

全國高職學生 106 年度專題暨創意製作競賽「專題組」

作品說明書



群別：電機與電子群

作品名稱：無線藍芽電動滑板

關鍵詞：滑板、藍芽、直流無刷馬達

## 目錄

無線藍芽遙控電動滑板.....	1
壹、摘要.....	1
貳、研究動機.....	2
參、研究方法(過程).....	3
一、研究步驟.....	3
二、軟體介紹.....	3
(一) APP inventor.....	3
(二) ArduinoSoftware IDE.....	3
(三) Altium designer .....	4
三、硬體介紹.....	4
(一) Arduino .....	5
(二) 直流無刷馬達.....	5
(三) 29.4 伏特電池.....	7
(四) TB6612FNG 馬達驅動板 .....	7
(五) 直流無刷馬達驅動器.....	7
(六) HC-05 藍芽模組.....	8
(七) 直滑式電位器.....	8
四、程式.....	9
(一) APP Inventor.....	9
(二) Arduino Software IDE .....	10
五、電路接線.....	13
肆、研究結果.....	14
伍、討論.....	17
陸、結論.....	18
柒、參考資料及其他.....	20

## 圖目錄

圖 3-1 研究過程流程圖 .....	3
圖 3-2 Arduino UNO 實體圖 .....	5
圖 3-3 直流無刷馬達構造 .....	6
圖 3-4 直流無刷馬達構造 .....	6
圖 3-5 直流無刷馬達驅動器 .....	7
圖 3-6 直滑式電位器 .....	8
圖 3-7 分段切速 APP 程式及介面 .....	9
圖 3-8 無段變速 APP 程式及介面 .....	9
圖 3-7 Arduino 電路接線圖及佈線圖 .....	13
圖 3-8 硬體接線及實體接線 .....	13
圖 4-1 功能流程圖 .....	14
圖 4-2 手機控制畫面 .....	15
圖 4-3 遙控器 .....	16
圖 6-1 根據不同體重操作時的速度折線圖 .....	18
圖 6-2 成品圖 .....	19

## 表目錄

表 3-1 材料表 .....	4
表 3-2 電池規格及實品圖 .....	7
表 3-4 藍芽模組 .....	8
表 3-5 馬達控制程式 .....	10
表 3-6 遙控程式 .....	12
表 6-1 成品與市面上電動滑板比較 .....	18

# 無線藍芽電動滑板

## 壹、摘要

於現代交通方式不勝枚舉，人們除了講求速度及安全外，也漸漸要求方便且具特色的交通工具。為了這個目標我們決定將滑板加入馬達並利用藍芽去做控制，來實現既方便又省力的目的。

本文一開始在講解我們的研究步驟，並分別介紹零件的功能及規格，其中也有最常用的 Arduino 及藍芽模組的說明，包括如何接線等等的注意事項。軟體部分介紹的是 APP 及 Arduino，這篇介紹中讀者可以學習到 APP 如何與藍芽做連結，以及如何控制馬達的快慢，考慮到剛開始不知道如何下手的初學者，在最後也附上了馬達及藍芽的接線圖，讓人既快速又有效率地完成這個作品。

## 貳、研究動機

為了增加城市中移動的便利性，小型單人交通工具於近幾年開始發展，不需要與汽車、公車、摩托車擠在大馬路上，而且十分環保，及很多優點於一身，是未來交通發展的重心。

解決了便利問題，智能平衡車卻出現了重大的安全危機。根據國外網站的報導，平衡車的玩家有 38% 留下永久傷害。當單人交通工具作為一項新型的科技開始進入人們的生活，社會制度卻還來不及配合發展，也因此形成了平衡車的安全爭議。

對於時速過快、使用者的平衡感不夠以及不擅操作所帶來的安全問題，我們不希望完全以使用者的體感來控制，而是以搖桿去控制，再用 App 加以輔助，希望能改善電動滑板的安全問題

我們所學的專業知識裡，其中有個單元是直流無刷馬達，相較之前學到的有刷直流機的缺點，它沒有電刷磨損及噪音的問題，而且速度與轉矩有更大的控制範圍，而電動滑板需要的正是噪音跟速度控制。付諸所學，將以前的知識實際操作一遍。

## 參、研究方法(過程)

### 一、研究步驟

研究過程如圖 3-1 所示：

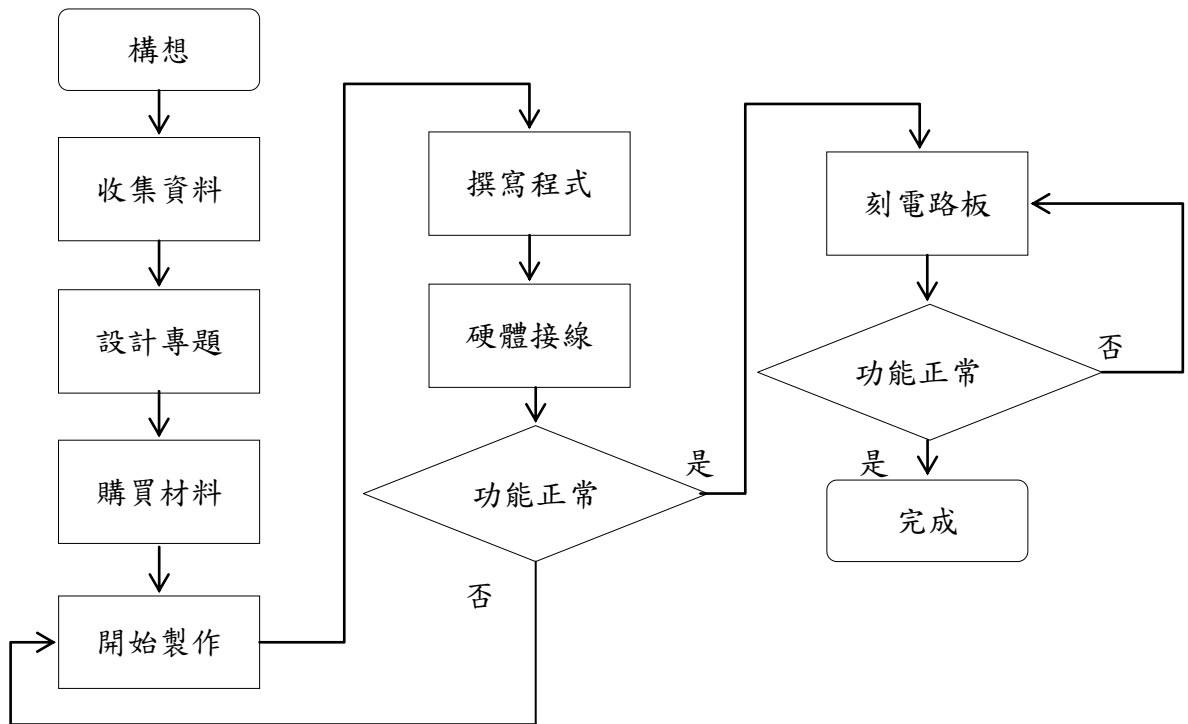


圖 3-1 研究過程流程圖

### 二、軟體介紹

#### (一)APP inventor

網站上提供的免費軟體，特點在於容易編寫，可利用拖放圖塊創造 Android 作業系統的應用軟體。類似 Scratch 語言界面。

#### (二)ArduinoSoftware IDE

它是被設計給不熟悉程式設計的人，使用的程式寫法和 C 語言相似，並且提供了常見軟體函式庫。C 語言需要自己去定義然後寫很多副程式來滿足動作需求，而 Arduino Software IDE 不需要，它只需要將寫好的程式打出來並設定即可使用。

### (三)Altium designer

容易使用的電路板程式，僅需要將電路圖和元件設定好便可開始佈線，把零件放在模擬板上即可自動佈線，不僅如此，它也可以進行 3D 零件模型模擬，進一步確認零件錯誤與否。

## 三、硬體介紹

本專題所使用的材料如下(如表 3-1 所示)

表 3-1 材料表

品名	數量	單價	運費	合計
單驅直流無刷馬達輪組	1	1400 元	60 元	3810 元
29.4V 鋰電池 LG 電芯+充電器	1	2000 元		
電池和控制板一體的外盒	1	350 元		
直流無刷馬達驅動器	1	590 元	60 元	650 元
TB6612FNG 馬達驅動板	1	56 元	70 元	126 元
HC-05 藍牙模組	2	118 元	60 元	476 元
推桿式可變電阻	1	150 元		
Arduino UNO	1	180 元	60 元	420 元
Arduino Micro	1	180 元		
9V 電池	4	30 元	0 元	120 元
合計				5602 元



### (一) Arduino

這是一款開程式碼的開發平台，常應用在製作專案中。由於它簡化過的控制硬體流程，可縮短專案設計的時間，花更多精力在建構創意。我們幾乎不需要了解其內部硬體結構和暫存器設定，僅僅知道它的接腳作用即可如圖 3-2。Arduino 平台具有完整的電子周邊模組，例如感應器、控制器或其他輸出裝置，也可以獨立運作成為一個可以跟軟體溝通的輸入裝置。。



圖 3-2 Arduino UNO 實體圖

### (二) 直流無刷馬達

直流無刷馬達的構造和一般直流馬達不同，如圖 3-3 所示，既沒有「碳刷」也沒有「整流子」之類的機械性接點，而是使用由電晶體等電子元件構成的電子迴路（驅動器迴路），透過電氣性的電流切換來使馬達轉動。

有點像是交流三相馬達一樣，必需由三條線圈，依序送入正負電源，及各相差 60 度角，一個循環共有 6 個步驟，所以又稱為六步方波，他的主要驅動線路如下圖表示，A、B、C 分別為馬達的三條電源線，由至少 6 個 MOSFET 做成的開關，控制 A、B、C 一點何時要接到 B+，何時要接到 B-，何時不導通，另外無感應式的無刷馬達，也靠 A、B、C 一點的回饋，讓控制器知道目前轉子的向位，以送出讓馬達轉動的相對應訊號。

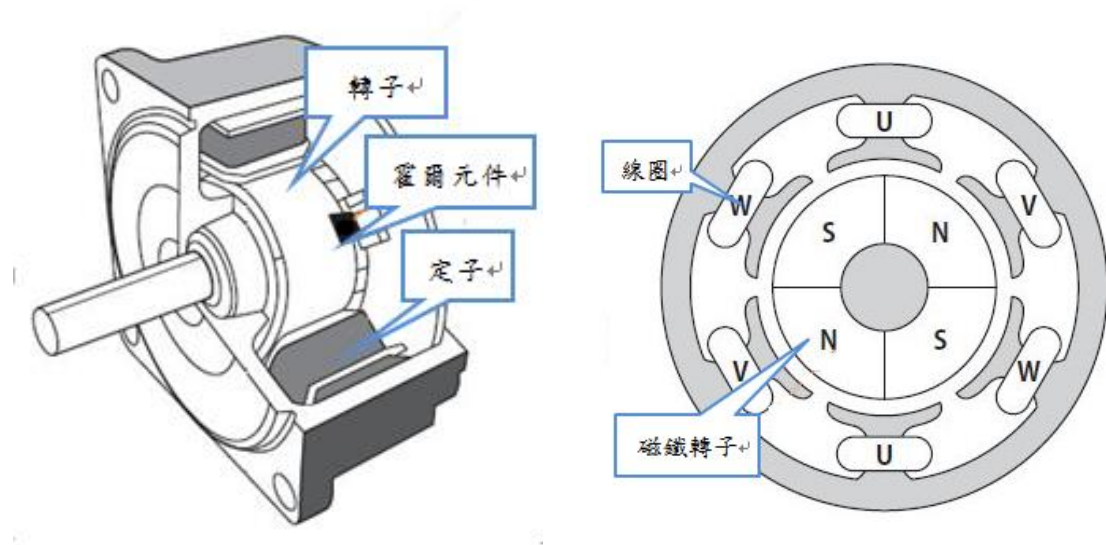


圖 3-3 直流無刷馬達構造

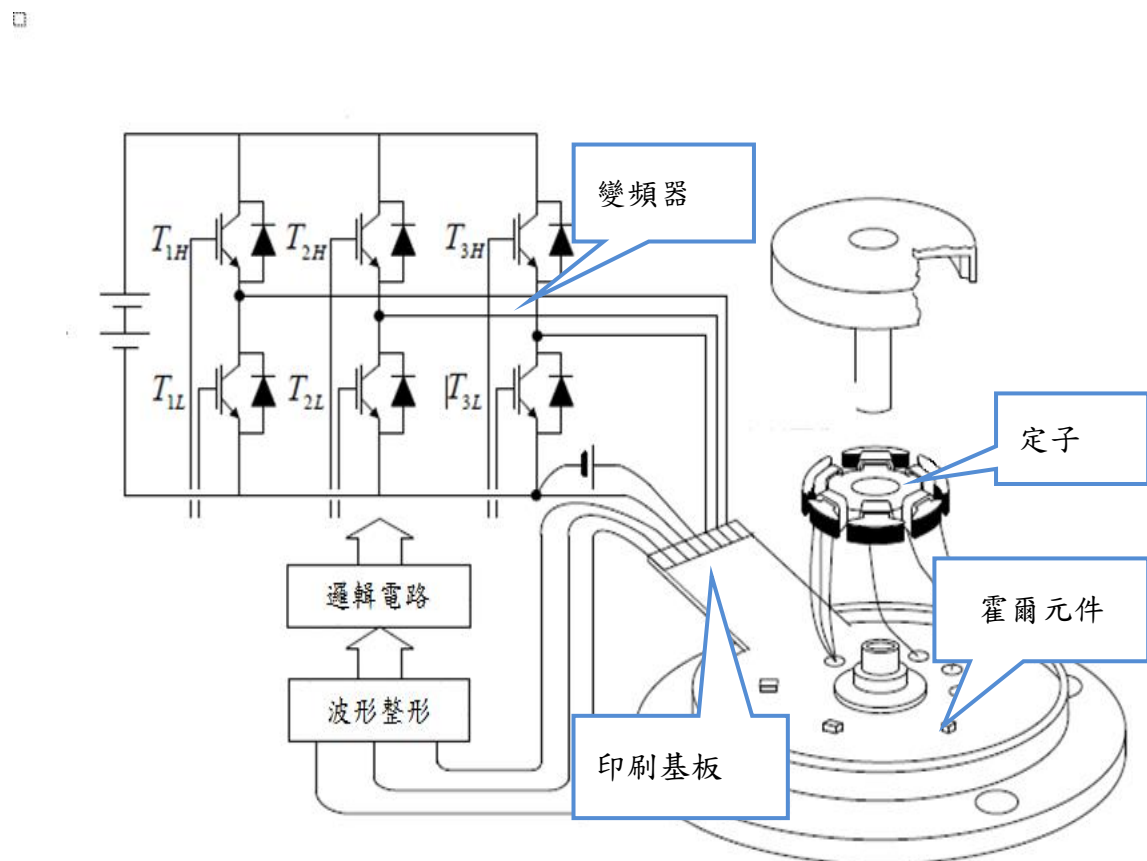



圖 3-4 直流無刷馬達構造

### (三) 29.4 伏特電池

需供給足夠電力，於是選擇較大容量的電池，詳細資料如表 3-2。

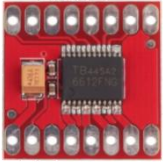
表 3-2 電池規格及實品圖

標準電壓	25.2V	
額定輸出	110.88Wh	
容量	4400mAh	

### (四) TB6612FNG 馬達驅動板

為了利用 Arduino 來控制馬達轉速，須利用驅動板來將 PWM 訊號轉換為類比電壓輸出訊號供給無刷馬達。我們使用如表 3-3 所示的 TB6612FNG，它更優於 L298N。原因是我們所需要的電壓值並不是太高，所以選擇使用 MOSFET 操作的 TB6612FNG 效率更高，驅動板規格。

表 3-3 驅動板規格及實品

電源電壓	15V	
輸出電流	1A	
最大輸出電流	3.2A	

### (五) 直流無刷馬達驅動器

透過內部電路可輸出 6 步方波控制直流無刷馬達轉動速度。

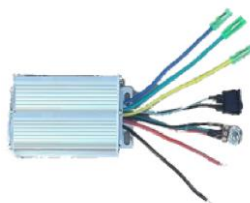
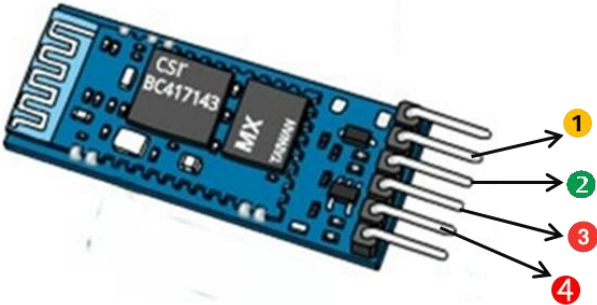


圖 3-5 直流無刷馬達驅動器

#### (六) HC-05 藍芽模組

HC-05 藍芽模組有 6 隻接腳，我們主要用到的接腳分別為 VCC、GND、TXD、RXD。如表 3-4 所示。

表 3-4 藍芽模組

1	VCC(正電源)	
2	GND(接地)	
3	TXD(傳送端)	
4	RXD(接收端)	

#### (七) 直滑式電位器

直滑式電位器最大電阻值為 10K 歐姆，將控制桿從一端滑到另一端，電位器兩端的電壓值將在 0 到 VCC 間變化。如圖 3-6 所示：



圖 3-6 直滑式電位器

## 四、程式

### (一) APP Inventor

起初我們使用分段變速的方式控制滑板，但我們發現在切換的瞬間滑板會震動且會影響使用者的平衡，於是最後改成滑桿式控速，便解決此問題。

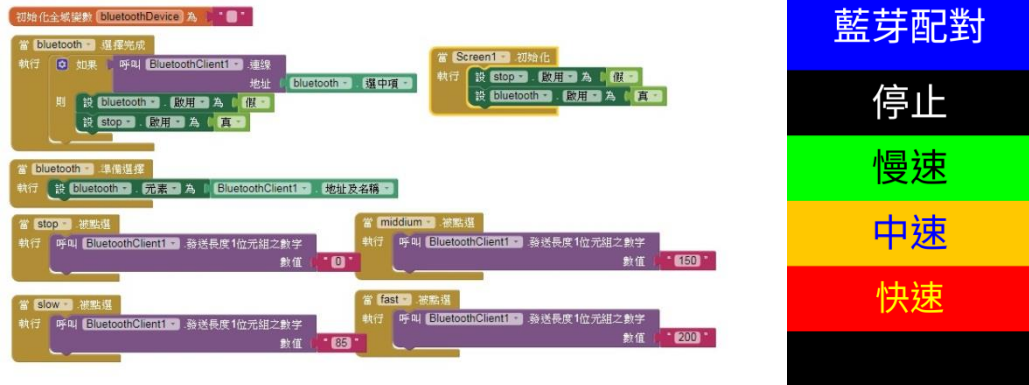


圖 3-7 分段切速 APP 程式及介面

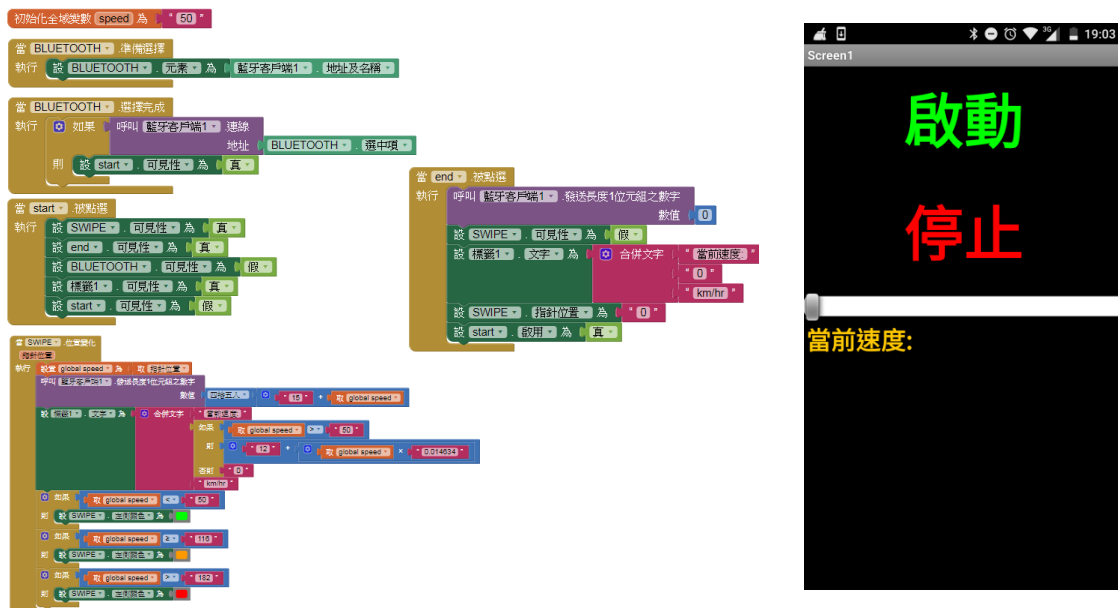


圖 3-8 無段變速 APP 程式及介面

## (二) Arduino Software IDE

表 3-5 為控制直流無刷馬達的程式，表 3-6 為遙控器的程式。

表 3-5 馬達控制程式

```
#include <SoftwareSerial.h>

//-----藍芽接收
SoftwareSerial BT(8,9);
char buffer [100];
unsigned char cControlByte;

//----- PWM 控制輸出
int dir1PinA = 13; // Arduino pin13 方向設定
int dir2PinA = 12; // Arduino pin12 方向設定
int speedPinA = 10; // Arduino pin10 PWM
int PWMout = 0;
int dir = 1;

void setup()
{
  Serial.begin(9600);
  BT.begin(9600);

  cControlByte = 0;
  pinMode(dir1PinA, OUTPUT);
  pinMode(dir2PinA, OUTPUT);
  pinMode(speedPinA, OUTPUT);
  PWMout = 0; //初始化速度
  dir = 1; //初始化方向
  if ( dir == 1 ) { //正轉
    digitalWrite(dir1PinA, HIGH);
    digitalWrite(dir2PinA, LOW);
  }
}

//=====
void loop()
{
```

```

if(BT.available()){
cControlByte = (unsigned char)BT.read();
PWMout = cControlByte;
//sprintf(buffer,"cControlByte=%d PWMout=%d", cControlByte,
PWMout);
//Serial.println(buffer);

if ( dir == 1 ) { //正轉
digitalWrite(dir1PinA, HIGH);
digitalWrite(dir2PinA, LOW);
//156~180
if ( PWMout< 50 ) {
PWMout = 0;
} else if ( PWMout>= 50 ) {
PWMout = map(PWMout,50,255,189,235);
}
analogWrite( speedPinA, PWMout);
sprintf(buffer,"cControlByte=%d PWMout=%d", cControlByte,
PWMout);
Serial.println(buffer);
}
}
}

```

表 3-6 遙控程式

```
#include <SoftwareSerial.h>
int xpotPin = 0;
int xval = 0;

char buffer [100];
SoftwareSerial BT(8,9);
unsigned char cControlByte;

void setup()
{
  Serial.begin(9600);
  //Serial1.begin(9600);
  BT.begin(9600);
}
//=====
void loop()
{
  xval = analogRead(xpotPin);

  cControlByte = (unsigned char)map(xval,0,1023,0,255);
  sprintf(buffer,"X=%d cControlByte=%d",xval, cControlByte);

  Serial.println(buffer);
  //Serial1.println(buffer);
  BT.write(cControlByte);
  delay(200);
}
```



## 五、電路接線

由於我們須將電路板在滑板下方，空間相當有限，因此我們將 Arduino 板需要的部分留著，重新刻一塊新的電路以縮小面積。

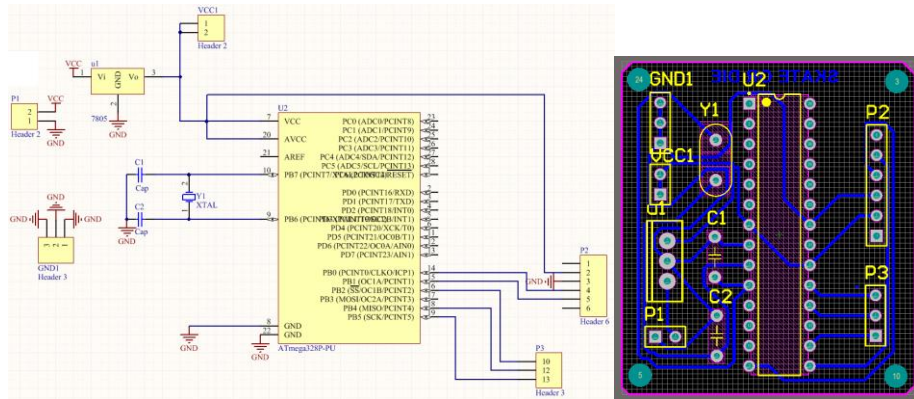


圖 3-7 Arduino 電路接線圖及佈線圖

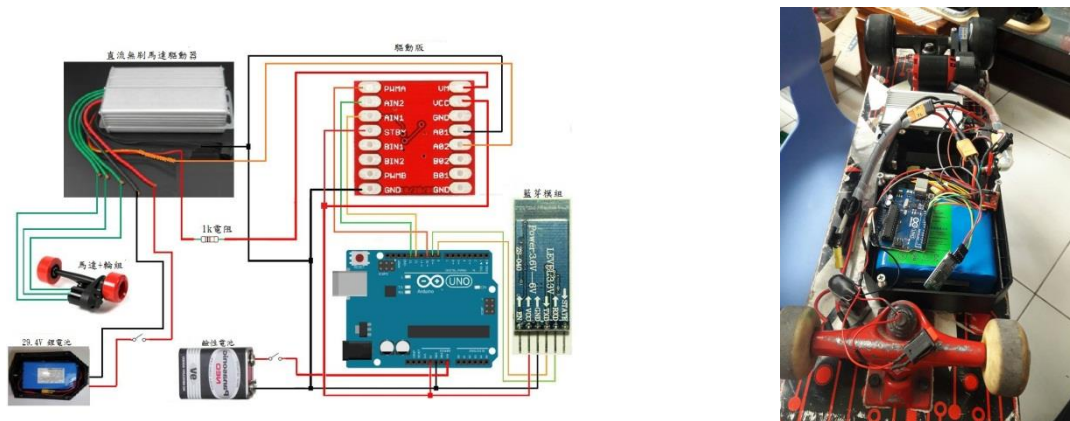


圖 3-8 硬體接線及實體接線

## 肆、研究結果

經過多次研究後，我們能夠成功使用遙控器或是手機控制滑板速度，以下是我們專題的功能介紹及流程圖。

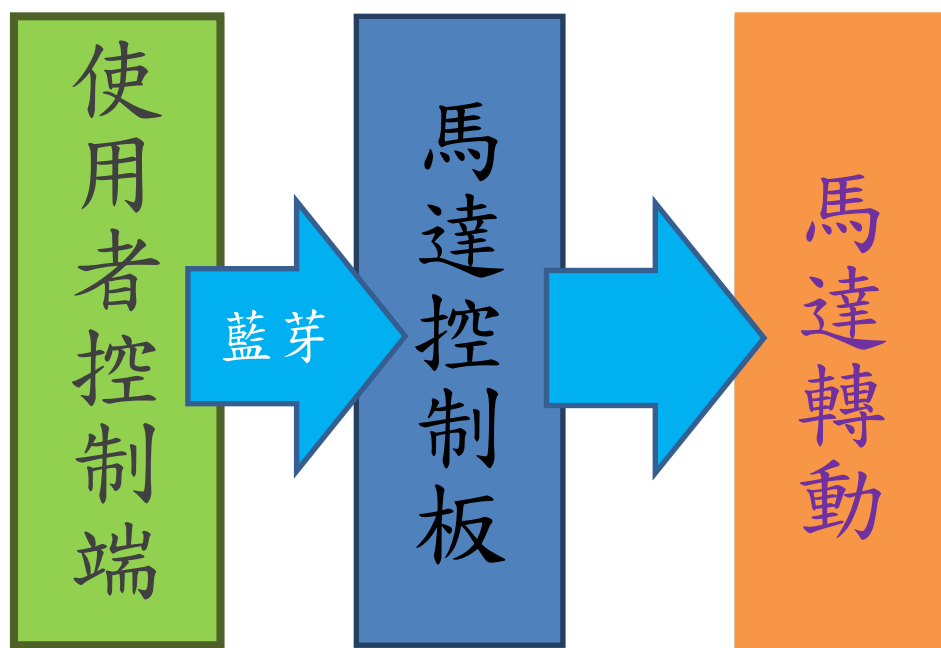


圖 4-1 功能流程圖

### 一、開機初始狀態

- (一) 滑板底部控制板 LED 燈閃爍
- (二) 連線後停止閃爍，常亮

## 二、開始控速

### (一)手機控制

- 1.打開馬達及控制板電源
- 2.手機開啓藍牙後進入 app 點選滑板連線進行配對
- 3.站穩踏板
- 4.按下啟動後出現滑桿
- 5.調速後滑桿下方會出現當前速度且滑桿顏色隨速度產生變化
- 6.按下停止後滑桿復歸，速度歸 0

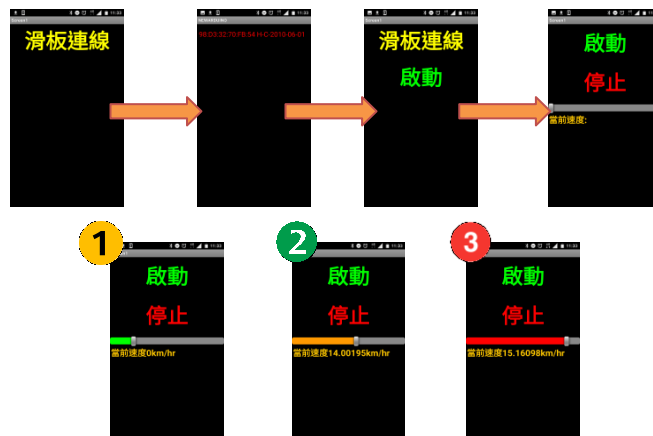


圖 4-2 手機控制畫面

## (二)遙控器控制

- 1.打開馬達及控制板電源
- 2.站穩踏板
- 3.使遙控器連線至滑板後即可開始調整滑桿位置加速
- 4.將滑桿回歸原位即停止



圖 4-3 遙控器

## 伍、討論

### 一、藍芽斷線導致無法控制

在 Arduino 做控制，當未接收到藍芽訊號時，馬達會自動斷電。

### 二、馬達運轉不順

剛開始 PWM 設定脈波週期錯誤，導致正常運轉範圍很小，經過一步步試驗及修正後，馬達已經可以理想運轉。

### 三、啟動時轉矩不足

由於真正在運作時並非無載，所以我們將啟動時的訊號提高，也就是最低速並非是 0，而是剛好在前進與不前進的臨界點，這樣一來只要稍微提高訊號就可以正常運作了。

### 四、Arduino 及驅動板接線雜亂

由於 Uno 板體積大，我們所用到的接腳以及零件只是其中的一小部分，因此我們用 Altium designer 自己刻劃電路板，解決體積問題。再將有接線部分移到電池盒內，使它看起來不過於凌亂。

### 五、負載過大造成電流超出額定

我們所使用的 29.4V 電池持續放電電流可以達到 15A，但為了安全起見，我們在電源測加裝了 10A 保險絲，萬一負載過重也不會造成電池或馬達的損壞。

## 陸、結論

目前市面上的電動滑板普遍偏貴，一般人通常會因為價格而不去選擇，我們的電動滑板成本大約 5000 元，市面上的售價大約是 7000-8000，我們做的其實更便宜、更人性化，且添加了手機遙控這個功能。

表 6-1 成品與市面上電動滑板比較

	無線藍芽遙控電動滑板	市面上電動滑板
價格	較便宜	稍貴 2~3000 元
穩定性	啟動稍不穩定	穩定
實用性	體積小，易攜帶	寬度稍寬，重量略重
難易度	速度慢，易上手	速度快，需多加練習方能上手
耐用性	兩腳踩踏間距小，不易斷裂	兩腳踩踏間距大，易斷裂

在控制上，我們把它精簡化，遙控器只需將推桿向前推便可讓馬達加速，手機控制的部分只要連線後按下啟動便可開始控速，按下停止即可漸漸停下；使用者只須在踏板上平衡，輕而易舉就能向前走。而為了安全，煞車並不是完全將馬達卡住，而是讓它逐漸停下，否則會因為慣性而跌倒；另一方面，我們在 UNO 板寫入了藍芽斷線便將馬達斷電的程式，以防使用者摔倒後滑板仍繼續往前；手機程式也有按下啟動後只顯示滑桿跟停止的功能以防按錯按鈕。

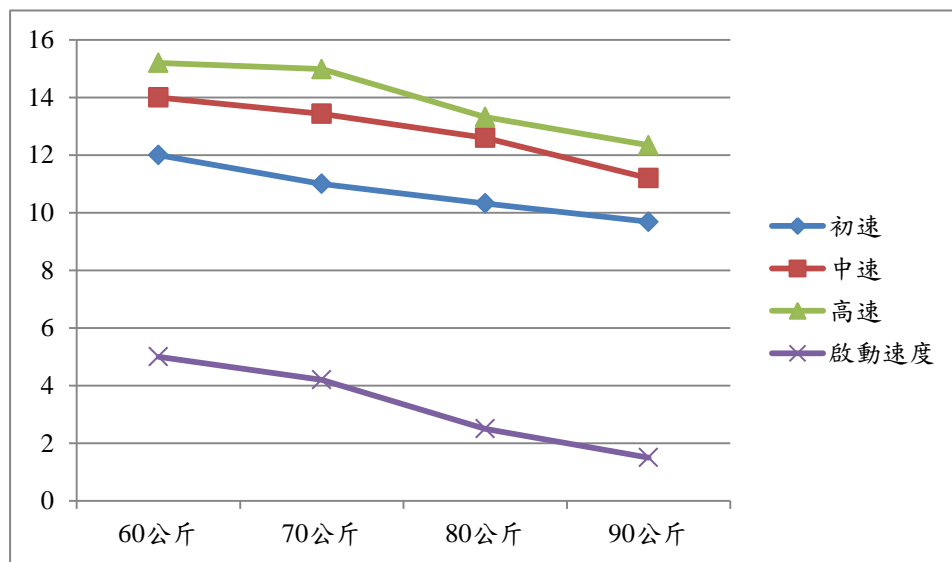


圖 6-1 根據不同體重操作時的速度折線圖

為了能做到每個人都能使用的電動滑板，未來我們希望能夠前方加上超音波感測器，以便在緊急時刻能夠及時反應並斷電；然後將馬達啟動時的不順修正，以防止啟動時重心的不穩。



圖 6-2 成品圖

## 柒、參考資料及其他

王安邦(2014)。MIT App Inventor 2 易學易用開發 Android 應用程式。台北市。上奇資訊。

張義和(2016)。新例說 ALTIUM DESIGNER 第二版。新北市。新文京開發出版股份有限公司。

趙英傑(2016)。超圖解 Arduino 互動設計入門第 3 版。台北市。旗標出版社。

民國 102 年 11 月 8 日，[雙 A 計劃] Part0：App Inventor 透過藍牙傳送訊號給 Arduino。取自:教育團隊技術部落

格:<http://blog.cavedu.com/programming-language/appinventor/appinventorandarduinowithbluetooth/>