

臺北市立大安高級工業職業學校
電子科

專題報告

藍芽兩輪平衡車

Bluetooth two-wheel balance car

學生 組長：陳定言
組員：簡丞志
組員：陳奕均
組員：王世薰

指導老師：黃建中老師

中華民國 107 年 1 月

中文摘要

現今社會交通忙碌，大多數人為了方便而自行開車，但是在都市中想要停車，位置幾乎都一位難求，因此有了兩輪平衡車的發明，不但使用電力能源而且汙染低，幾乎沒有噪音。讓我們能在都市中進行短暫的旅途，達到最高的效益。本專題主要是研究藍芽兩輪平衡車如何自行保持平衡，並且透過陀螺儀感測器及藍芽遙控來控制任何想行進的方向，最後再利用 Arduino 程式的撰寫，完成本專題的製作。

平衡自走車是未來交通工具的趨勢，所以我們想探討其中的理論，但因為在製作過程中遇到了許許多多的瓶頸，導致在一開始著手時有點不知所措，不小心把馬達模組燒壞及車體機構設計不好，不過我們在與指導老師討論及上網蒐集更多的資料後，針對遇到的問題再重新整理、理解，也學會了更多以往所沒接觸過的範圍，使我們獲益良多。

關鍵字：Arduino、平衡車、陀螺儀

英文摘要

Today's social transportation is busy, most people drive by themselves for convenience, but in the city, they want to park, and almost all of them are hard to find. Therefore, with the invention of two-wheeled balance car, not only the use of electric energy but also low pollution, there is almost no noise. Let us make a short trip in the city to achieve the highest efficiency. This topic is mainly to study how the Bluetooth two-wheeled balance car can maintain its own balance, and control any direction that wants to travel through the gyroscope sensor and the Bluetooth remote control. Finally, the Arduino program is used to complete the production of this topic.

Balanced self-propelled vehicles are the trend of future vehicles, so we want to explore the theory, but because of the many bottlenecks encountered in the production process, I was a bit overwhelmed when I started, accidentally put the motor module Burnout and car body design is not good, but after discussing with the instructor and collecting more information online, we have reorganized and understood the problems encountered, and learned more scopes that we have never touched before. We have benefited a lot.

目錄

中文摘要	I
英文摘要	II
目錄	III
圖目錄	V
表目錄	VI
第一章 前言	- 1 -
1-1 專題製作背景及目的	- 1 -
1-2 預期成果	- 1 -
第 2 章 理論探討	- 2 -
2-1 機構部分	- 2 -
2-2 元件與硬體	- 2 -
2-2-1 編碼器減速電機 AB 雙向直流馬達	- 2 -
2-2-2 十軸加速度計 JY901B	- 5 -
2-2-3 TB6612FNG 直流馬達驅動/控制板	- 6 -
2-2-4 Arduino mega2560	- 7 -
2-2-5 HC-05 藍芽模組	- 8 -
2-2-6 OLED 模組	- 8 -
2-2-7 HC-SR04 超音波	- 9 -
2-2-8 電源零件	- 10 -
2-3 平衡原理	- 11 -
2-4 PID 控制原理	- 11 -
2-5 軟體部分	- 12 -
2-5-1 Arduino IDE	- 12 -
2-5-2 EAGLE 電路版設計	- 13 -
第 3 章 專題設計	- 14 -
3-1 系統架構圖	- 14 -
3-2 系統流程圖	- 14 -
3-3 甘特圖	- 15 -
第 4 章 專題成果	- 16 -
4-1 自主平衡	- 16 -
4-2 可爬坡 10.9 度	- 16 -
第 5 章 結論和建議	- 17 -

5-1 結論.....	- 17 -
5-2 問題與解決.....	- 17 -
5-3 建議.....	- 18 -
參考文獻.....	- 19 -
附錄.....	- 20 -
附錄一 設備清單.....	- 20 -
附錄二 材料清單.....	- 21 -
附錄三 程式碼.....	- 22 -
成員簡歷.....	- 34 -

圖目錄

圖 1 車體架構示意圖	- 2 -
圖 2 編碼器減速電機 AB 雙向直流馬達外觀.....	- 3 -
圖 3 編碼器減速電機 AB 雙向直流馬).....	- 3 -
圖 4 編碼器減速電機 AB 雙向直流馬達尺寸.....	- 3 -
圖 5 編碼器減速電機 AB 雙向直流馬達內部機構.....	- 4 -
圖 6 JY901B 正面零件圖及背面接腳圖.....	- 5 -
圖 7 TB6612 馬達驅動 IC.....	- 6 -
圖 8 TB6612 馬達驅動 IC.....	- 6 -
圖 9 ARDUINO MEGA2560 板.....	- 7 -
圖 10 HC-05 藍芽模組.....	- 8 -
圖 11 SSD1306 OLED 模組	- 9 -
圖 12 HC-SR04 超音波.....	- 9 -
圖 13 電池盒	- 10 -
圖 14 車體平衡示意圖.....	- 11 -
圖 15 PID 概念圖.....	- 12 -
圖 16 ARDUINO 軟體介面.....	- 12 -
圖 17 EAGLE 程式設計介面.....	- 13 -
圖 18 系統架構圖	- 14 -
圖 19 系統流程圖	- 14 -
圖 20 甘特圖	- 15 -
圖 21 經人為影響仍可立即平衡.....	- 16 -
圖 22 可爬坡 10.9 度	- 16 -
圖 23 設備清單	- 20 -

表目錄

表 1 編碼器減速電機 AB 雙向直流馬達的電機參數以及編碼器參數	- 4 -
表 2 JY901B 相關參數	- 5 -
表 3 TB6612 馬達驅動 IC 參數表	- 6 -
表 4 TB6612 馬達驅動 IC 運作表	- 7 -
表 5 ARDUINO MEGA2560 參數表	- 8 -
表 6 SSD1306 參數	- 9 -
表 7 HC-SR04 超音波參數	- 10 -
表 8 材料清單	- 21 -

第一章 前言

1-1 專題製作背景及目的

人類的科技不斷進步，隨著人類的需求也更加多樣化，代步工具不斷推陳出新。最近常常在老街或路上看到有人使用二輪平衡車，引起我的注意力，讓我好奇其中的原理。經過我上網查尋後，發現那項產品名字叫—賽格威(Segway)是一種交通工具被稱作電動個人輔助機動裝置 (Electric Personal Assistive Mobility Device, EPAMD)，俗稱電動代步車。了解二輪平衡車的由來後，因此我們善用從高一至高三所學的專業技巧與理論，希望能夠自己做出一台二輪平衡車，雖然沒辦法載人，但至少我們希望能夠用手機遙控的方式來操縱它，達成我們的需求。

1-2 預期成果

本專題預計要完成一台「兩輪藍牙平衡車」，透過手機連接藍牙控制兩輪平衡車的行走方向：前進、轉彎及原地打轉，同時能夠藉由馬達的正反轉，使得平衡車在行駛的過程中能保持平衡。

在這次的研究中，透過組員們互相溝通與學習後，有了老師認真的教學下，我們針對理論進行探討與概念理解，其中包括有 Arduino、EAGLE PCB Designer 以及各項資源，幫助我們度過各式各樣的難關。

第 2 章 理論探討

2-1 機構部分

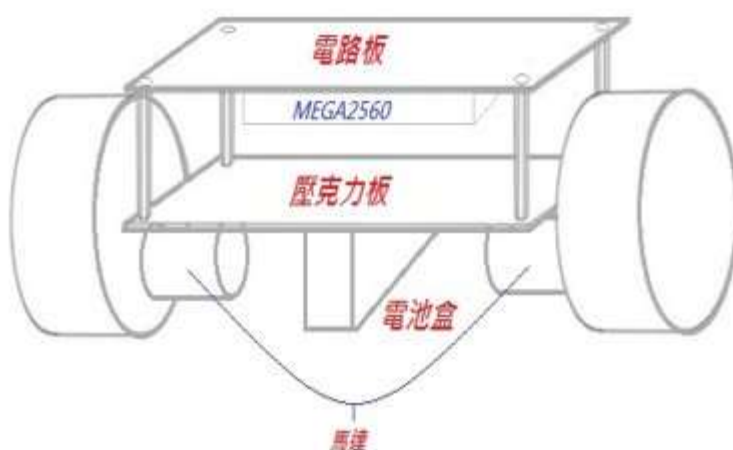


圖 1 車體架構示意圖

2-2 元件與硬體

2-2-1 編碼器減速電機 AB 雙向直流馬達

旋轉編碼器也稱為軸編碼器，是將旋轉位置或旋轉量轉換成類比或數位訊號的機電裝置，一般裝設在旋轉物體中垂直旋轉軸的一面。旋轉編碼器用在許多需要精確旋轉位置及速度的場合等。

編碼器的主要用途是量測角度位置與轉速，另外還配備像是系統診斷與參數設定等常見功能編碼器是將信號或數據進行編製、轉換為可用以通訊、傳輸和存儲的信號形式的設備。另外也用來檢測機械運動的速度、位置、角度、距離或計數，除了應用在機械外，許多的馬達控制如伺服馬達 均需配備編碼器以供馬達控制器作為換相、速度及位置的檢出。

本專題使用的是編碼器減速電機 AB 雙向直流馬達，是一款使用霍爾感測器編碼器的測速模組，配有 13 線強磁碼盤，在 AB 雙向輸出共同利用下，通過計算可得出車輪轉一圈時，脈衝數可達到 $30 \times 13 \times 2 = 780$ 個，單相也可以達到 390

個，高精度，可以輕鬆操控平衡車。

編碼器減速電機 AB 雙向直流馬達自帶高精度磁編碼器，馬力強勁性能優秀，具有高精度，抗干擾能力強等特點，內部具備了霍爾測速碼盤及高品質的碳刷等等。



圖 2 編碼器減速電機 AB 雙向直流馬達外觀

編碼器減速電機 AB 雙向直流馬達規格與尺寸如下：

本專題所使用的是編碼器減速電機 AB 雙向直流馬達的電機參數以及編碼器參數如下所示：

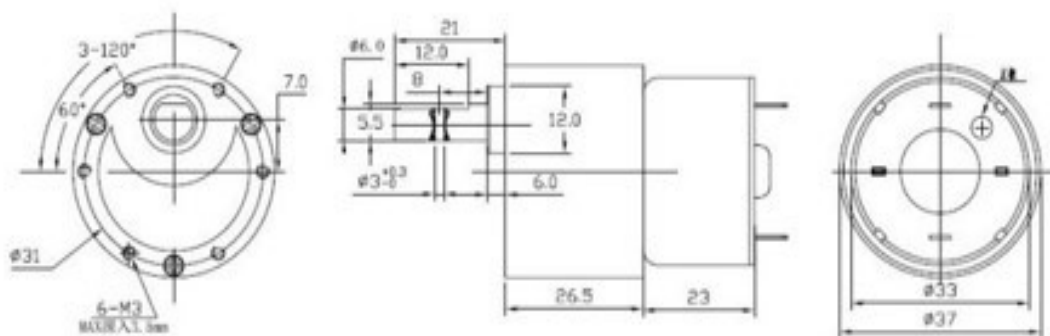


圖 4 編碼器減速電機 AB 雙向直流馬達尺寸

電機參數	編碼器參數
額定電壓：12V	類型：AB 相增量是霍爾編碼器
額定功率：4.8W	線速：390（以減速器輸出軸測量）
電機類型：永磁有刷	供電電壓：5.0V
額定電流：2.8A	特色功能：自帶上拉整形，單片機可以直接讀取
額定扭矩：10Kgf.cm	接口類型：PH2.0(標配連接線)
原始轉速：11000rpm	
減速比：1：30	

表 1 編碼器減速電機 AB 雙向直流馬達的電機參數以及編碼器參數

本專題所使用的編碼器減速電機 AB 雙向直流馬達的內部結構如下：



圖 5 編碼器減速電機 AB 雙向直流馬達內部機構

2-2-2 十軸加速度計 JY901B

JY901B 十軸感測器模組具備三圍加速度、三圍陀螺儀、三圍角度、三圍磁場以及氣壓高度，內部採用高性能的微處理器和先進的動力學演算法與卡爾曼動態濾波法，能夠快速運算出當前的狀態，並且支援串口速率 2400bps~921600bps，IIC 接口支持全速 400K 速率。具備有 GPS 連接能力，接受 NMEA-0183 標準的串口 GPS 數據，形成 GPS -IMU 組合導航單元。有關 JY901B 相關參數如下：

電壓：3.3~5V
電流：25<mA
量程：加速度:±2/4/8/16 g 角速度:±250/500/1000/2000 °/s 角度±180°
穩定性：加速度：0.01g 角速度 0.05°/s
數據輸出頻率 0.1Hz~200Hz
VCC 模块电源：3.3V 或 5V 輸入
RX：串行數據輸入
TX：串行數據輸出
GND：地線
SCL：I2C 時鐘線
SDA：I2C 數據線

表 2 JY901B 相關參數

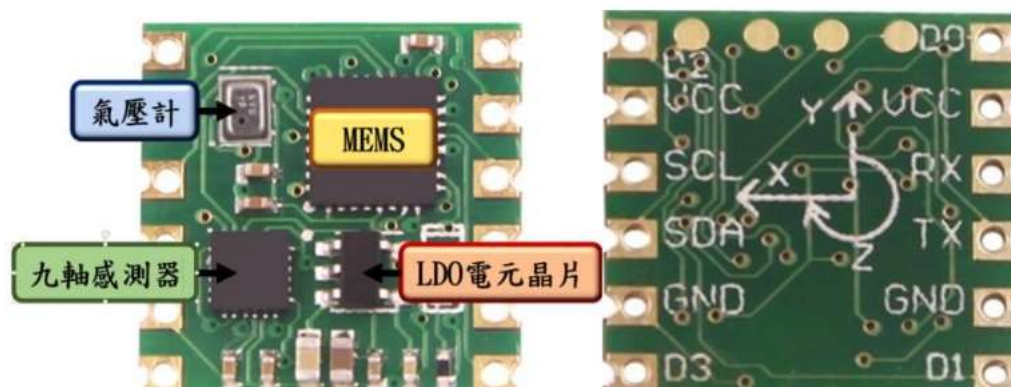


圖 6 JY901B 正面零件圖及背面接腳圖

2-2-3 TB6612FNG 直流馬達驅動/控制板

TB6612FNG 是東芝生產的馬達驅動與控制 IC，內部包含兩組 H 橋式電路，可驅動和控制兩個小型直流馬達，或者一個雙極性步進馬達。

1. 模組電源 2.7~5.5VDC (可由 Arduino 板子上的 5V 或 3.3VDC 供電)
2. 馬達電源 2.5~13VDC (由於馬達電流需求較大，必須由外部電源供電)



圖 8 TB6612 馬達驅動 IC

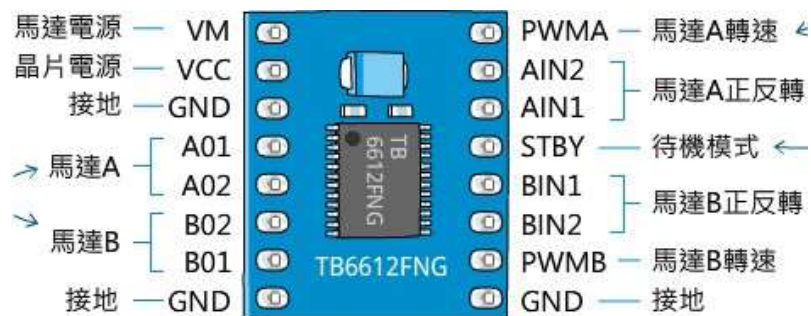


圖 7 TB6612 馬達驅動 IC

馬達工作電壓：	2.5V~13.5V
晶片工作電壓：	2.7A~5.5A
單一通道輸出電流：	1.2A(極限 3.2A)
H 橋式電路元件：	MOSFET
高速切換二極體：	晶片內建
高溫保護電路：	有
效率：	91.74%

表 3 TB6612 馬達驅動 IC 參數表

輸入			輸出		模式說明
AIN1	AIN2	STBY	A01	A02	
1	1	1	0	0	煞車(shortbrake)
0	1	1	0	1	逆時針方向旋轉
1	0	1	1	0	順時針方向旋轉
0	0	1	0	0	停止(stop)
0	0	0	0	0	待機(standby)

表 4 TB6612 馬達驅動 IC 運作表

2-2-4 Arduino mega2560

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

由於開放原代碼，以及使用 Java 概念（跨平臺）的 C 語言開發環境，讓 Arduino 的周邊模組以及應用迅速的成長。而吸引 Artist 使用 Arduino 的主要原因是可以快速使用 Arduino 語言與 Flash 或 Processing... 等軟體通訊，作出多媒體互動作品。



圖 9 Arduino MEGA2560 板

微處理器	<u>ATmega2560</u>
工作電壓	5V
輸入電壓（推薦）	7-12V
輸入電壓（限值）	6-20V
數字輸入/輸出引脚	54 路（其中 15 路可用於 PWM 輸出）
模擬輸入引脚	16

I/O 引腳電流	20mA
3.3V 引腳電流	50mA
Flash	256KB，其中引導程序占用 8KB
SRAM	8KB
EEPROM	4KB
時鐘頻率	16MHz

表 5 Arduino Mega2560 參數表

2-2-5 HC-05 藍芽模組

HC-05 模塊是一款易於使用的藍牙 SPP (串行端口協議) 模塊，專為透明無線串行連接設置而設計。

串口藍牙模塊是完全合格的藍牙 V2.0 + EDR (增強型數據速率) 3Mbps 調製，具有完整的 2.4GHz 無線電收發器和基帶。它採用 CSR Bluecore 04-External 單芯片藍牙系統，採用 CMOS 技術和 AFH (自適應跳頻功能)。它的佔地面積小至 12.7mmx27mm。



圖 10 HC-05 藍芽模組

2-2-6 OLED 模組

SSD1306 是一款單芯片 CMOS OLED / PLED 驅動器，帶有控制器，用於有機 / 聚合物發光二極管點陣圖形顯示系統。它由 128 個段和 64 個 commons 組成。這個 IC 是專為共陰極型 OLED 面板設計。

SSD1306 嵌入了對比度控制，顯示 RAM 和振盪器，減少了數量外部元件和功耗。它具有 256 級亮度控制。數據/命令通過通用硬件可選的 6800/8000 系列兼容並行接口從通用 MCU 發送，I2C 接口或串行外設接口。它適用於許多緊湊型便攜式應用，例如手機分顯示器，MP3 播放器和計算器等。



圖 11 SSD1306 OLED 模組

尺寸：	0.96 吋
分辨率高：	128*64
顏色：	白色
低功耗：	0.04W
額定電壓：	3.3V~5V
工作溫度：	-30 ° C~80 ° C
驅動晶片：	SSD1306
通信方式：	IIC

表 6 SSD1306 參數

2-2-7 HC-SR04 超音波

超聲波測距模塊 HC-SR04 提供 2cm~400cm 非接觸式測量功能，測距精度可達 3mm。這些模組包括超聲波發射器，接收器和控制電路，會發射一連串 40 kHz 的聲波並且從離它最近的物體接收回音。



圖 12 HC-SR04 超音波

工作電壓：	DC 5 V
工作電流：	15mA
工作頻率：	40Hz
最大距離：	4m
最小距離：	2cm
測量角度：	15°
觸發輸入信號：	10uS TTL 脈衝
迴聲輸出信號：	輸入 TTL 桿信號和範圍比例
尺寸：	45*20*15mm

表 7 HC-SR04 超音波參數

2-2-8 電源零件

為解決車的供應電源問題，我們使用了兩顆 3.7V 的電池，再通過電池盒串聯，形成 7.4V 的電源，以利驅動馬達和 Arduino 單晶片



圖 13 電池盒

2-3 平衡原理

平衡車的平衡主要靠陀螺儀與驅動電路，利用陀螺儀（包括加速度傳感器和角速度傳感器）與馬達作互相的配合，速度則是由車體的傾斜程度來控制的。

陀螺儀是平衡車的核心部件，其原理是這樣的：「一個旋轉物體的旋轉軸所指的方向在不受外力影響時，是不會改變的。」，就像我們騎自行車，輪子轉得越快越不容易倒，因為車軸有一股保持水平的力量。

通過感應車體重心的變化來調整方向車體，車體前傾前進，後傾後退，站立保持平穩，透過 JY901B 的偵測，使馬達帶動輪子調整使其拉回平衡點。如所示。

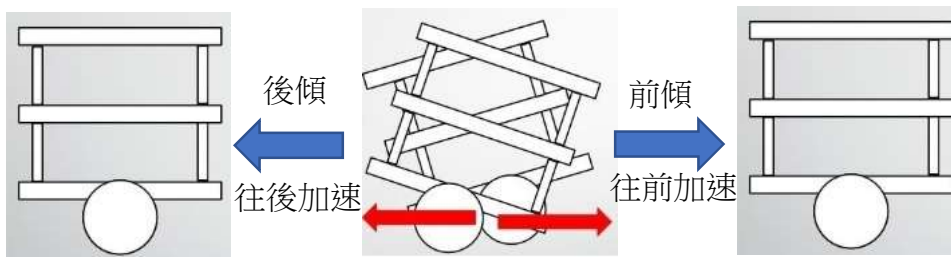


圖 14 車體平衡示意圖

2-4 PID 控制原理

所謂的 PID 控制器（比例-積分-微分控制器），由比例單元(P)、積分單元(I)和微分單元(D)組成。透過 K_p ， K_i 和 K_d 三個參數的設定。PID 控制器主要適用於基本上線性，且動態特性不隨時間變化，而 PID 控制是將控制器設定的數值和收集到的值相減的到誤差，後將誤差進行 PID 運算，使系統之後的數據接近或重合設定點。PID 控制系統為工業上常見之閉迴路式控制系統，其概念圖如所示，其說明如下：

- (一)比例(Proportional)：單位間的轉換，依比例放大。將所得誤差乘上常數 K_p 。
- (二)積分(Integral)：積分控制考慮過去誤差，將過去一段時間誤差再乘以一個正值的常數 K_i 。可從過去的誤差和來找到系統的輸出結果和預定值的平均誤差。通過加上負的平均誤差值，平均系統誤差值就會漸漸減少。
- (三)微分(Derivative)：微分控制考慮將來誤差，計算誤差的一階段，並和常數 K_d 相乘。導數結果越大，那麼控制系統就對輸出結果作出更快速的反應。 K_d 參數對改善控制器瞬間的變化很有幫助。

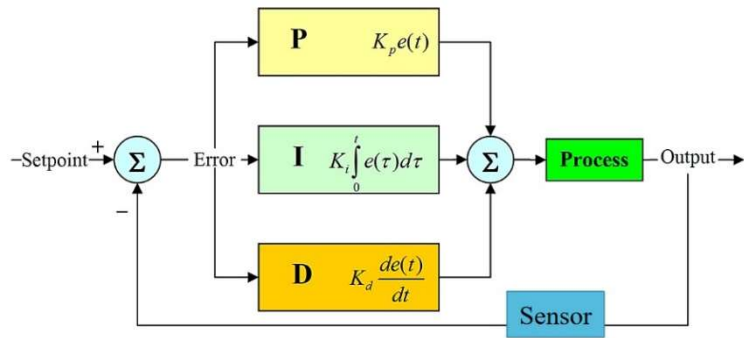


圖 15 PID 概念圖

2-5 軟體部分

2-5-1 Arduino IDE

Arduino 的軟體開發環境是開放源始碼，可以在 Arduino 官網免費下載，它所用的程式語言語法類似於 C/C++，具備文字編輯介面、常用工具欄、圖形化控制介面及錯誤編輯器，利用這項軟體進行程式的上傳，以便於和各項感應器的溝通。

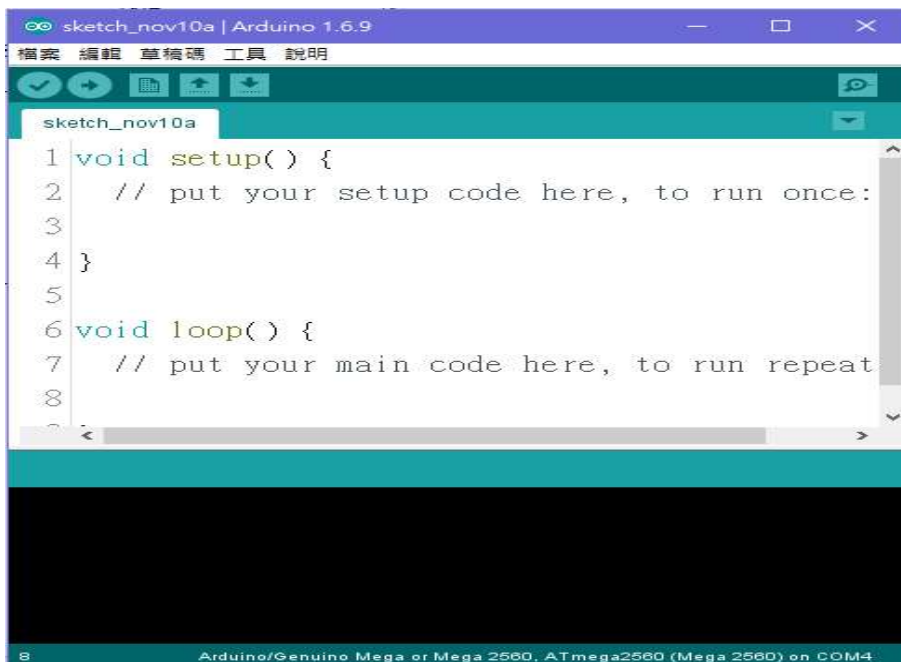


圖 16 Arduino 軟體介面

2-5-2 EAGLE 電路版設計

由於網路上的印刷電路板的設計的軟體十分多元，我們考慮到方便性與實用性，因此選擇了由 Autodesk 收購下來 EAGLE 的這一款 PCB LAYOUT 軟體，他支援跨平台的支持，雖然使用免費版只能使用 2 個示意圖、2 個信號或平面層及板面積為 80cm²，但是它擁有精簡化的介面以及方便的零件庫供我們使用，使我們不用考慮太複雜的使用程序。

EAGLE 中的 PCB 設計分為兩步。首先設計 SCH 原理圖，然後根據原理圖佈置 PCB 的位置。由於 EAGLE Free 版有限定版子的區域面積，而我們板子的面積不夠大，因此本專題花錢購買 EAGLE Standard 版，來獲得更多的額外功能。

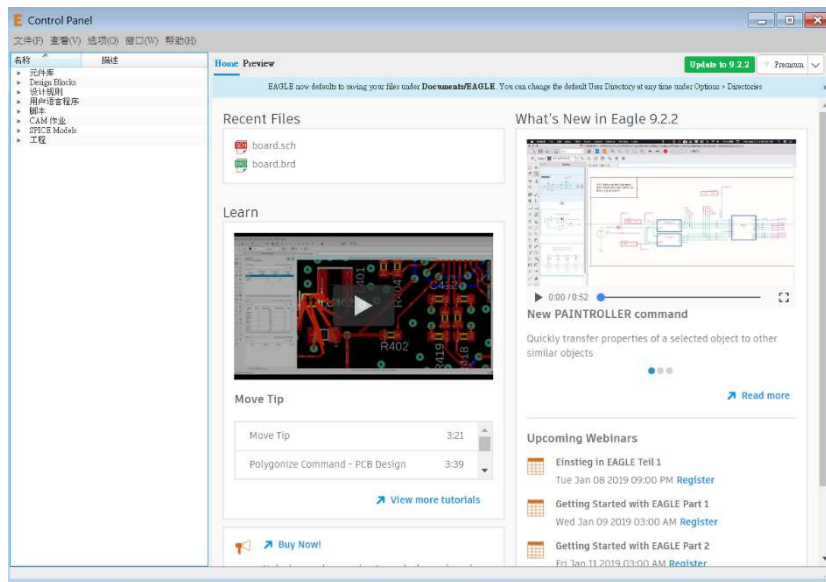


圖 17EAGLE 程式設計介面

第 3 章 專題設計

3-1 系統架構圖



圖 18 系統架構圖

3-2 系統流程圖

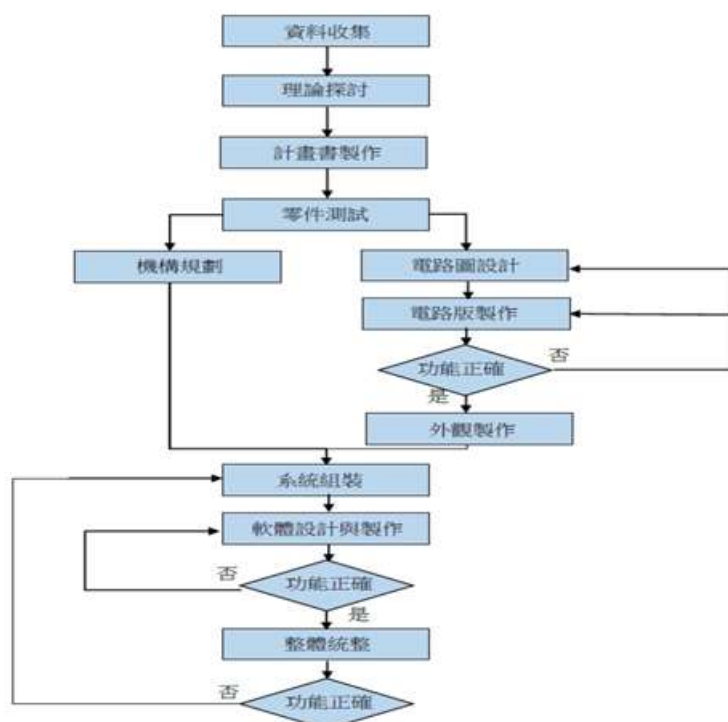


圖 19 系統流程圖

3-3 甘特圖

工作項目	週次																				負責成員
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
蒐集資料	■	■	■	■	■	■															全員
理論探討	■	■	■	■	■	■															全員
計畫書製作				■	■	■															全員
零件測試				■	■	■	■	■													全員
機構規劃				■	■	■															王世薰 陳奕均
電路圖設計				■	■	■															陳定言 簡丞志
電路板製作				■	■	■															王世薰 陳奕均
外觀製作					■	■	■	■	■												陳定言
系統組裝								■	■	■											簡丞志 陳定言
軟體設計與製作									■	■	■	■	■	■	■						全員
整體統整														■	■	■					陳定言
完成成品																■	■	■	■		全員
撰寫報告						■	■	■	■	■		■	■	■	■	■	■	■	■	■	全員
口頭報告									■	■	■	■	■	■	■	■	■	■	■	■	全員
預定進度	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	累積百分比 %

圖 20 甘特圖

第 4 章 專題成果

4-1 自主平衡

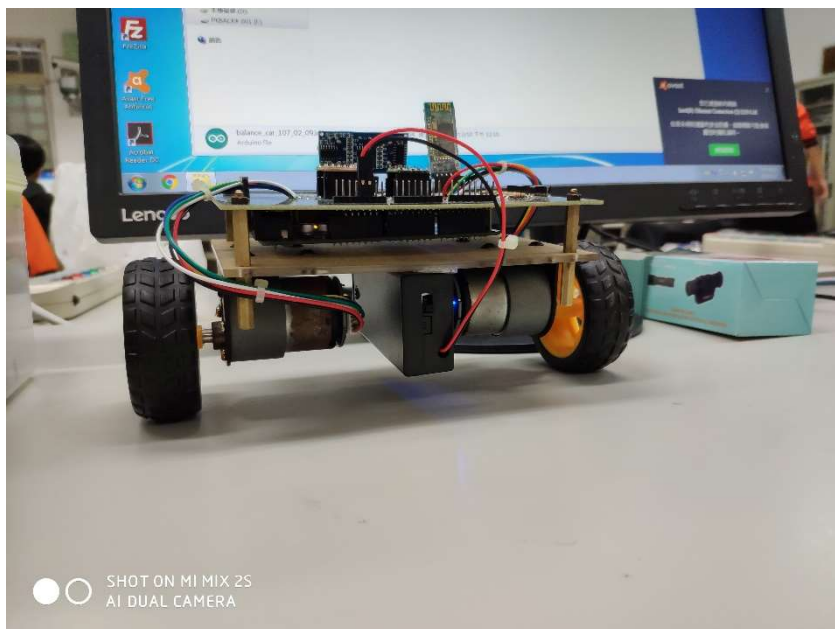


圖 21 經人為影響仍可立即平衡

4-2 可爬坡 10.9 度

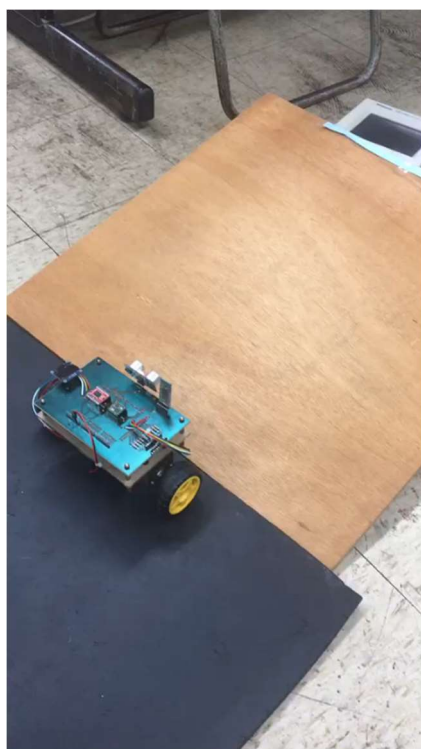


圖 22 可爬坡 10.9 度

第 5 章 結論和建議

5-1 結論

在這次的研究中，我們發現剛開始的構想和實際執行上會有很大的差異，要實際將它表現出來，確實不太容易，對於程式一開始摸不著頭緒，常常會因為一些地方影響了整個環節的流暢度。但透過組員們之間的討論與學習，加上老師的認真教學下，針對理論進行探討與概念理解，不斷尋找資源，並思考之後將會遇到的問題，逐一擊破，最後才成功。

平衡自走車是未來交通工具的趨勢，所以我們想探討其中的理論，但因為在製作過程中遇到了許許多多的瓶頸，導致在一開始著手時有點不知所措，不小心把馬達模組燒壞及車體機構設計不好等等，不過我們在與指導老師討論及上網蒐集更多的資料後，針對遇到的問題再重新整理、理解，也學會了更多以往所沒接觸過的範圍。

在軟體程式及感測器模組上，雖然網路資訊非常的豐富，但是相關的文件並不是很多，我們有部分程式參考了國外的分享專區，測試各個零件的實際運作情況，修改了部分程式碼後，燒錄並且執行，都能得到不錯的結果。

5-2 問題與解決

(一) 電路板操作及元件腳位大小：

一開始我們對電路版的製作非常的不熟悉，以至於我們在洗電路板這部分花了許久的時間來製作，尤其是設計部分，我們是使用 Autodesk EAGLE 來設計整體的電路，剛開始還不太熟悉這個軟體的介面，花了許久才了解。並洗了三次才成功，中間因為電路設計的地方沒對齊，才導致重洗的結果。

(二) 程式功能不明確：

本專題主要是圍繞在 PID 的原理上，其中包還了一些物理相關的東西，對於我們來說十分陌生，編寫程式時，還要去了解 PID 的內容。在 PID 中有個東西叫「卡爾曼濾波」，這部分有些深奧，於是就暫時不作進一步的討論。

5-3 建議

函式庫是個好東西，本專題對於函式庫的運用比較少，因此整體程式看起來較為複雜，假如把某些重複性較高的內容以函式庫的方式來撰寫，將會省時省空間。另外在於PID的部分，需要多次實驗，找到最穩定的參數，會讓結果更加順暢。

硬體的方面，雖然考慮到將電池和放置於底部能增加整體穩定性，但在爬坡以及衝刺時，電池盒會與地面磨擦，導致車子減速或翻覆，如果將來有機會繼續研究，可以將電池盒的位置往整體上方移動，讓行走時更加順利。並且在採購零件時，應事先確認好需求與規格，不然後面遇到問題時，才不會因為硬體方面而另外花時間去做採購。

參考文獻

- [1] 編碼器減速電機 AB 雙向直流馬達。亞博智能。
<https://item.taobao.com/item.htm?spm=a230r.1.14.32.296eea98iauQQv&id=550128802200&ns=1&abbucket=19#detail>
- [2] TB6612 馬達驅動 IC。
<https://swf.com.tw/?p=1066>
<https://www.playrobot.com/dc-/1246-motor-driver-dual-tb6612fng-1a.html>
- [3] Arduino MEGA2560 板。
<http://taiwanarduino.blogspot.com/2014/08/arduino-mega-2560.html>
- [4] HC-05 模組。
<https://www.taiwaniot.com.tw/product/hc-05-%E4%B8%B2%E5%8F%A3%E8%97%8D%E7%89%99%E6%A8%A1%E5%A1%8A-%E5%B8%B6%E5%BA%95%E6%9D%BF-%E4%B8%BB%E6%A9%9F%E5%BE%9E%E6%A9%9F%E5%8F%AF%E8%A8%AD%E5%AE%9A-%E9%99%84-arduino-%E7%AF%84%E4%BE%8B/>
- [5] OLED 模組。
<https://www.dx.com/p/ssd1306-0-96-128-x-64-white-oled-screen-display-board-module-white-2066748#.XCc6OegzaUk>
- [6] HC-SR04 超音波。
<https://www.electroschematics.com/8902/hc-sr04-datasheet/>
- [7] 電池盒範例。
<http://www.100y.com.tw/product/58612.htm>

附錄

附錄一 設備清單

類別	設備、軟體名稱	應用說明
硬體	電腦	工作應用
硬體	雷射印表機	列印電路版圖
硬體	手機	遙控自走車、照相
軟體	Arduino1.6.9	控制 JY-901B、TB6612
軟體	Eagle autodesk	繪製電路板
軟體	AppInventor2	藍芽遙控_手機介面
軟體	PowerPoint2013	簡報報告製作
軟體	Processing3.3.5	繪製傾斜平面座標圖
軟體	Word2013	書面報告製作

圖 23 設備清單

附錄二 材料清單

類別名稱	材料名稱	單位	數量	應用說明
整合開發環境	MEGA2560	個	1	核心主體
積體電路 U1	JY-901B	只	1	感測器
積體電路 U2	TB-6612	只	1	馬達控制板
積體電路 U3	HC-05	只	1	藍芽模組
積體電路 U4	HC-SR04	只	1	音波感測器
電機材料	520 編碼器減速電機馬達	個	2	配合輪胎使用
電子材料	LEDbar	只	1	顯示傾角
電子材料	330Ω-1/4W 電阻	顆	10	保護 LEDbar
電子材料	Panasonic 3.7V 電池	顆	2	提供馬達電源
電子材料	單芯線	捆	1	導線
消耗材料	壓克力板	塊	3	車身
消耗材料	銅柱	根	8	架高電路板
消耗材料	輪胎	個	2	配合馬達使用
電子材料	OLED	個	1	資料顯示

表 8 材料清單

附錄三 程式碼

```

//*****
// 107-02-05
// 車輪軸:Y 軸，前進軸:正 X 軸
// pitch 是 X 軸對地面的夾角，roll 是 Y 軸對地平面的夾角
//*****
#include <Wire.h>
#include<JY901.h>
#include<FlexiTimer2.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
//*****
//OLED
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

#define NUMFLAKES 10
#define XPOS 0
#define YPOS 1
#define DELTAY 2

#define LOGO16_GLCD_HEIGHT 16
#define LOGO16_GLCD_WIDTH 16
//*****
//motor
int PWMA = 4;
int AIN1 = 23;
int AIN2 =22;
int STBY = 27;
int BIN1 = 24;
int BIN2 = 25;
int PWMB = 12;
//*****
//角度,速度,旋轉控制參數
double kp = 35 , ki, kd = 0.4;
double ksp = 7, ksi = 0.1058, ksd = 0.0;//7 0.1058

```

```

// double kp_turn = 5, ki_turn = 0.0, kd_turn = 0.26;
double kp_turn = 5, ki_turn = 0.0, kd_turn = 0.52;
//kp_turn 不能超過 9   kd_turn 不能超過 0.52
int speed_cnt;
int turn_cnt;
int pulse_left;
int pulse_right;
volatile int cnt_right;
volatile int cnt_left;
//*****
//JY901
double ax, ay, az;
double gyro_x, gyro_y, gyro_z;
double pitch, yaw, roll;
double angle0 = -1.00; //機械平衡角
double k1 = 0.05;      //一階濾波器對加速度的取值權重
double dt = 0.005;
double angle6;
double pitch_offset, roll_offset, yaw_offset;
double gyro_x_offset, gyro_y_offset, gyro_z_offset;
//*****
//angle_out
double angle_output;
double error_angle;
//*****
//speed_out
float  speeds;
float  speeds_filterold;
float  speeds_filter;
float  positions;
double speeds_output;
int    front; //BT 控制
int    back;  //BT 控制
//*****
//turn_out
int    spinonce;
float turn_speed;

```

```

float rotationratio = 0;
float turnout;
float turn_output;
int    turnmax, turnmin;
int    turnleft_flag, turnright_flag;
int    spinleft_flag, spinright_flag;
//*****
//pwm_out
double pwm_r = 0, pwm_l = 0;
//*****
//cnt_pulse
int r_pluse;
int l_pluse;
int stop_r;
int stop_l;
int flag1;
//*****
unsigned long lastT=millis();
//*****
//*****
void setup() {
    //=====
    /*display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);*/
    //=====
    pinMode(AIN1, OUTPUT);
    pinMode(AIN2, OUTPUT);
    pinMode(PWMA, OUTPUT);
    pinMode(STBY, OUTPUT);
    pinMode(BIN1, OUTPUT);
    pinMode(BIN2, OUTPUT);
    pinMode(PWMB, OUTPUT);
    //=====
    digitalWrite(AIN1, HIGH);
    digitalWrite(AIN2, HIGH);

```

```

digitalWrite(BIN1, HIGH);
digitalWrite(BIN2, HIGH);
digitalWrite(STBY, HIGH);
//=====
Serial1.begin(57600);
Serial.begin(9600);
delay(2000);
JY901.startIIC();
for (int i = 0; i < 50; i++) {
    pitch_offset = pitch_offset + JY901.getPitch();
    roll_offset = roll_offset + JY901.getRoll();
    gyro_x_offset = gyro_x_offset + JY901.getGyroX();
    gyro_y_offset = gyro_y_offset + JY901.getGyroY();
    gyro_z_offset = gyro_z_offset + JY901.getGyroZ();
    delay(20);
}
//取平均值
pitch_offset /= 50;
roll_offset /= 50;
gyro_x_offset /= 50;
gyro_y_offset /= 50;
gyro_z_offset /= 50;
delay(100);
FlexiTimer2::set(5, compute);
FlexiTimer2::start();
//=====
}
//*****
//*****
void loop() {
    attachInterrupt(0, cnt_l, FALLING);
    attachInterrupt(1, cnt_r, FALLING);
    blue_tooth();
    //OLED();
}
//*****
//*****

```



```

// 中斷定時 5ms 定時中斷 (角度是 PD 演算)
//*****
void compute() {
    sei();
    cnt_pluse();
    pitch = JY901.getPitch();
    roll = JY901.getRoll();
    yaw = JY901.getYaw();
    gyro_x = JY901.getGyroX();
    gyro_y = JY901.getGyroY();
    gyro_z = JY901.getGyroZ();
    //Serial1.println(pitch);
    //一階互補濾波器
    angle6 = k1 * (roll - roll_offset) + (1 - k1) * (angle6 + (gyro_x -
gyro_x_offset) * dt); // JY901B 安裝時 Y 軸對地面的夾角(姿態)
    //angle_output = kp * (pitch + angle0) + ki * error_angle + kd * gyro_y; //
gyro_y 是車輪軸的角速度
    angle_output = kp * (angle0 + pitch - pitch_offset) + kd * (gyro_y -
gyro_y_offset); // gyro_y 是車輪軸的角速度
    //angle_output = kp * (angle0 - pitch) + kd * gyro_y; //
test1:(angle0-pitch) 結果
    speed_cnt++;
// 50ms 進入速度 PI 控制迴圈
    if (speed_cnt >= 10) {
        speeds_out();
        speed_cnt = 0;
    }
    turn_cnt++;
    if (turn_cnt > 3)
{ // 10ms 進
入旋轉 PD 控制迴圈
        turn_out();
        turn_cnt = 0;
    }
    pwm_out();
}
//*****

```

```

//*****
//speeds_out
void speeds_out() {
    speeds = (pulse_left + pulse_right) * 1.0; // 車速脈衝值
    pulse_left = 0;
    pulse_right = 0;
    //一階互補濾波器
    speeds_filterold = speeds_filterold * 0.7;
    speeds_filter = speeds_filterold + speeds * 0.3;
    speeds_filterold = speeds_filter;

    positions = positions + speeds_filter + front + back;
    positions = constrain(positions, -6550, 6550); // 抗積分飽和 -6550 ~ 6550
    speeds_output = ksi * (0.00 - positions) + ksp * (0.00 - speeds_filter);

    // 速度環控制 PI
    if (flag1 == 1) {
        positions = 0;
    }
}
//*****
//*****
//turn_out
void turn_out() {
    spinonce = 0;
    turn_speed = 0;
    turn_output = 0;
    rotationratio = 0;
    if (turnleft_flag == 1 || turnright_flag == 1 || spinleft_flag == 1 || spinright_flag
== 1) {
        if (spinonce == 0) {
            turn_speed = pulse_right + pulse_left;
            spinonce++;
        }
        if (turn_speed < 0) {
            turn_speed = -turn_speed;
        }
    }
}

```

```

    if (turnleft_flag == 1 || turnright_flag == 1) {
        turnmax = 5;
        turnmin = -5;
    }
    if (spinleft_flag == 1 || spinright_flag == 1) {
        turnmax = 8;
        turnmin = -8;
    }
    rotationratio = 55 / turn_speed;
    if (rotationratio < 0.5) {
        rotationratio = 0.5;
    }
    if (rotationratio > 5) {
        rotationratio = 5;          //5
    }
    //Serial1.println(turn_speed);
}
else {
    rotationratio = 0.5;
    spinonce      = 0;
    turn_speed    = 0;
}

if (turnleft_flag == 1 || spinleft_flag == 1) {
    turnout += rotationratio;
}
else if (turnright_flag == 1 || spinright_flag == 1) {
    turnout -= rotationratio;
}
else turnout = 0;
if (turnout > turnmax) turnout = turnmax;
if (turnout < turnmin) turnout = turnmin;
turn_output = -turnout * kp_turn - (gyro_z - gyro_z_offset) * kd_turn;
//Serial1.println(turn_output);

}
//*****

```

```

//*****
//pwm_out
void pwm_out() {
  if (turnleft_flag == 1 || turnright_flag == 1 || spinleft_flag == 1 || spinright_flag
== 1) {
    pwm_r = -angle_output - speeds_output - turn_output;          // 右馬達
PWM 輸出值
    pwm_l = -angle_output - speeds_output + turn_output; // 左馬達 PWM 輸出
值
  }
  else {
    pwm_r = -angle_output - speeds_output;          // 右馬達 PWM 輸出值
    pwm_l = -angle_output - speeds_output; // 左馬達 PWM 輸出值
  }
  if (pwm_r > 255) pwm_r = 255;
  if (pwm_r < -255) pwm_r = -255;
  if (pwm_l > 255) pwm_l = 255;
  if (pwm_l < -255) pwm_l = -255;
  if (pitch > 30 || pitch < -30) { //傾斜角過大關閉馬達
    pwm_r = 0;
    pwm_l = 0;
    Serial1.println("error");
  }
  if (angle6 > 3 || angle6 < -3 && turnleft_flag == 0 && turnright_flag == 0 &&
spinleft_flag == 0 && spinright_flag == 0 && front == 0 && back == 0) {
    if (stop_l + stop_r > 1500 || stop_l + stop_r < -1500) {
      pwm_r = 0;
      pwm_l = 0;
      flag1 = 1;
    }
  }
  else {
    stop_l = 0;
    stop_r = 0;
    flag1 = 0;
  }
  // 右馬達正反轉判斷

```

```

if (pwm_r >= 0) {
    digitalWrite(AIN1, LOW);
    digitalWrite(AIN2, HIGH);
    analogWrite(PWMA, pwm_r);
}
else {
    digitalWrite(AIN1, HIGH);
    digitalWrite(AIN2, LOW );
    analogWrite(PWMA, -pwm_r);
}
// 左馬達正反轉判斷
if (pwm_l >= 0) {
    digitalWrite(BIN1, LOW);
    digitalWrite(BIN2, HIGH);
    analogWrite(PWMB, pwm_l*0.985);
}
else {
    digitalWrite(BIN1, HIGH);
    digitalWrite(BIN2, LOW);
    analogWrite(PWMB, -pwm_l*0.985);
}
}
//*****
//*****
void cnt_l() {
    cnt_left++;
}
//*****
//*****
void cnt_r() {
    cnt_right++;
}
//*****
//*****
//cnt_pluse
void cnt_pluse() {
    r_pluse    = cnt_right;
}

```

```

l_pluse    = cnt_left;
cnt_right = 0;
cnt_left  = 0;
if ((pwm_r < 0) && (pwm_l < 0)) {
    r_pluse = -r_pluse;
    l_pluse = -l_pluse;
}
else if ((pwm_r > 0) && (pwm_l > 0)) {
    r_pluse = r_pluse;
    l_pluse = l_pluse;
}
else if ((pwm_r < 0) && (pwm_l > 0)) {
    r_pluse = -r_pluse;
    l_pluse = l_pluse;
}
else if ((pwm_r > 0) && (pwm_l < 0)) {
    r_pluse = r_pluse;
    l_pluse = -l_pluse;
}
stop_r += r_pluse;
stop_l += l_pluse;

pulse_right += r_pluse;
pulse_left  += l_pluse;
}
/*void OLED(){
display.setCursor(0,0);
display.clearDisplay();
display.print("Acc=");
display.print(JY901.getAccX()); display.print(" ");
display.print(JY901.getAccY());display.print(" ");
display.println(JY901.getAccZ());display.println();

display.print("Gyro=");
display.print(JY901.getGyroX());display.print(" ");
display.print(JY901.getGyroY());display.print(" ");
display.println(JY901.getGyroZ());display.println();
}

```

```

display.print("Angle=");
display.print(JY901.getRoll());display.print(" ");
display.print(JY901.getPitch());display.print(" ");
display.println(JY901.getYaw());
display.display();
}*/
//*****
//*****
void blue_tooth() {
char bt_char;
if (Serial1.available()) {
bt_char = Serial1.read();
Serial.print("BT command = ");
Serial.println(bt_char);
switch (bt_char) {
case '0': front = 600;          break; //前進
case '1': back = -600;         break; //後退
case '2': turnleft_flag  = 1; break; //左轉
case '3': turnright_flag = 1; break; //右轉
case '4': spinleft_flag  = 1; break; //左旋轉
case '5': spinright_flag = 1; break; //右旋轉
case '6': turnleft_flag  = 0; turnright_flag = 0; spinleft_flag  = 0;
spinright_flag = 0; back = 0; front = 0;          break; // 按鍵鬆開停車
case '7': turnleft_flag  = 0; turnright_flag = 0; spinleft_flag  = 0;
spinright_flag = 0; back = 0; front = 0;          break; // 按鍵鬆開停車
case '8': turnleft_flag  = 0; turnright_flag = 0; spinleft_flag  = 0;
spinright_flag = 0; back = 0; front = 0; turn_output = 0; break; // 按鍵鬆開停車
case '9': kp_turn += 1; Serial1.println(kp_turn); break;
case 'a': kp_turn -= 1; Serial1.println(kp_turn); break;
case 'b': kd_turn += 0.01; Serial1.println(kd_turn); break;
case 'c': kd_turn -= 0.01; Serial1.println(kd_turn); break;
default : break;
/*case '0': kp += 1; Serial1.println(kp); break;      //前進
case '1': kp -= 1; Serial1.println(kp); break;      //後退
case '2': kd += 0.01; Serial1.println(kd); break; //左轉
case '3': kd -= 0.01; Serial1.println(kd); break; //右轉

```

```

    case '4': kd += 0.1; Serial1.println(kd); break; //左旋轉
    case '5': kd -= 0.1; Serial1.println(kd); break; //右旋轉
    case '6': kp += 0.1; Serial1.println(kp); break; // 按鍵鬆開停車
    case '7': kp -= 0.1; Serial1.println(kp); break; // 按鍵鬆開停車
    case '8': ksp += 0.1; Serial1.println(ksp); break; // 按鍵鬆開停車
    case '9': ksp -= 0.1; Serial1.println(ksp); break;
    case 'a': ksi += 0.01; Serial1.println(ksi); break;
    case 'b': ksi -= 0.01; Serial1.println(ksi); break;
    default : break;*/
  }
}
}
//*****
//*****

```


成員簡歷

姓名	陳定言	班級	電子三甲	
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 程式設計(VisualBasic, VHDL, Arduino, C++) 5. 電腦繪圖 (Altium Designer) 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 蒐集資料 2. 理論探討 3. 計畫書製作 4. 外觀製作 5. 軟體規劃 6. 軟體設計與製作 7. 整體統整 8. 完成成品 9. 撰寫報告 10. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 105 學年工業電子丙級技術士 2. 105 學年擔任基電小老師 3. 105 學年擔任實習工場廠長 4. 106 學年簡版急救技能訓練 5. 106 學年參加全國技能競賽 應用電子 初賽 			

姓名	簡丞志	班級	電子三甲	
曾修習專業科目	1. 基本電學實習 2. 電子學實習 3. 數位邏輯實習 4. 程式設計(VisualBasic, VHDL, Arduino, C++) 5. 電腦繪圖 (Altium Designer)			
參與專題工作項目	1. 理論探討 2. 蒐集資料 3. 軟體與零件測試 4. 計畫書製作 5. 外觀製作 6. 軟體規劃 7. 軟體設計與製作 8. 整體統整 9. 完成成品 10. 撰寫報告			
經歷簡介	105 學年工業電子丙級技術士 105 學年擔任衛生股長 106 學年擔任副班長 106 學年參加北科大程式設計先修課程 107 學年簡版急救技能訓練			

姓名	陳奕均	班級	電子三甲	
曾修習 專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. 程式設計(VisualBasic, VHDL, Arduino, C++) 7. 電腦繪圖 (Altium Designer) 			
參與專題 工作項目	<ol style="list-style-type: none"> 1. 蒐集資料 2. 理論探討 3. 計畫書製作 4. 機構規劃 5. 外觀製作 6. 硬體規劃 7. 整體統整 8. 完成成品 9. 撰寫報告 10. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 105 學年工業電子丙級技術士 2. 107 學年擔任圖書股長 3. 107 學年簡版急救技能訓練 			

姓名	王世薰	班級	電子三甲	
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 程式設計(VisualBasic, VHDL, Arduino, C++) 5. 電腦繪圖 (Altium Designer) 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 蒐集資料 2. 理論探討 3. 計畫書製作 4. 外觀製作 5. 完成成品 6. 撰寫報告 7. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 105 學年工業電子丙級技術士 2. 105 學年擔任實習工場之安全與衛生股長 3. 106 學年擔任輔導股長 4. 107 學年簡版 CPR 急救技能訓練 			