

中華民國第 55 屆中小學科學展覽會

作品說明書

高中組 生活與應用科學科

佳作

040807

藍牙方程式賽車--繪圖車設計

學校名稱：國立新竹高級中學

| | |
|-------------------------|--------------|
| 作者： 高二 柯皓恩 高二 林 翔 | 指導老師： 曾聖超 |
|-------------------------|--------------|

關鍵詞：方程式、繪圖、藍牙控制

摘要

為增加數學課程的互動性，我們設計了一台可遙控的「方程式」繪圖車，可以依所輸入的座標來移動，精準的繪出所有的二次函數和其他圖形。

這台繪圖車裝備兩顆步進馬達，透過 Arduino 微控制器，指揮繪圖車的行進路徑。如果把所有圖形都看成是平面上點的連結，我們的構想是將座標系套用在平面上，而繪圖車在座標系裡從點到點之間行進的過程中，車上的筆就可以描出不同的圖形。我們自己寫了一個 App，讓使用者可以用手機與繪圖車連線，增加了無線控制的方便性。此一裝置，可讓教學者只需拿起手機，就可以輕鬆又準確的呈現各種二次函數圖形，而學習者也可以藉由操控繪圖車，更清楚掌握數學函數的原理。此種繪圖車的原理，可以廣泛應用到各種圖形的輸出與展示。

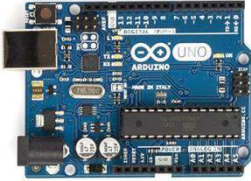


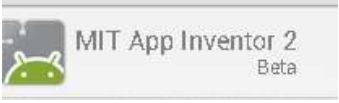


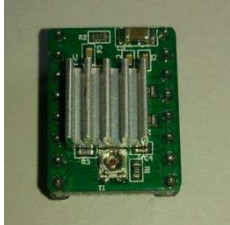

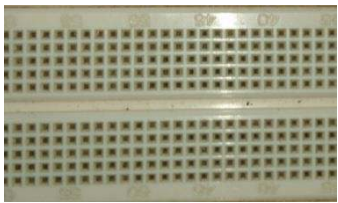
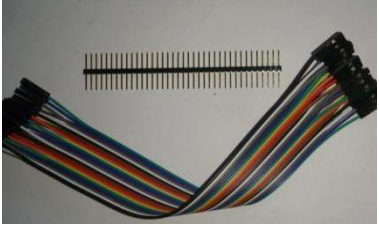


壹、研究動機

在上數學課時，我們發現函數這個概念非常抽象，要透過圖形來理解。許多數學老師花很多時間在黑板上用粉筆畫出各種函數圖形，但擦擦改改，還是很難呈現出標準的圖形。不準確的圖形，很可能會造成許多同學觀念混淆或是無法理解函數的概念。為了改善這個狀況，我們想到可以設計一台方便操控的繪圖車，來執行這個教學現場的任務。希望這個裝置可以在不同材質的平面上行進，如同一個點在座標平面上精準移動，並且表現出函數一個 X 值可以對應到一個 Y 值的概念。希望這個方法，也能用在各種圖形的繪製上。

貳、研究目的

- 一、用 Arduino 製作出一台能夠在座標系上精準行走的繪圖車。
- 二、使繪圖車能夠畫出二次函數。
- 三、自行開發 App 並在繪圖車上加裝藍牙模組，使我們可以藉由藍牙遙控繪圖車。
- 四、探討此一繪圖車的誤差，及減低誤差的方法。

參、研究設備及器材

| | | | |
|----------------------------|---|----------------------------------|---|
| <p>Arduino-Uno</p> |  <p>取自官方網站</p> | <p>Arduino IDE 開發程式</p> |  |
| <p>42 型步進馬達</p> |  | <p>App Inventor2(安卓 APP 開發器)</p> |  |
| <p>藍牙模組 HC-06</p> |  | <p>7.4V 鋰電池 12V 電瓶</p> |  |
| <p>步進馬達驅動器 (A4988)</p> |  | <p>變壓電路板 (轉 5V)</p> |  |
| <p>麵包版</p> |  | <p>電線與針</p> |  |
| <p>個人電腦、 筆記型電腦</p> |  | <p>Android 手機</p> |  |

肆、研究過程及方法

為了要達成我們的研究目的，我們著手開始設計車體，而在設計的過程中，我們首先必須篩選出一種精準的馬達，編寫程式碼來控制馬達運轉。為了提高使用者操控的方便性，增加了手機藍芽遙控。

一、使用步進馬達的原因：

我們為了要驅動繪圖車，曾經嘗試過許多其他種馬達，試圖找出最佳的機械組合。一開始我們使用小型 DC 馬達加上齒輪組，再在上面接上自製的旋轉編碼器來定位，此組合的優點為體積小且較便宜，但是在驅動的過程中，我們發現直流馬達轉速過於不穩定，因此不符合我們的要求。

我們所試的第二種馬達，是破解過的伺服馬達，加上自製的旋轉編碼器。這種組合可以結合伺服馬達的高扭力和準確性，而經過破解後的伺服馬達可以連續轉動，而且可以讓我們調整它的速度。但是在實驗的過程當中，我們發現伺服馬達在破解過後會變得不穩定，而且反應速度過慢，同樣不符合我們的要求。

最後我們選中了步進馬達，因為步進馬達透過無數顆小型的電磁鐵，可以將一圈分割成無數「步」，使我們可以將此裝置的精確度分割至單位小於毫米的等級。由於我們選擇的步進馬達驅動器可以透過 PWM 將一步分割成十六部分，使裝置的精確度更加提升。為了控制步進馬達，我們也自己寫了一些函數。

二、使用 Arduino 控制：

因為我們的目標是讓繪圖車能在座標上依照我們要求的方向行進，所以需要透過我們自己寫的程式來驅動，為此我們選用 Arduino 單晶片微控制器。Arduino 讓我們可以直接以熟悉的 C 語言寫程式，再加上他有已經架好的燒錄系統，大幅增加了實驗的方便性，也降低使用上的難度。

三、使用藍牙連接

在實驗的初期，我們都必須將 Arduino 透過傳輸線連接來控制繪圖車。傳輸線雖然直接又穩定，但是牽制了繪圖車運作的範圍。為了增加使用者的方便性，我們決定用藍牙控制，而經過了一些資歷搜查，我們在網路上看到了一篇介紹如何以 Android 手機連接 Arduino 的文章。(參考資料二)

在文章中介紹了如何使用 App Inventor 2 來自己寫出一個 App，並用這個 App 和在 Arduino 上連接的 HC-06 藍牙模組傳送"hello world"到 Arduino 上並在電腦上顯示。我們以這篇文章為基礎，反覆實驗並修改，寫出了我們自己的 App 能讓使用者選擇功能和輸入數值，並將使用者輸入的資料傳至 Arduino，使繪圖車能夠照著使用者的想法來運行。

伍、研究成果

一、Arduino 控制碼

```
1/* Astep 函數                                     }
    使 A 輪走一格*/                                3/* ABstep 函數
                                                    使 A 輪 B 輪 一起各走一格*/
void Astep(){                                       void ABstep(){
    digitalWrite(A_Step,HIGH);                       digitalWrite(A_Step,HIGH);
    digitalWrite(A_Step,LOW);                         digitalWrite(B_Step,HIGH);
    delayMicroseconds(N);                             digitalWrite(A_Step,LOW);
}                                                    digitalWrite(B_Step,LOW);
*****                                             delayMicroseconds(N);
2/* Bstep 函數                                     delayMicroseconds(N);
    使 B 輪走一格*/                                }
void Bstep(){                                       }
    digitalWrite(B_Step,HIGH);
    digitalWrite(B_Step,LOW);
    delayMicroseconds(N);
```

4/* fr 函數

如果傳入的數值為正值 代表往前走

把 A 輪方向調成順時針(HIGH)

把 B 輪方向調成逆時針(LOW)

之後走出所需的步數 呼叫 ABstep 函數

反之 則都相反即是*/

```
void fr(int t){
    if(t>0){
        digitalWrite(A_Dir,HIGH);
        digitalWrite(B_Dir,LOW);
        for(int i=0;i<t;i++){
            ABstep();
        }
    }
    else{
        t*=-1;
        digitalWrite(A_Dir,LOW);
        digitalWrite(B_Dir,HIGH);
        for(int i=0;i<t;i++){
            ABstep();
        }
    }
}
```

5/* tu 函數

如果傳入的數值為正值代表往右轉

把 A 輪方向調成逆時針(LOW)

把 B 輪方向調成逆時針(LOW)

之後走出所需的步數，呼叫 ABstep 函數

反之 則都相反即是*/

```
void tu(int t){
    if(t>0){
        digitalWrite(A_Dir,LOW);
        digitalWrite(B_Dir,LOW);
        for(int i=0;i<t;i++){
            ABstep();
        }
    }
    else{
        digitalWrite(A_Dir,HIGH);
        digitalWrite(B_Dir,HIGH);
        t*=-1;
        for(int i=0;i<t;i++){
            ABstep();
        }
    }
}
```

6/* atanEX 函數

使用內建函數 atan2(傳回值為 $-2\pi\sim 2\pi$)，最後傳回值換成 0~360 度*/

```
double atanEX(double Y,double X){
    double tmp=atan2(Y,X)*180/PI;
    return tmp<0 ? tmp+360 : tmp
}
```

7/* TurnTo 函數

先算出要順時針轉還是逆時針轉。如果順時針轉的角度大於 180 度則逆時針轉，反之則順時針因為無法轉出所要的完全準確的角度，因此我們算出最為接近的角度，把能轉的角度轉換成步進馬達的格數傳入函數 tu 函數，最後更新目前角度*/

```
void TurnTo(double next){
    double tmp=nDir-next;
    while(tmp<0) tmp+=360;
    if(abs(tmp)>180){
        if(tmp<0) tmp+=360.0;
        else tmp-=360.0;
    }
    int actual_step=(int) (tmp/DEG);
    tu(actual_step);
    nDir-=(double)actual_step*DEG;
    if(nDir>360) nDir-=360;
    else if(nDir< -360) nDir+=360;
}
```

8/* FrontTo 函數

因為無法走出所要的完全準確距離，因此我們算出最接近的距離把能走出的距離傳入函數 fr 函數最後更新目前座標位置*/

```
void FrontTo(double distance,double dir){
    int actual_dis=(int) distance*stepfar;
    fr(actual_dis);
    double tmp=(actual_dis)/stepfar;
    nX+=cos(dir*PI/180)*tmp;
    nY+=sin(dir*PI/180)*tmp;
}
```

9/* toXY 函數

算出遠本的點到要達到的點所需的的角度 (呼叫函數 atanEX 函數)把角度傳到函數 TurnTo 函數算出原本的點和要到達的點的距離把距離值傳到函數 FrontTo 函數*/

```
void toXY(double X,double Y){
    double
    dist=sqrt((nX-X)*(nX-X)+(nY-Y)*(nY-Y));
    double tmp=atanEX(Y-nY,X-nX);
    TurnTo(tmp);
    delayMicroseconds(30000);
    FrontTo(dist,nDir);
}
```

10/* getval 函數

傳入完整得的 2 次函數 及 X 值,把 X 值帶入函數式
內,最後傳回所得的 Y 值*/

```
double getval(double i,double x2,double  
x1,double x0){  
    return i*i*x2+i*x1+x0;  
}
```

二、繪製各種圖形方式：

(一)、前進：

先呼叫 FrontTo()函數,將想要走的步數輸入此函數,此函數就會再呼叫 fr()函數並且更新最新位置。

(二)、原地旋轉：

呼叫 Tu()函數,輸入想要旋轉的步數。如果輸入的值為正,則順時針轉,若為負值則逆時針轉。

(三)、左右輪轉：

先判斷正負值來改變方向,接下來藉由呼叫 Astep()/Bstep()來使馬達轉動。

(四)、座標：

呼叫 toXY 函數,它就會走到我們想要的(X,Y)。

(五)、二次函數：

我們運用函數的基本原理,也就是帶入一個 X 值就會對應一個 Y 值,透過帶入無數個非常密集的 X,藉此畫出一個多項式。它運作的方式是把讀入的所有參數記起來之後從一開始

給的初始點 X 帶入所需函數，求出 座標(X,Y)，帶入 toXY 函數，之後 X 增加 0.5 個單位，再次丟入 toXY，重複上述步驟直到到達所需的末點 X。

三、藍牙控制

為了使我們的繪圖車能夠脫離電腦的限制，使他不必要依賴電腦和傳輸線來進行控制，我們決定要透過無線的方式來對他下達指令，而這時我們就想到了現今越來越普及的智慧型手機來進行遙控。為了達成此一目標，我們在繪圖車上加裝了 HC-06 藍牙控制模組，並用 App Inventor 來寫出控制繪圖車用的 App。

我們寫的 App 執行的工作，主要就是提供使用者介面來控制繪圖車繪出多邊形、圓形或是函數圖形，而方法如下。

```
when BTlist .AfterPicking
do
  if
    call BluetoothClient1 .Connect
      address BTlist . Selection
  then
    set BTlist . Enabled to false
    set Disconnect . Enabled to true
    set VerticalArrangement1 . Visible to true
```

首先我們的程式必須和 Arduino 建立連線和產生使用者介面，此 App 會顯示一個藍芽裝置表供使用者選擇並建立連線，

```

when Spinner1 .AfterSelecting
  selection
do
  call Notifier1 .ShowAlert
  notice get selection
  if
  then set global select to " 0 "
  else if
  then set global select to " 1 "
  else if
  then set global select to " 2 "
  else if
  then set global select to " 3 "
  else if
  then set global select to " 4 "
  open another screen screenName " Screen2 "
  else if
  then set global select to " 5 "

```

接下來我們的 App 會建立一個選單讓使用者可以選擇執行/原地轉/動左輪/動右輪/繪製二次函數/走到點等功能，並使使用者可以輸入值來控制繪圖車。

```

when send .Click
do
  set global btCommand to command . Text
  call BluetoothClient1 .SendText
  text get global select
  call BluetoothClient1 .SendText
  text get global btCommand
  call Notifier1 .ShowAlert
  notice " sent "

```

在選好功能並輸入完值之後，手機就會將指令傳至 Arduino。

四、誤差

繪圖車在繪圖的過程中，我們發現它會出現一些誤差，因此我們就進行了幾個實驗想要了解這些誤差。而由於我們的繪圖車的行進方式為直走和旋轉所組成的，因此我們決定就針對這兩個動作進行實驗。

(一)實驗一：原地旋轉

在本實驗中我們使用 tu 函數使車子旋轉 7000 步。選擇 7000 的原因，是因為它會使車子非常接近旋轉一圈，而我們在車子轉完之後，測量輪子的位置和我們在旋轉前所標記的位置相差多少毫米。而我們分別對正轉和反轉、輪子上油前和上油後各進行了五次實驗。

| | 左輪偏移量(mm) | 右輪偏移量(mm) | |
|---------|-----------|-----------|-----|
| 正轉 7000 | 4.5 | 3.5 | 0 |
| 上油前 | 5 | 3 | 1 |
| | 4.5 | 3 | 0 |
| | 4 | 4.5 | 0 |
| | 2 | 7 | 1.5 |

表格一：正轉 7000 步、上油前

| | 左輪偏移量(mm) | 右輪偏移量(mm) | 平移量 |
|---------|-----------|-----------|-----|
| 正轉 7000 | 4 | 3 | 0 |
| 上油後 | 3 | 4 | 0 |
| | 5 | 2 | 0.5 |
| | 3.5 | 4 | 1 |
| | 3.5 | 3.5 | 0.5 |

表格二：正轉 7000 步、上油後

| | 左輪偏移量(mm) | 右輪偏移量(mm) | 平移量 |
|---------|-----------|-----------|-----|
| 反轉 7000 | 4.5 | 2 | 3.5 |
| 上油前 | 2 | 4 | 1 |
| | 3 | 3 | 1.5 |
| | 5 | 2.5 | 3 |
| | 5 | 2 | 4 |

表格三：反轉 7000 步、上油前

| | 左輪偏移量(mm) | 右輪偏移量(mm) | 平移量 |
|---------|-----------|-----------|-----|
| 反轉 7000 | 4 | 3 | 0 |
| 上油後 | 5 | 2 | 1 |
| | 5 | 2 | 1 |
| | 5 | 3 | 1 |
| | 5.5 | 3 | 0 |

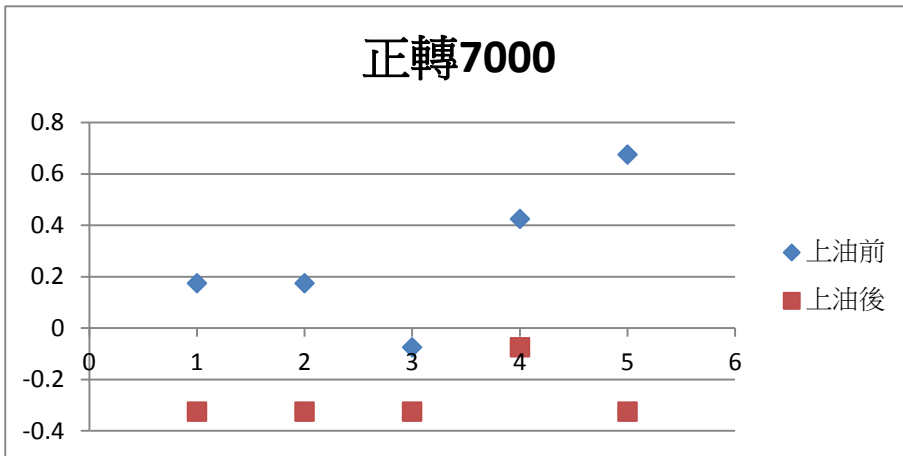
表格四：反轉 7000 不、上油後

我們算出正轉的平均偏移量為 3.8mm，而反轉的平均偏移量為 3.5mm，而為了比較上油前和上油後的偏移量，我們將每次實驗的偏移量減掉平均偏移量，並算出兩輪的平均。

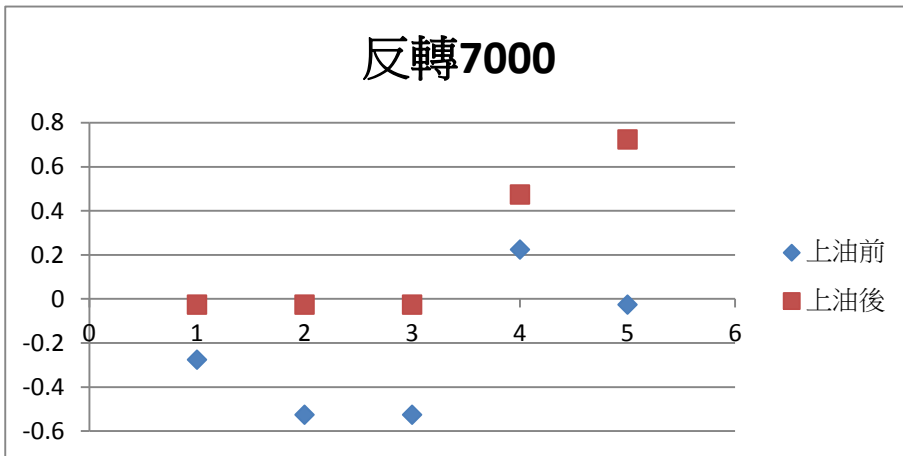
| 正轉 7000 | | | | 反轉 7000 | | | |
|---------|-----------|-----------|--------|---------|-----------|-----------|--------|
| 上油前 | 左輪偏移量(mm) | 右輪偏移量(mm) | 兩輪平均誤差 | 上油前 | 左輪偏移量(mm) | 右輪偏移量(mm) | 兩輪平均誤差 |
| | 0.675 | -0.325 | 0.175 | | 0.975 | -1.525 | -0.275 |
| | 1.175 | -0.825 | 0.175 | | -1.525 | 0.475 | -0.525 |
| | 0.675 | -0.825 | -0.075 | | -0.525 | -0.525 | -0.525 |
| | 0.175 | 0.675 | 0.425 | | 1.475 | -1.025 | 0.225 |
| | -1.825 | 3.175 | 0.675 | | 1.475 | -1.525 | -0.025 |
| 上油後 | 左輪偏移量(mm) | 右輪偏移量(mm) | 兩輪平均誤差 | 上油後 | 左輪偏移量(mm) | 右輪偏移量(mm) | 兩輪平均誤差 |
| | 0.175 | -0.825 | -0.325 | | 0.475 | -0.525 | -0.025 |
| | -0.825 | 0.175 | -0.325 | | 1.475 | -1.525 | -0.025 |
| | 1.175 | -1.825 | -0.325 | | 1.475 | -1.525 | -0.025 |
| | -0.325 | 0.175 | -0.075 | | 1.475 | -0.525 | 0.475 |
| | -0.325 | -0.325 | -0.325 | | 1.975 | -0.525 | 0.725 |

表格五

為了使差異清楚呈現，我們分別取兩輪平均誤差上油前和上油後的數據製成圖表。



圖表一：正轉 7000 步上油前和上油後比較



圖表二：反轉 7000 步上油前和上油後比較

由圖表一和圖表二，我們可以看出上油後轉的長度，較上油前正轉時平均多了 0.55mm，反轉時平均多了 0.45mm，很明顯的上油後轉得較多。而平移量在上油前後都相差不大。

(二)實驗二：直行

本實驗為控制繪圖車直走 1000 步，並測量最終車輪位置和最初的位置差了多少毫米，重複進行五次實驗，得出數據如下：

| | | | | | | |
|-----------|----|------|----|------|------|----------|
| 向前 1000 步 | | | | | | |
| 位移量(mm) | 78 | 77.8 | 79 | 77 | 77.4 | 76.4 |
| | | | 平均 | 77.6 | 標準差 | 0.894427 |

表格六：直行 1000 步位移量

由實驗數據當中可以看出直行 1000 步平均長度為 77.6mm 也就是平均一步長度為 77.6 微米，而我們實驗所得出的標準差為 0.894427。

(三)實驗三：加重與快慢

我們使用 tu 函數使車子正轉 1000 步再逆轉 1000 步重複 8 次，重複進行五次實驗，得出數據如下：

| | (1)加重，每步 800 MS 偏移量(mm) | (2)加重，每步 200MS 偏移量(mm) | (3)未加重，每步 800MS 偏移量(mm) |
|----|----------------------------|---------------------------|----------------------------|
| 1 | 0 | 25 | 11 |
| 2 | 0 | 29 | 8 |
| 3 | 3 | 23 | 9 |
| 4 | 0 | 13 | 13.5 |
| 5 | 2 | 44 | 13.5 |
| 平均 | 1 | 26.8 | 11 |

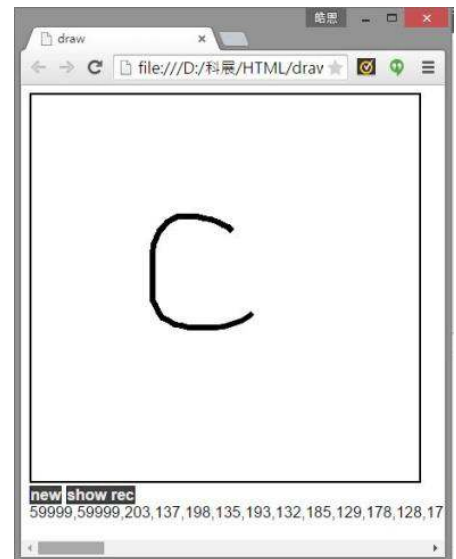
表格七：(順轉 1000 步逆轉 1000 步)*8

由(1)和(2)可以明顯的看出速度快會使在正轉逆轉交換時產生的振動比較大使誤差變大，而由(1)和(3)可以看出未加重會使在正轉逆轉交換時產生的振動比較大使誤差變大。由此實驗我們可以知道透過加重和放慢速度可以明顯減少繪圖車的誤差。

五、手繪功能

透過前面的研究過程，我們有效的減少了誤差並可以讓車子精確的走至我們所指定的座標，而我們就想利用這個能力更進一步，讓他繪圖的能力不限於只有函數圖形，因此我們就新增了手繪功能。而畫圖的方式有兩種：

(一)、網頁：我們用 Javascript 寫了一個網頁(如右圖)，使用者在上面畫完他所要的圖形後，按下 show rec 之後就會跑出這個圖形所經過的點座標，接下來再把這些數字複製，輸入到 Arduino 當中，Arduino 會把記錄存成 TXT 檔讓使用者可以隨時將這個圖形畫出來。

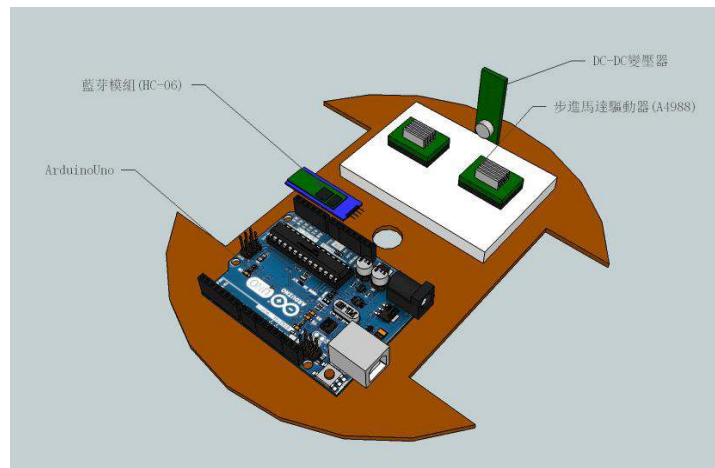


(二)、APP：雖然用我們寫的網頁已經能夠讓繪圖車畫出我們畫的圖形了，但是我們認為這樣太麻煩了，因此我們決定改寫我們的 APP，讓使用者可以直接在 APP 裡面完成所有操作。在更新後的 APP 當中使用者只需要到繪圖頁面當中畫完他要的圖形，按下 send 之後，手機就會自動將紀錄透過藍牙傳到 Arduino 中讓使用者可以隨時輸出。

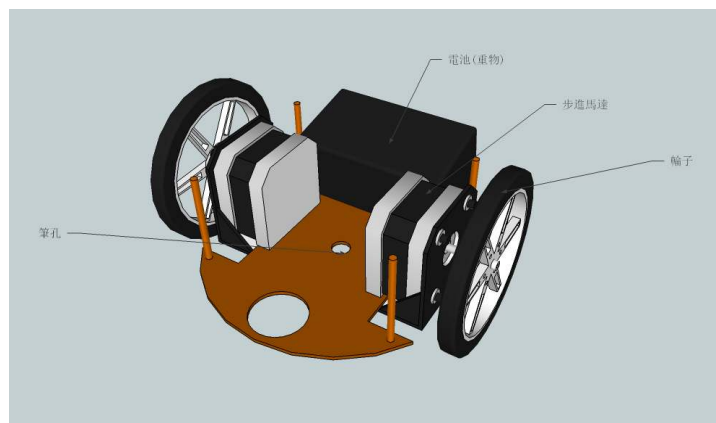


透過新增手繪的功能我們使繪圖車的方便性增加了，除了畫函數之外使用者也可以讓繪圖車描出任何他畫得出來的圖形。

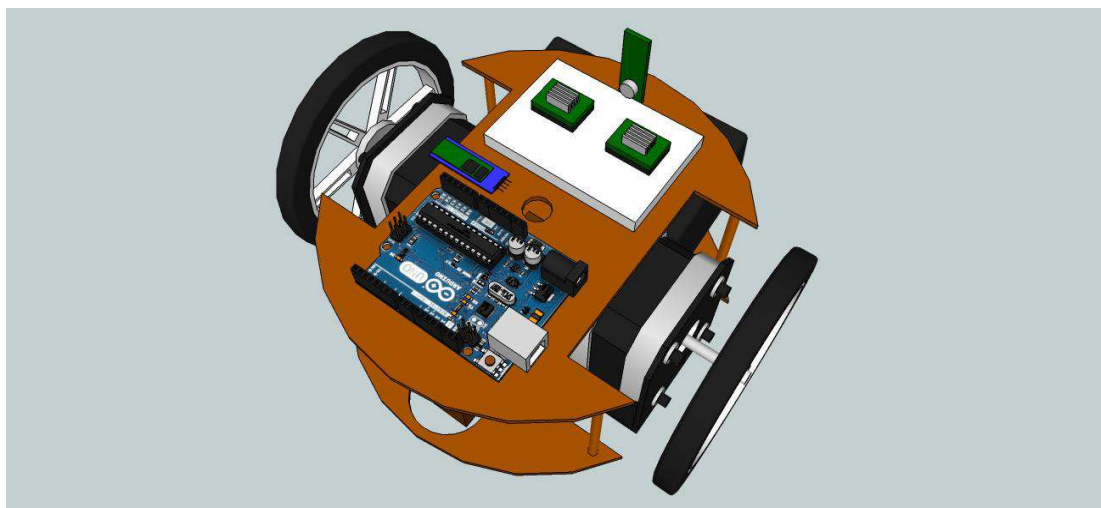
六、車體設計



在繪圖車的第一層主要是負責控制的裝置，包括 Arduino、步進馬達驅動器，也放了負責和手機連線的藍牙模組(HC-06)，和負責供應電力的 DC-DC 變壓器，而中間的圓孔是我們將筆插下去的位置，它的位置一定要在步進馬達軸的連線上面，如此在轉彎時筆才會在原地。

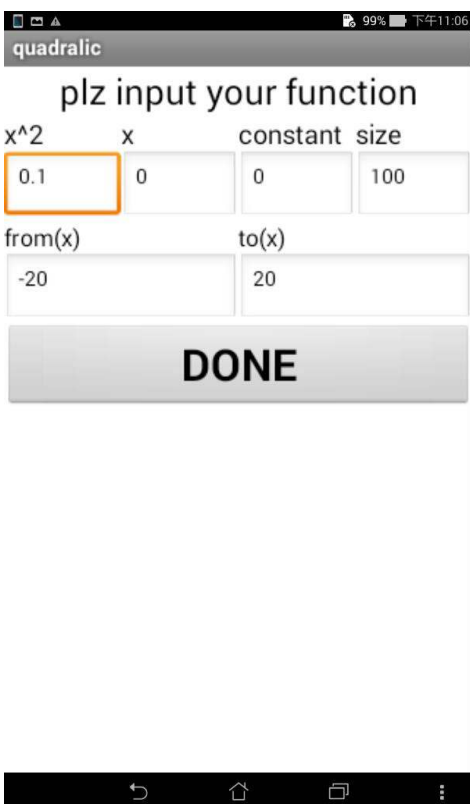
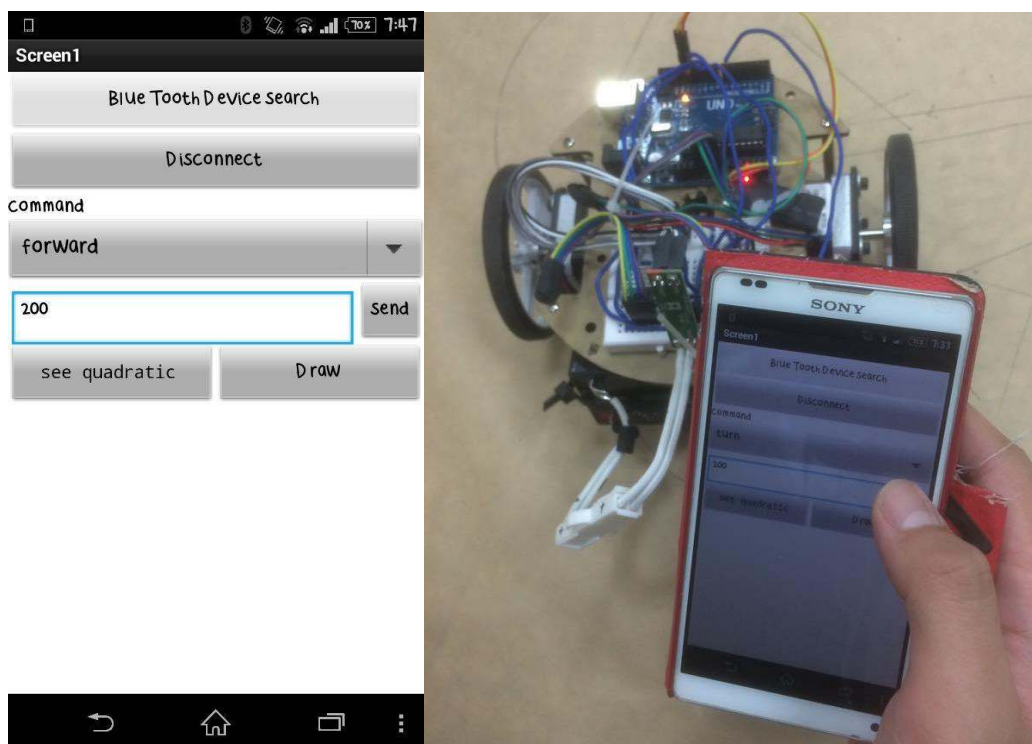


繪圖車的第二層是負責提供動力的，包括一個電池和兩顆步進馬達。



七、最終結果

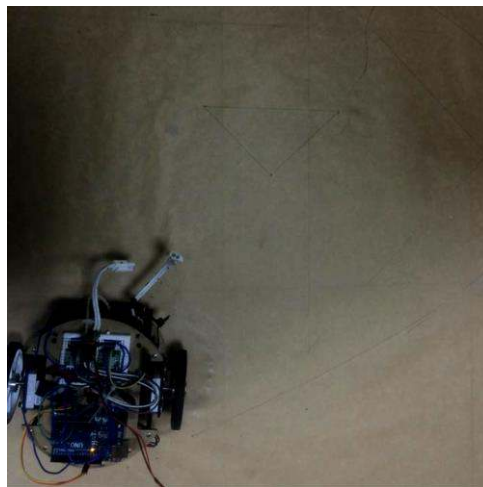
(一)自行撰寫 App 程式透過藍牙控制繪圖車



左圖為輸入二次函數的畫面

使用者可以輸入二次函數的各項係數、圖形大小和繪圖範圍，再將此二次函數傳至繪圖車，就可以繪製出任何二次函數。

(二)、我們可以使繪圖車畫出二次函數和圓形



(三)、我們可以使繪圖車走到我們指定的任意點(圖中為輸入三點使其走出一三角形)

陸、討論

一、誤差來源

在繪圖時，我們發現執行一段時間後，會出現一定大小的誤差，因此我們進行討論試圖找出誤差的來源如以下：

(一)、計算誤差

一開始我們在繪圖車每跑一格時，直接當作它已經跑到我們預設的那個點了，但是實際上因為步進馬達每次最小會轉一步，造成理論和時機有所偏離，也使繪圖車在跑的時候會有累進誤差。為了解決此誤差，我們在程式當中加入了 nX , nY , $nDir$ 分別用來記錄現在實際上的座標和角度，藉此避免累進誤差。

(二)、輪子

由於現實世界當中的輪子和地面接觸並非只有一個點，而是一個面，這就會使繪圖車每旋轉一次就會多出一次無法估計的誤差。

(三)、震動

由於繪圖車跑路徑的方式為直行-轉彎-直行-轉彎，因此在轉彎時，靠內側的論子會不停的前進-後退-前進-後退，當切換頻率過快時，就會造成車體震動，而這個震動也會造成無法估計的誤差。目前我們所採取的解決方式有：

1.在車上加裝重物，藉由重物穩定車身，降低震動。

2.降低速度，降低速度就會降低加速度，使車子沒有那麼容易震動。

但是我們認為以上方法並不能夠真正有效的根除震動的問題，因此目前在研究第三種解決方式：

3.換一種轉彎方式，由於早成震動的原因主要是轉彎時輪子必須不斷前進-後退-前進-後退，因此我們正在研究另一種轉彎方式，也就是在轉彎時，內側輪子不是前進-後退-前進-後退而是前進-停-前進-停，這將可以大幅降低車體的震動。

二、誤差大小

當繪圖車在會圖時，摩擦力會對車子有一定程度的影響，而透過在輪子上上油可以確定縮小此一誤差。

柒、結論

我們成功的以 Arduino 控制兩顆-步進馬達來製成繪圖車，而此繪圖車也能透過直走到不同座標的方式來繪出圓形、直線、多邊形和函數圖形，雖然有些許誤差，但透過幾種改善方法得以縮小許多。而我們利用他的這項功能進一步的加工，使獲圖車可以畫出任何我們手繪的圖形。而從這項功能中也可以證明我們設計的繪圖車走法可以畫出大部分的圖形。

捌、應用與未來展望

本作品除了應用在課堂上幫助數學老師的教學之外，相信也可以在其他領域上有所貢獻。因為這輛車是沿著座標跑，再加上它沒有任何範圍的限制，因此可以將圖畫得很大，可以應用來繪製海報或大型圖畫。同樣的，它也可將大型運動場的邊界或分隔線精準繪製。再者，相較於傳統印表機，此裝置輕便許多，因此可以被當作可攜型的大型印表機。而配合 App 增加的手繪功能的話，就可以使身障者透過操作手機隨心所欲的繪製複雜圖形，成為得力的輔具。

玖、參考資料及其他

一、Arduino UNO 介紹：<http://www.arduino.cc/en/Main/arduinoBoardUno>

二、控制步進馬達的參考程式碼：<http://arduino.cc/en/Reference/Stepper?from=Tutorial.Stepper>

三、Android 與 Arduino 的藍牙通訊

<http://blog.xuite.net/chycahock/EDA/225365055-Android%E8%88%87Arduino%E7%9A%84%E8%97%8D%E8%8A%BD%E9%80%9A%E8%A8%8A>

四、A4988 步進馬達驅動器 Datasheet：

https://www.pololu.com/file/download/a4988_DMOS_microstepping_driver_with_translator.pdf?file_id=0J450

五、App Inventor 2 安卓 APP 開發器：<http://ai2.appinventor.mit.edu/>

【評語】 040807

本作品利用藍芽傳輸與 APP 製作一可遙控之繪圖車，在作品的實現上，頗有新意。但在應用情境的設定與整體價值上，宜更注意實用性與價值的創造。若能解決定位問題應更能提升本作品價值。