

第 19 屆 TDK 盃全國大專院校創思設計與製作競賽

參賽報告書

參賽組別：自動組

隊伍編號：A15

隊名：鋼彈盪單槓

學校名稱：國立臺灣科技大學

科系名稱：機械工程系

指導老師：林紀穎老師

參賽學生：楊宗翰、馬匡緯、蘇冠瑋、洪桓駿

中華民國 104 年 7 月 21 日

一、 機器人特色摘要說明

本組所設計的機器人是以前簡約為主旨，設計用最少的結構來完成任務，以此概念設計出既可夾球又可夾筆的機器手臂可以說是本組最大的特色，使用滾珠螺桿的滑塊前後滑動來帶動夾爪的張開和夾緊。

機台因為配合需要寫字的需求，用滾珠螺桿滑軌來帶動夾爪而不使用現在比較流行的關節式的設計，是因為穩定性和程式編寫難度的考量，用兩條互相垂直的滑軌來在平面上寫字。

二、 機構設計

由於這次的比賽需要有寫字與投籃的功能，我們組打算使用 XYZ 滑軌的機械手臂，其重量比起以往做得機器人要來的重，因此底盤的設計相當的重要，必須要能夠穩固，循線、感測、寫字、投球才能夠更加準確。 本組這次是選用3cm × 3cm的鋁擠條來當整體的底盤骨架材料，如下圖所示：



圖. 機器人底盤骨架

由於這次比賽寫字的範圍需要有60cm長、帶球過障礙物的距離為140cm，因此我們底盤骨架設計成80cm × 80cm的矩形，讓機器人能夠穩定且順利的通過關卡。

至於底座的馬達，為了要能夠帶動如此沉重的機台，我們選用了65W DC24V1800R 4.5A的馬達，輪子的部份我們也決定使用2個輪胎當作我們的行進裝置，如下圖所示：



圖. 機器人選用的輪胎



圖. 底座選用的馬達

但由於買來的輪胎內孔為圓孔，與馬達的孔無法配合，因此我們將馬達的軸加熱，穿過塑膠材質的孔，以達到目的，如下圖所示：



圖. 馬達軸的形狀



圖. 加工後輪胎的孔

再來就是把馬達固定在底座上，為了方便拆裝與位置調整，我們一樣使用鋁擠將馬達裝置再



底座上，其結構設計如下圖

圖. 將馬達固定的結構

為了能夠穩固，不讓馬達有鬆動產生，這結構鎖緊時的調整也是花了不少時間，穩固後，再整個裝置到底座上鎖緊完成。

由於我們只使用 2 個輪胎，整個底座還需要更多的支撐點，因此我們在機台的四周圍加裝萬向輪，其裝置的方法如下圖：



圖. 萬向輪裝置側視圖

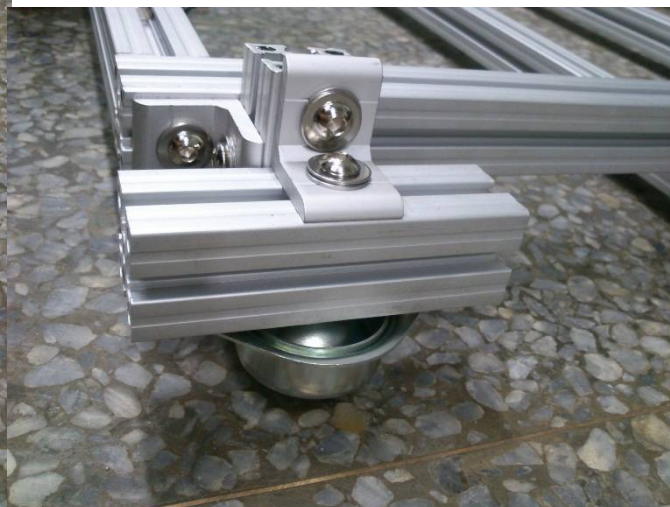


圖. 萬向輪裝置正視圖

最後經過一連串的裝配與校正過程，完成了穩固的底座基本構造，如下圖所示：

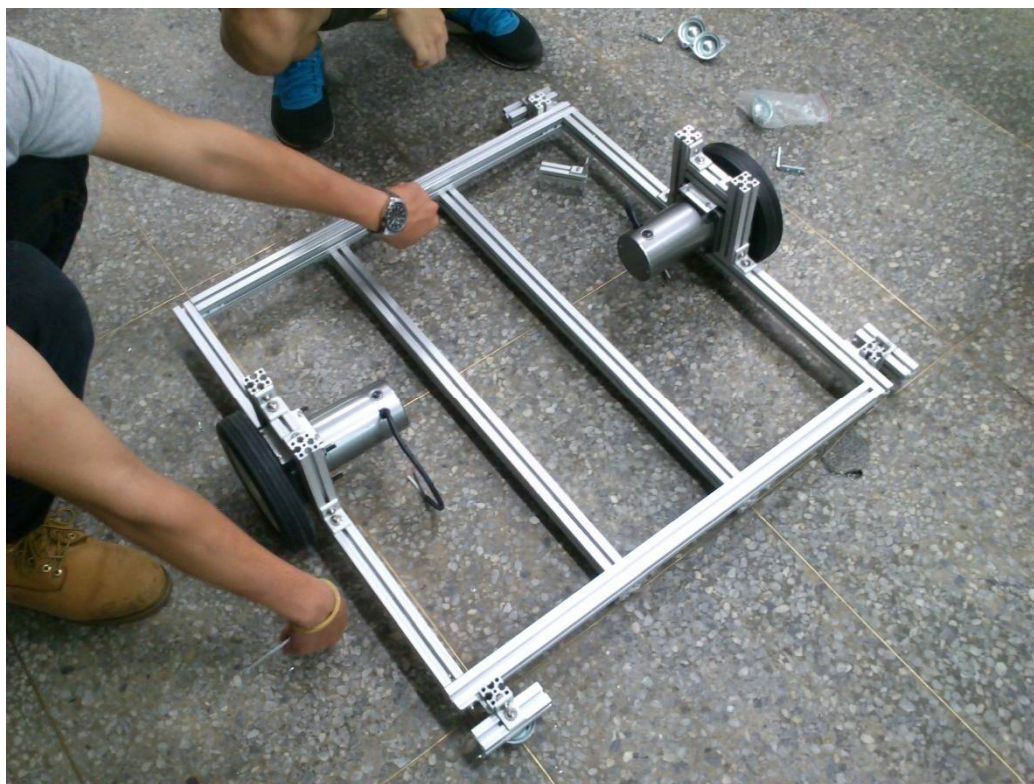
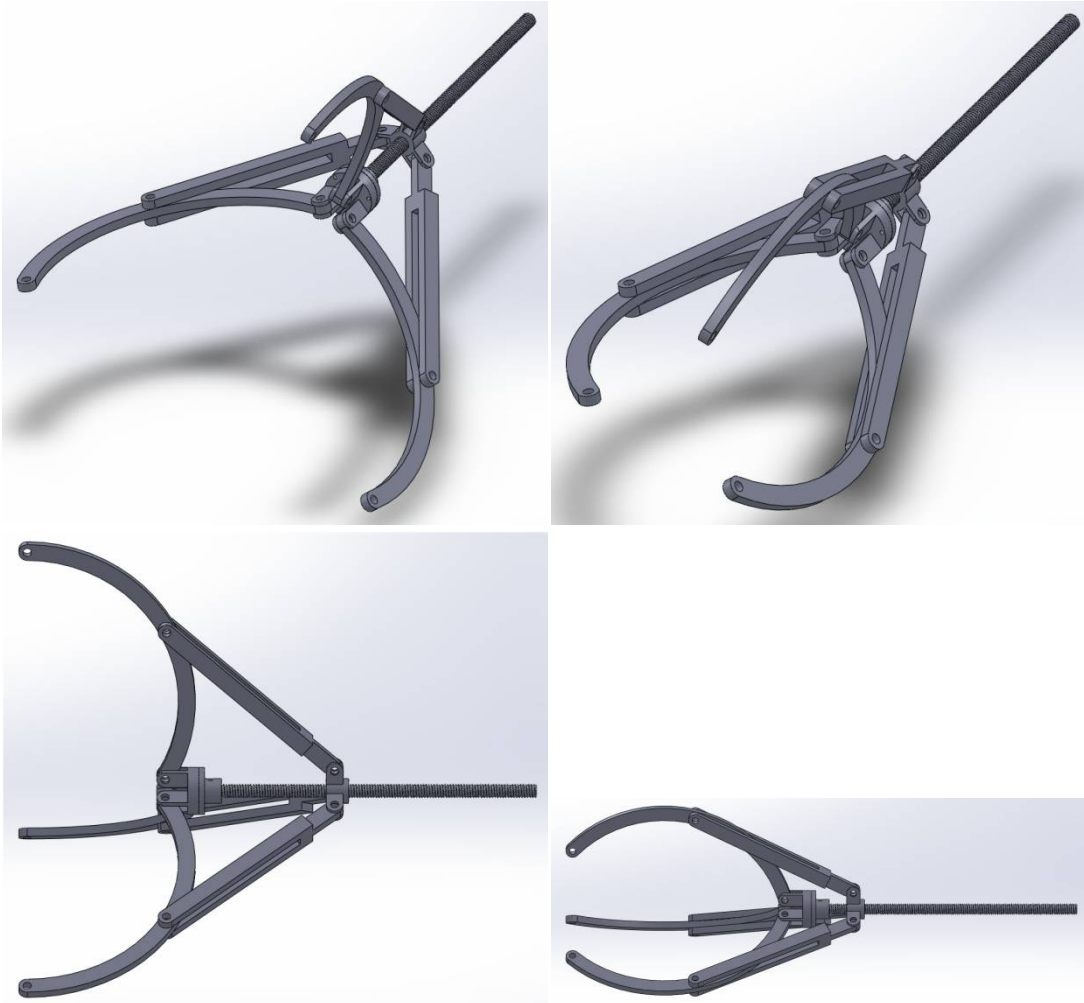


圖. 底座成品圖

夾爪設計

第一代夾爪設計：

自動組夾爪需求為：夾持圓筒狀色筆、籃球，兩者徑向皆為圓形，所以設計上採用三爪結構，驅動方式以步進馬達轉動螺桿，驅動滑前後移動以帶動夾爪運作

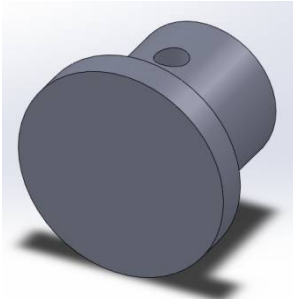


零件：

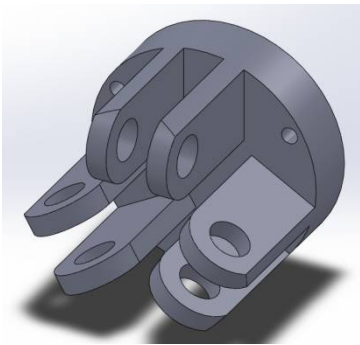
1. 螺桿



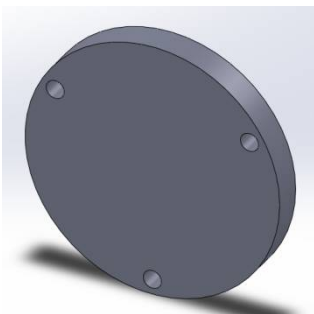
2. 螺桿端固定件



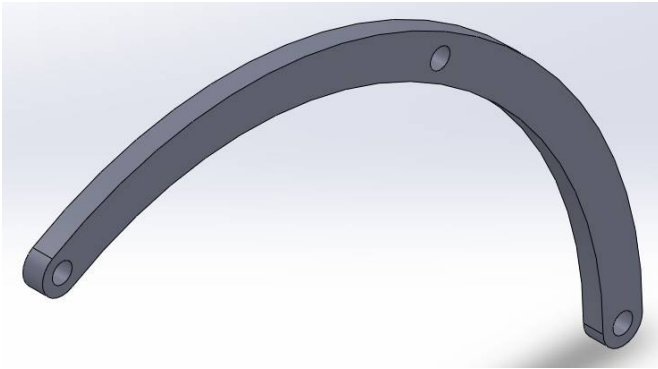
3. 螺桿端轉動件



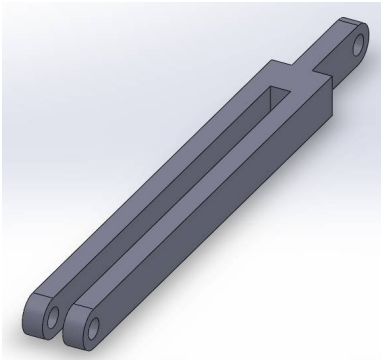
4. 螺桿端轉動件蓋



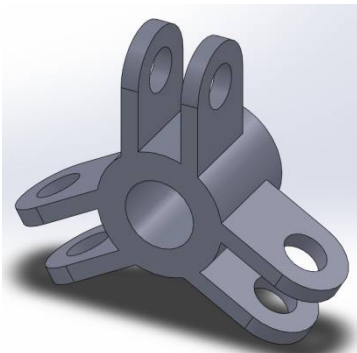
5. 夾爪



6. 浮桿



7. 滑塊

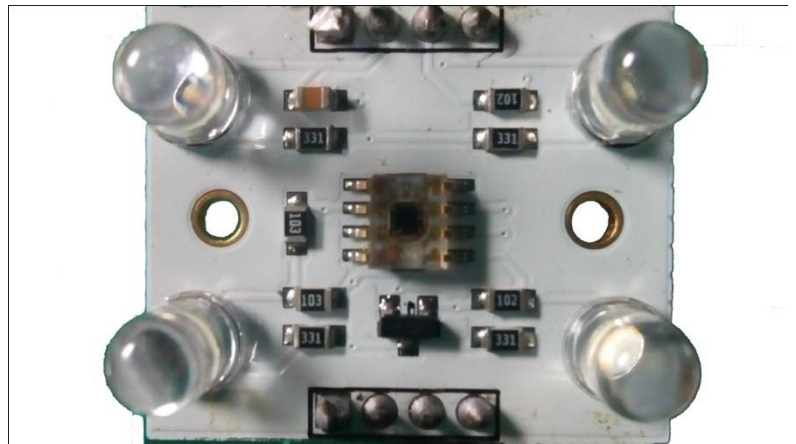


三、電控設計

循線，辨色方法：

(1) 機器人辨色功能-

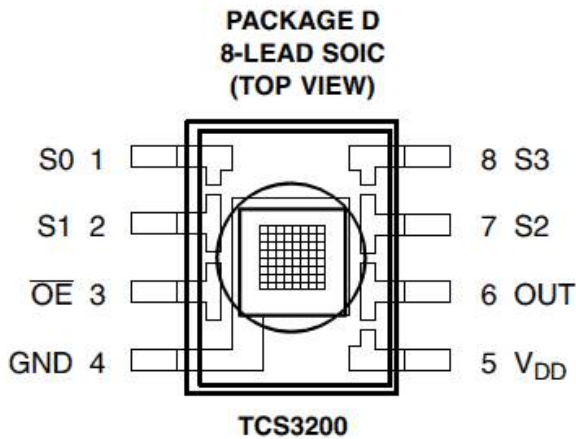
使用晶片:TCS3200



辨色原理：

因為知道三原色的值我們就可以判斷出物體的真實顏色，TCS3200 有三種濾波器，分別對應三種顏色的感測，假設當選擇紅色濾波器時，他只允許紅色的光通過，藍色和綠色都被阻止，這樣就可以得到紅色光的光強；同理選擇其他的濾波器，就可以得到藍色光和綠色光的光強。通過這三個值，就可以分析投射到 TCS3200 感測器上的光的顏色。

datasheet:



Terminal Functions

TERMINAL NAME	NO.	I/O	DESCRIPTION
GND	4		Power supply ground. All voltages are referenced to GND.
\overline{OE}	3	I	Enable for f_o (active low).
OUT	6	O	Output frequency (f_o).
S0, S1	1, 2	I	Output frequency scaling selection inputs.
S2, S3	7, 8	I	Photodiode type selection inputs.
V_{DD}	5		Supply voltage

Table 1. Selectable Options

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

由
上
圖
知，

我們可以藉由調整 S0、S1 來改變輸出值的速度，若需要判別很接近的顏色或需要快速判斷顏色時，可以把 S0 和 S1 都輸入高電位，S2 和 S3 可以改變濾波器濾波後剩下的顏色，在寫程式時需要把三種顏色的強度都測完後才判斷真實物體的顏色。

arduino 程式：

```
int s0=22;//設定 arduino 對應晶片的腳
int s1=24;
int s2=26;
int s3=28;
int OUT=2;
int count=0;//給待會會用到的變數
int selectcolor=0;
int countR=0 ,countG=0,countB=0;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(s0, OUTPUT);
    pinMode(s1, OUTPUT);
    pinMode(s2, OUTPUT);
    pinMode(s3, OUTPUT);
    pinMode(3, OUTPUT);
}
void tcs3200()// tcs3200 晶片程式
{
    selectcolor=0;
    digitalWrite(s0, HIGH); //頻率使用最高
    digitalWrite(s1, HIGH);
    digitalWrite(s2, LOW);//從紅色的強度開始測量
    digitalWrite(s3, LOW);
    attachInterrupt(0, countpause, CHANGE);// 當有訊號改變時執行程式， 用
    attachInterrupt 可以不漏掉任何訊號
    timer0_init();
}
void countpause()
{
    count++;//每次訊號改變都加一用來判斷顏色強度(最後此值越高強度越強)
}
```

```

void timer0_init(void)// 設定 timer 讓每個濾波顏色測試的時間都一樣長
{
TCCR2A=0x00;
TCCR2B=0x07;
TCNT2=100;
TIMSK2=0x01;
}
ISR(TIMER2_OVF_vect)//輸出量測到的顏色對應強度
{
    TCNT2=100;
    selectcolor++;
    if(selectcolor==1)
    {
        countR=count;
        Serial.print("red=");
        Serial.println(countR);
        digitalWrite(s2, HIGH);//紅色測完測綠色
        digitalWrite(s3, HIGH);
    }
    else if(selectcolor==2)
    {
        countG=count;
        Serial.print("green=");
        Serial.println(countG);
        digitalWrite(s2, LOW);//綠色測完測藍色
        digitalWrite(s3, HIGH);
    }
    else if(selectcolor==3)
    {
        countB=count;
        Serial.print("blue=");
        Serial.println(countB);
        Serial.println("\n");
        digitalWrite(s2, LOW);//設定回測量紅色
        digitalWrite(s3, LOW);
    }
}

```

```

else if(selectcolor==4)
{
  selectcolor=0;
}
count=0;
}
void loop() {
  tcs3200();
while(1);
}

```

量測結果:

測量藍色時的值

red=350

green=335

blue=1118

測量紅色時的值

red=769

green=253

blue=1222

測量綠色時的值

red=227

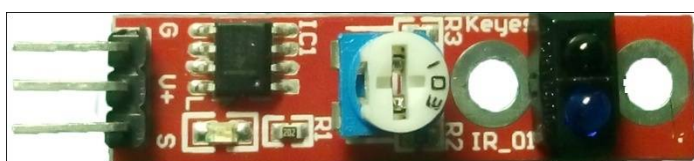
green=222

blue=720

由實驗結果看出，測量紅色和綠色時，測量物體的顏色想對應的強度都不是最強的，反而是藍色的值最強，若要改善此現象可以在測量時先對此晶片做白平衡，讓每種顏色的強度在感測白色時都相等，再來測量帶測物。

(2) 機器人巡跡功能-:

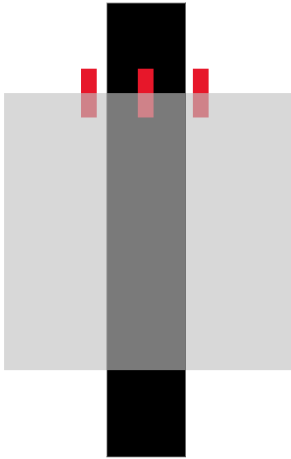
晶片選用: TCRT5000:



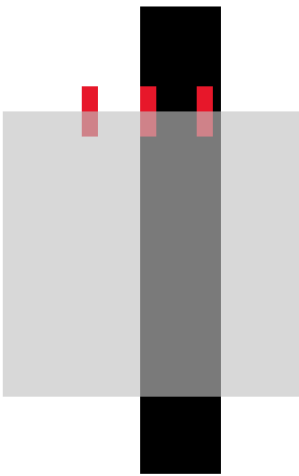
此晶片會發射紅外線，當感測器接受到反射回來的紅外線時(物體不為黑色)S角輸出低電位，當沒有接收到反射回來的紅外線時(物體為黑色)S角輸出高電位，有亦可變電阻可以調整感測的靈敏度，適應不同環境。

循跡原理:

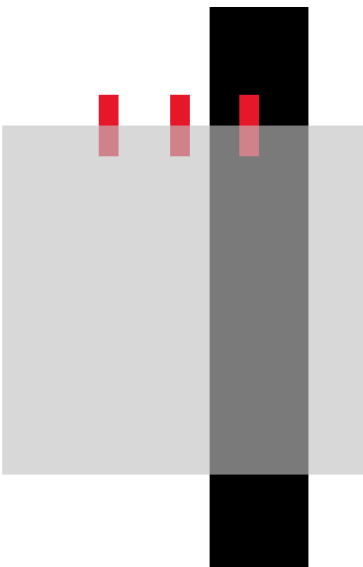
當兩邊的感測器輸出低電位 0，中間的感測器輸出高電位 1 時，執行向程式。



當中間和右邊的感測器輸出為高電位 1，左邊為低電位 0 時，知道自走車稍微偏右，執行向右微調程式，需左轉微調時道理想同。



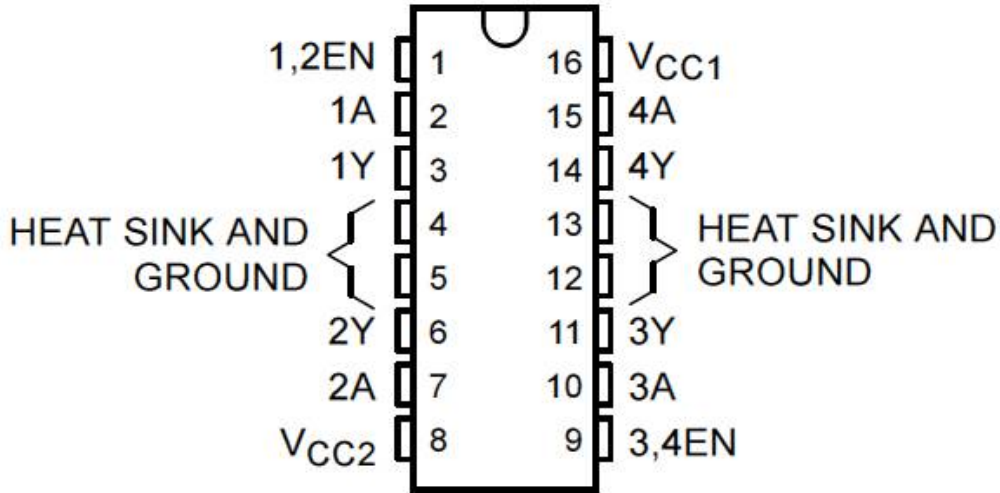
當左邊和中間感測器輸出為低電位 0，右邊為高電位 1 時，知道自走車需右轉，執行右轉程式，需左轉時道理相同。



自走車馬達控制晶片選用:1293d

晶片腳位功能:

**N, NE PACKAGE
(TOP VIEW)**



**FUNCTION TABLE
(each driver)**

INPUTS†		OUTPUT Y
A	EN	
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)

† In the thermal shutdown mode, the output is
in the high-impedance state, regardless of
the input levels.

使用方法：

由晶片的 datasheet 知若 EN 角為高電位時給 A 角高電位對應編號的 Y 角也會變為高電位，給 A 角低電位，對應編號的 Y 角也會變為低電位，所以若要調整馬達轉向，只要把相對應的 A 角高電位和低電位互相調換就可以達到，若在 EN 角位輸入 PWM 的訊號可以達到調整馬達轉速快慢的效果，若要暫停時只要把 EN 角為輸入低電位就可以達到。

1.4arduino 循跡程式：

```
int forward=0;// 設定常數方便程式閱讀
```

```

int right=1;
int rright=2;
int left=3;
int lleft=4;
int rightsensor, leftsensor, midsensor;
//arduino pin
int EN12=6;//定義 ic 晶片和感測器對應到 arduino 的腳位
int A_1=52;
int A_2=53;
int EN34=7;
int A_3=50;
int A_4=51;
int sensorr=22;
int sensorl=23;
int sensorm=24;
void setup() {
    // put your setup code here, to run once:
pinMode(EN12, OUTPUT);
pinMode(A_1, OUTPUT);
pinMode(A_2, OUTPUT);
pinMode(EN34, OUTPUT);
pinMode(A_3, OUTPUT);
pinMode(A_4, OUTPUT);
pinMode(sensorr, INPUT);
pinMode(sensorl, INPUT);
pinMode(sensorm, INPUT);
Serial.begin(9600);
}
int witch_act(void)//判斷動作的子程式
{
    if (rightsensor==0&leftsensor==0&midsensor==1)
return 0;//向前
    else if (rightsensor==1&leftsensor==0&midsensor==1)
return 1;//向右微調
    else if (rightsensor==1&leftsensor==0&midsensor==0)
return 2;//向右

```

```

else if(rightsensor==0&leftsensor==1&midsensor==1)
return 3;//向左為調
else if(rightsensor==0&leftsensor==1&midsensor==0)
return 4;//向左
}
void forwardact(void)// 執行向前動作時控制 1293d 晶片的子程式
{
    analogWrite(EN12, 200);
    analogWrite(EN34, 200);
    digitalWrite(A_1, HIGH);
    digitalWrite(A_2, LOW);
    digitalWrite(A_3, HIGH);
    digitalWrite(A_4, LOW);
}
void rightact(void)// 執行右轉微調動作時控制 1293d 晶片的子程式
{
    analogWrite(EN12, 200);
    analogWrite(EN34, 0);
    digitalWrite(A_1, HIGH);
    digitalWrite(A_2, LOW);
    digitalWrite(A_3, HIGH);
    digitalWrite(A_4, LOW);

}
void rrightact(void)// 執行右轉動作時控制 1293d 晶片的子程式
{ analogWrite(EN34, 200);
    analogWrite(EN12, 200);
    digitalWrite(A_1, HIGH);
    digitalWrite(A_2, LOW);
    digitalWrite(A_3, LOW);
    digitalWrite(A_4, HIGH);

}
void leftact(void)// 執行左轉微調動作時控制 1293d 晶片的子程式
{analogWrite(EN34, 200);
analogWrite(EN12, 0);

```

```

    digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);
    digitalWrite(A_3, HIGH);
digitalWrite(A_4, LOW);

}
void lleftact(void)// 執行左轉動作時控制 1293d 晶片的子程式
{analogWrite(EN34, 200);
    analogWrite(EN12, 200);
digitalWrite(A_1, LOW);
digitalWrite(A_2, HIGH);
digitalWrite(A_3, HIGH);
digitalWrite(A_4, LOW);

}

void loop() { // 主要程式
    analogWrite(EN34, 0);
    analogWrite(EN12, 0);

    int act;
while(1)
{
    rightsensor=digitalRead(sensorr); //感測器讀取
    leftsensor=digitalRead(sensorl);
    midsensor=digitalRead(sensorm);
    act=witch_act();//執行動做判斷程式
    Serial.println(act);
    switch(act)
    {
    case 0:
    {
        forwardact();
        break;
    }
    case 1:

```



```

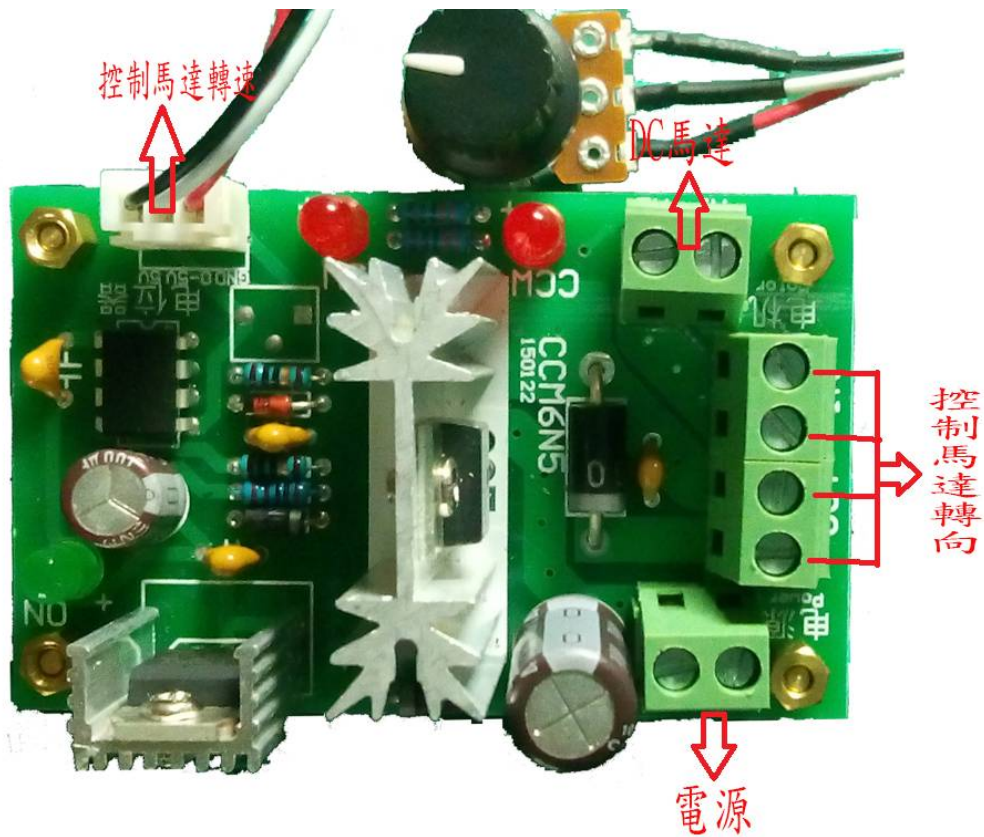
rightact();
break;
case 2:
rrightact();
break;
case 3:
leftact();
break;
case 4:
lleftact();
break;
}
}
}

```

機器人輪胎控制-:

使用已模組化的控制器來改造

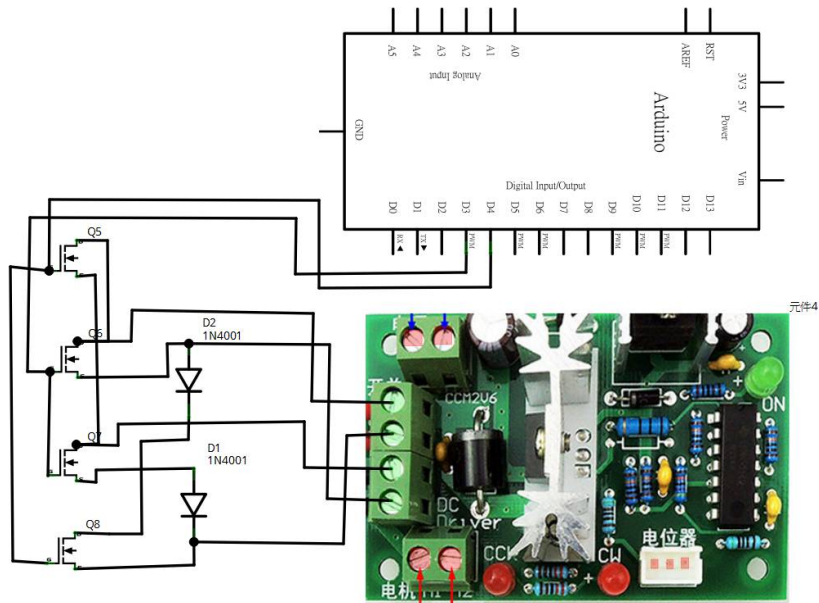
主要控制器:



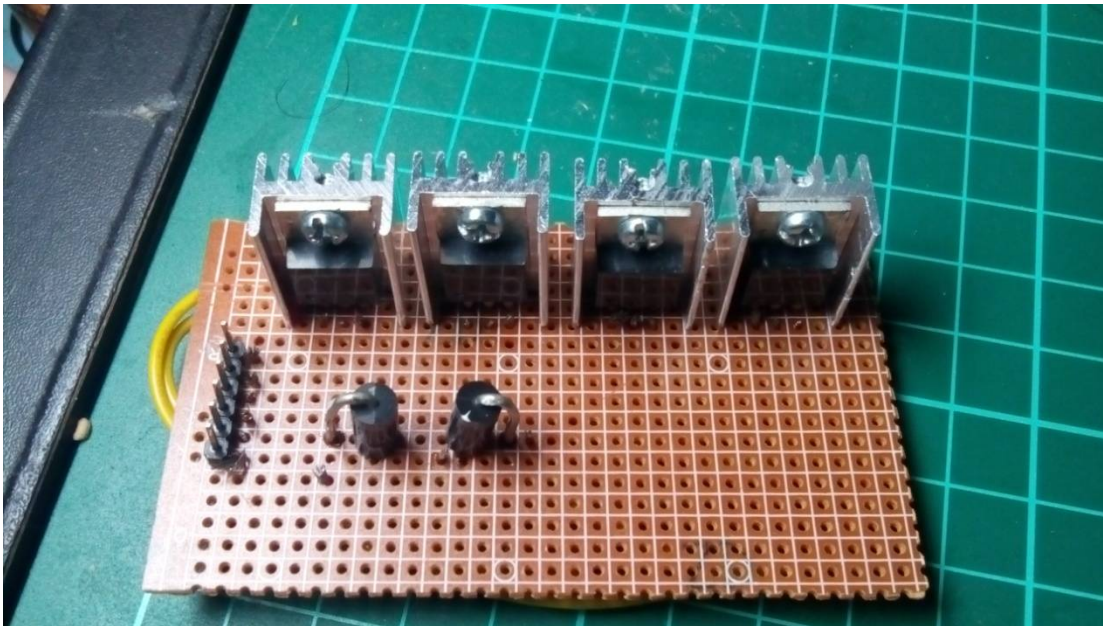
改善馬達轉向的控制:

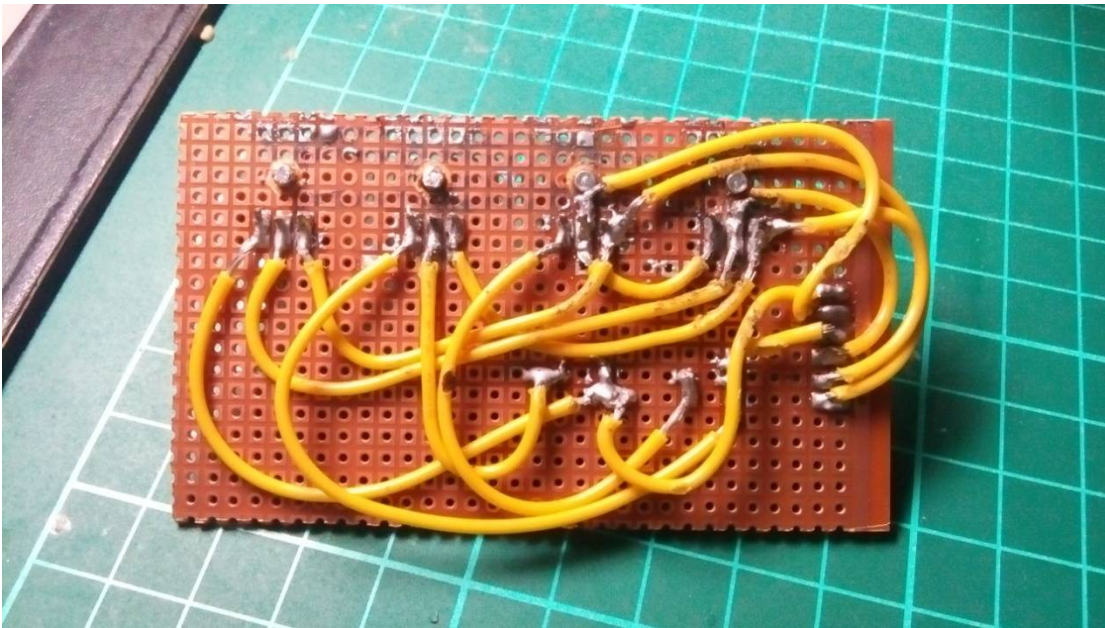
此模組的轉向和纂速都是用外部的控制裝置來控制，為了可以達到完全自動，可以把控制轉向的按鈕改為電晶體方便使用 arduino 控制。

電路圖：



成品照片：





改善馬達轉速的控制：

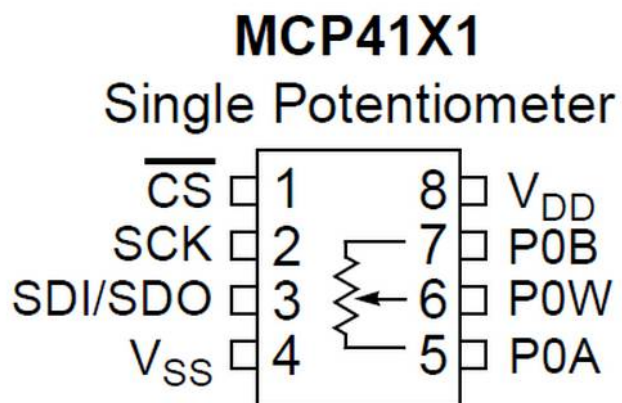
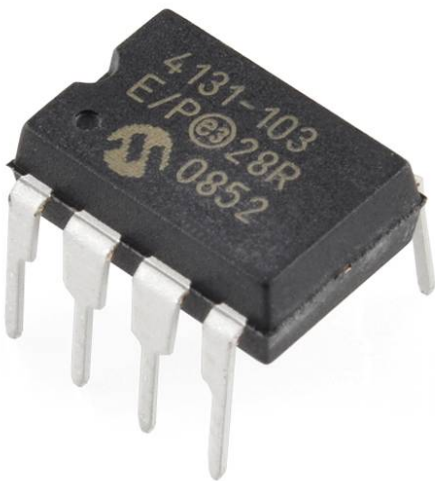
此晶片的馬達轉速可以靠輸入 0-5v 的電壓來控制，輸入 5v 時轉速最高，輸入 0v 時轉速最低。

改良方法：

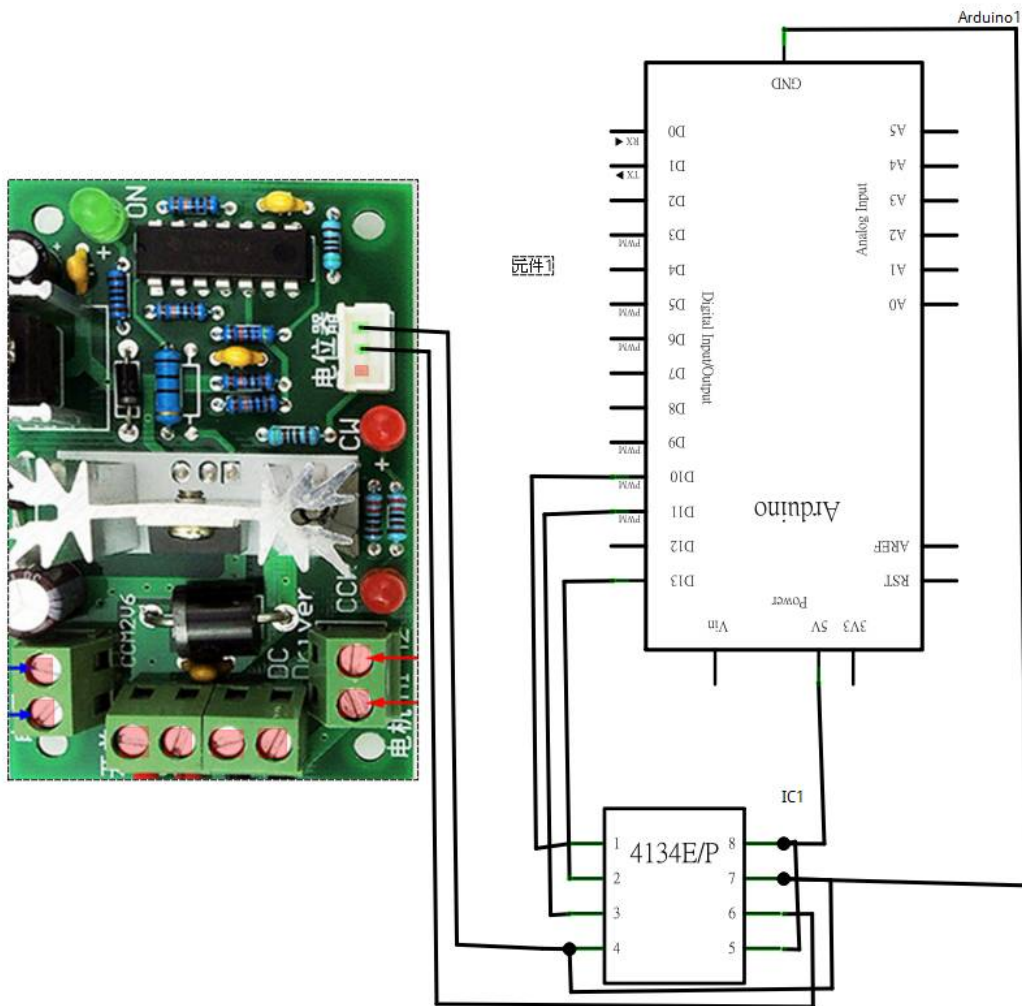
此處的改良方法是用一個數位電位計來取代可變電阻，讓整個系統偷可以藉由 arduino 控制。

使用 MCP4134 晶片：

此晶片可以控制輸出電壓從 1.8V 到 5.5V，分為 129 步從 0 到 128，是使用 SPI 的通訊方法。



MCP4131 和主要馬達控制器配線圖



arduino 控制 MCP4131 控制程式：

```
#include <SPI.h> //此晶片用 SPI 的通訊方法
byte address = 0x00;
int CS= 10;
int digitalPotWrite(int value) // 和 MCP4134 晶片溝通的副程式
{
    digitalWrite(CS, LOW);
    SPI.transfer(address);
    SPI.transfer(value);
    digitalWrite(CS, HIGH);
}
void setup()
{
    pinMode (CS, OUTPUT);
```

```

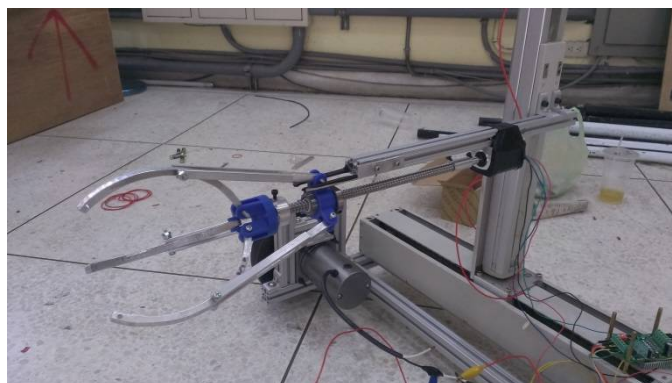
pinMode (2, OUTPUT);
pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
SPI.begin();
}

void loop() {
  digitalPotWrite(0);
  for(int i=0;i<=128;i++)//讓晶片的電位從最小慢慢升到最大
  {
    digitalPotWrite(i);
    delay(30);
  }
  for(int i=128;i>=0;i--)//讓晶片的電位從最大慢慢降到最小
  {
    digitalPotWrite(i);
    delay(30);
  }
  delay(1000);
}

```

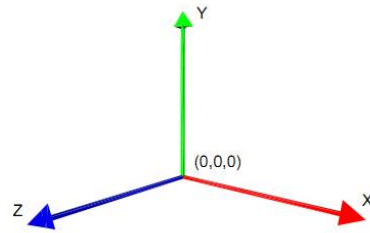
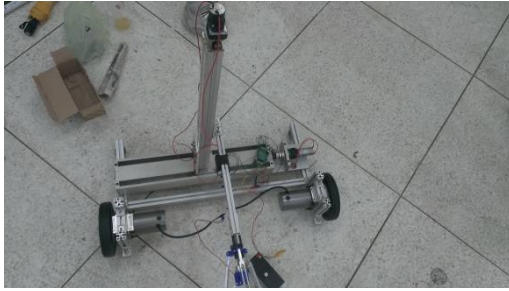
四、創意與科技人文整合說明

我們的機器人，目的是要設計出具備行進、感測辨色、取球、投球與寫字功能之全自主式機器人，我們利用娃娃機夾爪的構思，設計出可以取球跟拿筆的爪子(如圖一所示)，透過螺桿驅使爪子的開與闔，來完成整個比賽之指定動作，其



圖一

機器人具備 X、Y、Z 三軸向的移動方向，機台上面裝有兩組線性滑軌加上移動機台的移動裝置所構成三軸向之運動(如圖二)，在設計的過程中，為了避免機台過



圖二

重導致移動裝置難以移動，把整個機台框架，盡量所小範圍，比賽的目的是需要做一些平常人在做的事情，把拿筆寫字簡化成我們設計出的機器人。

五、遭遇困難

1. 整體太大

一開始設計是 80cmx80cm 的大小，但我們的機器人決定只單靠手臂完成所有關卡，如果是 80cmx80cm 的話會有太多不必要的空間，加上為了克服關卡上的設計，最後選擇使用大約 35cmx80cm 的長方形。

2. 輪胎的放置位置

本機器人設計只單靠手臂完成寫字、放筆、取球、灌籃的動作，但必須考慮到電控的方便、寫字的距離、與籃框的距離等考量，於是我們決定選用 2 顆輪子來操控機器人，且放置在整台機器人的最前端，不僅有效的運用空間，更解決了加上夾爪後的機台可會超出 1m 限制的條件。

3. 夾爪自由度

我們組別想了很多種夾爪的運動機構，有考慮使用 3 個旋轉軸的手臂、也考慮 XYZ 軸移動的手臂、甚至是 XYZ 軸手臂加上夾爪 A 軸旋轉，但考慮金費、重量太重可能馬達帶不動等因素，因此我們機器人只單靠 XY 軸的滑軌來控制夾爪移動。

4. 夾爪太重

為了能寫字、灌籃，決定後的夾爪須向外延伸至少 48cm，且我們材料都選用鋁、再加上馬達的重量，整個上機台後，滑軌會撐不住重量而往下滑。因此我們更改了些設計，把不必要的部分拿掉，設計出更簡單的方式來定位夾爪。

5. 滑軌動太慢

本組原先使用步進馬達來控制滑軌，扭力夠支撐夾爪的重量、電控上也較為方便，但是就是太慢了。為了解決這問題，且為了以更快的速度完成關卡，我們決定使用直流馬達來控

制，因此我們克服了速度上的問題、但未來也可能會因為太多感測器而產生其他的問題，但相信一定會有解決辦法的。

六、未來規劃

我們所參與的是自動組，因此很多電路的部分必須等實際的場地出來後才可動作，現在能做的就是把硬體的部分給做好，且在家測試一些簡單的電路模擬。

未來希望在比賽前可以在1至2個禮拜前就完工，測試、解決可能會發生的問題，甚至是對機器人作美化達到加分果，最後修正BUG讓比賽一切順利。比賽過程也希望能夠穩定快速的解決所有關卡，贏得冠軍，但不管得獎與否，最重要的是能夠藉由這次的比賽，讓未來求職求學都有加分的作用，且能將所學運用，讓未來人生更加順利美好。

七、團隊成員分工說明

楊宗翰	電控設計、進度控制
蘇冠瑋	材料加工、電路測試
馬匡緯	結構設計、材料加工
洪桓駿	材料加工、電路測試

附錄：工作週報

工作週報

填寫日期 | 104 年 4 月 11 日

底盤構想設計

由於這次的比賽需要有寫字與投籃的功能，我們組打算使用 XYZ 滑軌的機械手臂，其重量比起以往做得機器人要來的重，因此底盤的設計相當的重要，必須要能夠穩固，循線、感測、寫字、投球才能夠更加準確。

底盤以 30mm*30mm 的鋁擠為材料，4 根 100cm，8 根 97cm。

工作週報

填寫日期 | 104年4月19日

底盤結構材料購買、式裝

用3cm × 3cm的鋁擠條來當整體的底盤骨架材料



工作週報

填寫日期 | 104 年 4 月 25 日

地盤改良

因為發現底盤骨架過大，所以又做裁切的加工，讓整個機台容易搬運。



工作週報

填寫日期 | 104年5月2日

地盤改良，機台移動用馬達購買和組裝

為了讓之後加裝東西更方便，我們把底做組裝了方式再作調整，改成用搭接的方式。



機器人移動用的馬達，我們選用直流馬達來帶動整台機器人。
固定方式是用鋁擠和鋁片夾住整個馬達。



工作週報

填寫日期 | 104 年 5 月 10 日

地盤輔助輪設計組裝

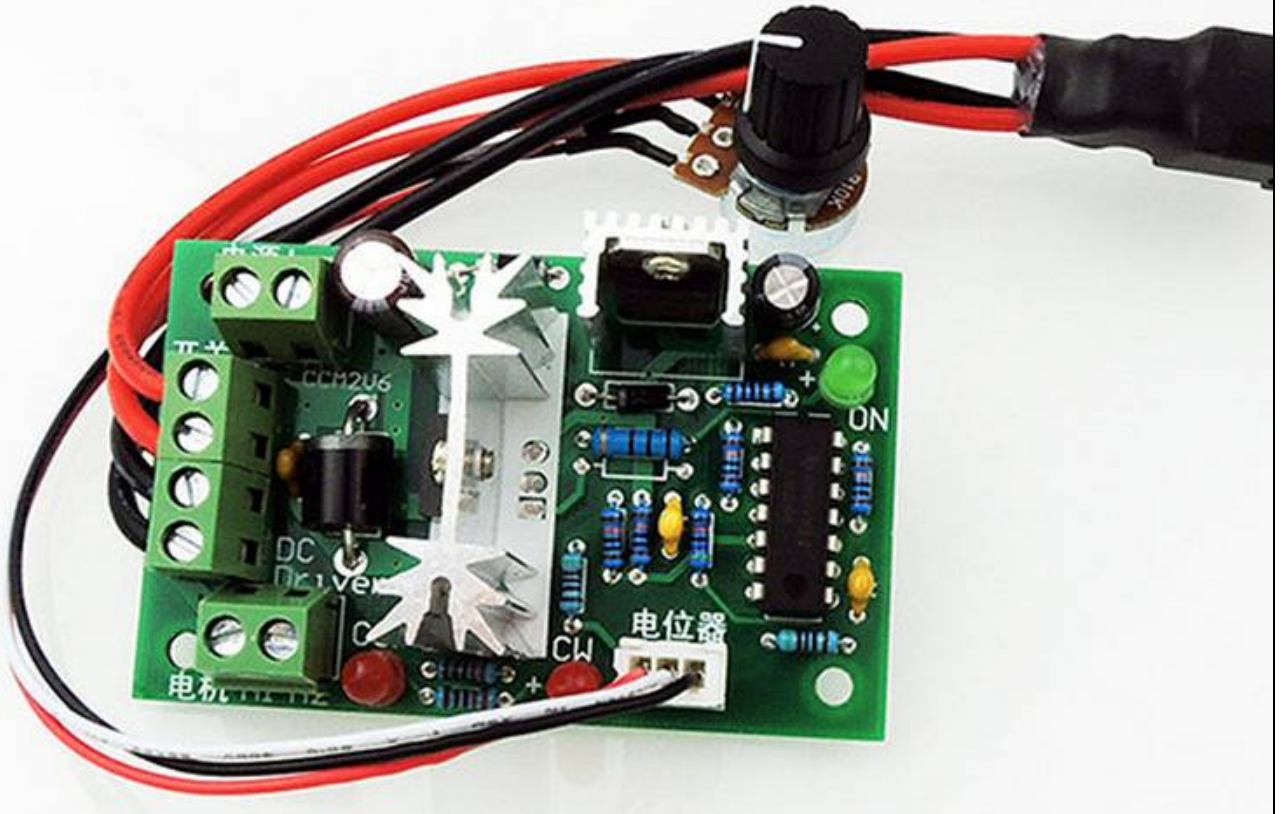
應為我們的機器人設計底盤只有兩個輪子帶動，所以在設計上要在四個編裝上補助輪，穩定機台。



工作週報

填寫日期 | 104年5月17日

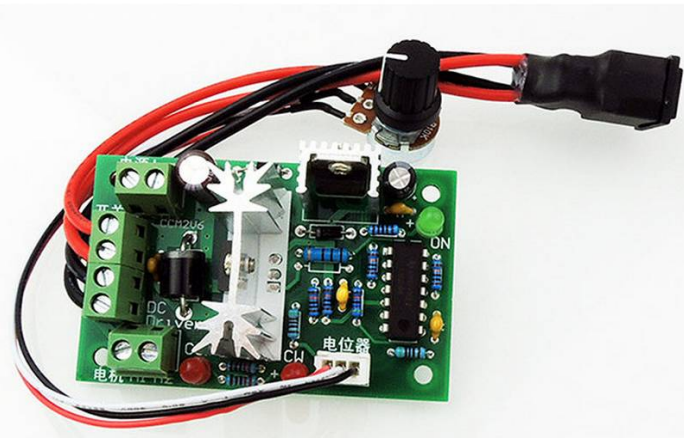
底座馬達用控制器購買，了解功能



此晶片可以做到直流馬達的轉向和轉速控制，但目前轉向是用按鈕改變，轉速是用可變電阻控制，對自動組來講條件並不理想。

填寫日期 | 104 年 5 月 24 日

底座馬達用控制器改良設計



因為我們的控制機轉向和速度的調整都不能直接用程式控制，所依需要改變一下他控制的方法。

轉向的部分比較簡單，可以改成用繼電器控制。



轉速應為控制器是使用輸入 0-5v 的電壓來調整速度，所以在網路上找到了 MCP4131 這個 IC 可以用程式控值其輸出的電壓。



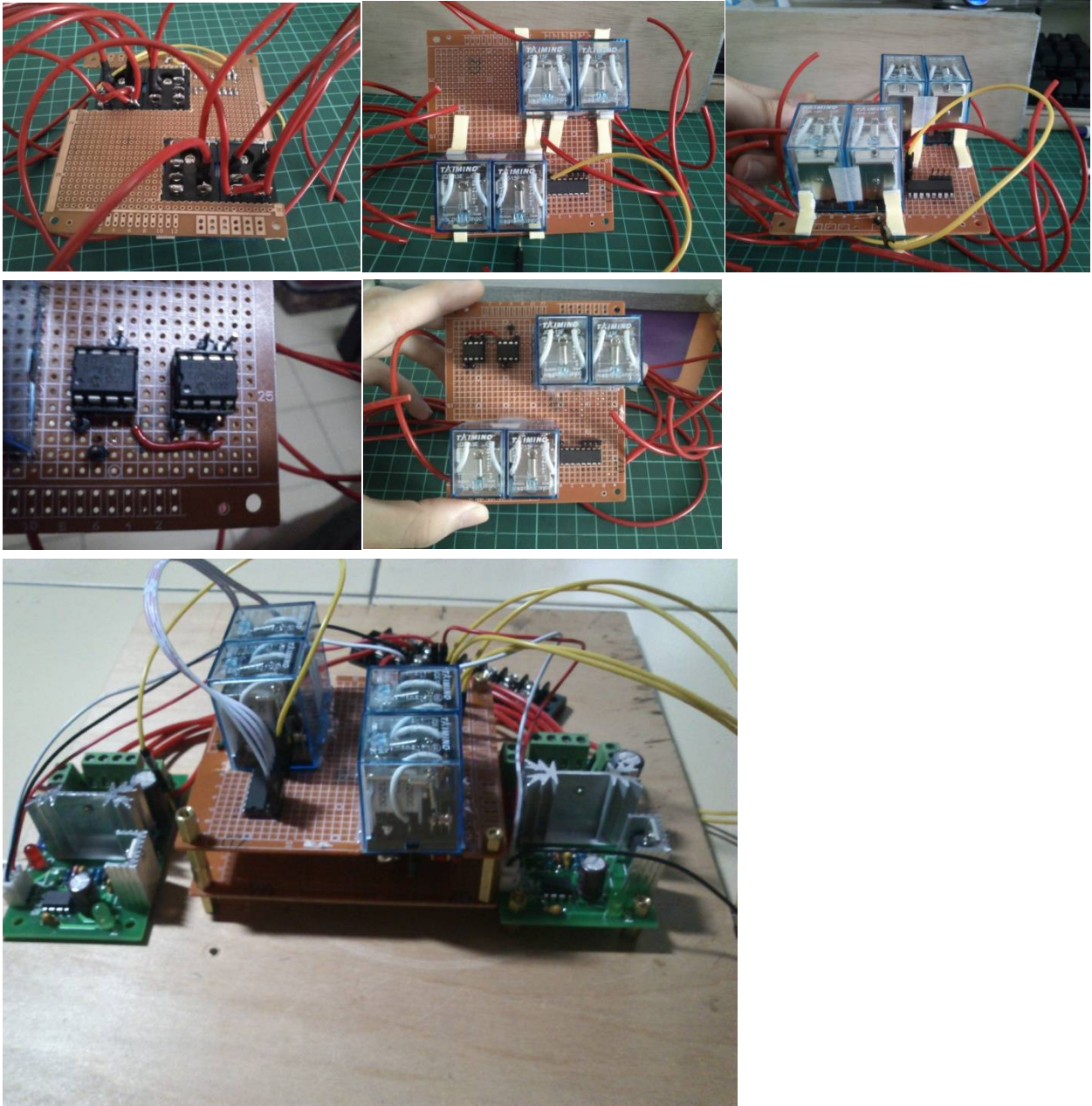
工作週報

填寫日期 | 104 年 5 月 30 日

底座馬達用控制器改良設計

底座馬達的控制用繼電器和 MCP4131 晶片。

此繼電器需要 12V 來激磁所以使用 ULN2003 晶片做控制，此晶片內有達靈頓電路。



工作週報

填寫日期 | 104年6月6日

底座輪胎購買



填寫日期 | 104年6月13日

底座輪胎組裝設計

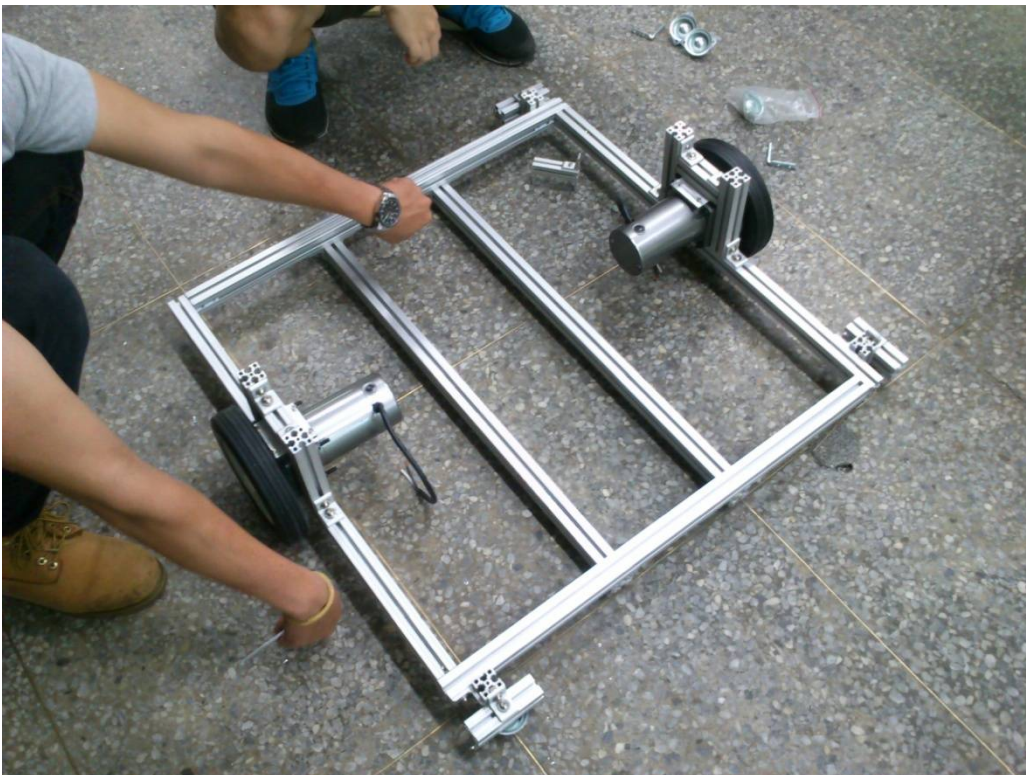
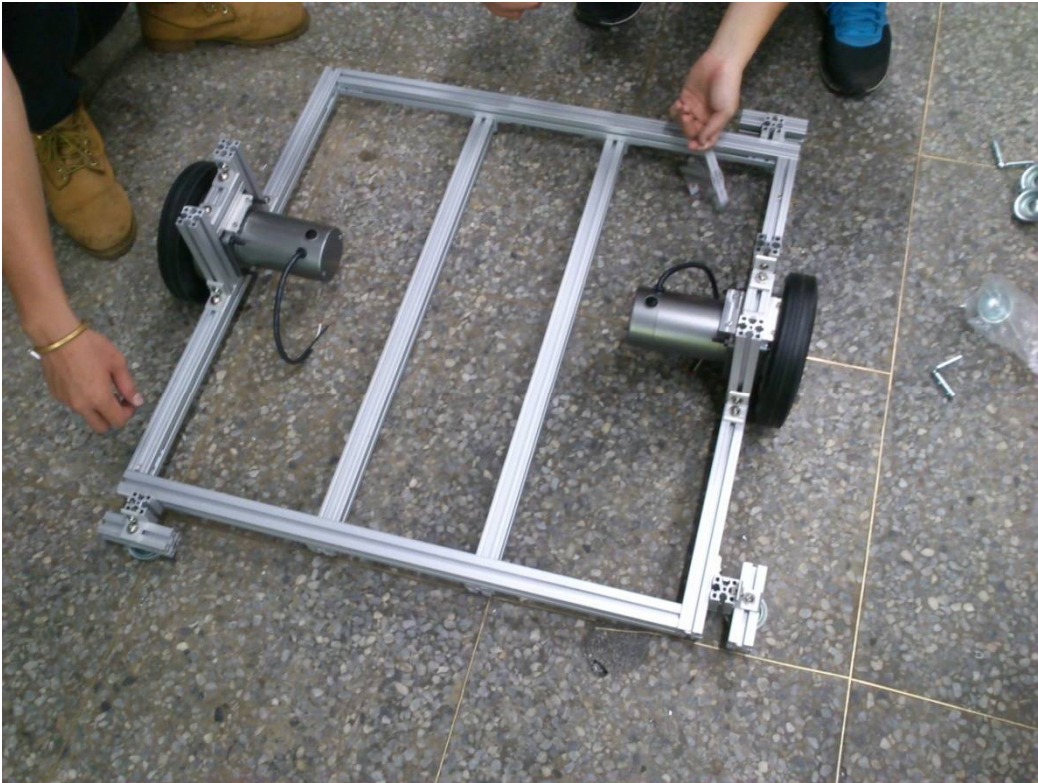


因為輪胎的材質為熱塑性塑膠，所以我們先把馬達上的鍵加熱用熔解的方法在輪胎上做出方形的凹槽。

工作週報

填寫日期 | 104年6月20日

底座輪胎組裝

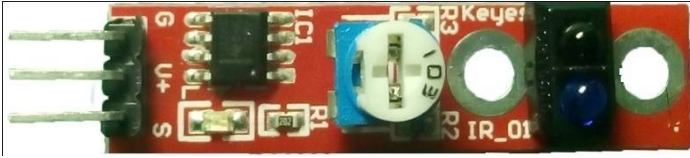


工作週報

填寫日期 | 104 年 6 月 27 日

循線設計

晶片用 TCRT5000



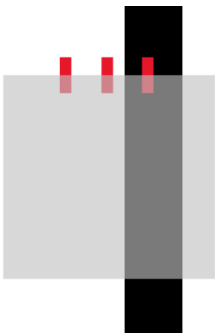
當兩邊的感測器輸出低電位 0，中間的感測器輸出高電位 1 時，執行向前程式。



當中間和右邊的感測器輸出為高電位 1，左邊為低電位 0 時，知道自走車稍微偏右，執行向右微調程式，需左轉微調時道理想同。



當左邊和中間感測器輸出為低電位 0，右邊為高電位 1 時，知道自走車需右轉，執行右轉程式，需左轉時道理相同。



工作週報

填寫日期 | 104 年 7 月 4 日

巡跡程式設計

```
arduino 循跡程式:  
#include <SPI.h>  
byte address=0x00;  
int CS1=26;  
int CS2=27;  
int forward=0;// 設定常數方便程式閱讀  
int right=1;  
int rright=2;  
int left=3;  
int lleft=4;  
int rightsensor,leftsensor,midsensor;  
//arduino pin  
int EN1=25;//定義 ic 晶片和感測器對應到 arduino 的腳位  
int A_1=24;  
int A_2=22;  
int EN2=23;  
int sensorr=28;  
int sensorl=30;  
int sensorm=32;  
void setup() {  
    // put your setup code here, to run once:  
    pinMode(EN1,OUTPUT);  
    pinMode(A_1,OUTPUT);  
    pinMode(A_2,OUTPUT);  
    pinMode(EN2,OUTPUT);  
    pinMode(CS1,OUTPUT);  
    pinMode(CS2,OUTPUT);  
    pinMode(sensorr,INPUT);  
    pinMode(sensorl,INPUT);  
    pinMode(sensorm,INPUT);  
    SPI.begin();  
    Serial.begin(9600);
```

```

}
int motorspeed1(int value)//4131 晶片控制馬達速度
{
digitalWrite(CS1,LOW);
SPI.transfer(address);
SPI.transfer(value);
digitalWrite(CS1,HIGH);
}
int motorspeed2(int value)
{
digitalWrite(CS2,LOW);
SPI.transfer(address);
SPI.transfer(value);
digitalWrite(CS2,HIGH);
}
int witch_act(void)//判斷動作的子程式
{
if (rightsensor==0&leftsensor==0&midsensor==1)
return 0;//向前
else if (rightsensor==1&leftsensor==0&midsensor==1)
return 1;//向右微調
else if (rightsensor==1&leftsensor==0&midsensor==0)
return 2;//向右
else if(rightsensor==0&leftsensor==1&midsensor==1)
return 3;//向左為調
else if(rightsensor==0&leftsensor==1&midsensor==0)
return 4;//向左
}
void forwardact(void)// 執行向前動作時控制 l293d 晶片的子程式
{
digitalWrite(EN1,HIGH);
digitalWrite(EN2,HIGH);
digitalWrite(A_1,LOW);
digitalWrite(A_2,LOW);
motorspeed1(100);
motorspeed2(50);
}

```

```

}
void rightact(void)// 執行右轉微調動作時控制 l293d 晶片的子程式
{
    digitalWrite(EN1,HIGH);
    digitalWrite(EN2,LOW);
    digitalWrite(A_1,LOW);
    digitalWrite(A_2,LOW);
    motorspeed1(50);
    motorspeed2(100);
}
void rrightact(void)// 執行右轉動作時控制 l293d 晶片的子程式
{ digitalWrite(EN1,HIGH);
  digitalWrite(EN2,HIGH);
  digitalWrite(A_1,HIGH);
  digitalWrite(A_2,LOW);
  motorspeed1(100);
  motorspeed2(50);
}
void leftact(void)// 執行左轉微調動作時控制 l293d 晶片的子程式
{digitalWrite(EN1,LOW);
  digitalWrite(EN2,HIGH);
  digitalWrite(A_1,LOW);
  digitalWrite(A_2,LOW);
  motorspeed1(30);
  motorspeed2(30);
}
void lleftact(void)//執行左轉動作時控制 l293d 晶片的子程式
{digitalWrite(EN1,HIGH);
  digitalWrite(EN2,HIGH);
  digitalWrite(A_1,LOW);
  digitalWrite(A_2,HIGH);
  motorspeed1(80);
  motorspeed2(80);
}

void loop() { // 主要程式

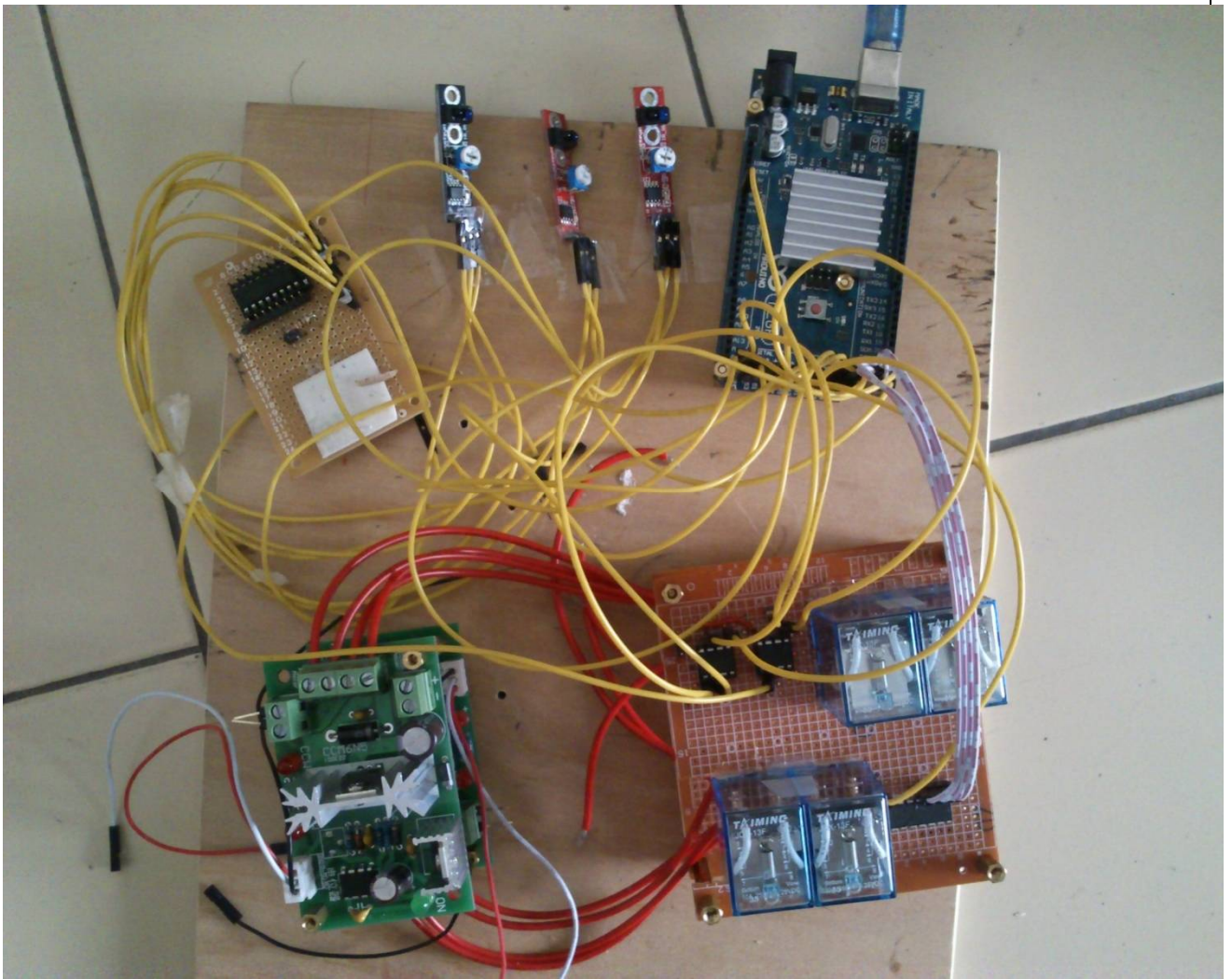
```

```

digitalWrite(EN1,LOW);
digitalWrite(EN2,LOW);

int act;
while(1)
{
rightsensor=digitalRead(sensorr); //感測器讀取
leftsensor=digitalRead(sensorl);
midsensor=digitalRead(sensorm);
act=witch_act();//執行動做判斷程式
Serial.println(act);
switch(act)
{
case 0:
{
forwardact();
break;
}
case 1:
rightact();
break;
case 2:
rrightact();
break;
case 3:
leftact();
break;
case 4:
lleftact();
break;
}
}
}

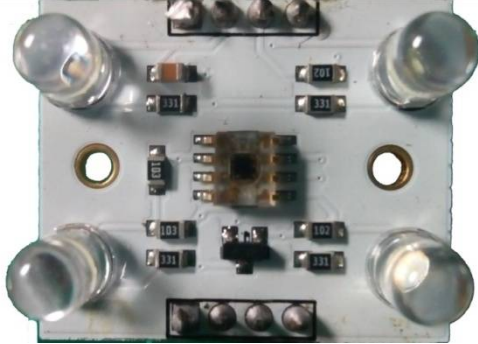
```



填寫日期 | 104 年 7 月 11 日

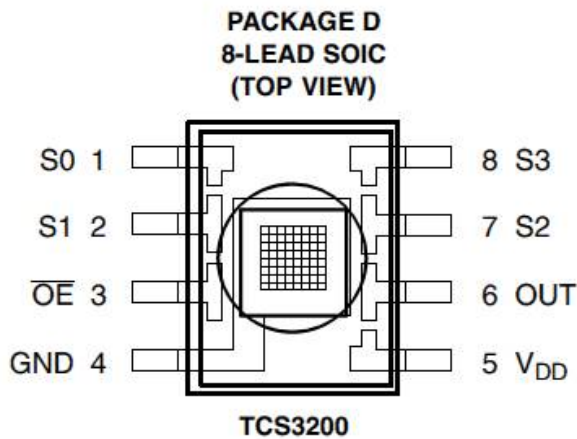
辨色設計

使用晶片:TCS3200



因為知道三原色的值我們就可以判斷出物體的真实顏色，TCS3200 有三種濾波器，分別對應三種顏色的感測，假設當選擇紅色濾波器時，他只允許紅色的光通過，藍色和綠色都被阻止，這樣就可以得到紅色光的光強；同理選擇其他的濾波器，就可以得到藍色光和綠色光的光強。通過這三個值，就可以分析投射到 TCS3200 感測器上的光的顏色。

(2)datasheet



arduino 程式

```
int s0=22;//設定 arduino 對應晶片的腳
int s1=24;
int s2=26;
int s3=28;
int OUT=2;
int count=0;//給待會會用到的變數
int selectcolor=0;
```

```

int countR=0 ,countG=0,countB=0;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(3, OUTPUT);
}
void tcs3200()//  tcs3200 晶片程式
{
  selectcolor=0;
  digitalWrite(s0, HIGH); //頻率使用最高
  digitalWrite(s1, HIGH);
  digitalWrite(s2,LOW);//從紅色的強度開始測量
  digitalWrite(s3,LOW);
  attachInterrupt(0,countpause,CHANGE);// 當有訊號改變時執行程式， 用
attachInterrupt 可以不漏掉任何訊號
  timer0_init();
}
void countpause()
{
  count++;//每次訊號改變都加一用來判斷顏色強度(最後此值越高強度越強)
}
void timer0_init(void)// 設定 timer 讓每個濾波顏色測試的時間都一樣長
{
  TCCR2A=0x00;
  TCCR2B=0X07;
  TCNT2=100;
  TIMSK2=0x01;
}
ISR(TIMER2_OVF_vect)//輸出量測到的顏色對應強度
{
  TCNT2=100;
  selectcolor++;
}

```

```

if(selectcolor==1)
{
  countR=count;
  Serial.print("red=");
  Serial.println(countR);
  digitalWrite(s2,HIGH);//紅色測完測綠色
  digitalWrite(s3,HIGH);
}
else if(selectcolor==2)
{
  countG=count;
  Serial.print("green=");
  Serial.println(countG);
  digitalWrite(s2,LOW);//綠色測完測藍色
  digitalWrite(s3,HIGH);
}
else if(selectcolor==3)
{
  countB=count;
  Serial.print("blue=");
  Serial.println(countB);
  Serial.println("\n");
  digitalWrite(s2,LOW);//設定回測量紅色
  digitalWrite(s3,LOW);
}
else if(selectcolor==4)
{
  selectcolor=0;
}
count=0;
}
void loop() {
  tcs3200();
  while(1);
}

```

(4)量測結果

測量藍色時的值

red=350

green=335

blue=1118

測量紅色時的值

red=769

green=253

blue=1222

測量綠色時的值

red=227

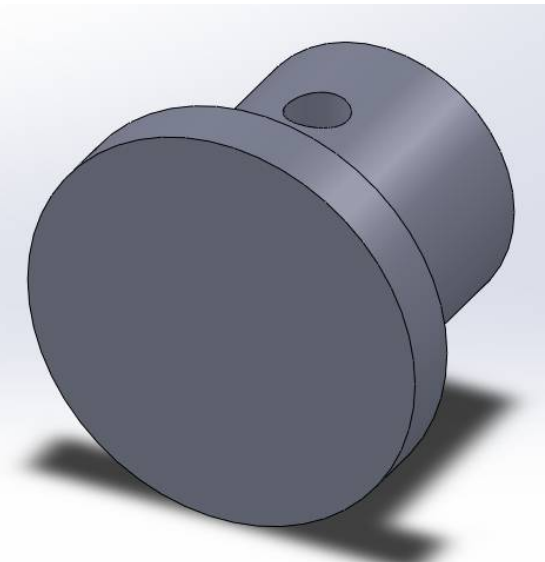
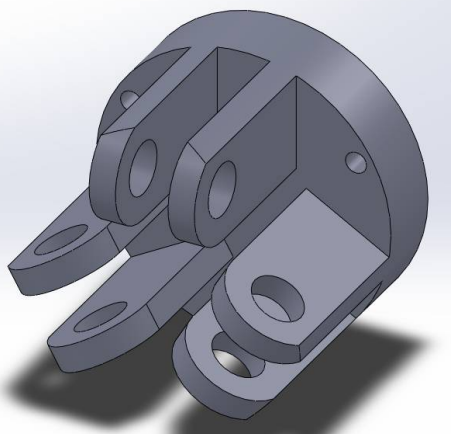
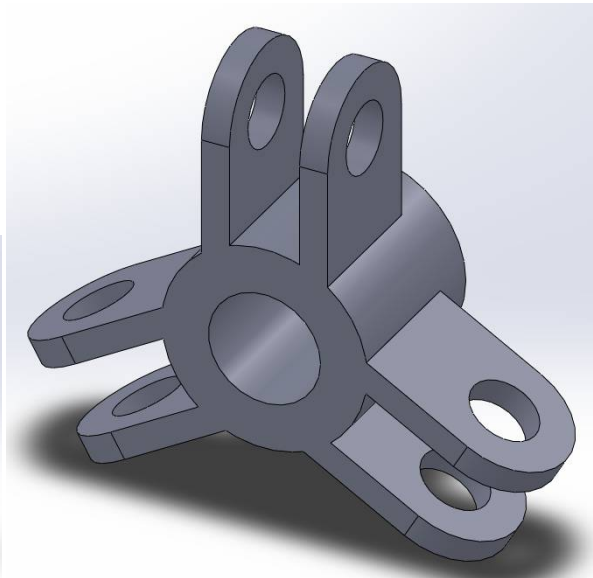
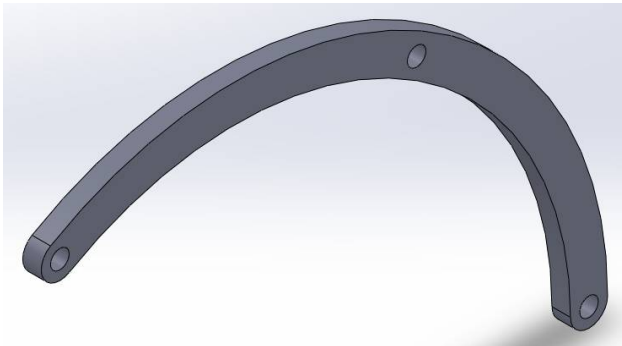
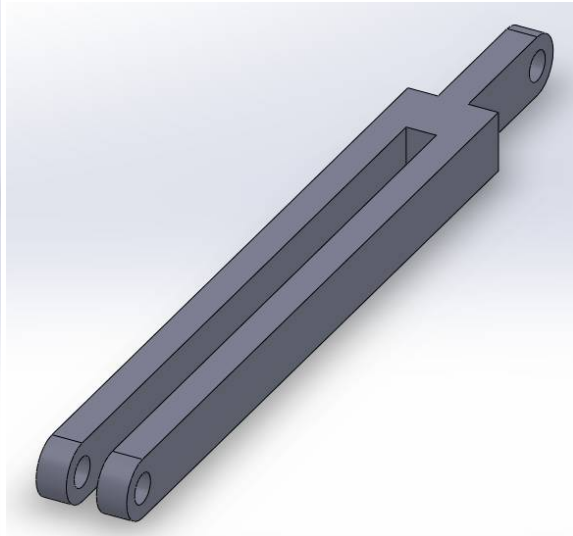
green=222

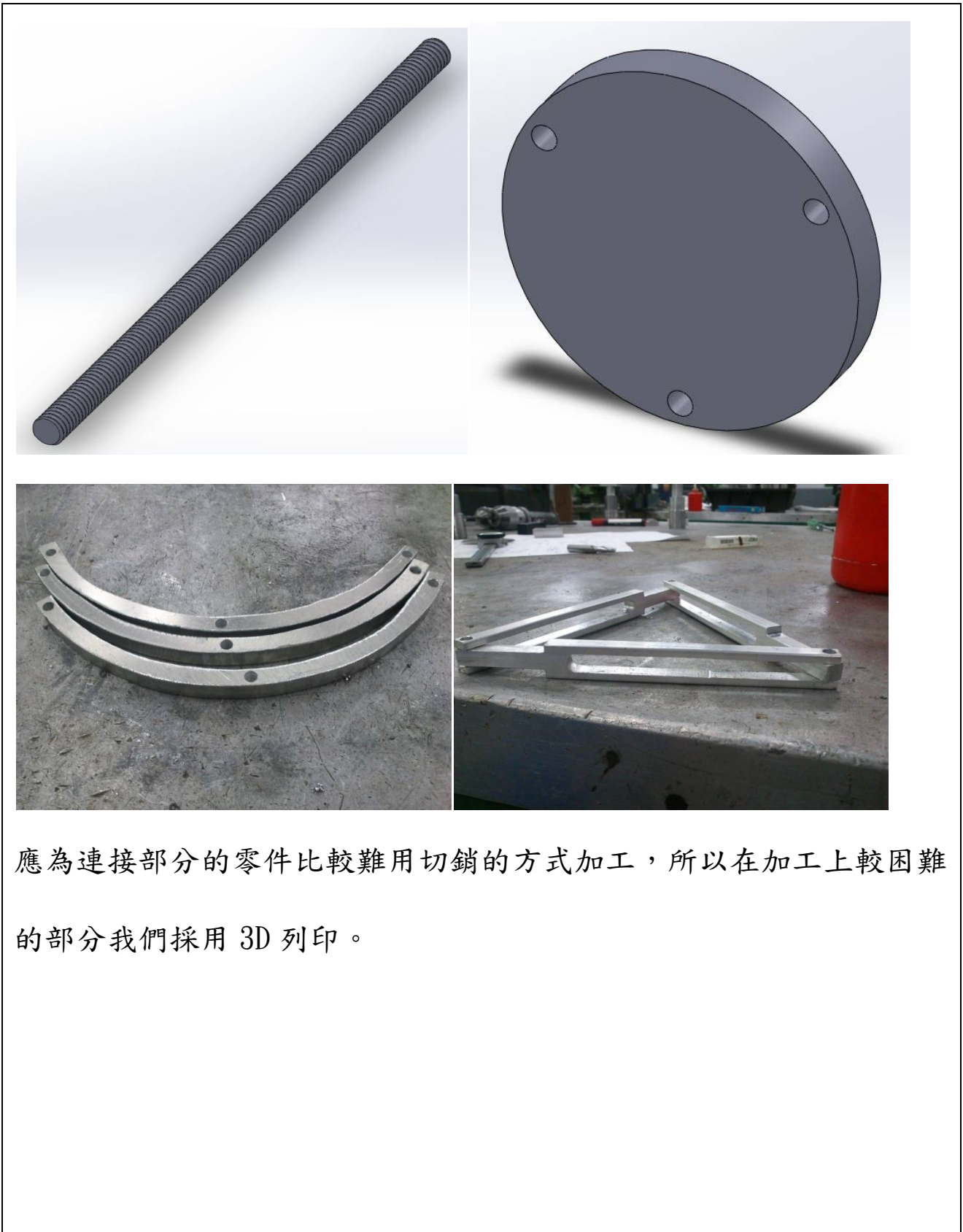
blue=720

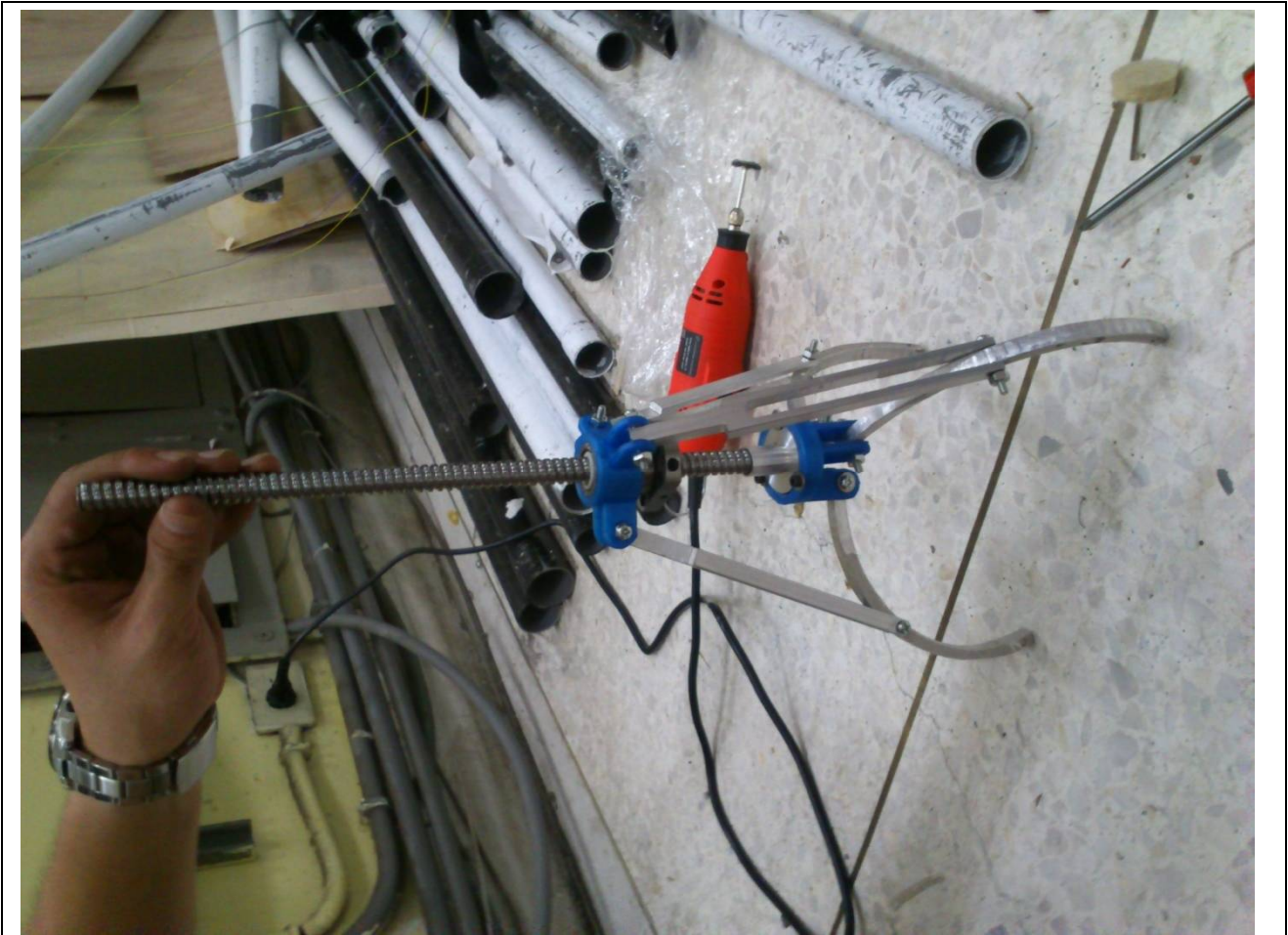
工作週報

填寫日期 | 104年7月18日

夾爪設計、製造



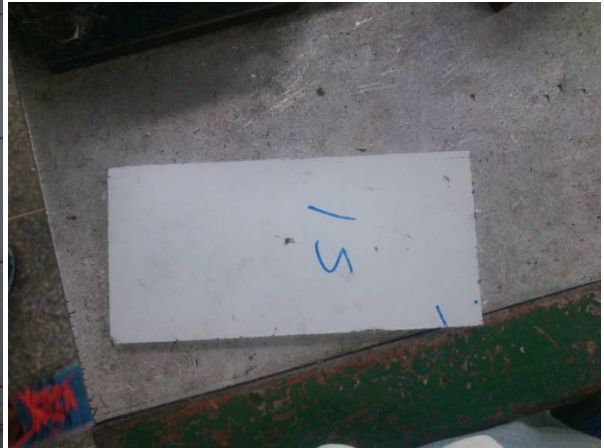




工作週報

填寫日期 | 104 年 7 月 25 日

線性滑軌購買、連接





工作週報

填寫日期 | 104年8月1日

底盤改造

因為發現自動組在開始時機台以今夾筆且比的長度也算在機台長度中，所以底盤又在改造。



