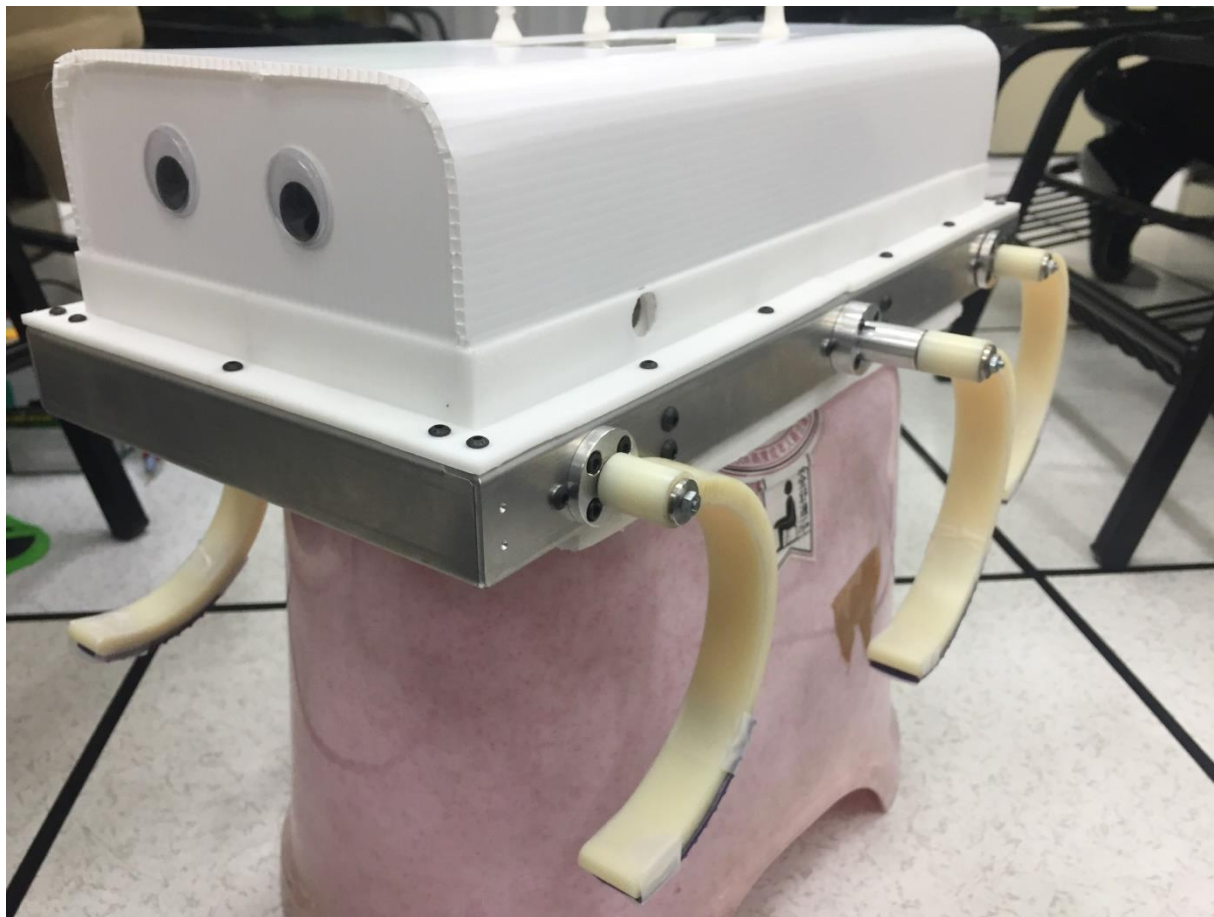


全國高級中等學校專業群科 107 年專題及創意製作競賽
「專題組」



群 別：電機與電子群

作品名稱：電子寵物機器人

關 鍵 詞：居家照顧、三腳步態、機器人

目錄

目錄.....	I
表目錄.....	II
圖目錄.....	II
壹、 摘要.....	- 1 -
貳、 研究動機.....	- 1 -
一、 專題製作背景與目的.....	- 1 -
參、 研究方法（過程）.....	- 1 -
一、 製作流程.....	- 1 -
二、 設備介紹.....	- 2 -
(一) 伺服馬達.....	- 2 -
57RH41.....	- 2 -
(二) 自製電刷.....	- 3 -
(三) MAX7219 點矩陣模組.....	- 3 -
(四) HC-06 藍芽模組.....	- 4 -
(五) 仿生腳.....	- 5 -
(六) ATMEL ATMEGA328 單晶片.....	- 5 -
肆、 研究結果.....	- 6 -
一、 成品實體.....	- 6 -
二、 程式架構.....	- 7 -
(一) 點矩陣.....	- 7 -
(二) 指定轉速.....	- 8 -
(三) 停止.....	- 11 -
(四) 自製電刷掃描程式.....	- 12 -
(五) 直走.....	- 13 -
伍、 討論.....	- 14 -
陸、 結論.....	- 14 -
柒、 參考資料及其他.....	- 15 -
一、 書籍.....	- 15 -
二、 三腳步態.....	- 15 -
三、 伺服馬達(程式).....	- 15 -
四、 藍芽(程式).....	- 15 -
五、 LED 點矩陣(程式).....	- 16 -
六、 3D 列印.....	- 16 -
七、 圖片.....	- 16 -

表目錄

表(一) 馬達比較表	- 2 -
------------------	-------

圖目錄

圖(一)	- 2 -
圖(二)	- 3 -
圖(三)	- 4 -
圖(四)	- 4 -
圖(五)	- 5 -
圖(六)	- 5 -

壹、摘要

隨著科技日新月異、遠端操控的技術逐漸普及，以及人類的生活中幾乎離不開智慧產品；像是看家機器人，看起來可靠，但家裡多多少少會有些高低落差、小門檻，此時機器人的行走就會受到限制。本篇專題為了讓機器人通行不受限制，因此參考了昆蟲三腳步態的行走方式，使其能夠克服地形對傳統圓輪型機器人所帶來的種種阻礙。

貳、研究動機

一、專題製作背景與目的

近年來，兒女長大與父母同住的意願越來越低，相對的老人們獨自在家的時間則提高。老人獨自在家也是會感到孤單，想要有個陪伴，但限於行動的不方便，寵物對他們來說可能不方便照顧，說不定還有可能是種負擔。我們因此而發想，想要讓老人靠著智慧型手機就能互動的電子寵物機器人。而且他不僅只是陪伴老人，偶爾想看看家裡各處的狀況也可以利用這個機器人。

這個機器人它不同於一般輪子機器人，他的腳可克服一些有門檻、有高地落差的地方，前面有可放置鏡頭的地方可透過視訊看他所在位置的狀況。

參、研究方法（過程）

一、製作流程

1. 擬定各期計畫表
2. 擬定所需材料
3. 測試藍芽及伺服馬達並編寫控制程式
4. 雕刻電路板並整合電路
5. 組裝成品
6. 完成

二、 設備介紹

(一) 伺服馬達

	直流馬達	步進馬達	伺服馬達		比較
			180 度 (控制角度 型)	360 度 (連續旋轉 型)	
型號	SKU：80002	57RH41	MG995		
價格(NTD)	430	307	250	250	
保持轉矩 (N-cm)	趨近 0 	63	13	13	保持轉矩可使機器人在靜止不動或斷電時仍保持站立，以免摔落進而損傷電子零件
空載電流(A)	0.2	2.4 	0.3	0.3	如使用六顆步進馬達，鋰電池無法供應如此大電流

表(一) 馬達比較表

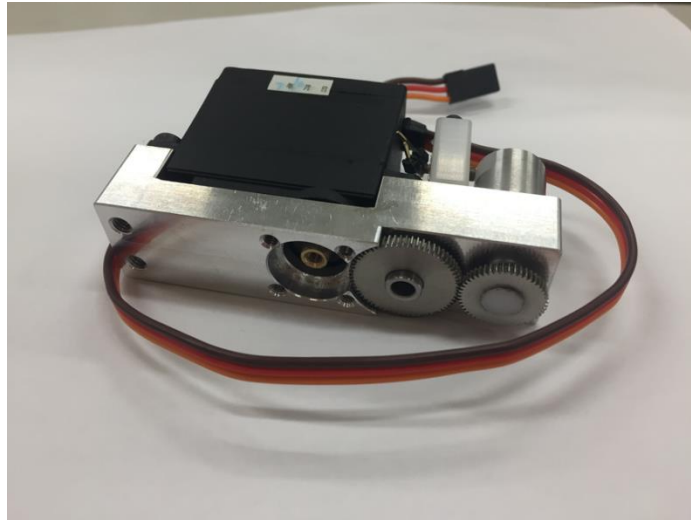
比較之下，伺服馬達性能雖屬中庸，但能符合所有必需條件



圖 (一)

(二) 自製電刷

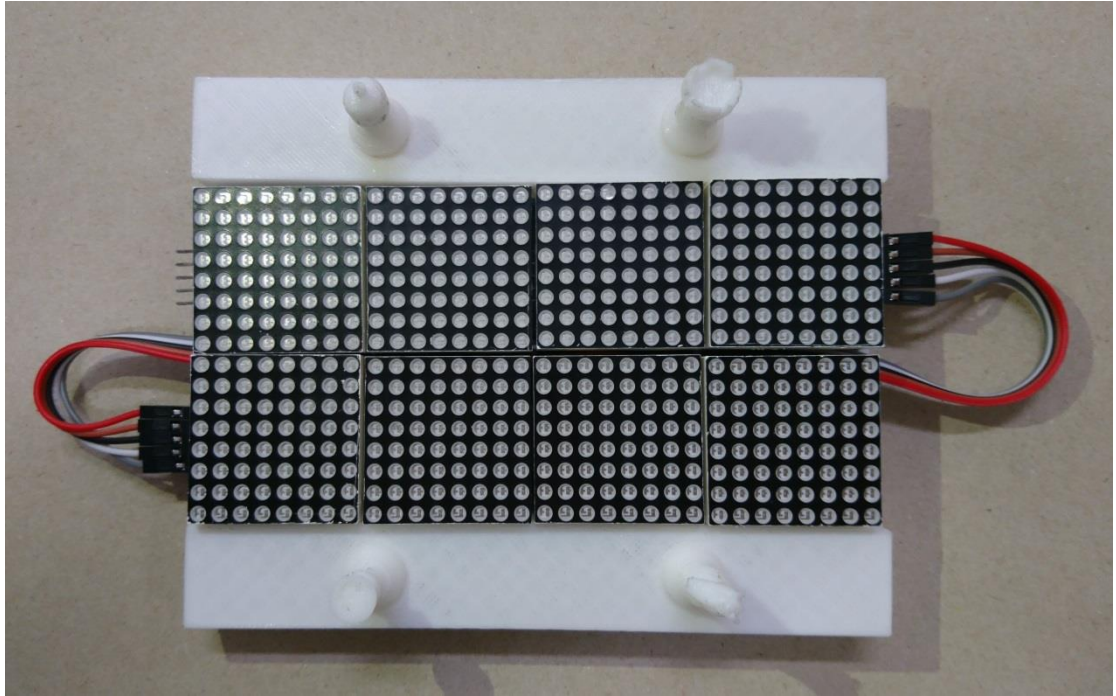
因為每具伺服馬達轉速並非完全相同，若利用程式延時控制，即會產生嚴重失步。因此我們利用電刷進行定位，使馬達可以在正確位置做出應有動作。



圖（二）

(三) MAX7219 點矩陣模組

MAX7219 是一種集成化的串列輸入/輸出共陰極顯示驅動器,它連接微處理器與 8 位元的數位 7 段數位 LED 顯示，且該晶片具有 10MHz 傳輸率的 Serial Peripheral interface (SPI) 三線串列介面，並可以和任何微處理器（包括單晶片）相連，另外只需一個外接電阻即可限制所有 LED 的電流。



圖（三）

(四) HC-06 藍芽模組

用於與手機藍芽溝通之媒介。以手機應用軟體 App Inventor2 自製 APP 後，手機只要開啟藍芽，便可輕鬆與其連接。



圖（四）

(五) 仿生腳

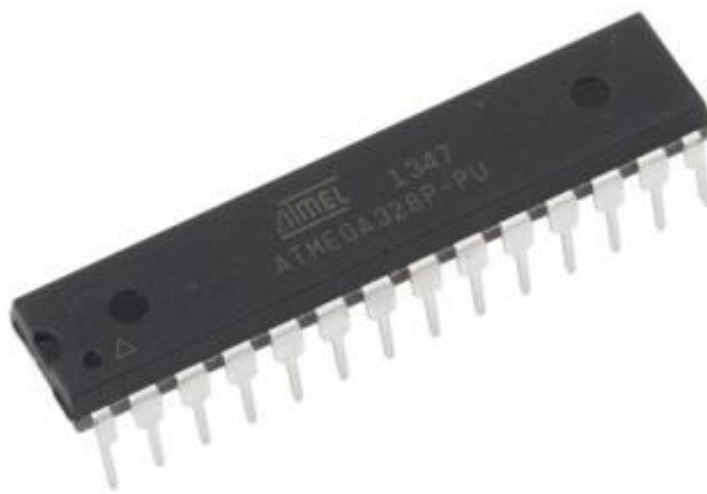
在參考了各種資料後我們決定製作出一隻仿生腳，材質選用如下:第一，要有彈性，第二，有可塑性，這兩點特性期望使腳壓縮時所產生的彈力位能轉成往上的動能。基於成本及種種因素的考量，材質使用尼龍。



圖（五）

(六) ATMEL ATMEGA328 單晶片

ATmega328 是整個電路的核心，用於控制或接收所有的訊號，無論是馬達、自製電刷、點矩陣、藍芽...等，都由它來控制



圖（六）

肆、 研究結果

一、 成品實體



圖（九）



圖 (十)

二、 程式架構

因程式共 404 行過於冗長，故僅挑重點說明

(一) 點矩陣

cmmd[0]: 藍芽送出訊號

cmmd[0]=79, ledk=0, 顯示 "你好"

cmmd[0]=80, ledk=4, 顯示 "真的"

cmmd[0]=81, ledk=8, 顯示 "好的"

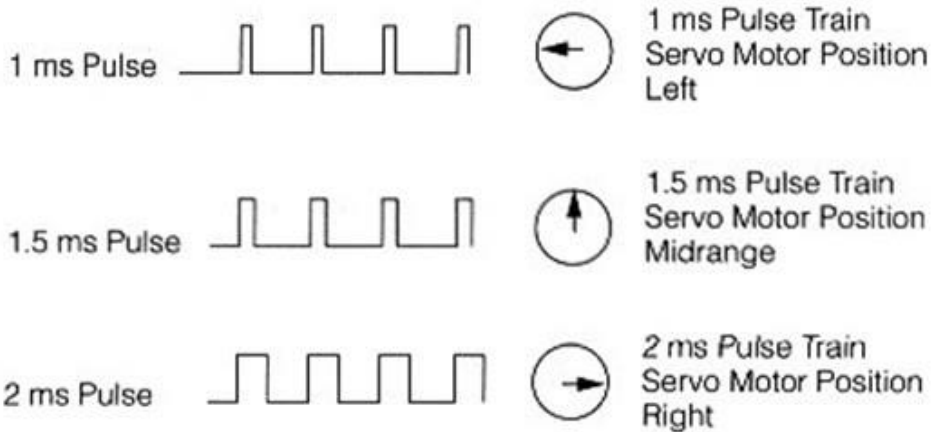
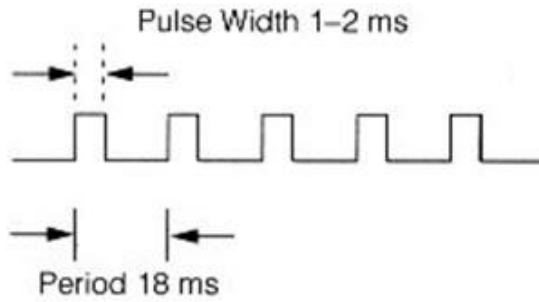
```

Martix $
1 #include "LedControl.h"
2 LedControl lc=LedControl(2,4,3,8); //2.3.4:指定接腳 //8:2點陣陣數目
3
4 int ledk; //;edk:LED顯示用
5 byte ledshow[8][2]={
6   {{0X00,0X00},{0X00,0X90},{0X00,0X90},{0X00,0X90},{0X7F,0XC8},{0X44,0X48},{0X24,0X2C},{0X04,0X0A}},//你
7   {{0X14,0X88},{0X24,0X88},{0X24,0X48},{0X44,0X48},{0X44,0X28},{0X05,0X08},{0X02,0X08},{0X00,0X00}},
8   {{0X00,0X00},{0X00,0X00},{0X3F,0X04},{0X10,0X24},{0X08,0X24},{0X04,0X7B},{0X04,0X24},{0X04,0X24}},//好
9   {{0X7F,0XA4},{0X04,0X14},{0X04,0X18},{0X04,0X08},{0X04,0X14},{0X05,0X22},{0X02,0X00},{0X00,0X00}},
10  {{0X00,0X00},{0X00,0X80},{0X7F,0XFE},{0X00,0X40},{0X0F,0XF0},{0X08,0X10},{0X0F,0XF0},{0X08,0X10}},//真
11  {{0X0F,0XF0},{0X08,0X10},{0X0F,0XF0},{0X08,0X10},{0X7F,0XFE},{0X0C,0X30},{0X30,0X0C},{0X00,0X00}},
12  {{0X00,0X00},{0X02,0X08},{0X02,0X08},{0X02,0X04},{0X3F,0X3B},{0X21,0X22},{0X20,0XA2},{0X22,0X62}},//的
13  {{0X24,0X3B},{0X24,0X22},{0X20,0X22},{0X20,0X22},{0X20,0X3B},{0X24,0X22},{0X18,0X00},{0X00,0X00}},
14  {{0X00,0X00},{0X00,0X00},{0X3F,0X04},{0X10,0X24},{0X08,0X24},{0X04,0X7B},{0X04,0X24},{0X04,0X24}},//好
15  {{0X7F,0XA4},{0X04,0X14},{0X04,0X18},{0X04,0X08},{0X04,0X14},{0X05,0X22},{0X02,0X00},{0X00,0X00}},
16  {{0X00,0X00},{0X02,0X08},{0X02,0X08},{0X02,0X04},{0X3F,0X3B},{0X21,0X22},{0X20,0XA2},{0X22,0X62}},//的
17  {{0X24,0X3B},{0X24,0X22},{0X20,0X22},{0X20,0X22},{0X20,0X3B},{0X24,0X22},{0X18,0X00},{0X00,0X00}}
18 };
19
20 void setup() {
21   lc.shutdown(0,false); // 關閉省電模式
22   lc.setIntensity(0,7); // 設定亮度為 8 (介於0~15之間)
23   lc.clearDisplay(0); // 清除螢幕
24
25   .
26   .
27   .
28   .
29
42   lc.shutdown(7,false);
43   lc.setIntensity(7,7);
44   lc.clearDisplay(7);
45 }
46
47 void leddisplay()
48 {
49   ledk=4*(cmmid[0]-79);
50   for(int j=0;j<4;j++)
51   {
52     for(int i=7;i>-1;i--)
53     {
54       lc.setColumn(j,i,ledshow[3-j+ledk][i%8][1]);
55       lc.setColumn(j+4,i,ledshow[3-j+ledk][i%8][0]);
56       Serial.print (3-j+(ledk+i)/8);
57       Serial.print ((ledk+i)%8);
58       Serial.print (" ");
59     }
60     Serial.print ("\n");
61   }

```

(二) 指定轉速

360 度伺服馬達沒有專用 Library,故使用 180 度伺服馬達 Library 前應先了解其運作方法:



180 度伺服馬達是利用光學編碼器偵測角度，當送入脈波與偵測角度相差越大，馬達轉速將越快以達到訊號角度。

360 度伺服馬達是將 180 度伺服馬達偵測角度的可變電阻換成兩顆相等電阻，故除了訊號送 90 度，不管馬達轉到哪，電阻都無法到達訊號角度。但如同 180 度伺服馬達(當送入脈波與偵測角度相差越大，馬達轉速將越快以達到訊號角度)，360 度伺服馬達可以以訊號角度決定轉速及轉向。

所以使用<servo.h>程式庫時:

```
a[z].write(90);    //代表 a[z]馬達停止
```

```
a[z].write(0~89);  //代表 a[z]馬達正轉
```

```
a[z].write(91~180); //代表 a[z]馬達反轉
```

此語法一旦寫入，arduino 即可連續送出脈波，不必以外部中斷方式一段一段送出。因三腳步態有特定三腳一起移動的特性，故將指定轉速包成副程式，即可輕鬆在主程式運用，程式如下：

```
servospeed $
1 #include <Servo.h>
2 Servo a[7];
3 // 右下a[1]，左下a[2]，右中a[3]，左中a[4]，右上a[5]，左上a[6]
4
5 void servospeed (int x ) //y=所選取速度
6 {
7     int z;
8     int speed[ ][7]={    {0,90,90,90,90,90,90},    //停
9                           {0,50,90,90,130,50,90},    //腳A
10                          {0,90,130,50,90,90,130},    //腳B
11                          {0,50,130,50,130,50,110},    //投降
12                          {0,140,40,140,40,140,40},    //直立
13                          {0,90,130,90,90,90,130},    //右轉
14                          {0,50,90,90,90,50,90}        //左轉
15                      };
16     for(z=1;z<7;z++)
17     {
18         a[z].write(speed[x][z]);
19     }
20 }
21
22 EX: servospeed(4); =>6顆馬達同時上轉
```

(三) 停止

360 度伺服馬達有兩種停止方式,一種為訊號送 1.5ms 脈波(及光學編碼器兩端電阻值相等,180 度伺服馬達轉到 90 度之訊號),但因定位精度的問題,此脈波並不能保證馬達停止,另一種是不送訊號。使用 detach()可防止 arduino 再送訊號給 arduino,故再次使用時須再次 attach() 伺服馬達。

```
#####  
1 void servostop()  
2 {  
3   for(int i=1;i<7;i++)  
4   {  
5     a[i].detach();  
6   }  
7 }  
8  
9 void servorun()  
10 {  
11   a[1].attach(A0);  
12   a[2].attach(A1);  
13   a[3].attach(A2);  
14   a[4].attach(A3);  
15   a[5].attach(A4);  
16   a[6].attach(A5);  
17 }
```

(四) 自製電刷掃描程式

```
sketch_jan20b $
1 void turn360 (int b1, int b2, int b3, int b4, int b5, int b6 ,int turn)
2 {
3
4   for(int i=1;i<7;i++)
5   {
6     value[1]=b1; value[2]=b2; value[3]=b3; value[4]=b4; value[5]=b5; value[6]=b6;
7     if(value[i]==1)
8     {
9       switch(command[i])
10      {
11        case 0:
12          if(digitalRead(encorder[i])==turn)
13          { command[i]=1; }
14          break;
15        case 1:
16          if(digitalRead(encorder[i])!=turn)
17          {
18            command[i]=2;
19          }
20          else {command[i]=1;}
21          break;
22        case 2:
23          delayMicroseconds(100);
24          if(digitalRead(encorder[i])!=turn)
25          { command[i]=3; }
26          else {command[i]=1;}
27          break;
28        case 3:
29          delayMicroseconds(100);
30          if(digitalRead(encorder[i])!=turn)
31          {
32            command[i]=4;
33            a[i].write(90);
34          }
35          break;
36      } //switch
37    } //if
38    else
39    {
40      command[i]=4;
41    }
42  } //for
43  if(command[1]==4 && command[2]==4 && command[3]==4 && command[4]==4 && command[5]==4 && command[6]==4)
44  {
45    for(int i=1;i<7;i++)
46    {
47      value[i]=0;
48      command[i]=0;
49    }
50    stepper=stepper+1;
51  } //if
52 }
53
54 EX: turn360(1,1,1,1,1,1,HIGH); =>掃描
```

上圖範例為診測六顆馬達(1,1,1,1,1,1)是否正轉(HIGH)一圈

(五) 直走

如註解所示，不在贅述。

```
run $
1 switch(stepper)
2 {
3   case 0 : //(腳A移動0.1s,腳B轉1圈)
4     servorun();
5     servospeed(3);
6     delay(100);
7     servospeed(1);
8     delay(750);
9     stepper++;
10    break;
11    case 1 : //掃描腳B是否轉一圈
12      turn360(1,0,0,1,1,0,HIGH);
13      break;
14      case 2 : //(腳B移動0.1s,腳A轉1圈)
15        servospeed(3);
16        delay(100);
17        servospeed (2);
18        delay(750);
19        stepper++;
20        case 3 : //掃描腳A是否轉一圈
21          turn360(0,1,1,0,0,1,HIGH);
22          break;
23          case 4 : //移動2步
24            stepper=0;
25            if(sw==1)
26            {
27              sw=0; cmmd[0]=0; servostop();
28            }
29            break;
30 }
```


伍、 討論

在專題的製作過程中，遇到許多技術上的困難與採買材料買錯的窘境，經過一再的測試調整與重製的過程中我們學到了許多理論外的知識，以下為無法克服之問題：

- 一、 在攀爬樓梯時，因為扭力不足，導致有時馬達無法撐住機體，使機體下墜，可能因此跌下樓梯。
- 二、 使用的控制器為利用 App Inventor 中進行藍芽連線單晶片以控制馬達，但藍芽連線距離有限，因此無法進行遠距離控制。
- 三、 因為各具馬達品質不一，有時在停止狀態下仍會旋轉，導致需要利用控制程式來使其停止。
- 四、 在鋰電池電量偏低時，有時會出現控制不順，甚至無法控制的情況

陸、 結論

- 一、 在這次的專題製作中，不只學習到課程中所教授的知識，也學習了許多課程以外的知識。
- 二、 在經過了無數次的測試後，我們發現：在爬樓梯時，因為伺服馬達的扭力不如預期，導致在攀爬樓梯時，因為扭力不足無法支撐整個機體，可能導致機器人跌落台階。
- 三、 經過了這一次的專題，我們驗證了可本中所學的理论，儘管有時會失敗，但找到了失敗的原因並解決他，也從其中學到寶貴的經驗。另外也感謝指導老師及惦記科所有老師的協助，讓我們可以順利的完成此次的專題製作。

柒、參考資料及其他

一、書籍

《輕鬆入門 Arduino 範例分析與實作設計》

《超圖解 Arduino 互動設計入門》

二、三腳步態

TVBS 新聞--科學不一樣 六足機器人仿昆蟲「三腳步態」未來可勘災

<https://www.youtube.com/watch?v=W9iUZNuWgX8>

分析多足動物+機器人步態

<http://hexapodalpha.blogspot.tw/2010/12/blog-post.html>

四足仿生機器人之設計與製作

[http://ijait.coinfo.cyut.edu.tw/IJAIT%208\(1\)/IJAIT%208\(1\)_05.pdf](http://ijait.coinfo.cyut.edu.tw/IJAIT%208(1)/IJAIT%208(1)_05.pdf)

仿生六足機器人之虛擬控制系統開發

<http://www.me.cycu.edu.tw/csme2007/pdf/D05-0027.pdf>

三、伺服馬達(程式)

Arduino 練習：伺服馬達以 Tower Pro SG90 為例

<http://yehnan.blogspot.tw/2013/09/arduinotower-pro-sg90.html>

Arduino 練習：連續旋轉伺服馬達 GWS S35

<http://yehnan.blogspot.tw/2013/09/arduinogws-s35.html>

Cooper Maa: 用 pulseIn 量測 button 被按了多久

<http://coopermaa2nd.blogspot.tw/2011/04/pulsein-button.html>

四、藍芽(程式)

[雙 A 計畫]

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

App Inventor 透過藍牙傳送訊號給 Arduino

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

將 Arduino Uno 當做一台 ISP 線上燒錄器，燒錄到麵包板上的 ATmega328P-PU 晶片

<http://yehnan.blogspot.tw/2013/02/arduino-unoispbootloadersketchatmega328>

五、LED 點矩陣(程式)

MAX7219-datasheet

<https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>

Arduino 筆記(十四)：使用 MAX7219 控制 8x8 LED Matrix

<http://atceiling.blogspot.tw/2017/03/arduino-max7219-8x8-led-matrix.html>

GitHub - wayoda/LedControl: An Arduino library for MAX7219

<https://github.com/wayoda/LedControl>

Arduino Playground – LedControl

<https://playground.arduino.cc/Main/LedControl>

六、3D 列印

Tinkercad | Create 3D digital designs with online CAD

<https://www.tinkercad.com>

[3D 列印] 3D 列印問題排除指南！ - 三帝瑪 3DMart

<https://3dmart.com.tw/news/3d-pinting-troubleshooting-guide>

3D 列印不再印出馬賽克！

<https://www.techbang.com/posts/3d-printers-use-be-reduced-to-70-micrn>

七、圖片

MAX7219 點矩陣模型

<https://uge-one.com/image/cache/data/new%20items2/max7219%20serial%20dot%20matrix%204%20module-500x375.jpg>

ATMEL ATMEGA328 單晶片

https://media.rs-online.com/t_large/F7589339-01.jpg