

國立龍潭高級中學學生專題製作競賽

製作專題名稱：Arduino 四軸飛行器

科別：電機科

班級：三年級

學生姓名：賴思齊、

指導老師：劉哲雄老師

一、現況調查

二、四軸飛行器又稱四旋翼、四轉子，是一種多軸飛行器，有四個旋翼來懸停、維持姿態及平飛。和固定翼飛機不同，它通過旋翼提供的推力使飛機升空。它的四個旋翼大小相同，分布位置接近對稱。對於簡單的設計來說，僅僅通過調整不同旋翼之間的相對速度來調節不同位置的推力，並克服每個旋翼之間的反扭力矩，就可以控制飛機維持姿態、或完成各種機動飛行。這一點和直升機不同，常見的直升機有兩個旋翼，尾槳只起到抵消主旋翼產生的扭矩，控制飛機偏航運動的功能。

三、早期飛機設計中，四軸飛行器被用來解決旋翼機的扭矩問題。主副旋翼的設計也可以解決扭矩問題，但副旋翼不能提供升力，效率低。且使用尾槳的設計在結構上比使用多旋翼要複雜很多。因此四軸飛行器是最早的一批比空氣重的垂直起降飛行器。^[1]但是早期的型號性能很差，難於操控和大型化。^[2]尤其是油門的控制難以做到精確和迅速。更加上傳統的直升機構型在巡航時的效率要優於四軸飛行器。故此在直升機的技术問題得以解決後，四軸飛行器迅速從飛行器設計方式中銷聲匿跡了

四、近來四軸飛行器在無人機領域獲得了新生。隨著智慧型手機的發展，帶動電子陀螺儀、GPS、電傳飛行控制系統的發展，並以及油門響應速度迅速的

電動機作為動力系統，克服了四軸飛行器的主要缺點。四軸飛行器飛行穩定，操控靈活，可以在戶內和戶外使用。^{[3][4]}和直升機相比，它有許多優點：它的旋翼角度固定，結構簡單。^[5]每個旋翼的葉片比較短，葉片末端的線速度慢，發生碰撞時衝擊力小，不容易損壞，對人也更安全。有些小型四軸飛行器的旋翼有外框，避免磕碰。^[6]

五、因為四軸飛行器體積小、重量輕，攜帶方便，能輕易進入人不易進入的各種惡劣環境。常用來製作模型，也用來執行航拍電影取景、實時監控、地形勘探等飛行任務。^{[7][8]}在 2009 年電影《三傻大鬧寶萊塢》中，就有描述當時剛開始萌芽的多軸飛行器，其中男主角從垃圾桶撿回後來拍到學生慘劇的那部，便是四軸飛行器的一種。

(引自 wiki)

設計動機與目的

我們的設計構想是來自有一次在瀏覽外國網站時,有一個名為 TEDGlobal 的機器人實驗室剛好發布了一個新影片,我們便迫不及待的點進去看,在那部影片中的四軸飛行器他的動作流暢度跟那個靈敏度,感覺實在不像是機器反而比較像是他從小養到大的寵物一樣,在當時我們就想說我們也要做一個自己的四軸飛行器,所以才有了這次的專題

二,創意發想歷程



一. 決定專題內容:在剛開始時我們也是很苦惱的思索了好久,但是當我們一看到那部介紹影片時,在當下就有一種[對,就是這個了]的感覺,然後我們便毫不猶豫地開始專題準備手續

二. 思考專題基本架構:先在腦中構想四軸飛行器的大製概略,再在紙上畫出基本的圖形

三. 資料查詢及材料購買:在網路上搜尋四軸飛行器,就會有許多的相關資料出現,我們就從他們的資料中得到我們所需要的程式構想;而我們所需的材料大部分是從露天商場購買的,而有部分的組件則是從電子材料行所進購買的

四. 程式及硬體設計:硬體是利用 MegaSise 的 Arduino 在接收端,也就是四軸飛行器的那一邊做為基底, 旋槳的加速與減速則是用 Gy80 的加速度計模組,再接一塊 apc220 來與遙控器做訊號的交流,,而遙控器是用中型的 Arduino 做為發送端,一樣也要接上一個 apc220 來跟飛行器上的 apc220 做交流

五. 加工-->完成

三.應用潛能分析:依據現有的組件及電子零件是可以將此項專題完成的,而在未來的發展及應用上也是非常有用處的,像是在戰區進行索敵.在崎嶇山區

進行受傷人員的搜索.還有在災區進行物資的補給等等

組件明細:

| 名稱 | 數量 | 金額 |
|-------------------------------|----|------|
| 高硬紅白 450mm 四軸機 架 | 1 | 350 |
| 電池 3s 11.1v 5800mah 25C | 1 | 950 |
| 多軸專用 20A 電 調 B_LHeli 軟體 | 4 | 1200 |
| 110~240VAC 電壓充電器 | 1 | 299 |
| APC220 | 2 | 1200 |
| GY 80 | 1 | 360 |
| XXD 新西達 2212/1400KV | 4 | 800 |
| 1045 兩葉正反 轉槳_送轉換圈 | 2 | 60 |

總價:5270NTD

程式解析:

遙控端

```
int pinLX = A0;
```

```
int pinLY = A1;
```

```
int pinRX = A2;
```

```
int pinRY = A3;
```

```
int valLX, posLX;
```

```
int valLY, posLY;
```

```
int valRX, posRX;
```

```
int valRY, posRY;
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
valLX = analogRead(pinLX);
```

```
valLY = analogRead(pinLY);
```

```
valRX = analogRead(pinRX);
```

```
valRY = analogRead(pinRY);
```

```
posLX = map(valLX, 0, 923, 3,-3);
```

```
posLY = map(valLY, 0, 1023, 3,-3);
```

```
posRX = map(valRX, 0, 1023, 3,-3);
```

```
posRY = map(valRY, 0, 923, 3,-3);
```

```
Serial.print(posLX);
```

```
Serial.print(",");
```

```
Serial.print(posLY);
```

```
Serial.print(",");
```

```
Serial.print(posRX);
```

```
Serial.print(",");
```

```
Serial.println(posRY);
```

```
delay(100);
```

```
Serial.println("n");
```

```
}
```

飛行器端

```
#include <Servo.h>
```

```
Servo servoM1,servoM2,servoM3,servoM4;
```

```
int posM1,posM2,posM3,posM4;
```

```
int valLX, posLX, pasLX;
```

```
int valLY, posLY, pasLY;
```

```
int valRX, posRX, pasRX;
```

```
int valRY, posRY, pasRY;
```

```
int P;
```

```
void setup() {
```

```
servoM1.attach(2);
```

```
servoM2.attach(3);
```

```
servoM3.attach(4);
```

```
servoM4.attach(5);
```

```
servoM1.write(30);
```

```
servoM2.write(30);

servoM3.write(30);

servoM4.write(30);

delay(1500);

servoM1.write(360);

servoM2.write(360);

servoM3.write(360);

servoM3.write(360);

Serial.begin(9600);

Serial1.begin(9600);

}

void loop() {

P = (posM1+posM2+posM3+posM4)/4;

Serial.println(P);

servoM1.write(posM1);

servoM2.write(posM2);

servoM3.write(posM3);

servoM4.write(posM4);

while(Serial1.available()>0);
```

```
{  
  
valLX = Serial1.parseInt();  
  
valLY = Serial1.parseInt();  
  
valRX = Serial1.parseInt();  
  
valRY = Serial1.parseInt();  
  
}  
  
posM1 = posM1 + valLX;  
  
posM2 = posM2 + valLX;  
  
posM3 = posM3 + valLX;  
  
posM4 = posM4 + valLX;  
  
if(valRX > 0){  
  
posM3 = posM3 + valRX;  
  
posM4 = posM4 + valRX;  
  
posRX = posRX + valRX;  
  
}  
  
if(valRX < 0){  
  
posM1 = posM1 - valRX;  
  
posM2 = posM2 - valRX;  
  
pasRX = pasRX - valRX;
```

```
}  
  
if(valRX == 0){  
  
    posM3 = posM3 - posRX;  
  
    posM4 = posM4 - posRX;  
  
    posRX = 0;  
  
    posM1 = posM1 - pasRX;  
  
    posM2 = posM2 - pasRX;  
  
    pasRX = 0;  
  
}  
  
if(valRY > 0){  
  
    posM1 = posM1 + valRY;  
  
    posM3 = posM3 + valRY;  
  
    posRY = posRY + valRY;  
  
}  
  
if(valRY < 0){  
  
    posM2 = posM2 - valRY;  
  
    posM4 = posM4 - valRY;  
  
    pasRY = pasRY - valRY;  
  
}
```

```
if(valRY == 0){

posM1 = posM1 - posRY;

posM3 = posM3 - posRY;

posRY = 0;

posM2 = posM2 - pasRY;

posM4 = posM4 - pasRY;

pasRY = 0;

}

if(vallY > 0){

posM2 = posM2 + vallY;

posM3 = posM3 + vallY;

posLY = posLY + vallY;

}

if(vallY < 0){

posM1 = posM1 - vallY;

posM4 = posM4 - vallY;

pasLY = pasLY - vallY;

}

if(vallY == 0){
```

```
posM2 = posM2 - posLY;
```

```
posM3 = posM3 - posLY;
```

```
posLY = 0;
```

```
posM1 = posM1 - pasLY;
```

```
posM4 = posM4 - pasLY;
```

```
pasLY = 0;
```

```
}
```

```
if(posM1<30){
```

```
posM1=30;
```

```
}
```

```
if(posM2<30){
```

```
posM2=30;
```

```
}
```

```
if(posM3<30){
```

```
posM3=30;
```

```
}
```

```
if(posM4<30){
```

```
posM4=30;
```

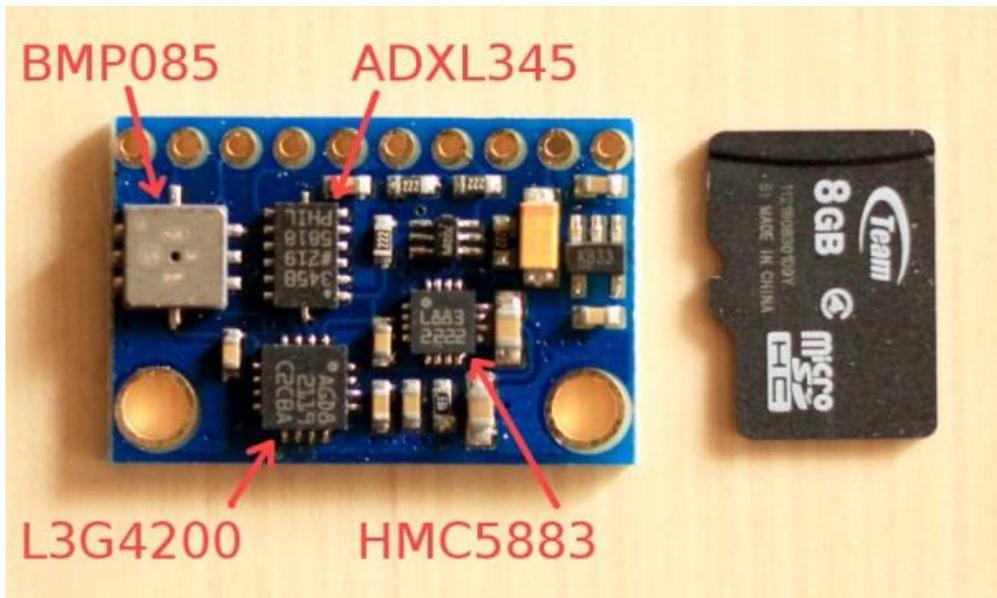
```
}
```

```
if(posM1>360){  
  
posM1=360;  
  
}  
  
if(posM2>360){  
  
posM2=360;  
  
}  
  
if(posM3>360){  
  
posM3=360;  
  
}  
  
if(posM4>360){  
  
posM4=360;  
  
}  
  
if(Serial1.read()=='n'){ }  
  
}
```

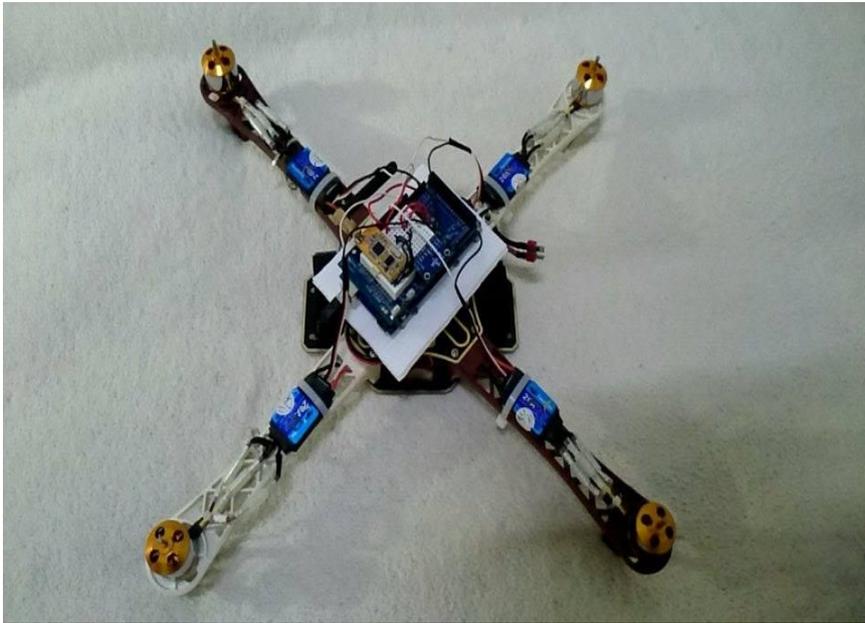
APC220(如下圖



GY 80(如下圖)



作品說明圖說

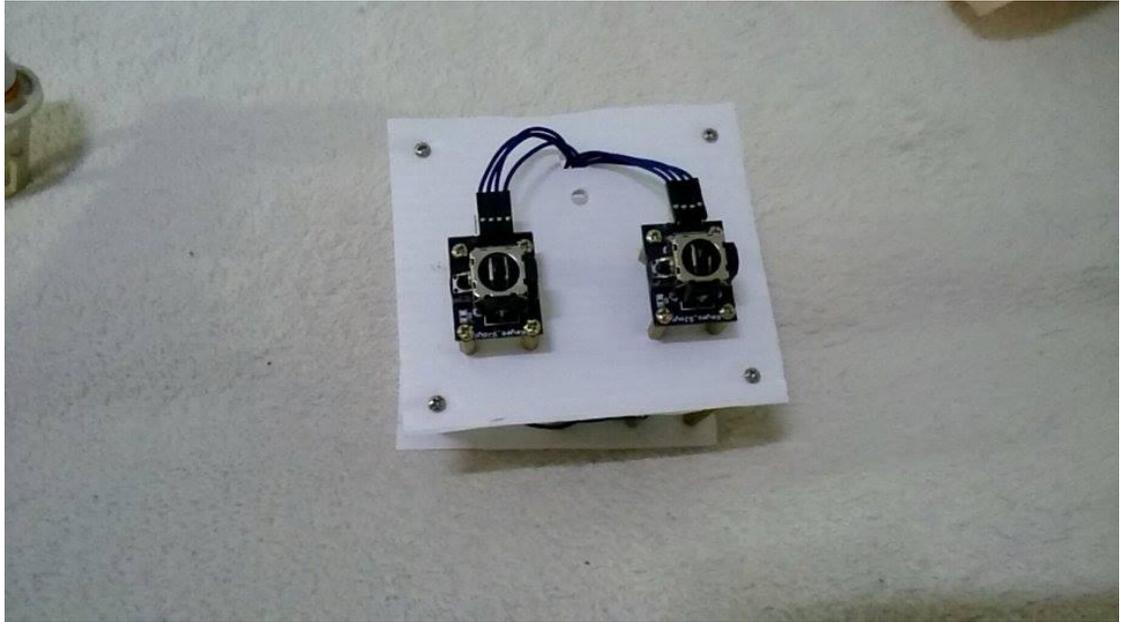


在四個角落的就是新西達 2212/1400KV 的馬達,出力很夠;

在中間的就是 Arduino MEGA,在上面有一個像天線的組件,那一顆就是

apc220;在四個支架中央的就是 20A 電調,有足夠的持續輸出能力及短時間

的大電流供應



這就是遙控器,現在看到的是 XY 搖桿,左邊的是油門;上是加速,下是減速;右邊的是傾斜方向;而在他底下也有一顆 Arduino 那顆就是讓 apc220 發送端裝置在上面的

參考資料:

四軸飛行器:

<http://zh.wikipedia.org/wiki/%E5%9B%9B%E8%BD%B4%E9%A3%9E%E8%A1%8C%E5%99%A8>

GY-80 模組:

<http://wukcsoft.blogspot.tw/2014/06/read-gy-80.html>