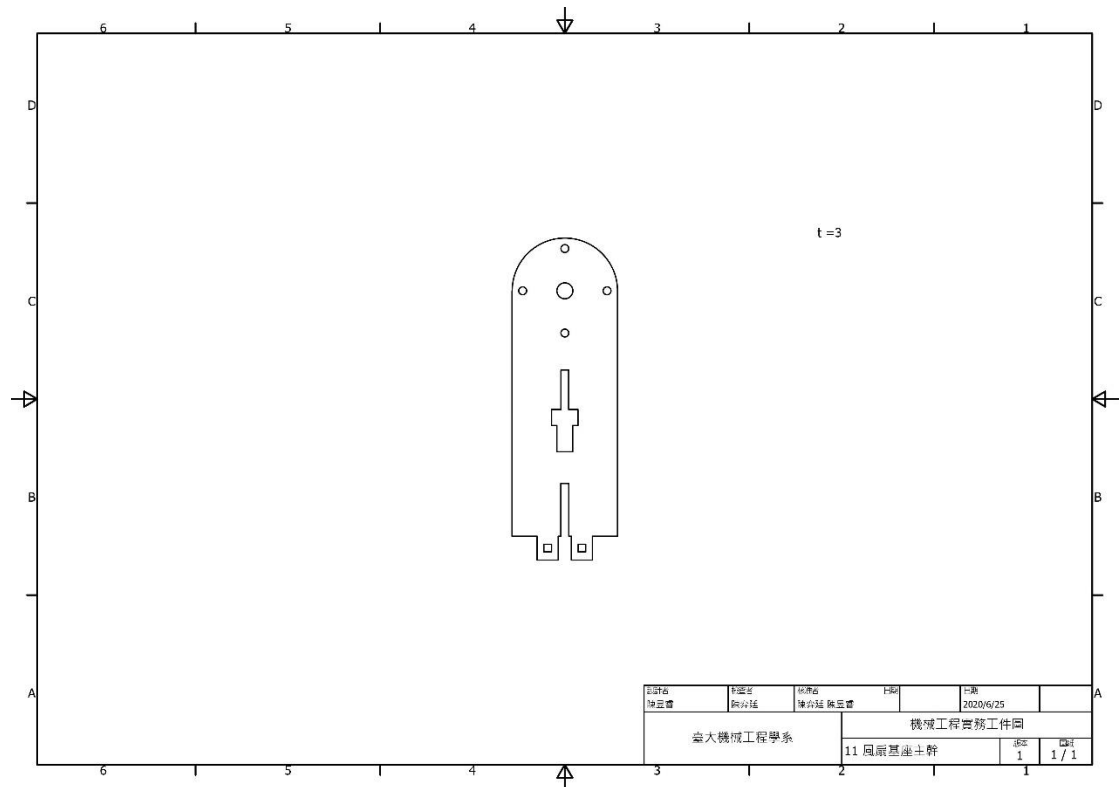


測速計卡準 2D 工程圖

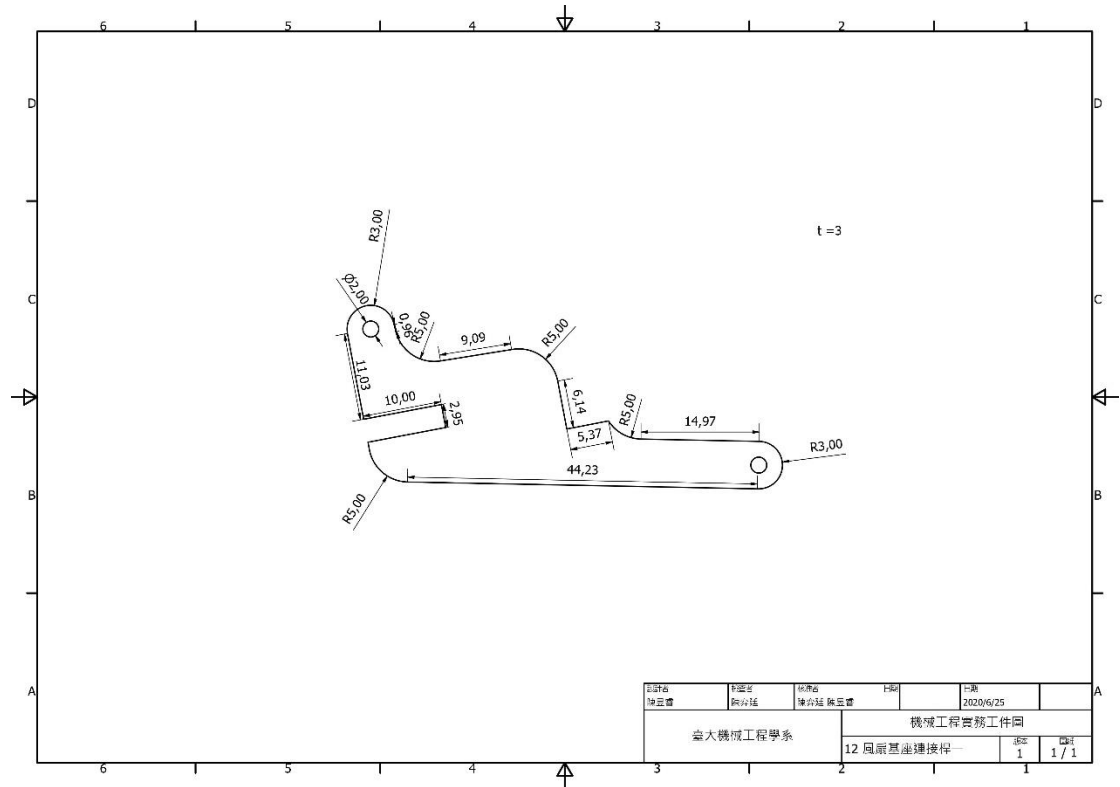


後輪支撐架 2D 工程圖

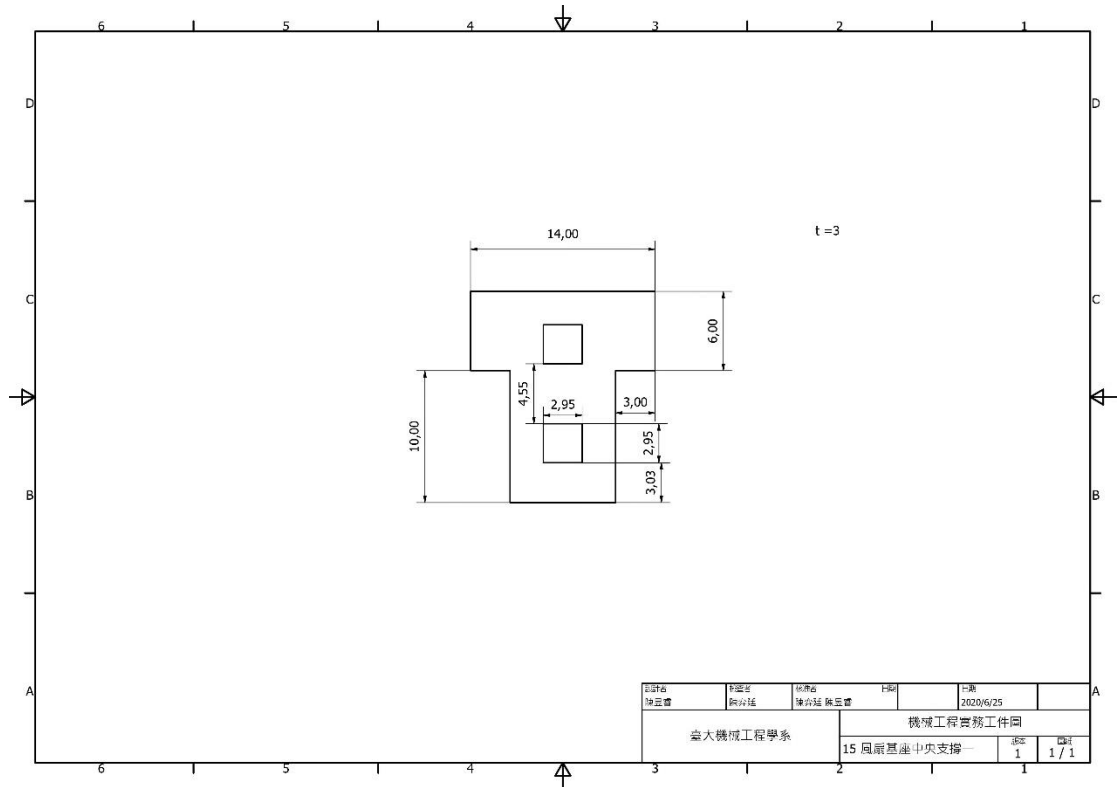
5.2.3 風扇架與風扇罩



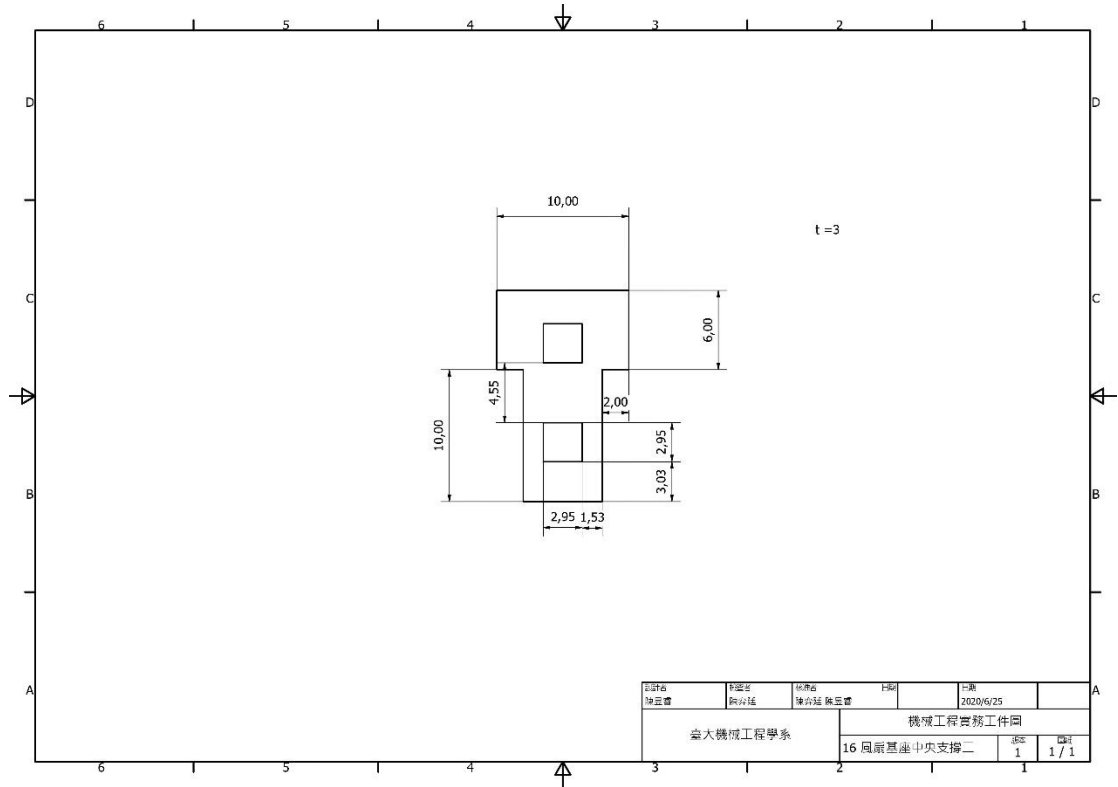
風扇主軸 2D 工程圖



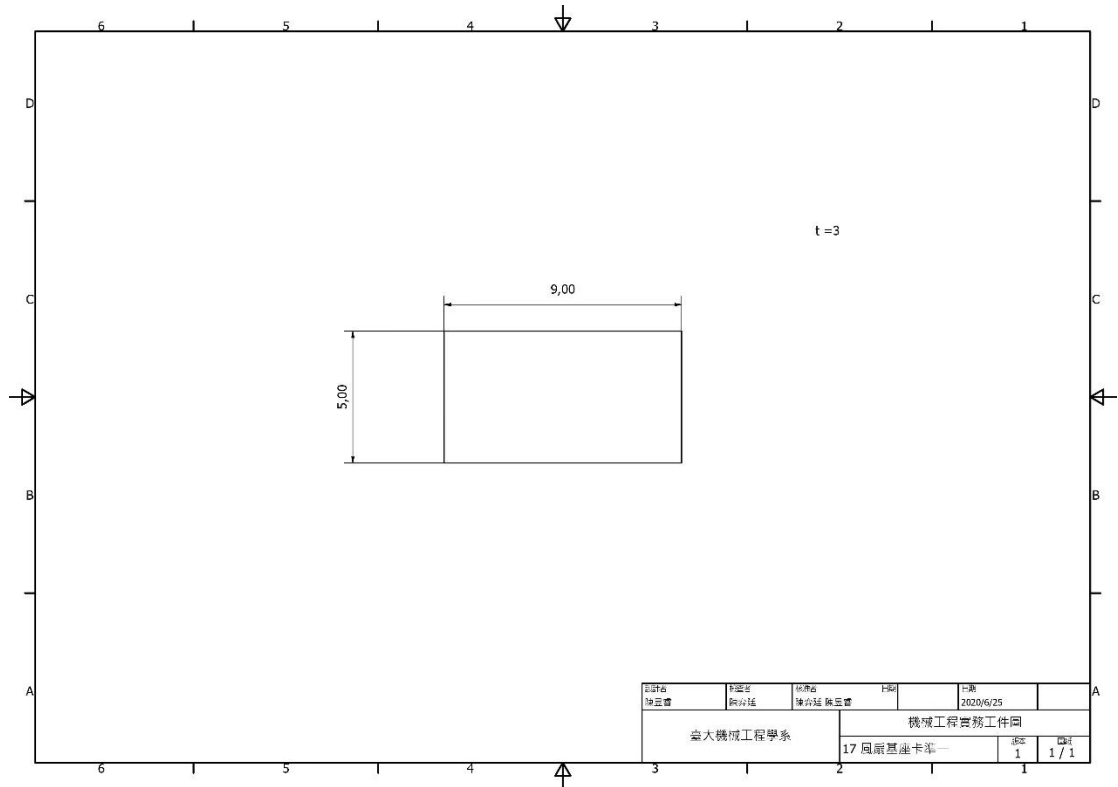
風扇前固定座 2D 工程圖



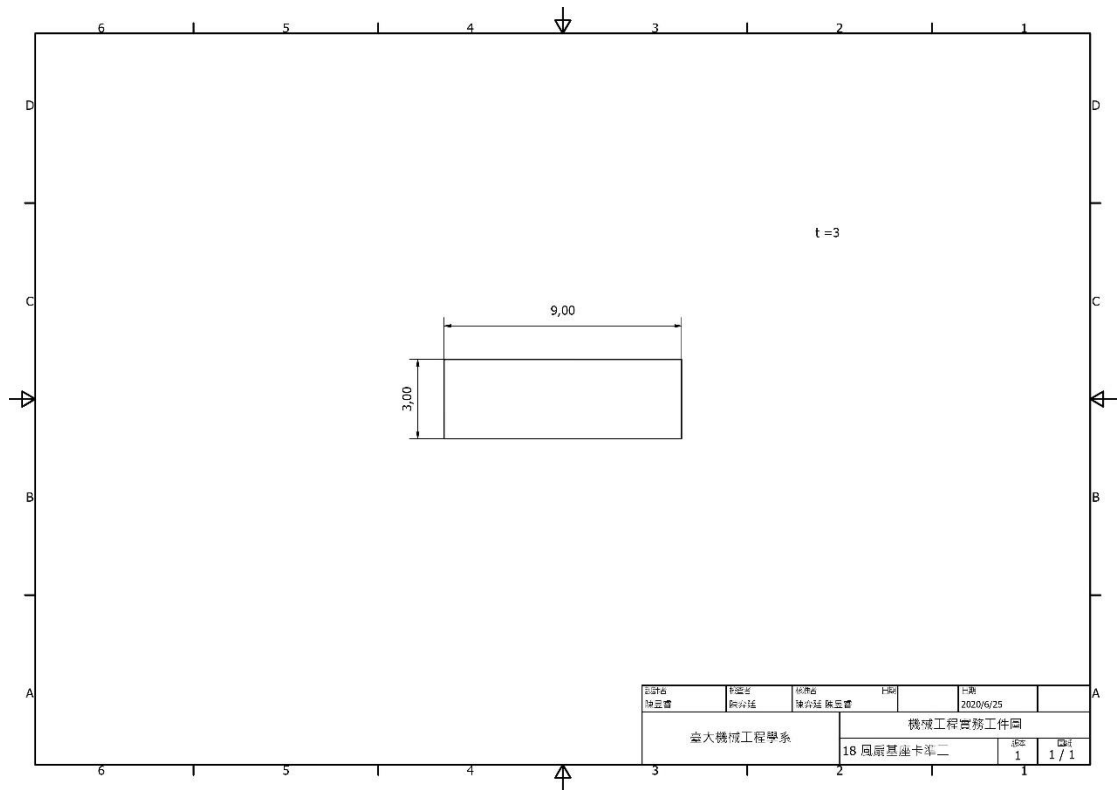
風扇前固定座卡準 2D 工程圖



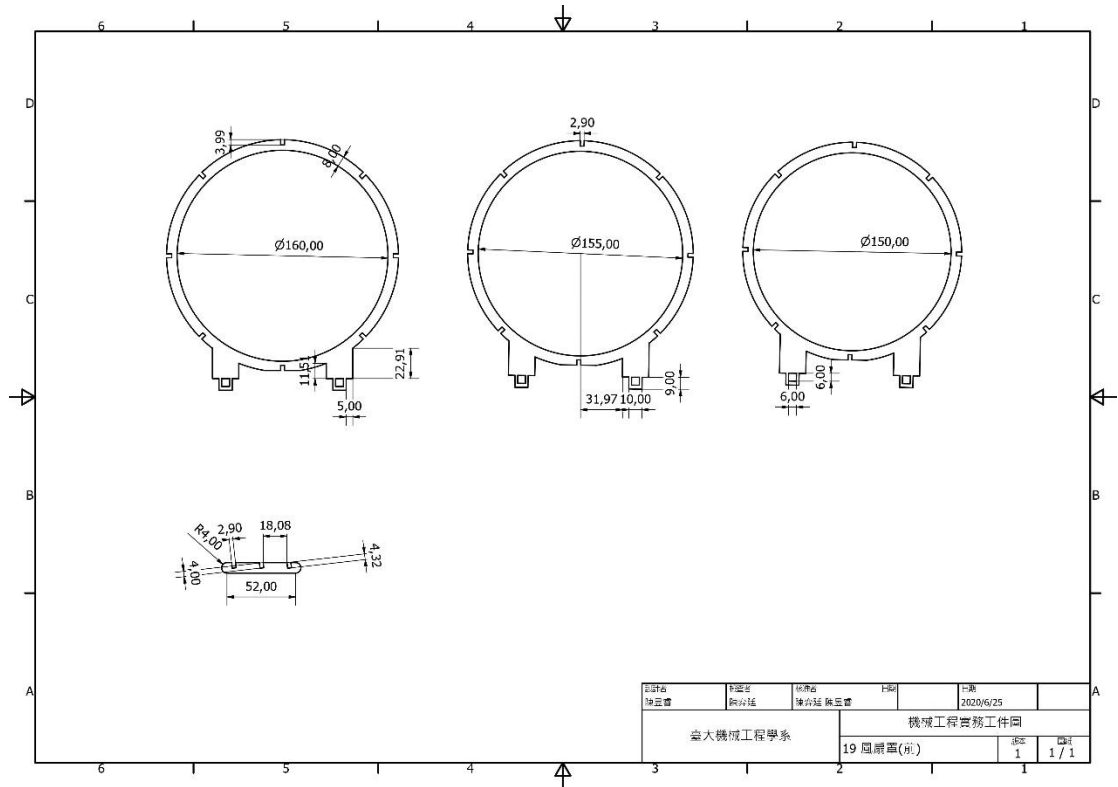
風扇後固定座卡準 2D 工程圖



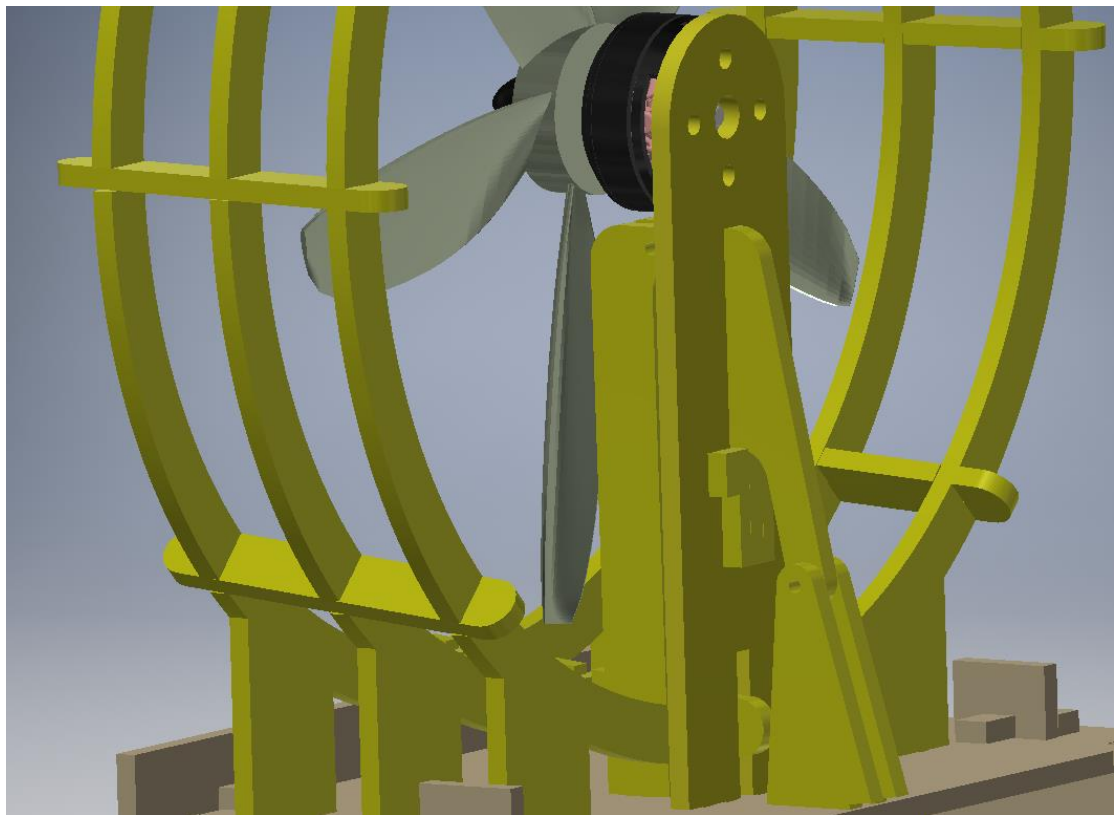
風扇-上板固定卡準 2D 工程圖



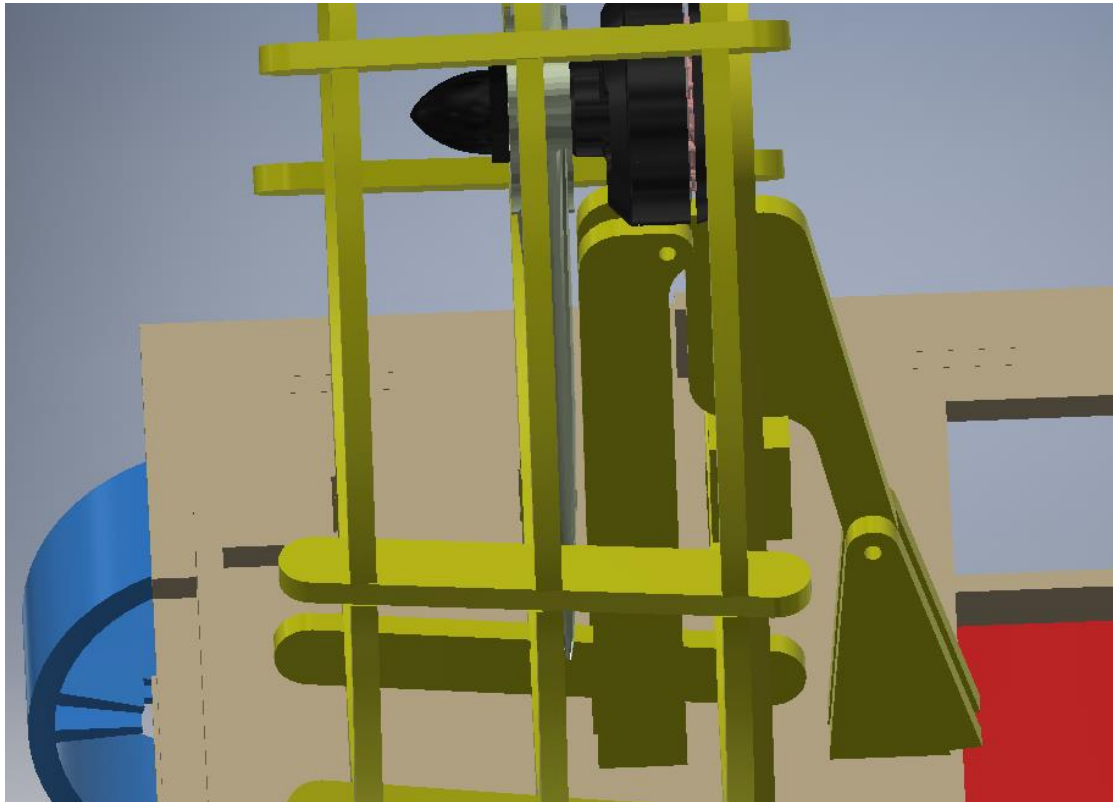
風扇側卡準 2D 工程圖



風扇罩 2D 工程圖



圖八十四：風扇罩與支架 3D 放大圖



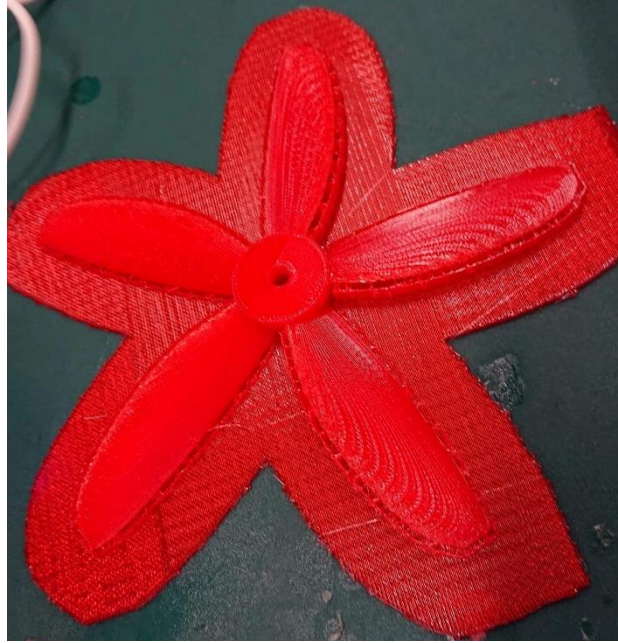
圖八十五：風扇罩與支架 3D 放大圖

5.3 製造

車體除輪子外皆由切割密集板製作而成，而輪子與煞車則部分使用密集板，部分使用 3D 列印。風扇則全由 3D 列印製造。在下一節將會驗證此兩種方式的尺寸誤差。

5.3.1 扇葉

材料是使用實作中心原有的 3D 列印 PLA 材料，而我們這組一開始使用 Kingssel 3D 印表機，但發現它印出來的精度並沒有很好，除了不容易拆支撐材外，扇葉的邊緣也較多不平順的地方，故我們接著試試看 23 萬 (Kraftmaker) 3D 印表機，發現不僅支撐材好拆非常多，扇緣的平滑度也好很多，而且材料的結構也較穩固，只有風扇正面會有少許顆粒需要以銼刀磨掉，故我們最後選擇 23 萬來印製我們期中測試的風扇。但它的缺點是常常停機，且列印時間為 Kingssel 的 2~3 倍。



圖八十六：以 23 萬印製且還未拆掉支撐材的最新風扇

5.3.2 車體

車體的結構主要為密集板的結構，主要因為較容易取得以及進行加工，也較便宜，在實作中心即可進行購買與切割，但缺點是容易受潮、bending 的狀況較明顯，需要較堅固的力學結構支撐。並且由下表可發現密集板與 PLA 的性值還是很接近的，只需承受 1kg 左右的車體，密集板是足夠的。而製作工具主要以實作中心的雷射切割機加以加工，雖然切割機容易受本身參數與功率的影響，使得每一台切出來的結果不一樣，甚至同一機台不同時間切割出的成本精度也有差，尤其在插銷的部分，有時會太緊，有時卻會太鬆，但由於我們能取得的資源又不須額外花錢的也只有實作中心的雷射切割機，因此還是選擇以此做為加工工具。

材料	密集板	PLA
降伏強度	$1 \times 10^7 Pa$	$3.6 \times 10^7 Pa$ [14]
抗拉強度	$1.8 \times 10^7 Pa$	$2.4 \times 10^7 Pa$
抗壓強度	$1 \times 10^7 Pa$	
彈性模數	$4 \times 10^9 Pa$	$3.5 \times 10^9 Pa$
Poisson 比	0.25	
質量密度	750 kg/m^3	1300 kg/m^3
剪力模數	$1.6 \times 10^9 Pa$	$1.6 \times 10^9 Pa$

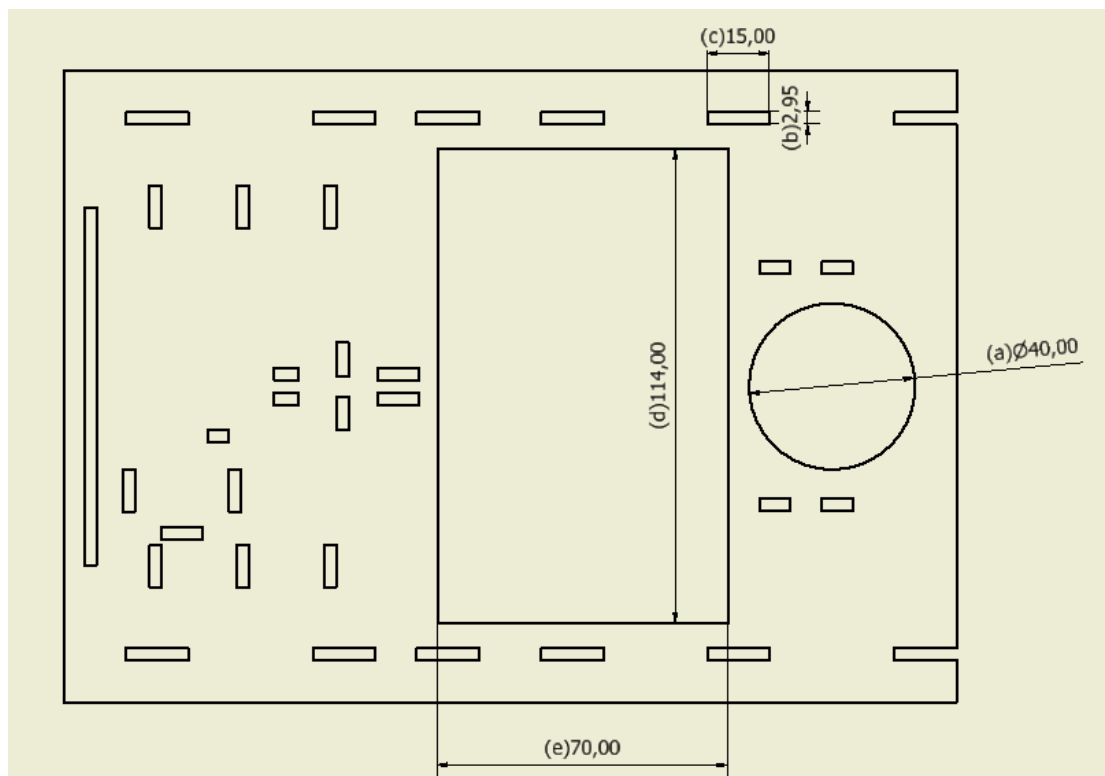
圖八十七：密集板與 PLA 的材料性質(參考林潔茵學姊的報告資料)

5.4 驗證

除了進行設計與製造外，我們也要將製造出的結果進行尺寸驗證，觀察其是否符合理論值，並記錄其誤差，推斷是否為儀器的系統誤差。由以下可看出，雷射切割機在小尺寸的精度上表現得較為不好。

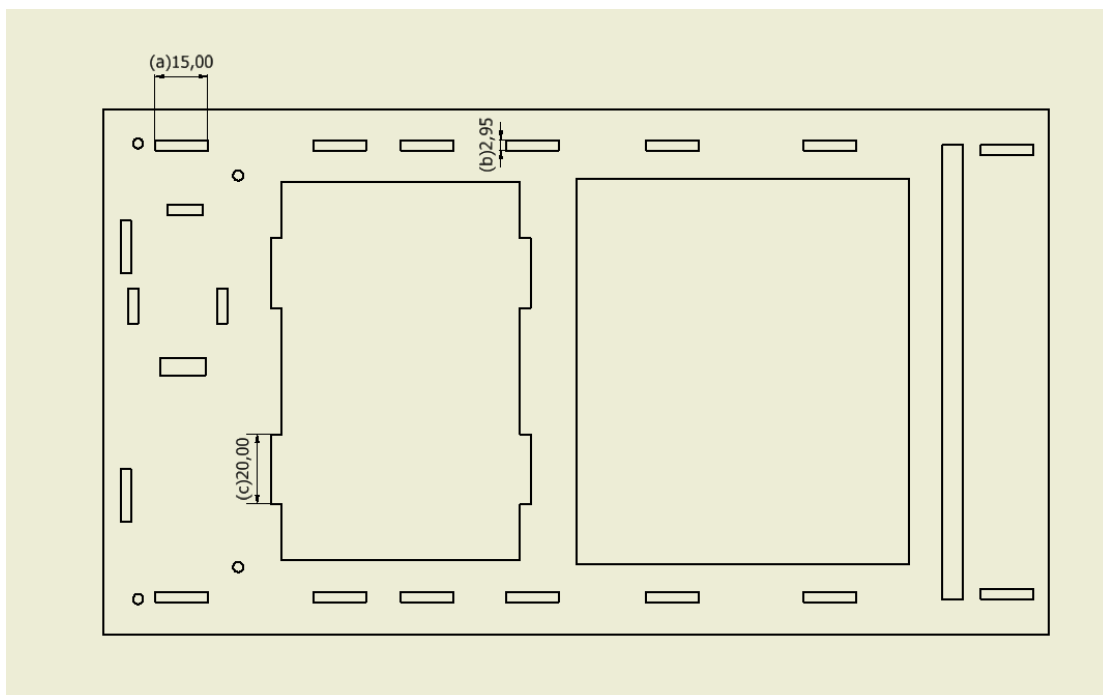
5.4.1 車體

1. 車體上板



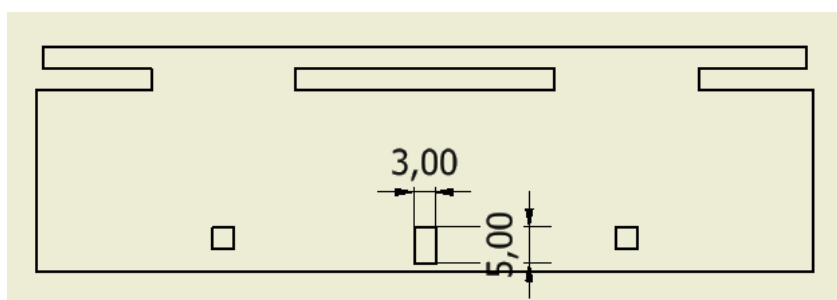
	設計尺寸	實際尺寸	誤差
a	40	40	0.0%
b	2.95	3	1.7%
c	15	15	0.0%
d	114	114	0.0%
e	70	70	0.0%

2. 車體底板



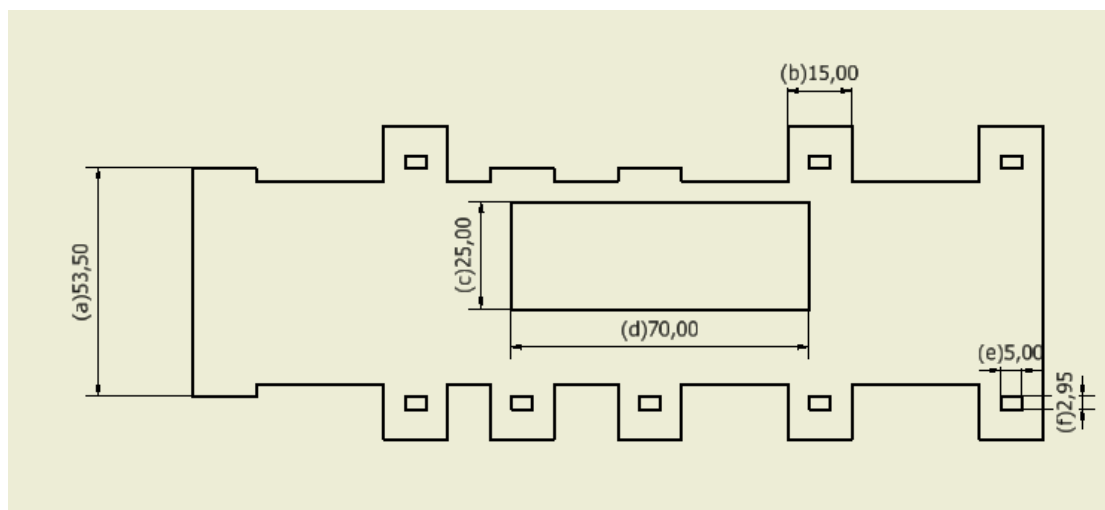
	設計尺寸	實際尺寸	誤差
a	15	14.92	-0.5%
b	2.95	3.1	5.1%
c	20	20.1	0.5%

3. 車體支架



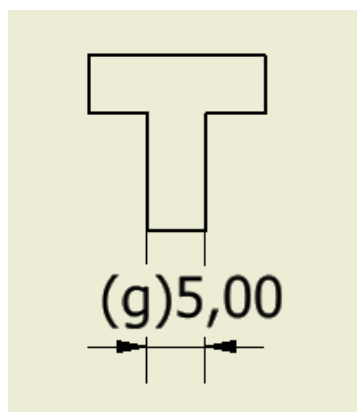
	設計尺寸	實際尺寸	誤差
a	3	3.1	3.3%
b	5	5.1	2.0%

4. 車體側板



	設計尺寸	實際尺寸	誤差
a	53.5	53.4	-0.2%
b	15	15	0.0%
c	25	25.1	0.4%
d	70	70.4	0.6%
e	5	5.1	2.0%
f	2.95	3.1	5.1%

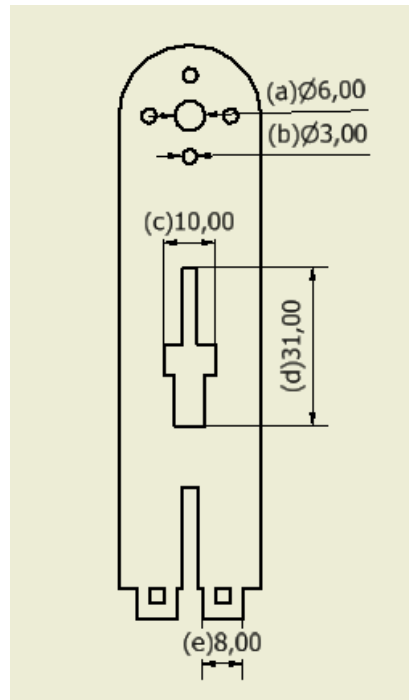
5. 底盤卡榫



	設計尺寸	實際尺寸	誤差
a	5	4.9	-2.0%

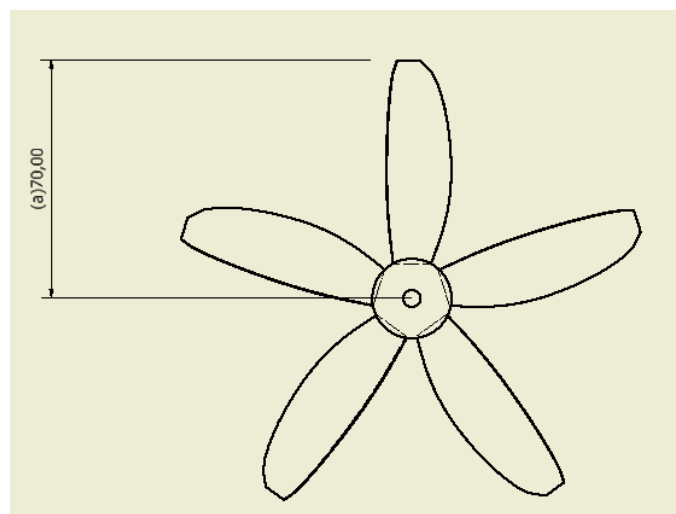
5.4.2 扇葉

1. 風扇主架



	設計尺寸	實際尺寸	誤差
a	6	6	0.0%
b	3	3	0.0%
c	10	10.3	3.0%
d	31	30.9	-0.3%
e	8	7.9	-1.3%

2. 風扇



	設計尺寸	實際尺寸	誤差
a	70	70.6	0.9%

5.5 實驗

由於時間的關係，加上我們這組最後組合的車體時間較晚，在期中前我們只有進行量測車體重量(1.127kgw)和進行風洞實驗，期中過後將會補上其它實驗，以驗證理論、實驗與實測結果間的異同，並建立實驗目標，將各個部分的實驗結果推演至整個系統。

期末分別進行了最大靜摩擦力實驗與側滑摩擦力實驗，目標將實驗結果所量測之數據整理至實際控制風扇功率之調整，與車體速度控制上。

5.5.1 風洞螺旋槳推力實驗

目的：於可視化風洞中量測車體前後流場速度，並根據 Actuator Theory 求得螺旋槳推力。

器材：風洞系統、超音波水霧產生器、雷射裝置、熱線測速儀

假設：假設馬達中心軸以上和以下為對稱分佈，在 inlet 以及 throat 面積內速度為平均分佈。

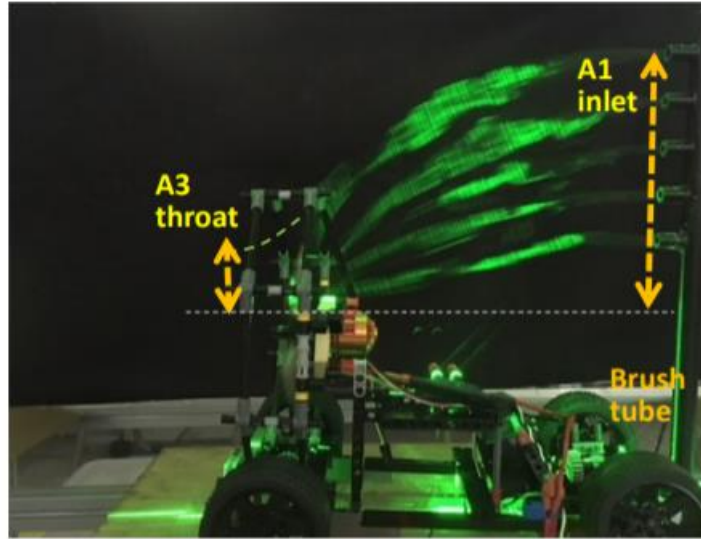
原理：利用流場可視話、動量理論，並根據量測實驗講義 Actuator Theory 可求得以下公式並進一步求得推力：

$$T = \frac{1}{U_2} \frac{\rho}{2} [U_3^3 A_3 - U_1^3 A_1] = \frac{\rho A_2}{2} [U_3^2 - U_1^2]$$

步驟：

1. 開啟風洞風扇，測量背景風速
2. 設定無刷馬達 Duty 值 1，啟動螺旋槳
3. 開啟煙機、雷射，並關燈，量測上游收斂範圍處煙線的流速
4. 量測下游喉部三條煙線的流速
5. 無刷馬達 Duty 值改為 2，重複步驟 C、D...

由於時間的關係，我們只有測試在兩種轉速下的風扇推力。



圖八十八：裝置示意圖

Measurement data(期中)

- 馬達：MT2204 2300kv 風扇：第 5 代風扇
- Blade dimension：0.07m(r)
- Background flow generated by the fan array：Duty：13%，0.27m/s

	ω_1 6400(rpm) V ₁ 11.4V Duty=50%		ω_2 7400(rpm) V ₂ 11V Duty=62.5%	
Height(cm)	U ₁ (m/s)	U ₃ (m/s)	U ₁ (m/s)	U ₃ (m/s)
#1 21.5 15.5	0.58	5.97	0.95	7.97
#2 24.0 19.5	0.38	3.88	0.70	7.18
#3 27.0 20.5	0.42	2.36	0.81	4.48
#4 29	0.32	---	0.55	---
#5 32	0.32	---	0.67	---
Mean(m/s)	0.40	4.07	0.74	6.54
Mean-背景(m/s)	0.13	3.80	0.47	6.27
U ₂	1.965		3.785	

結果討論：

當天室溫為 22°C，查表並內插可得 $\rho = 1.196 \text{ kg/m}^3$ ，體積流率 $Q = A_2 * U_2$ ，由上式與上表整理計算的結果於下表：

平均角速度(rpm)	6400	7400
推力(N)	0.13	0.36

檢討與改進：

由於組員們對於風洞實驗還不是很熟悉，可能在熱線風速計上的量測並不是很精準，並且在測量高度上沒有考慮風扇的中心高度，可能有點過高，此實驗的結果需要其它實驗的完成才能驗證。

Measurement data(期末)

- 馬達：A2212 2200kv 風扇：第 6 代風扇
- Blade dimension：0.07m(r) 中心高度：17.2cm
- Background flow generated by the fan array：Duty：10%，0.18m/s

	ω_1 4286(rpm) V ₁ 12.3V Duty=7.5%		ω_2 5666(rpm) V ₂ 12.53V Duty=11.25%	
Height(cm)	U ₁ (m/s)	U ₃ (m/s)	U ₁ (m/s)	U ₃ (m/s)
#1 13 15	0.81	6.31	0.59	6.12
#2 15 17	0.72	5.24	0.59	5.5
#3 17 19	0.60	5.61	0.60	6.92
#4 19	0.52	---	0.63	---
#5 21	0.51	---	0.54	---
Mean(m/s)	0.63	5.72	0.59	6.18
Mean-背景(m/s)	0.45	5.54	0.41	6.00
U ₂	2.995		3.205	

結果討論：

當天室溫為 25°C 左右，查表可得 $\rho = 1.184 \text{kg/m}^3$ ，體積流率 $Q = A_2 * U_2$ ，由上式與上表整理計算的結果於下表：

平均角速度(rpm)	4286	5666
推力(N)	0.28	0.33

選擇這兩個馬達轉速(Duty 值)的原因為，經過測試，在 Duty=7.5% 的時候，車子的推力接近於動摩擦力，無法啟動車子，但用手輕推一下時，時而會動，時而不動，假設實驗的結果是精準的，那麼車子的動摩擦力應該在 0.28(N) 左右。而會選擇 Duty=11.25% 的原因為，這是我們一開始啟動向前衝的 Duty 值，當時程式的設計為當速度=40cm/s 時，結束單一的 11.25%Duty，降至 Duty=7.5% 並進行控速，而經過影片的估計，當 Duty=11.25% 時，加速到速度=40cm/s 時所用的是間大概是 8 秒左右，也就是加速度=5cm/s² 左右，而如果我們前面假定動摩擦力為 0.28(N) 是正確的，我們將 0.33-0.28(N)，並除以車子的質量 1.2263kg，可得到加速度=4.08cm/s²，以數量級來說是很接近的，因此此動摩擦地的假設應很接近於理想值。

在期中測試後，我們除了將馬達改成扭矩較大的規格外，也將風扇的扭轉角條大，為了增加推力、降低轉速，從期中與期末的結果比較中可以看到，在期中時轉速為 6400rpm，推力為 0.13(N)，而在期末的結果 4286rpm 就已經有超過其 2 倍的推力，當轉速為 5666rpm 時，也與期中轉速為 7400rpm 時有相當的推力，很明顯的當轉速相同時，推力明顯變大。

與 4.2.5 節的理想推力進行比較，不管是期中還是期末的結果，數量級都差了 10 倍左右，這可能與風扇的精度、風扇附近的流場、量測時的不精確…等等的

原因都有關係，因此還是需要進行實驗的驗證，不能以理想值作為標準。

檢討與改進：

應該以流場可示話所評估出的 A_1 與 A_2 另用另一種方法進行二次驗證，但由於肉眼觀察與拍攝的狀況很不清楚，因此只能大概推估流線的分布，可能造成實驗的不夠精確，除此之外，只做兩個轉速下的推力實驗造成數據點不足，不易觀察出推力與轉速的關係。在使用熱線風速計時，前方的孔必須與流場方向垂直，所得到的值才會較精確，但因為是以人手支撐，加上以肉眼判斷，可能造成風速都過小，使用最大值可能比平均值還來的準確。

其它可能造成誤差的原因有：

- 流場誤差：流場中有煙管、風扇架等物品使流場受干擾。
- 風洞誤差：從量側實驗中可發現風洞流場並非均勻，而本次實驗假設 ambient flow 為 laminar，減去的背景風速可能也因實驗沒量測多次而造成誤差。

5.5.2 風洞風阻實驗

目的：將車體置於風洞中測量流場速度，求得行進時的風阻。

器材：風洞系統、熱線測速儀。

假設：背景流速與本身之行車速度相同

原理：將車體視為一控制容積(Control Volume)，則求得出入(上下游)的流速分布後，可根據 Momentum Equation 得出車體所受的風阻 F_D ，如下式：

$$F_D = \int \rho U U U(y) dy - \int \rho U D U D(y) dy \approx \rho \Delta y \sum [U U^2(y) - U D^2(y)]$$

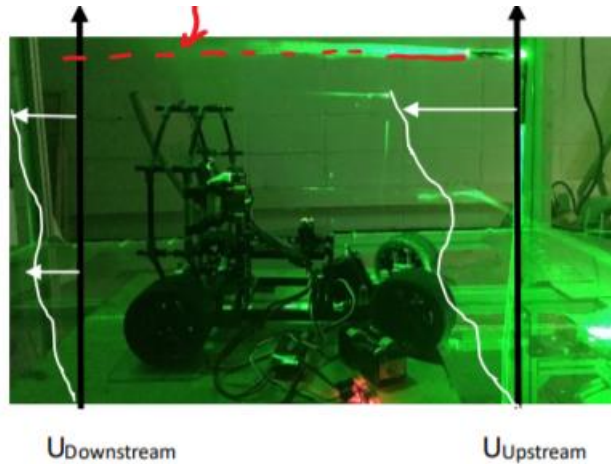
並進一步得到：

$$C_D = \text{Total drag} / (1/2 \rho U^2 A_B)$$

$$Re = \frac{\rho \times D \times U}{\mu}$$

步驟：

1. 量測車體尺寸、氣溫
2. 將車體放入風洞中央處，並調整風洞風扇 duty 為 20
3. 待流場穩定後，測量流場上游(車頭)風洞一半寬處不同高度的流速($U_U(y)$)
4. 測量流場下游(車尾)風洞一半寬處不同高度的流速($U_D(y)$)
5. 將風洞風扇 duty 調整為 40 與 60，重複步驟 C、D



圖八十九：裝置示意圖

Measurement data(期末)

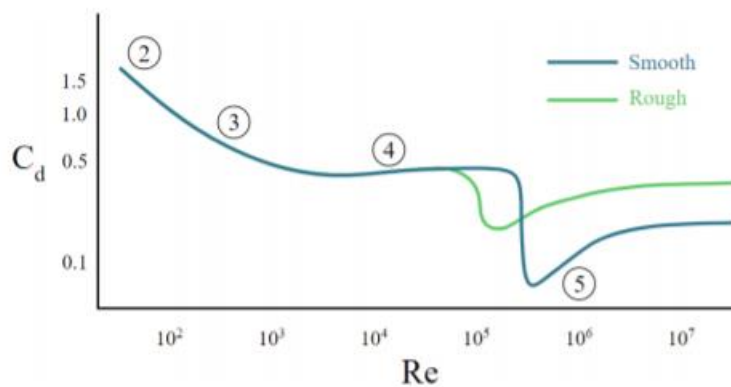
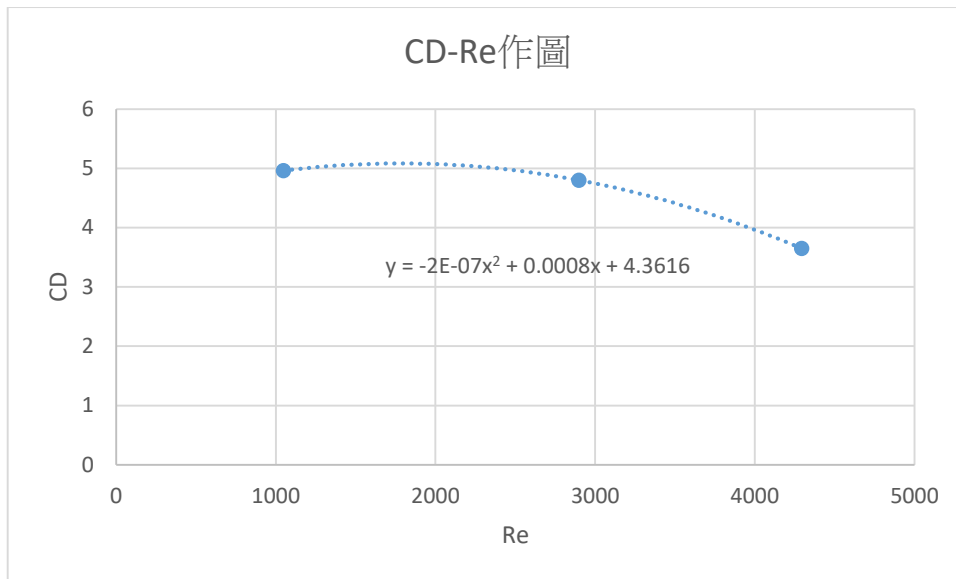
室溫 \approx 25°C	Duty Z=20%		Z=40%		Z=60%	
Y_i	U_U (m/s)	U_D (m/s)	U_U (m/s)	U_U (m/s)	U_U (m/s)	U_D (m/s)
36cm	0.38	0.22	0.99	0.93	1.33	1.46
32cm	0.58	0.38	1.15	0.91	1.55	1.22
28cm	0.35	0.3	1.08	0.88	1.66	0.76
24cm	0.4	0.11	1.42	0.32	1.93	1.34
20cm	0.52	0.21	1.22	0.51	1.96	1.43
16cm	0.5	0.05	1.21	0.08	1.64	0.33
12cm	0.44	0.07	1.43	0.05	1.91	0.07
8cm	0.49	0.03	1.39	0.02	1.77	0.01
4cm	0.34	0.04	1.16	0.15	1.56	0.16

查表可得 $\rho = 1.184 \text{ kg/m}^3$, $\mu = 1.849 \times 10^{-5} \text{ kg/m} \cdot \text{s}$, W 為控制體積(車體)寬度: 0.185m, 車子尺寸 28cm(L) \times 18.5cm(W) \times 26.5cm(h), 可計算出可計算出 $D^* =$

$\frac{W \times h}{2(W+h)} = 0.0545$, 並利用上式將結果整理於下表:

Duty(%)	阻力(N)	阻力係數	上游平均流速 (m/s)	體積流率 (m^3/s)	Re
20	0.013	4.96	0.44	0.0216	1047
40	0.096	4.80	1.23	0.0603	2897
60	0.16	3.65	1.70	0.0833	4293

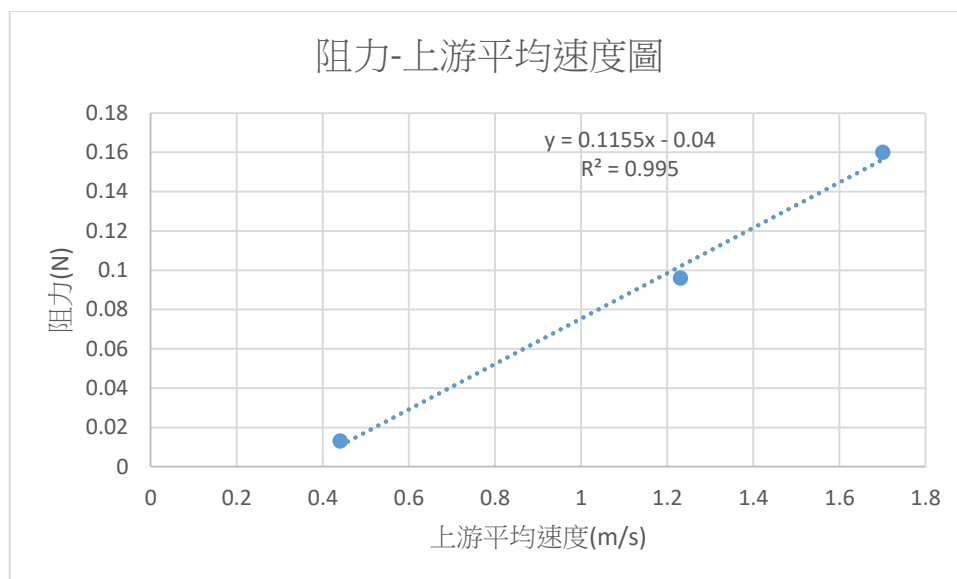
由上表可發現, 當 Duty 越大(車速越快), 阻力越大。且阻力係數越低, 比較下圖與圖(九十)可觀察到, 在雷諾數所對應到的範圍($10^3 \sim 5 \times 10^3$), C_d 值應差不多, 並且實際阻力係數的值也過大, 整體曲線有往右移動的趨勢, 但由於只有 3 個點的緣故, 還無法準確評估趨勢。



圖九十： C_d 對 Re 作圖(量測講義)

由於 $Duty=20\%$ 相當於行進速度為 45cm/s 左右，而我們期末測試的一、三圈車速也剛好是 45cm/s 左右，因此考慮此值的阻力，如果此值是接近正確值的，那麼在風洞推力實驗所得到的結果應為動摩擦力+車體阻力，可以推斷在行進速度為 45cm/s 左右，動摩擦力為 0.267(N) ，而空氣阻力大約為 0.013(N) 。至於在一開始加速的部分並不好評估，因為阻力的值是在變動的，但可以肯定的是一定比 0.013(N) 還要小，因此大約可以忽略不計。

照理來說，阻力值應與上游平均流速成正比，由下圖可以發現，其結果與理論值相符，非常接近線性。可推估實驗的結果應該還算是準確的。



檢討與改進：

在測量下游風速時，有些地方會被車體擋住而造成測到的流速非常低，但我們並沒有將這些數值排除，可能導致最後的阻力結果過大。

其它可能造成誤差的原因有：

- 我們選擇的是以離散點進行計算，並非利用積分，因此結果可能與實際會有誤差。
- 因為所取的控制容積邊界並不完全是 $V(x)=0$ ，因此還是有流體由上方流出控制容積。
- Boundary Layer 影響：Base hole 較 top hole 更為接近風道邊界，所受邊界之黏性與阻力影響較大，隨測量位置遠離邊界而流速變大，需考量 base hole 所量測風速受邊界層影響較大。
- 流場並非均勻。

5.5.3 風扇風速與轉速測試

目的：了解馬達訊號與風扇轉速間的關係。

原理：在風扇上貼反光貼紙，開啟馬達後，以紅外線轉速計量測紅外線被反射的次數即為轉數，轉數與時間相比即可知道轉速，再對比程式馬達輸入值，即可知道馬達控制與轉速的關係。

結果討論：

在理想狀態、馬達沒有承受任何負載的情況下，我們將馬達輸出的電壓上限設為 180，下限設為 20，並實際量測在風扇運轉下，其得到的電壓，因為我們這組期中前使用的是 2300kv 無刷馬達，而期中後為 2200kv，也就是說輸出 1 電壓，可讓馬達增加 2300 或 2200rpm，故理想的情況下，風扇的轉速應為電壓

$(V) \times \text{Duty} \left(\frac{x-20}{160} \right) \% \times 2300 \text{ or } 2200 (\text{rpm}/V)$ ，我們將理論與實際測試到的值整理於下

表：

(期中)

電壓(V)	11.4	11
Duty(%)	50	62.5
理想轉速	13110	15813
實際轉速	6400	7400

由上表可以發現，實際轉速與理論轉速的誤差超過 2 倍以上，可能是我們量測到的電壓有誤(馬達實際得到的電壓更小)、程式設定錯誤、電池的輸出並不理想，或者馬達承受過大負載造成的，而實際的原因還需後續進行檢測，但我們認為是馬達承受過大負載的原因比較有可能，因此我們在期末的變革中，更換了更大、扭矩較大的馬達。

(期末)

電壓(V)	11.96	12.07
Duty(%)	7.5	11.25
理想轉速	1973	2987
實際轉速	4286	5666

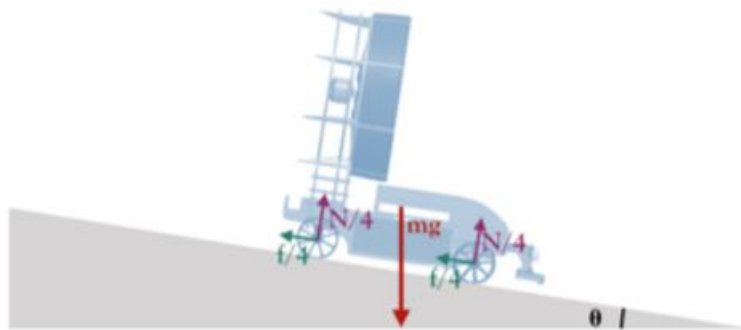
在期末的測試中我們發現，反而與期中的結果相反，實際轉速比理想轉速還來得高，而我們的推測是，馬達在低轉速下會有一個下限值，當轉速越快後才會越接近理想值，比期中的狀況較合理，但我們並沒有實際做實驗驗證，只是一個推測，但比起期中的馬達，扭矩的方面似乎改善不少，風扇的負載並沒有超過馬達所負荷的量。

由於數據點只有兩個，無法推估馬達轉速與電壓、Duty 的關係，但可以確定的是，Duty 越高轉速越快，可見程式與馬達應該是正常運作的。此外，馬達的持續運轉會造成電壓下降，應取測試前後的平均值進行計算，或者實測時間要短點。

5.5.4 車體最大靜摩擦力實驗

在實驗中，我們以車體的輪胎和永齡比賽場地之地板做為模擬，將車體置於一斜坡上，計算其傾斜角度 θ ，可以得到靜摩擦係數 $U_s = \tan(\theta)$ 。

測量到之實驗值與數據



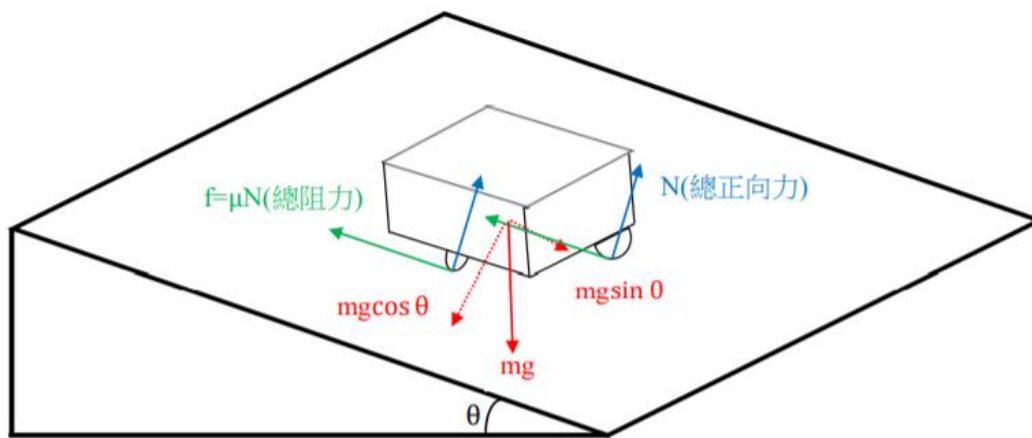
圖九十一：車體力平衡示意圖

傾斜角度 theta(deg)	靜摩擦係數 $U_s = \tan(\theta)$
1.5547	0.0272
1.6365	0.0285
1.8410	0.0321
2.0454	0.0357
2.2497	0.0392
theta 平均值	平均值 U_s
1.8655	0.0326

從上表可以大概讀出摩擦係數約為 0.0326，而車體之重量為 1226.3 克，可以大概算出車體之最大靜摩擦力約為 0.392N，透過判讀該項數值，我們可以對風扇功率進行推力量測，從而找到等速時之所需功率，方便我們調整系統之速度上下限。

5.5.5 車體側滑之最大靜摩擦力實驗

與前項實驗雷同，我們以車體的輪胎和永齡比賽場地之地板做為模擬，將車體橫向置於一斜坡上，並事前利用電腦校正前輪之舵機，使其事先維持筆直向前之角度，接著開始量測斜坡所需抬升之高度，計算其傾斜角度 theta，可以得到靜摩擦係數 $U_s = \tan(\theta)$ 。



圖九十二：氣動車置於斜面上的自由體圖

傾斜角度 theta(deg)	靜摩擦係數 $U_s = \tan(\theta)$
21.90	0.4035
21.41	0.3921
23.03	0.4250
21.44	0.3928
21.38	0.3914
theta 平均值	平均值 U_s
21.83	0.4009

從上述數據我們可以計算出車體側向打滑所需之最大靜摩擦力，藉由該項方式，測向最大靜摩擦力大約為 4.82N，可以大概計算前輪轉向之速度，並免轉向過快，導致車體打滑，從而影響循跡之情形發生。

5.5.6 紅外線感測器實驗

目的：尋找紅外線感測器閾值

器材：紅外線感應器、地板黑線

場地：實作中心

方式：使紅外線感應器在跑道黑線邊緣來回移動，觀察有無感測到黑線，並觀察其類比回傳值。

結果：

感測器編號(由左至右)	黑色膠帶	一般地面
1	767	318
2	691	191
3	677	174
4	625	113
5	655	165
6	674	103

根據實驗結果可以看到黑色膠帶以及一般地面的類比輸入值，因此在選擇閾值時需選擇介於兩者之間，但根據我們實際測試的結果，將閾值設定的離黑色膠帶的輸入值近一點比較保險，避免產生雜訊。

5.6 討論

在期中前由於我們只有做推力與轉速實驗，故在此僅限於討論兩者，由期中測試的最大加速度要求 3.75cm/s^2 、我們的車體質量為 1.127kg，將兩者相乘可得我們需要的淨合力為 0.042 牛頓，而我們上面實驗所測得期中的最大推力為 0.36 牛頓，為所要求的淨合力 10 倍左右，如考慮摩擦力且忽略空氣阻力，再加上參考過去組別的數據，這個推力應該是足夠的，但也還是要進一步進行摩擦力實驗後才可得證。

而在期末後經由理論計算，新風扇啟動的推力為 6N 左右，而風洞實驗我們所測得 Duty=32 時的推力為 0.28N，Duty=38 時為 0.33N，經過實際測試，當 Duty=35 左右時能使車體啟動，因此實際測得的啟動推力應介於 0.28~0.33N 間，與理論的 6N 相差甚遠，在這邊我們認為理論值可能過大，因為所有條件都假定為最理想的狀態，加上弦長是用平均的方式計算，造成不準，但實際測得的值應該也過小，因為使用熱線風速計不可能一直維持與流場夾 90 度，加上一些風場的不完美，實際值應該要再更大一些，但總體來說還是會更接近風洞實驗所測得的值。

經過測試，當 Duty=32 時車體接近等速，因此我們假設車子的動摩擦力為

0.28N 左右，再經由上方量測最大靜摩擦力時所得的結果為 0.392N，動摩擦力小於最大靜摩擦力，符合理論的結果，但是最大靜摩擦力應與啟動推力相等，由上段可得知實際測得的啟動推力應介於 0.28~0.33N 間，由於最大靜摩擦力實驗為多次實驗的結果，我們認為其值應較為準確，且也符合風洞推力實驗理論上會測到較小值的結果，假設 Duty 與推力間(在 duty=32~38)為線性的，可以推算啟動 duty35 實際測得的推力應為 0.305N 左右，因此我們推斷風洞實驗整體大概低估了 $0.392-0.305=0.087(N)$ 左右，因此實際狀況 duty=32 時推力應為 $0.28+0.087=0.367(N)$ ，duty=38 時則為 $0.33+0.087=0.417(N)$ ，而動摩擦力也應變為 0.367(N) 左右，還是符合小於最大靜摩擦力的假設。由於風阻藉由風洞實驗應小於 0.013N，因此在此先將其忽略。

經由側向的最大靜摩擦力(打滑)實驗可得知，要大於 4.82(N) 的側向力才有打滑的可能，而我們風扇的 duty 上限為 42，最大也只有 0.5(N) 左右，因此不需要擔心打滑的問題。

最後統整我們認為本次期末測試氣動車的性質在下表：

最大靜摩擦力 (啟動力)	最大側向靜摩 擦力(滑動力)	動摩擦力	啟動推力	風阻 (V=45cm/s)
0.392(N)	4.82(N)	0.367(N)	0.417(N)	0.013(N)

由啟動推力與動摩擦力可得知，我們的啟動加速度為 $4\text{cm}/\text{m}^2$ 左右，以實際用相機錄製計算的 $5\text{cm}/\text{m}^2$ 接近。

6 工作進度與分工

6.1 基本規範

Meeting norms :

主要開會時間為每星期一於機械系系學會或永齡實做中心，時間當天決定，時長約一到二小時，盡量不請假，若有需要一起測試，會另行通知於群組。

Work norms :

陳昱睿與陳羿廷為車體組，陳佳好與簡揚開(期中後幫忙控制)為風扇組，鄭丞佑為控制組。我們每星期會在開會時報告進度、設立進度目標，並一起討論與檢討。若有人未達設定的工作目標，或是未請假缺席者，將會被其他四人高度關心。若對於成品有不同的意見時，我們將採取多數決。

Decision making :

我們會在開會上討論出大方向，而如何實際設計則由小組決定，再互相配合。若有人堅持己見，我們將投票以多數決決定。

在期末與第 10 組的合作中，都是以多數決的方式進行，並創立社群媒體群組，有任何問題或要約測試，皆在群組中進行，期中後的兩次開會後，即由各組自行發揮，並沒有特定的開會時間。

6.2 工作分工

車體組的詳細分工

由陳昱睿負責：設計、繪製並製作後輪；設計與製作煞車裝置；設計第一層底板與車手(飲料)座；設計後輪轉向機構(期末)；統整所有底盤並繪製整體工程圖；設計黑色擋板。

由陳羿廷負責：設計第二層底板與規劃車體空間；設計與製作前輪轉向機構；設計紅外線循跡感測儀支架；設計車體支撐結構；主要進行密集板雷切工作。

風扇組的詳細分工

由簡揚開負責：風扇的設計與繪製；風扇實驗；報告統整；協助控制組進行實測。

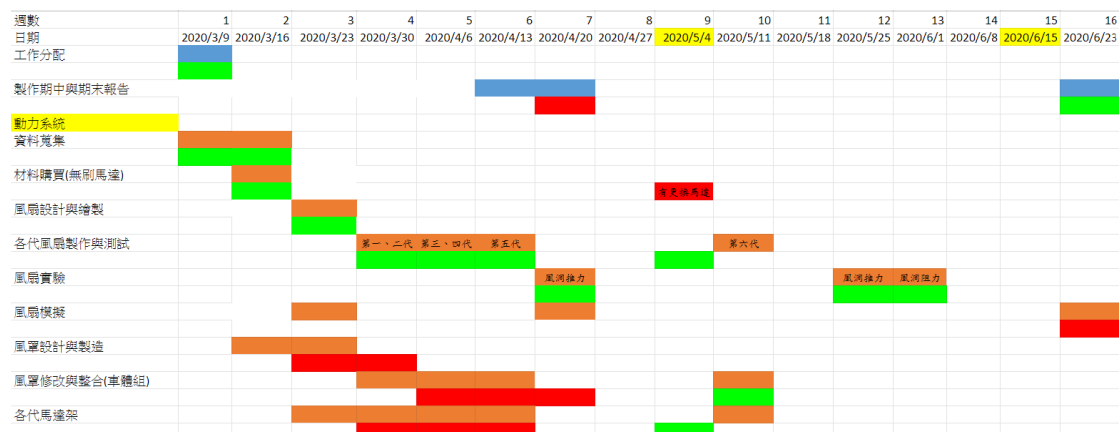
由陳佳好負責：設計馬達避震支架；風扇分析；設計與製作風罩；主要負責3D列印工作。

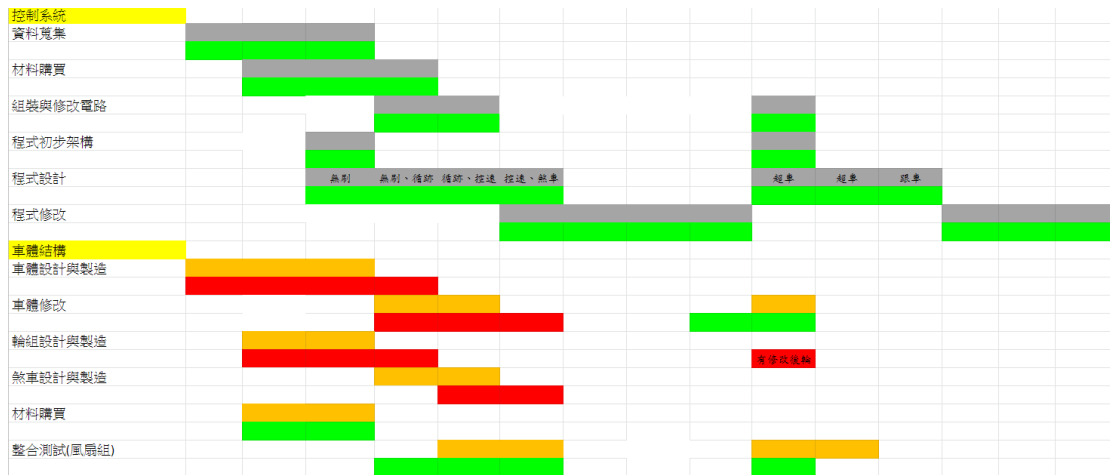
控制組的詳細分工

由鄭丞佑負責：所有機電硬體元件的組合、接線與控制實測。

6.3 甘特圖與完成圖

各顏色為期初規劃時的甘特圖，而第二列為實際完成時間的完成圖(以紅色與綠色標示，紅色：開始時間比規劃時間較晚或完成時間比規劃時間晚；綠色：提早開始或提早完成)





由上圖甘特與完成圖可以發現，本組在這學期的完成進度還在預期內，並沒有像上學期嚴重落後的情況發生。

6.4 期中後未來工作與期末檢驗

6.4.1 車體組

車體部分的空間雖大，但目前東西放置的規劃還不甚理想，杜邦線的走向也算凌亂，這部分的整理與計畫已成必要。(▲此項期末完成一半，雖然有改善了空間配置，但線材的凌亂並未解決)

後輪使用 3D 列印的輪胎與固定輪胎的支架之間以培林、螺絲和螺帽連接，但由於 3D 列印機臺的加工精度限制，後輪連接配合的不夠理想，使得輪胎有些許外傾的跡象，改善的方法是打算直接使用密集板組件的配合，由於雷射切割的精度頗高，經過適當配合件的設計，或許能優化後輪的狀態。(▲此項期末完成一半，雖然改成以密集板組件的方式設計後輪，但並沒有完全改善傾斜的問題) 至於煞車方面，雖然目前其作用能有效制動車子的運動，但並不能適當的調控力道，增加控制的困難。目前改善的可能方向是引入彈簧元件的使用，透過其不同伸長量產生不同大小的特質，期許能設計出容易控制力道的煞車裝置。(★此項期末不適用，期末並未使用煞車)

6.4.2 風扇組

風扇的部分已大致完成，有經過現場的實測發先推力對於期中測試的要求是足夠的，且馬達轉速也沒有超過 10000rpm，但缺點就是震動還稍微有點大且推力也不是非常大，在期中後將試著改變風扇的其他參數，以改善期末可能震動過大導致轉彎的循跡不穩定和推力不足的問題。(●此項期末有改善，改變風扇的扭轉角後，推力有顯著的提升，風扇 Duty 只需要 10%左右即可，也明顯改善震動與噪音的問題)

風罩的部分也大致上已經完成，之後會將風罩外圍纏起來，加強安全性跟導流效果。另外，待所有零件的位置擺放確定之後，可能會再視風扇被其他零件遮

擋的程度決定是否要增高馬達與風扇的位置。馬達架的設計則還需要進一步修改，目標是提高避震的效果，讓車體不會在風扇運轉時劇烈震盪、影響行進的速度和穩定性。目前規劃大致上是先找出第二版的四連桿為何沒有發揮避震效果，並盡快設計出下一個經過修正的版本。（●此項期末有改善，馬達支架的微調，使得震動有改善，而由於車體的設計改變，因此不需要變更風扇高度，風罩也以膠帶纏繞，不確定是否增加了導流效果，但安全與穩固性確實提高了）

6.4.3 控制組

由於期中與期末的目標完全不同，所以控制組的工作量會很大。這學期的目標主要是循跡以及控速，而期末的重點是與別台車進行跟車以及超車的行為。過了期中後，我們就要開始規劃對於期末的跟車以及超車的策略了。期末的賽道有轉彎的部分，也會是控制的難題。若這次的循跡能夠順利進行，期末我們就會有更多時間，也可能考慮使用影像辨識來進行對車子定位的作用。（●此項期末有改善，期末測試非常成功）

7 心得與專業建議

✓ 陳佳好

心得：從大一開始就聽學長姐不斷提起的一門課在不知不覺間竟然也默默走到尾聲了，在情況這麼特殊的一個學期修習這門課的感受很複雜；一邊實作一邊擔心疫情讓實作中心無法開放、大家直接和學分說再見的情況大概也不太常見了。這門課算是機械設計的進階版；更明確地感覺到理論和實作的差異，既知道模擬的重要、一邊卻又清楚模擬出來的結果和實際一定大相逕庭。這門課就是一個屢戰屢敗、屢敗屢戰的過程，彷彿看見自己未來身為工程師的生活……很高興可以跟我的組員們同一組，在合作的過程中看到大家都很強，也更意識到自己的不足之處。

建議：這學期期中前有一段時間大雷切機 universal 故障，因此大家需要排 beambox，雖然不知道為甚麼，但 beambox 的軟體與機台似乎都不是非常穩定，若有機會希望能找到保養或是優化它們的方式。

✓ 簡揚開

心得：過去只有聽說過這堂課會很累、測試前會熬夜，但在實際經過這學期的摧殘後才真正體驗到什麼叫累，尤其在期中測試結束的時候，真的是身心疲乏。雖然我是風扇組，主要在實作上的工作是設計與列印風扇，但當風扇確定後，工作量就變得比較少，因此除了承擔統整期中與期末的書面報告之外，也在期中測試前開始到控制組去幫忙邏輯思考與程式的修改。回想起期中測試前幾乎每天都與控制組的同學泡在永齡的實作中心內，不斷的修改與優化，還記得當時我們在期中測試的那一週，還設置了每天一定要達到的目標，從 90 到 95，每一天都要有所長進與突破，達到後才能去吃飯或回家，但沒想到在期中測試的時候，因為線路脫落加上一時緊張找不到問題所在，錯過了許多時間，

導致最後沒有時間跑完，分數從一直很穩定的 90 幾分掉到 70 幾，頓時覺得好像花那麼多時間努力都白費了，甚至一度想要放棄這種求好的精神。但很幸運的是我有幾位組員能夠一起加油打氣，願意付出很多的時間在氣動車上，加上期末是與其它組別的合作進行，因此我們並沒有放棄，反而更引此為戒。而期末測試又是另一個噩夢的開始，我們這組的進度很快，因為一開始我們是前車的關係，跟車的部分占比較少，加上我們程式的架構使得轉向、超車非常穩定，但另外一組一開始出現許多不知名原因的問題，例如循跡時好時壞、超車有時成功有時不成功、超音波的 PID 控制也不易調控，因此在測試前一週末，才臨時改成我們這組變成一開始是後車。好在合作的組別它們一一解決了前述所面臨到的問題，度過了驚心動魄的一週，在還沒測試前誰也不知道會發生什麼事，雖然在期末測試前熬了幾天夜，但最後期末測試還是很順利的在第一趟就拿了滿分。過完了一學期，只能說很幸運這學期沒有修很多堂課，因為花在這堂課的時間真的太多太多，除了要花時間在這堂課外，其它科也有許多的報告與考試，體驗到了難以忘懷的大三下生活，但是回頭想想，還是覺得這堂課過得很充實，除了在過程中了解風扇的特性與理論外，更重要的是讓我學習到如何解決問題與團隊合作的能力，在過程中我們常常遇到很多不知道怎麼解決的問題，我們到處詢問、不斷嘗試，最後還是都克服了這些困難，這是在其它課中沒有體驗到的感覺，讓我覺得在這堂課中是有所成長的。雖然不是每個組員都付出的一樣多，但在過程中一起設計、腦洞大開也讓我學習到很多，尤其在期中後與第 10 組的合作，很謝謝他們也很努力的在這堂課上，一起為滿分的目標前進。也謝謝一起熬夜奮鬥的組員，讓我了解到什麼是革命情感 xD。最後也謝謝這堂課的所有老師與助教，尤其在這學期疫情的干擾下格外辛苦，不管結果如何，大家都努力嘗試到最好了，這樣就足夠了。

建議：我覺得這堂課的負荷量真的有點偏重，如果是全組一起分擔，那倒還好，但專題的內容到了後期就幾乎都壓在控制組的身上，但又不可能所有人都變成控制組，難免會造成工作量不均的問題，主要負責與幫忙控制的同學，花的時間可能會是其它組員的好幾倍，因此在內容設計上，希望可以做些調整，讓控制的占比比較沒那麼重。另外一個建議是盡量能夠避免跨組合作的情形，我們這組很幸運，遇到了很好的合作組別，但很多組就沒那麼幸運了，做兩台車不是為了測試，而是因為另外一組真的不行，雖然最後有互評的機制，但多少還是可能會造成某些組別的心理不平衡。除此之外，氣動車的不確定因素太高，又加上總 BOM 成本線縮在 3000 以內，我們這組用的都不是特別高級的材料，實際的成本加一加就接近已 3000，更不要說那些壞掉或已被汰換掉的東西，在這樣的成本考量下，這限制了硬體設備的精密程度，造成結果很大的不確定性，希望能藉由修改專題內容，盡量去減少不確定性的問題，或者將成本上限提高與增加補助，以得到更好的專題品質。最後一點是期中+期末的兩個實測，雖然有些東西與期中相似，例如控速與循跡，但期末是一個完全不一樣的場地，對於控制來說，幾乎是藥棉林一個全新的挑戰，我認為負擔有點太重，

希望未來能夠將期中與期末的平均難度都調降，或者將兩次測試合為一個測試，期中只做基本的功能加以審查，比較符合 2 學分的這堂課。但不管如何，我認為開設這堂課的出發點是很好的，藉由遇到困難培養我們解決問題與團隊合作的能力，希望這個出發點能持續延續到未來每一年的專題中，每一年都能夠越來越好、越來越進步，學生也能學到更多。再次感謝這堂課的所有老師、助教和同學，收穫很多的一學期！

✓ 鄭丞佑

心得與建議：很感謝這學期一起辛苦的夥伴們，也感謝與我們合作的第十組。身為期初控制組的唯一負責人，我把我幾乎休閒的時光放在與這台車子上了。我在這門課學習了很多，不論是實作的能力，或是團隊合作的精神。我還記得學姐在學期初跟我說，要將一切都確定很穩定，才能夠安心。那時候的我還不重視穩定一直到期中在上場時突然有問題的時候。那時候的時間好像過得特別的快，結果花了幾分鐘 debug 的結果是 encoder 的線掉了，也導致我們原本穩定 90 分的成績變成 70 分。在那之後我就學會將一切確定的線路都給它焊起來，也深深體會了學姐那句話的意義。我認為這門兩學分的課實在是不好拿分，也有許多可以進步的空間，但我很感謝教授以及助教對於這堂課的用心。

✓ 陳昱睿

心得與建議：經過這門課的收穫確實豐碩，但它本身作為一門課所給予我的，卻少之又少。建議的部分，學長們的意見大都可直接沿用，也無須在此老調重彈。謝謝同組的同學們，大家真的辛苦了，我做的不多，但學到很多。

✓ 陳羿廷

心得：機械工程實務這堂課，算是大學以來少數有實體操作的必修課，我想這門課除了真的很花時間這項缺點之外，算是一個很好把之前必修所學的知識得以套用並發揮的課程，真的學到了很多，從設計一直到構圖，從機械元件到程式語言，包山包海，同時也能與系上同學分工合作，建立合作關係與分配，並學會制定規則，以及如何在團隊中對話與溝通，真的是非常扎實且有趣的一堂課，雖然礙於疫情影響，做車的體驗不佳之外，還是學到了不少，本人也很喜歡手做的感覺，除了很有成就之外，很能與系上同學交流，大家一起學習分享自己的設計，真的很棒。

建議：這堂課名義上只有 2 學分，但是花的時間太多，導致同學之間有些人會選擇混分而過或是不滿投資報酬率太低之類的，雖然提高學分數並沒有太大意義，但我認為或許可以新增其他課程與機械課程實務同時在大三下進行與互相配合，在同學自主摸索之餘，又有其他相關的知識能夠補充，我覺得在學習設計車子時，如果有教授那邊的意見或是看法能夠交流，對學生的幫助也比較大，同時也能提升學習意願與課程氛圍，我希望這堂課能成為一堂帶得走的課，不要是一堂大家提起時，都一笑置之的課，而是一堂可以讓我們有自信大聲說自己曾經上過這堂課，總之還是謝謝系上教授開授的這堂課，真的學到不少。

8 參考資料

- [1] 圖片來源：<https://www.orientalmotor.com/cooling-fans/technology/cooling-fan-structure-overview.html>
- [2] 圖片來源：<https://kknews.cc/news/qjmnklr.html>
- [3] 圖片來源：<http://tw.gigacircle.com/1816486-1>
- [4] 圖片來源：<https://www.chc-fan-blower.com/lab/direct.html>
- [5] 圖片來源：https://www.everflowtech.com/products_detail/15.htm
- [6] ENGR 5961 Fluid Mechanics I:Dr. Y. S. Muzychka Handout
- [7] 圖片來源：<https://nptel.ac.in/courses/101/101/101101001/>
- [8] Catching the Wind, Andy Hall, Dec 10, 2007@Stanford University
- [9] Aerodynamics of Wind Turbines, Emrah Kulunk@Institute of Mining & Technology
- [10] 圖片來源：
http://blog.udn.com/alpineatks/112070774?fbclid=IwAR024eSrDhhGR3Tg_dl3gOppaUriroHtdezX9EFA1UA2evXx5VySzg3dHq8
- [11] 圖片來源：<https://goods.ruten.com.tw/item/show?21609969456041>
- [12] 圖片來源：<https://goods.ruten.com.tw/item/show?21830395452541>
- [13] ABRA, A2212/6T 2200KV Brushless DC Motor For RC Quadcopters Planes Boats Vehicles and DIY Kits, <https://abra-electronics.com/quadcopters/motors/mot-qc-02.html>
- [14] Robert W.Fox, Philip J.Pritchard, Alan T.McDonald, INTRODUCTION TO FLUID MECHANICS 7th Edition, p.9-41.
- [15] 圖片來源：<https://goods.ruten.com.tw/item/show?21108045546196>

9. 附錄(程式碼)

9.1 期中測試

1. 連接 IR 模組及轉向伺服馬達的 arduino 板

```
#include<Servo.h>
```

```
Servo servoturn;
```

```
int turnpin = 10;
```

```
int IR[] = {A0, A1, A2, A3, A4, A5};
```

```
int IRread[6];
```

```
int a = 1.2;
```

```

int gate = 0;
boolean ra, rb, rc, rd, re, rf;
boolean IRfinal[6];

float intg = 0, der = 0;
float output = 0;      //turn angle
float error = 0, previouserror = 0, ppreviouserror = 0;
float ang;

//PID
float Kp = 1;
float Ki = 0;
float Kd = 0;
float pterm , iterm, dterm;

//param of angle correction
float e1 = 10;
float e2 = 15; //5
float e3 = 18;
float e4 = 22; //11
float e5 = 26;
float e6 = 30; //20

void setup() {
  // put your setup code here, to run once:
  Serial.begin(4800);
  for (int i = 0 ; i < 6; i++) {
    pinMode(IR[i], INPUT);
    servoturn.attach(turnpin);
  }
}

void loop() {
  for (int i = 0; i < 6 ; i++) {
    IRread[i] = analogRead(IR[i]);
    gate = gate + IRread[i];
    Serial.println(IRread[i]);
  }
}

```

```

}
gate = gate / 6 + 50;
for (int i = 0; i < 6 ; i++) {
    if (IRread[i] < gate) {
        IRfinal[i] = 1; //not black
    } else if (IRread[i] > gate) {
        IRfinal[i] = 0; //black
    }
}

}

if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] ==
1) && (IRfinal[4] == 1) && (IRfinal[5] == 0)) {
    error = e5 * a;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) &&
(IRfinal[3] == 1) && (IRfinal[4] == 0) && (IRfinal[5] == 0)) {
    error = e4 * a;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) &&
(IRfinal[3] == 1) && (IRfinal[4] == 0) && (IRfinal[5] == 1)) {
    error = e3 * a ;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) &&
(IRfinal[3] == 0) && (IRfinal[4] == 0) && (IRfinal[5] == 1)) {
    error = e2 * a;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) &&
(IRfinal[3] == 0) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = e1 * a;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 0) &&
(IRfinal[3] == 0) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = 0;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 0) &&
(IRfinal[3] == 1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = -e1 ;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 0) && (IRfinal[2] == 0) &&
(IRfinal[3] == 1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = -e2 ;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 0) && (IRfinal[2] == 1) &&
(IRfinal[3] == 1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = -e3 ;
}

```



```

    } else if ((IRfinal[0] == 0) && (IRfinal[1] == 0) && (IRfinal[2] == 1) &&
(IRfinal[3] == 1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
        error = -e4 ;
    } else if ((IRfinal[0] == 0) && (IRfinal[1] == 1) && (IRfinal[2] == 1) &&
(IRfinal[3] == 1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
        error = -e5 ;
    } else {
        if (previouserror == -e5 || previouserror == -e4) {
            error = -e6;
        } else if (previouserror == e5 || previouserror == e4) {
            error = e6 * a;
        } else if (previouserror == 0 || previouserror == -e1 || previouserror == e1) {
            error = 0;

        } else {
            error = previouserror;
        }
    }
}

//PID 計算角度誤差
intg += error;
der = error - previouserror;
pterm = Kp * error;
iterm = Ki * intg;
dterm = Kd * der;
output = (pterm + iterm + dterm) / (Kp + Ki + Kd);
//Serial.print(output);
previouserror = error;
ppreviouserror = previouserror;

//轉向
output = error;
ang = output + 90;
servoturn.write(ang);

gate = 0;
delay(1000);
}

```

2. 連接無刷馬達及計步器的 arduino

```
#include <Servo.h>

Servo brushless;
Servo brake;

int gate = 0;
int encoderpin = 2;
int brakepin = 3;
int brushlesspin = 9;

int timed;
int modee = 1;
int output = 20;
int routput[6] = {20, 40, 60, 80, 100, 120};
int switchmode = 0;
//編碼器 param
boolean encoder1 = false;
boolean pencoder1 = false;
boolean encoder2 = false;
int encodercount = 0;

int displacement = 0;
int wheelR = 2.4; //cm
int velocity = 0;
int accel = 0, a, aerror, v, verror;

int deltadisplacement = 0, pdisplacement = 0, deltavelocity = 0, pvelocity = 0,
deltatime = 0;
unsigned long presenttime, starttime;
unsigned long pasttime, stoptime, dstoptime;

int bangle = 120;
int nbangle = 90;

void setup()

{
```

```

    brake.attach(brakepin);
    Serial.begin(9600);
    brushlessset();

}

void brushlessset() {
    brushless.attach(9);
    brushless.write(180); //設定上界
    delay(2000);
    brushless.write(20); //設定下界
    delay(2000);
}

void loop() {
    // brushless.write(40);
    //brushless.writeMicroseconds(830);

    if (encodercount <= 30) {
        brushless.write(40);
    } else if (encodercount > 30 || encodercount <= 50) {
        brushless.write(100);
    } else if (encodercount > 50) {
        brushless.write(20);
    }

    encoder1 = digitalRead(encoderpin);
    if (encodercount = 1) {
        starttime = millis();
    }
    if (encoder1 != encoder2) {
        if (encoder1 == 1) {
            if (pencoder1 != encoder1) {
                encodercount = encodercount + 1;
                presenttime = millis();
                pdisplacement = displacement;
                displacement = encodercount / 20 * 2 * 3.14 * wheelR; //cm
                Serial.print(encoder1);
                Serial.print(",");
                Serial.print(encoder2);
            }
        }
    }
}

```



```

    brake.write(nbangle);
}

if (displacement >= 150) {
    modee = 2;
    switchmode = 1;
}
case 2: //v = 30 150-300
    if (switchmode = 1) {
        output = routput[4];
        switchmode = 0;
    }

    v = 30;
    if (verror > 0) { //
        output = routput[1];
        brake.write(bangle);
    } else if (verror < 0) {
        if (output < 100) {
            output = output + 0.5;
            brake.write(nbangle);
        }
    }
    if (displacement >= 300) {
        modee = 3;
        switchmode = 1;
    }
case 3: //a = -3 300-420
    if (switchmode = 1) {
        output = routput[1];
        switchmode = 0;
    }
    a = -3;
    if (aerror > 0) {
        if (output > 25) {
            output = 20;
            brake.write(bangle);
        }
    }

```

```

} else if (aerror < 0) {
    if (output < 120) {
        output = 50;
        brake.write(nbangle);
    }
}
if (displacement >= 420) {
    modee = 4;
    switchmode = 1;
}
case 4: //v = 0 420-480
    if (switchmode = 1) {
        output = routput[1];
        brake.write(bangle);
        stoptime = presenttime;
    }
    dstoptime = stoptime - presenttime;
    if (dstoptime = 8) {
        switchmode = 1;
        brake.write(nbangle);
        modee = 5;
    }
}
case 5: //a = 3.75 480-570
    if (switchmode = 1) {
        output = routput[3];
        switchmode = 0;
    }
}

a = 3.75;
if (aerror > -0.5 && aerror < 0) {
    if (output < 140) {
        output = output + 1;
        brake.write(nbangle);
    }
} else if (aerror > -1 && aerror < -0.5) {
    if (output < 140) {
        output = output + 1.5;
        brake.write(nbangle);
    }
}

```

```

    }
} else if (aerror > 0) {
    if (output > 25) {
        output = output - 2.5;
        brake.write(bangle);
    }
} else if (aerror < -1) {
    output = routput[6];
    brake.write(nbangle);
}

if (displacement >= 570) {
    modee = 6;
    switchmode = 1;
}
case 6: //v = 30 570-690
    if (switchmode = 1) {
        output = routput[4];
        switchmode = 0;
    }

v = 30;
if (verror > 0) {
    output = routput[1];
    brake.write(bangle);
} else if (verror < 0) {
    if (output < 60) {
        output = output + 0.5;
        brake.write(nbangle);
    }
}

if (displacement >= 690) {
    modee = 7;
    switchmode = 1;
}

case 7: //a = 3.75 690-840
    if (switchmode = 1) {

```

```

    output = routput[3];
    switchmode = 0;
}

a = 3.75;
if (aerror > -0.5 && aerror < 0) {
    if (output < 130) {
        output = output + 1;
        brake.write(nbangle);
    }
} else if (aerror > -1 && aerror < -0.5) {
    if (output < 130) {
        output = output + 1.5;
        brake.write(nbangle);
    }
} else if (aerror > 0) {
    if (output > 25) {
        output = output - 2.5;
        brake.write(bangle);
    }
} else if (aerror < -1) {
    output = routput[6];
    brake.write(nbangle);
}

if (displacement >= 840) {
    modee = 8;
    switchmode = 1;
}

case 8: //v = 45 840-1140
    if (switchmode = 1) {
        output = routput[4];
        switchmode = 0;
    }

v = 45;
if (verror > 0) {

```



```

    output = routput[1];
    brake.write(bangle);
} else if (verror < 0) {
    if (output < 80) {
        output = output + 0.5;
        brake.write(nbangle);
    }
}
if (displacement >= 1140) {
    modee = 9;
    switchmode = 1;
}

case 9: //a = -5.4 1140-1320
    if (switchmode = 1) {
        output = routput[1];
        switchmode = 0;
    }
    a = -5.4;
    if (aerror > 0) {
        if (output < 25) {
            output = output - 1;
            brake.write(bangle);
        }
    } else if (aerror < 0) {
        if (output < 120) {
            output = output + 1;
            brake.write(nbangle);
        }
    }
    if (displacement >= 1320) {
        modee = 10;
        switchmode = 1;
    }

case 10: //v = 9 1320-1410
    if (switchmode = 1) {
        output = routput[4];

```

```

        switchmode = 0;
    }

    v = 9;
    if (verror > 0) {
        output = routput[1];
        brake.write(bangle);
    } else if (verror < 0) {
        if (output < 60) {
            output = output + 1;
            brake.write(nbangle);
        }
    }
    if (displacement >= 1140) {
        modee = 11;
        switchmode = 1;
    }

case 11:    //v = 0    1470
    if (switchmode = 1) {
        output = routput[1];
        brake.write(bangle);
    }

}
}

```

9.2 期末測試

```

#include <Servo.h>
#include <MsTimer2.h>
#include <NewPing.h>

```

匯入必要的程式庫

```

Servo brushless;
Servo brake;
Servo servoturn;

```

宣告 Servo

```

//sonar param
int triggerf = 11;

```

超音波所需參數

```
int echof = 12;
int maxdis = 200;
unsigned long cardis = 40, pcardis;
int diserror = 0;    //車距
NewPing sonarf(triggerf, echof, maxdis);
//NewPing sonars(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
```

```
//pin
int encoderpin = A6;
int brakepin = 6;
int brushlesspin = 8;
int turnpin = 10;
unsigned long time;
```

模組腳位

```
int boostmode = 1;
float output = 20;
int realoutput = 20;
```

```
//encoder param
int encoder = 0;
int count = -1;
int z = 1;
int overtake = 0;
int r = 0;
int highlowlabel = 0;
int thresholdh = 700;
int thresholdl = 200;
int stop_check_count = 0;
```

測速模組參數

```
//speed cal param
float x = 0;
float wheelR = 24; //mm
float velocity = 0;
float realvelocity = 0;
float v, verror, verror;
unsigned long now , lasttime;
float dx = 0, px = 0, dv = 0, pv = 0, dt = 0;
unsigned long presenttime, starttime;
```

計算速度

```
unsigned long pt;
```

```
//mode
```

```
int slow = 0;
```

```
int boost1 = 1;
```

```
//IR
```

```
int IR[] = {A0, A1, A2, A3, A4, A5};
```

```
int IRread[6];
```

```
int IRgate = 0;
```

```
boolean IRfinal[6];
```

循跡模組參數

```
float intg = 0, der = 0;
```

```
float turnout = 0; //turn angle
```

```
float error = 0, previouserror = 0;
```

```
float ang;
```

```
//PID
```

```
float Kp = 0.7;
```

```
float Ki = 0;
```

```
float Kd = 0.4;
```

```
float pterm , iterm, dterm;
```

PID 參數

```
//param of angle correction
```

```
float e1 = 3;
```

```
float e2 = 7; //6
```

```
float e3 = 14;
```

```
float e4 = 18; //17
```

```
float e5 = 24;
```

```
float e6 = 27; //27
```

```
void setup() {
```

```
    MsTimer2::set(2, en_update);
```

```
    MsTimer2::start();
```

```
    brake.attach(brakepin);
```

```
    brake.write(180);
```

```
    Serial.begin(9600);
```

```

pinMode(brushlesspin, OUTPUT);
brushlessset();

for (int i = 0 ; i < 6; i++) {
    pinMode(IR[i], INPUT);
    servoturn.attach(turnpin);
    servoturn.write(90);
}
}

```

```

void brushlessset() {
    brushless.attach(8);
    brushless.write(180); //設定上界
    delay(2000);
    brushless.write(20); //設定下界
    delay(4000);
}

```

無刷馬達前置設定

```

void en_update() {
    sei();
    time = millis();
    r = analogRead(encoderpin);

    if (r > 20) {
        encoder = r;
    }

    if (encoder > thresholdh && highlowlabel != 1) {
        highlowlabel = 1;
        count = count + 1;
    } else if (encoder < thresholdl && highlowlabel != -1) {
        highlowlabel = -1;
        count = count + 1;
    } else {
        stop_check_count++;
    }
}
}

```

偵測編碼器

```

void loop() {

    IR_update();           //IR 循跡
    countvelocity();      //計算當下速度
    v_update();           //算理論速度與誤差
    outputerror_update(); //算出誤差值所需要的調整

    //=====跟車=====
    cardis = constrain(sonarf.ping_cm(), 0, 70); //車距

    if (x < 71300 && velocity < 520) {           //在與前車距離
        if (cardis != 0) {                       //若前方有東西
            if (cardis < 3) {
                diserror = -15;

            } else if (cardis < 6) {
                diserror = -6;

            } else if (cardis < 8) {
                diserror = 4;

            } else if (cardis < 15) {
                diserror = 5;

            } else if (cardis < 20) {
                diserror = 5;

            } else if (cardis < 25) {
                diserror = 6;

            } else if (cardis < 30) {
                diserror = 6;

            } else if (cardis < 35) {
                diserror = 7;

            } else {

```

```

        diserror = 8;

    }
} else if (cardis == 0 && x < 23000) { //候車追上
    diserror = 9;
} else if (cardis == 0 && x < 40000) { //前車控速    轉回時開啟
    diserror = 0;
} else if (cardis == 0 && x < 71300) { //後車追上
    diserror = 9;
} else {
    diserror = 0;
}
} else {
    diserror = 0;
}

output = chooseduty(x) + fanpid(1, 0.3, verror, verror) + diserror ; //輸出

//=====write fan=====

realoutput = constrain(output, 20, 40);

if (x < 71300 && boost1 == 0) {
    brushless.write(realoutput);           //一般情況
} else if (boost1 == 1) {
    brushless.write(realoutput + 5);      //跟車前段加速
    if (velocity > 420) {
        boost1 = 0;                       //到達目標速度停止加速
    }
} else {
    brushless.write(20);
}

delay(15);

}

void IR_update() {           //循跡

```

```

for (int i = 0; i < 6 ; i++) {

    IRread[i] = analogRead(IR[i]);

}
IRgate = IRgate / 6 + 350;
for (int i = 0; i < 6 ; i++) {
    if (IRread[i] < IRgate ) {
        IRfinal[i] = 1; //not black
    } else if (IRread[i] > IRgate) {
        IRfinal[i] = 0; //black

    }

}

if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] == 1) &&
(IRfinal[4] == 1) && (IRfinal[5] == 0)) {
    error = e5 ;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] ==
1) && (IRfinal[4] == 0) && (IRfinal[5] == 0)) {
    error = e4;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] ==
1) && (IRfinal[4] == 0) && (IRfinal[5] == 1)) {
    error = e3 ;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] ==
0) && (IRfinal[4] == 0) && (IRfinal[5] == 1)) {
    error = e2;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] ==
0) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = e1;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 0) && (IRfinal[3] ==
0) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = 0;
} else if ((IRfinal[0] == 1) && (IRfinal[1] == 1) && (IRfinal[2] == 0) && (IRfinal[3] ==
1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
    error = -e1 ;
}

```



```

    } else if ((IRfinal[0] == 1) && (IRfinal[1] == 0) && (IRfinal[2] == 0) && (IRfinal[3] ==
1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
        error = -e2 ;
    } else if ((IRfinal[0] == 1) && (IRfinal[1] == 0) && (IRfinal[2] == 1) && (IRfinal[3] ==
1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
        error = -e3 ;
    } else if ((IRfinal[0] == 0) && (IRfinal[1] == 0) && (IRfinal[2] == 1) && (IRfinal[3] ==
1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
        error = -e4;
    } else if ((IRfinal[0] == 0) && (IRfinal[1] == 1) && (IRfinal[2] == 1) && (IRfinal[3] ==
1) && (IRfinal[4] == 1) && (IRfinal[5] == 1)) {
        error = -e5;
    } else {
        if (previouserror == -e5 || previouserror == -e4) {
            error = -e6 ;
        } else if (previouserror == e5 || previouserror == e4) {
            error = e6;
        } else if (previouserror == 0 || previouserror == -e1 || previouserror == e1) {
            error = 0;
        }

    } else {
        error = previouserror;
    }
}

```

//PID 計算角度誤差

```

now = millis();
dt = now - lasttime;
intg += error * dt;
der = (error - previouserror) / dt;
pterm = Kp * error;
item = Ki * intg;
dterm = Kd * der;
turnoutput = pterm + item + dterm;
//Serial.print(turnoutput);
previouserror = error;
lasttime = now;

```

```

//轉向
ang = turnout + 97;
//Serial.print(turnoutput);

//=====超車&控速=====
if (x < 22500) { // 第一圈後車 11260
    slow = 6;
} else if(x < 24000){ //準備超車
    slow = 5;
} else if(x < 28000){ //超車
    slow = 7;
} else if(x < 34200){ //超車後至準備被超
    slow = 5;
} else if(x < 36000){ //給超前降速
    slow = 1;
} else if (x < 42900) { //給超 26960+11280
    if (x < 36500) {
        overtake= 1;
        ang = 107; //left turn
    } else if (x < 37500 && overtake == 1) {
        ang = 100;
        if (IRfinal[5] == 0) {
            ang = 85;
        } else if (IRfinal[4] == 0) {
            ang = 88;
        } else if (IRfinal[3] == 0 ) {
            overtake = 0;
            //outside = 1;
        }
    } else if (x < 39200) {
        //outside normal
        slow = 2;
    } else if (x < 41800) {
        slow = 1;
        if (x < 39800) {
            overtake = 1;
            ang = 88;
        } else if (x < 41800 && overtake == 1) {

```

```

    ang = 93;
    if (IRfinal[0] == 0) {
        ang = 108;
    } else if (IRfinal[1] == 0) {
        ang = 106;
    } else if (IRfinal[2] == 0) {
        overtake = 0;
        //inside = 1;
    }
}
} else if (x < 42900) {
    if (velocity < 400) {
        boost1 = 1;
    }
}
} else if (x < 47300) { //回到後車
    slow = 6;
} else if (x < 71300) {
    slow = 6;
} else {
    slow = 9;
}

IRgate = 0;

ang = constrain(ang, 80, 120);
servoturn.write(ang);
}

void countvelocity() {
    x = ((float)count * 2 * 3.14 * wheelR / 40);
    dx = x - px;
    dt = time - pt ;

    velocity = dx * 1000 / dt;
    if (velocity < 800) {
        realvelocity = constrain(velocity, 0, 800);
    }
}

```

```

px = x;

pt = time;
stop_check_count = 0;
}

void v_update() {
{
    if (stop_check_count > 100)
        realvelocity = 0;
}
verror = track_desire_v(x) - realvelocity;
}

//速度需求
float track_desire_v(float x) {          //output desire x dot mm/s
    if (x < 71300) {                    //22646
        if (slow == 0) {                //第二圈改 400*****
            return 410;
        } else if (slow == 1) {
            return 380;
        } else if (slow == 2) {
            return 250;
        } else if (slow == 3) {
            return 430;
        } else if (slow == 5){
            return 390;
        } else if (slow == 6){
            return 540;
        } else if (slow == 7){
            return 440;
        } else if (slow == 9) {
            return 0;
        }

    } else {
        return 0;
    }
}

```

```

/*if (overtake == 1) {

    }
    if (x < 0) return min_speed;
    else if (x < 1500) {
    return 400;
    }
    else return 0;*/

}

//底
int chooseduty(int x) {
    if (x < 71300) {

        return 32;

    } else{
        return 20;
    }
}

float fanpid(float P, float D, float pdata, float ddata) {
    int a = (int)(P * pdata + D * ddata);
    return a;
}

void outputerror_update() {

    if (verror > 0) {
        if (abs(verror) < 35) {
            verror = 1;

        } else if (abs(verror) < 75) {
            verror = 2;

        } else if (abs(verror) < 110) {
            verror = 3;

```

```
} else if (abs(verror) < 140) {  
    verror = 4;  
  
} else if (abs(verror) < 220) {  
    verror = 5;  
  
} else {  
    verror = 7;  
  
}  
} else if (verror < 0) {  
    if (abs(verror) < 25) {  
        verror = -2;  
  
    } else if (abs(verror) < 90) {  
        verror = -3;  
  
    } else if (abs(verror) < 180) {  
        verror = -5;  
  
    } else if (abs(verror) < 205) {  
        verror = -6;  
  
    } else if (abs(verror) < 250) {  
        verror = -7;  
  
    } else {  
        verror = -10;  
  
    }  
} else if (verror == 0) {  
    verror = 0;  
  
}  
  
}
```