

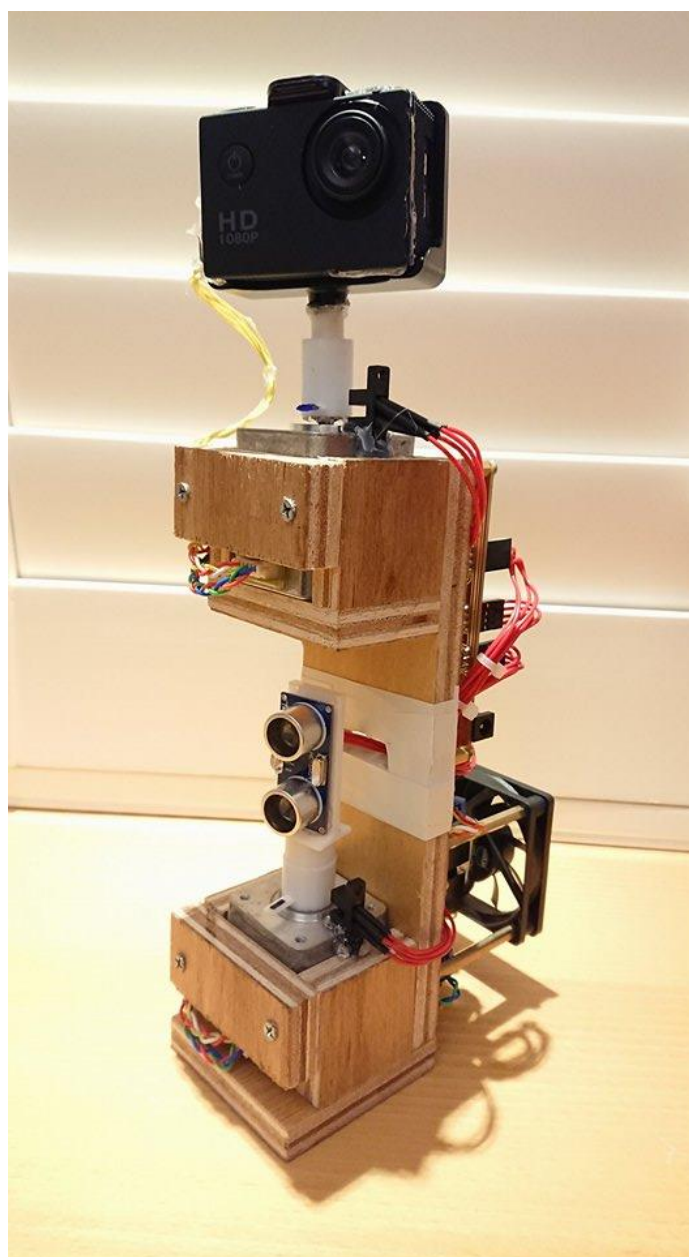
全國高級中等學校專業群科 107 年專題及創意製作競賽

「專題組」作品說明書

群 別：電機與電子群

作品名稱：自動追蹤攝影機

關 鍵 詞：超音波感測器、自動追蹤、攝影機



目錄

壹、 摘要	1
貳、 研究動機	1
參、 研究方法	2
一、元件介紹	2
二、研究過程	6
肆、 研究結果	16
伍、 討論	18
陸、 結論	19
柒、 參考資料及其他	19

自動追蹤攝影機

壹、摘要

操作攝影機，是一種既無聊又費時的工作，因此我們研製出[自動追蹤攝影機]，我們將超音波感測器裝在不斷正反轉的步進馬達上，因此超音波感測器能夠感知不同角度的距離，藉此推算物體的位置，偵測到物體後再將物體位置回傳給控制器，在經由步進馬達控制攝影機的旋轉角度，實現自動追蹤的功能，自動追蹤攝影機能夠完全自動操作，比傳統的人工操作更方便也更有效率。這個專題結合了距離感測、角度計算等技術。不但佔地小而且堅固耐用。可以運用在學校演講廳、教室、會議室等場所。不但不會干擾演講者和擋住觀眾視線，又可以自動調整攝影機視角，使攝影師的工作完全自動化。

貳、研究動機

在學校週會的時候，學校時常會請外面的講師到大禮堂進行各式各樣的演講。而每次在演講台下，都會需要負責攝影的人員無時無刻調整攝影機的拍攝角度，而且技術不佳的攝影師可能讓拍攝的畫面模糊不清或者是拍攝的視角完全錯誤。

為了提升周會的演講品質，我們決定研發能夠透過超音波感測器偵測演講者的方位，並自動控制攝影機視角與畫面縮放的裝置，使操作攝影器材所需的人力能夠被完全取代。

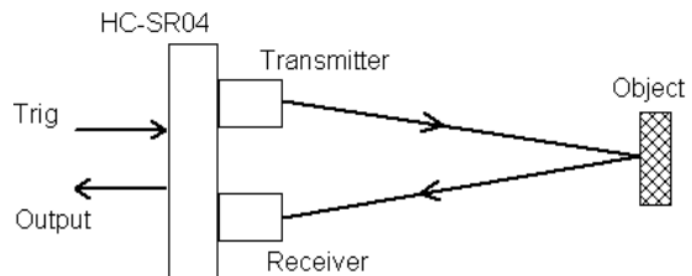
目前雖然有使用影像辨識進行自動追蹤的裝置，但影像辨識的裝置需要強大的處理器與演算法，而且容易受到光線的干擾，因此我們決定採用更簡單更可靠的方式來實現自動追蹤的功能，製作出成本能大幅下降，穩定度卻能提高的自動追蹤攝影機。

參、研究方法

一、元件介紹

(一) 超音波感測器(HC-SR04)

超音波感測器(見圖(1))其原理如圖(2)所示，發射端應用壓電效應產生超音波，再由接收端接收聲波訊號，藉由量測超音波在感應器與物體之間往返經過的時間，利用音速公式計算出物體的距離。由於超音波從發射到返回是兩段距離，因此在計算時必須將結果除以 2 才是正確的物體距離。音速非常的快(大約每秒 340 公尺)，所以可以在短時間內進行多次量測，對自動追蹤攝影機來說是非常適合的感測元件。



圖(2) 超音波感測器原理

(二) 步進馬達(DST28)

步進馬達如圖(3)可以藉由定子輪流激磁來旋轉。擁有旋轉角度精準與靜止轉矩大的特性，因此可以快速轉動也可以立即停止。



圖(3) 步進馬達圖

(三) 馬達驅動模組(L298N)

步進馬達消耗的功率很大，單晶片無法獨立驅動。因此需使用馬達驅動模組(如圖(4))透過 H 橋式電路控制步進馬達，並提供步進馬達所需的大電流。



圖(4)L298N

(四) 光遮斷器(LP-1952)

這是一個藉由光線的遮斷來啟斷的按鈕，分為發送和接收端兩部份，外觀如圖(5)。可當作無接觸點的極限開關，以減少因為摩擦造成系統不穩定的現象。



圖(5) LP-1952

(五) 攝影機(SJ8000+)

這是一款可拆卸的小型攝影機如圖(6)，擁有體積小、重量輕、電池續航力強以及畫質高等優點，我們修改了其內部電路，使攝影機可以透過外部電路控制。



圖(6) SJ8000+

(六) 光耦合器(PC-817)

光耦合器(見圖(7))透過光為媒介傳輸訊號，具有電器隔離的特性，就算電路準位不同也可以控制，非常適合用來控制外部電路，這次專題使用的攝影機就是使用此原件來控制攝影機的開機、錄影、放大、縮小。



圖(7) PC-817

(七) 風扇

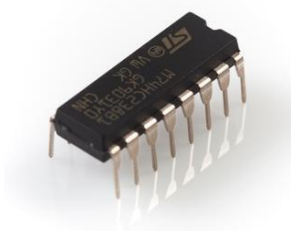
由於馬達驅動模組在運作的時候會產生高熱，因此必須要加裝風扇圖(8)強制空氣對流使驅動器溫度降低。



圖(8) 風扇外觀

(八) 三對八解碼器(74HC238)

由於單晶片的輸出接腳不足，因此使用三對八解碼器如圖(9)可以擴充可使用的接腳。按照真值表圖(10)的邏輯，可用三支接腳的訊號，控制八個輸出。



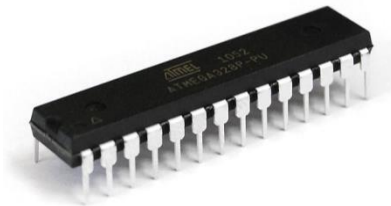
圖(9) 74HC238

Inputs						Outputs							
E1	E2	E3	A0	A1	A2	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	X	X	X	X	X	L	L	L	L	L	L	L	L
X	H	X	X	X	X	L	L	L	L	L	L	L	L
X	X	L	X	X	X	L	L	L	L	L	L	L	L
L	L	H	L	L	L	H	L	L	L	L	L	L	L
L	L	H	L	L	L	L	H	L	L	L	L	L	L
L	L	H	L	H	L	L	L	H	L	L	L	L	L
L	L	H	H	L	L	L	L	L	H	L	L	L	L
L	L	H	H	L	H	L	L	L	L	L	H	L	L
L	L	H	L	H	H	L	L	L	L	L	L	H	L
L	L	H	H	H	H	L	L	L	L	L	L	L	H

圖(10)三對八解碼器真值表

(九) 單晶片(ATMEGA328P-PU)

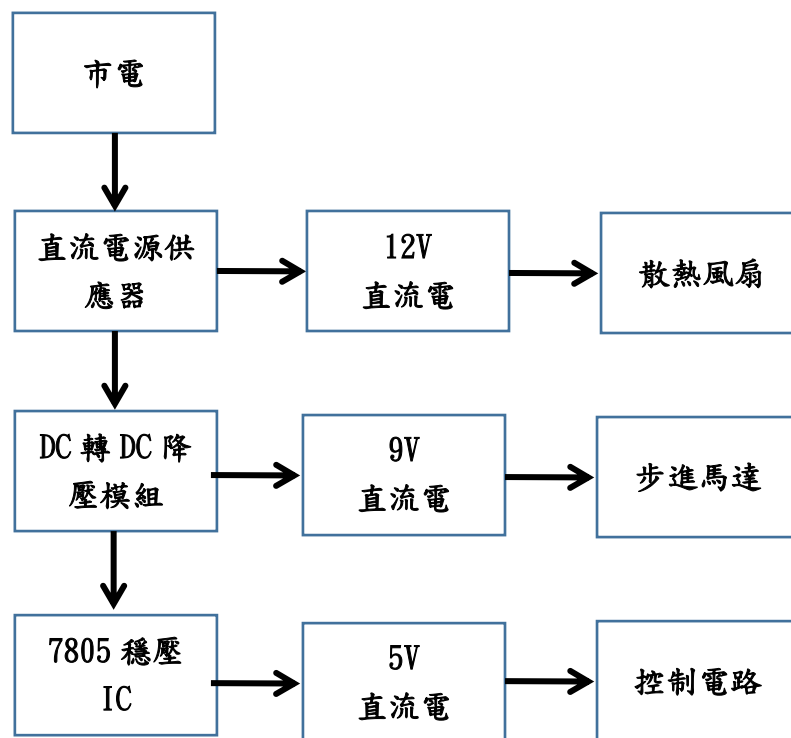
此元件是控制此裝置的核心元件，各個部件的動作都由此控制，其外觀如圖(11)。



圖(11) ATMEGA328P-PU

(十) 電源供應系統

由於機器中每個元件的工作電壓不相同，因此我們分成三階段來降壓，讓每個元件都能在最合適的電壓下工作，其供電系統流程如圖(12)，我們分別使用的電源供應器圖(13)、DC 轉 DC 降壓模組圖(14)、7805 穩壓 IC 圖(15)來進行降壓。



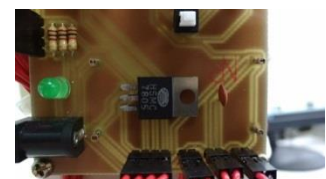
圖(12)供電系統流程圖



圖(13) 電源供應器



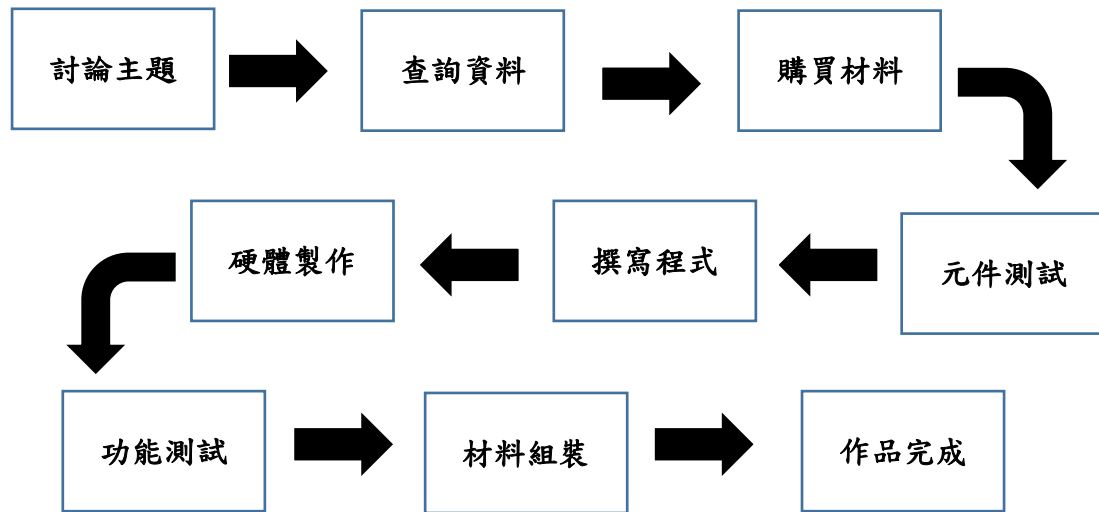
圖(14) DC 轉 DC 降壓模組



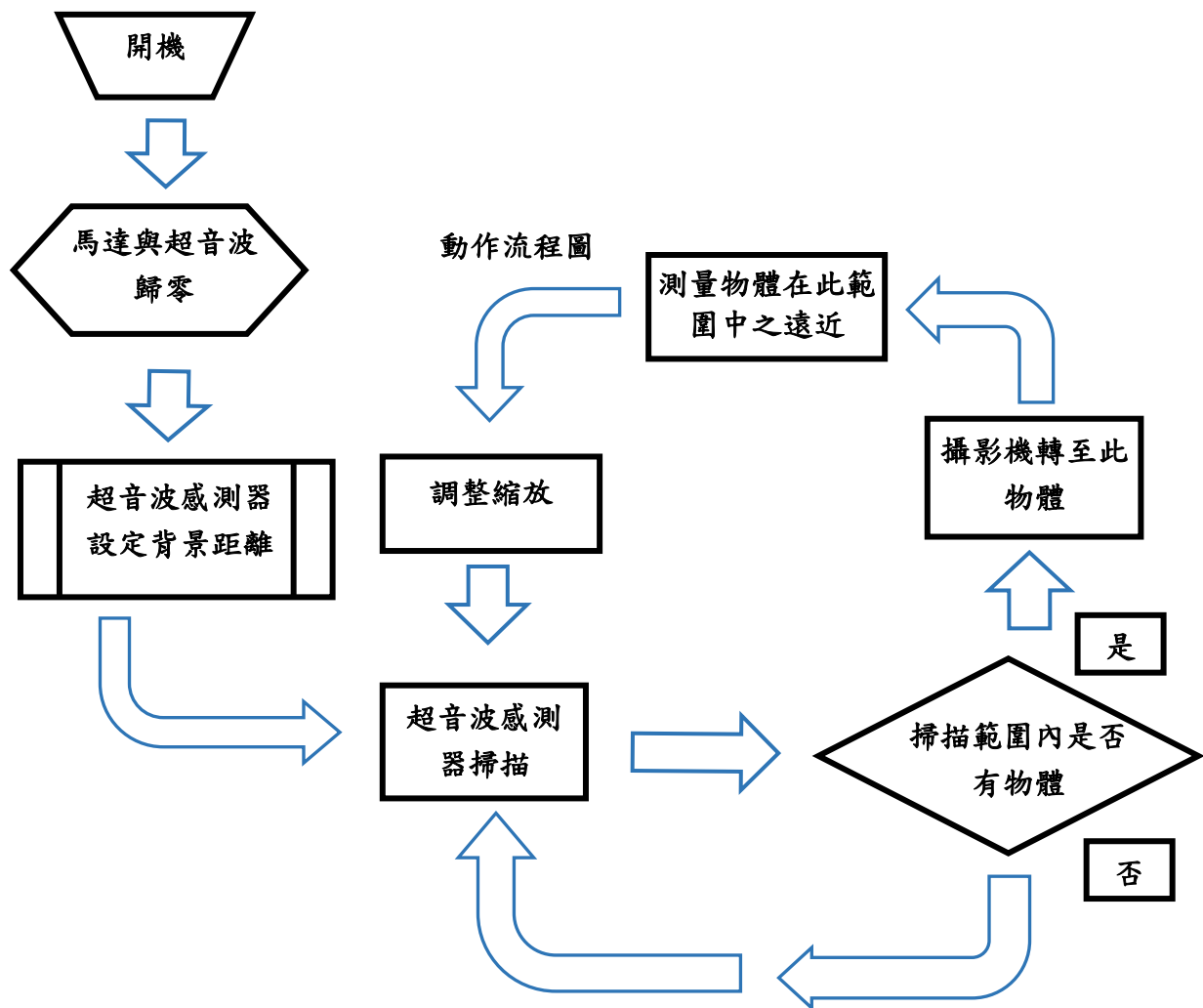
圖(15) 7805 穩壓 IC

二、研究過程

專題的製作過程如圖(16)，而我們設計的動作架構如圖(17)。



圖(16)製作流程圖



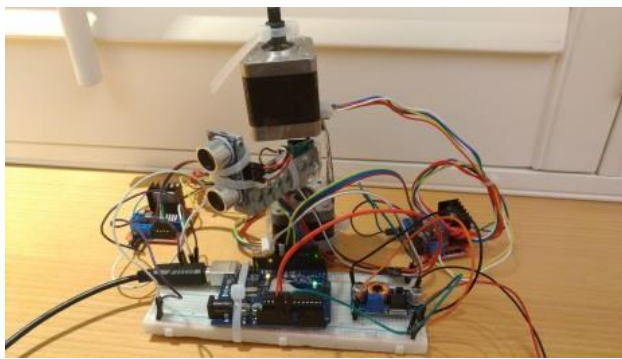
圖(17)動作流程圖

(一)硬體

硬體的部分大致分為三個部分，下層為超音波掃描裝置
上層為攝影機視角控制系統，後半部為控制電路

1. 木板製作

一開始用鐵架來支撐兩個步進馬達的重量如圖(18)，但後來發現發現兩顆步進馬達運作時，造成很大的震動，所以後來用木板來支撐改善問題如圖(19)。



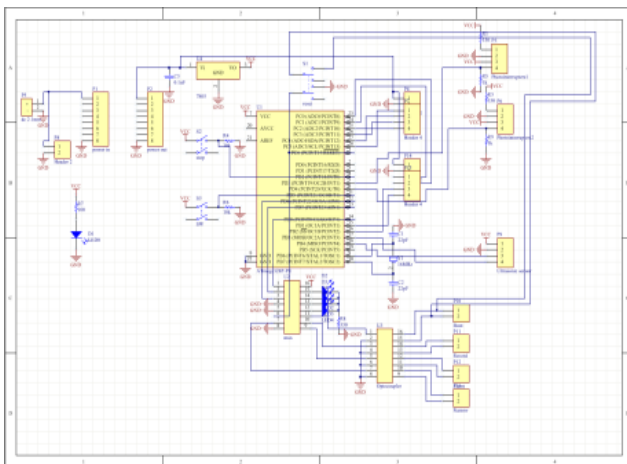
圖(18)結構改善前



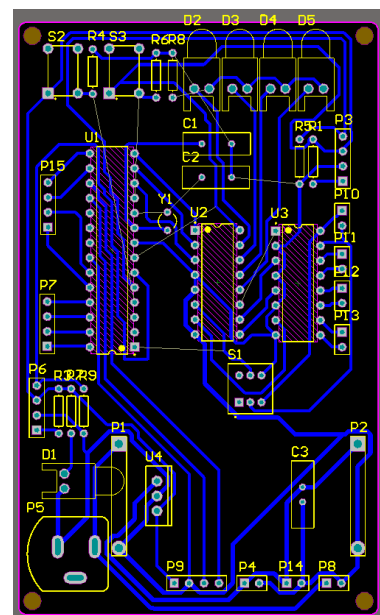
圖(19)結構改善後

2. 電路板製作

使用麵包線會有鬆脫的可能性，且體積也很佔空間，所以後來用了 Altium Designer 來設計電路圖如圖(20)、圖(21)，再由雕刻機來完成電路板的製作。



圖(20) AD 電路圖



圖(21) AD 線路圖

第一次刻電路板發現有些孔的位置不對，導致有些元件針腳無法插入，而且板子大小也超出我們預期的範圍，如圖(22)，所以後來再設計的時候就有將所有問題改善如圖(23)。因為我們的線很多，而且是單層版的製作，導致我們得跳線。



圖(22)電路改善前



圖(23)電路改善後

3. 3D 列印

製作 3D 列印是為了要和步進馬達能夠緊密連接，使運轉時能降低噪音，而 3D 列印精密度也非常高，可以輕易製作出想要的結構，也因為結構能自行設計，在兩個連接器底部多加了一個定位用棍子，使馬達能在每次啟動時歸位，不易造成誤差，結構改善前後如圖(24)、圖(25)。



圖(24) 使用夾子與馬達連結，但卻極為不穩固，且噪音大。



圖(25)使用 3D 列印改善，結構穩固，不易掉落，噪音也有明顯下降。

4. 攝影機改裝

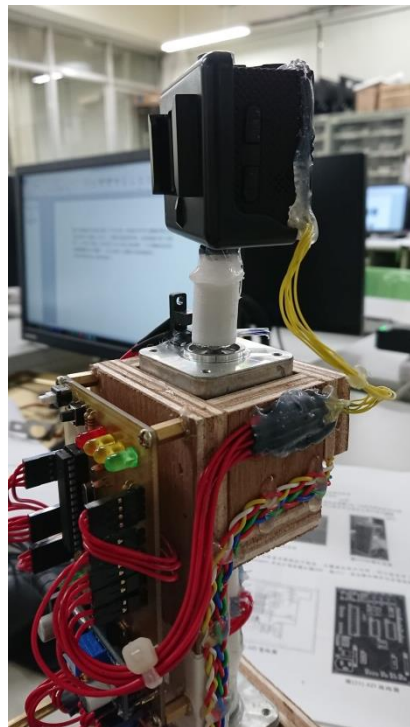
為了使單晶片能直接控制攝影機的開關機和縮放功能，我們把攝影機內部控制的電路用單芯線延長，透過光耦合器與控制面板連接如圖(27)、圖(28)，因此不須額外操控攝影機，攝影機會自動執行所有動作。



圖(26)攝影機改裝前



圖(27)攝影機改裝後



圖(28)攝影機改裝連接至單晶片

(二)軟體

程式分為以下四個架構，開機歸零、背景數值設定、自動追蹤、停止/開始，其詳述如下：

1. 開機歸零程式

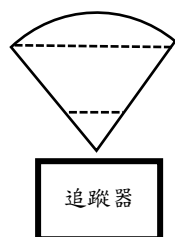
按下電源開關後，系統會先執行一連串的开機作業，首先，會先啟動攝影機，然後執行歸零程式，由於步進馬達為開迴路控制，無角度回授的裝置，而是藉由單晶片的運算來控制馬達轉動的角度，因此剛開機的時候無法得知馬達的角度，必須先歸零，才有基準點可以參考，歸零的方法是讓馬達不停反轉，直到裝在馬達軸承特殊機構阻擋了光遮斷的訊號，使馬達停止，即可完成歸零。

```
//=====歸零程式=====  
lightsw1 = digitalRead(light1); //超音波感測器角度歸零  
while(lightsw1==HIGH)  
{  
    lightsw1 = digitalRead(light1);  
    myStepper.step(-1);  
}  
lightsw2 = digitalRead(light2); //攝影機角度歸零  
while(lightsw2==HIGH)  
{  
    lightsw2 = digitalRead(light2);  
    myStepper2.step(-1);  
}
```

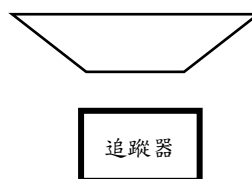
2. 背景數值設定程式

完成歸零後，指示燈會提醒使用者設定最遠極限距離，馬達會先旋轉到正中間的位置，然後使用者必須站在攝影機正前方，此時使用者按下設定按鈕，超音波感測器測定到的距離(使用者與機器間的距離)會自動設定成最遠極限距離，再來指示燈會提醒使用者設定最近極限距離，方法與前面相同，但使用者與機器間的距離會設定成最近極限距離，因此使用者可依照現場狀況自行調整上下限距離，以防止環境干擾所造成的誤差。

這個程式是非常重要的程式，自動追蹤攝影機工作的場所可能會有許多物體在超音波的掃描範圍內，也可能會有人在前後走動，造成不正確的動作，如果限定了追蹤的範圍，就能濾除不正確的干擾。其概念如圖(29)、(30)。



圖(29)超音波感測器
原本的掃描範圍



圖(30)濾除干擾區
域後的掃描範圍

```

//===== 最遠極限距離設定=====
    myStepper.step(52);          //超音波感測器角度控制馬達正轉至九十度
    myStepper2.step(49);         //攝影機角度控制馬達正轉至九十度
    sw1= digitalRead(pb1);       //按下設定按鈕，量測一次距離
    while(sw1==LOW)
    {sw1= digitalRead(pb1);}
do
{
    digitalWrite(fuction1, HIGH); //點亮[設定中]指示燈
    digitalWrite(fuction2, LOW);
    digitalWrite(fuction3, LOW);
    unsigned long d=ping()/58; //calculate distance //超音波感測器測量距離
    Serial.print(d);
    Serial.println("cm");
    delay(1);          //防彈跳
    cm_min=cm-5;        }      //最遠極限距離儲存
while(cm>200); //若超音波感測器性號失真，重複量測直到數值合理
    sw1= digitalRead(pb1);
while(sw1==HIGH)
    {sw1= digitalRead(pb1);}
delay(20);          //防彈跳
//===== 最近極限距離設定=====
sw1= digitalRead(pb1); //按下設定按鈕，量測一次距離
while(sw1==LOW)
{
    sw1= digitalRead(pb1);}
digitalWrite(fuction1, LOW); //點亮待機指示燈
digitalWrite(fuction2, HIGH);
digitalWrite(fuction3, LOW);
do
{
    unsigned long d=ping()/58; //calculate distance //超音波感測器測量距離
    Serial.print(d);
    Serial.println("cm");
    delay(1);          //防彈跳
    cm_min_short=cm; //最近極限距離儲存 }
while(cm>cm_min); //若超音波感測器性號失真，重複量測直到數值合理
myStepper.step(-30); //超音波感測器角度控制馬達反轉 36 度
myStepper2.step(-30); //攝影機角度控制馬達反轉 36 度

```

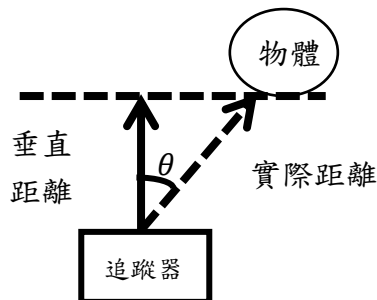
3. 自動追蹤程式

開機設定完成後，再按下設定按鈕，攝影機將開始錄影，超音波感測器開始掃描物體位置並進行追蹤。原理是藉由讀取超音波感測器在不同角度感測到的數值，判斷物體的位置，經過計算後使攝影機角度控制馬達轉動到面相被偵測物體的位置。

```
while(1)                                //無限迴圈
{
  for(i=1;i<7;i++)
  //超音波角度控制馬達正轉，每動 36 度偵測一次距離，共偵測六次
  {
    myStepper.step(10);                  //攝影機角度控制馬達轉 36 度
    delay(3);                           //等待機械延遲
    unsigned long d=ping()/58; //calculate distance //超音波感測器測量距離
    Serial.print(d);
    Serial.println("cm");
    delay(1);                           //防彈跳
```

超音波感測器量測的距離需要轉換成物體與追蹤器之間的垂直距離才能進行比較，如圖(31)。

$$\text{實際距離} \times \cos \theta = \text{垂直距離}$$



圖(31)距離轉換示意圖

```
rad= (3-i)*0.31413; //計算目前角度與中垂線之夾腳
cm_cosnot=abs(rad); //取絕對值
cm_cos=cm*cos(cm_cosnot); //計算物體取餘弦函數後的垂直距離
```

取得垂直距離後就可以進行距離比較，如果垂直距離小於最遠極限距離且大於最近極限距離，攝影機角度控制馬達就會進行追蹤。

```
if(cm_cos<cm_min) //若垂直距離小於最遠極限距離且大於最近極限距離
{
  if(cm_cos>cm_min_short)
  {
    x=i*10-a; //角度運算
    myStepper2.step(x); //攝影機角度控制馬達轉至超音波感測器當前角度
    a=a+x;
```

攝影機還會自動進行縮放控制，根據量測到的距離，畫面會縮放到合適的位置。

```
//=====攝影機縮放控制=====
if(cm<50) {aa=0;}
else if((cm<50)&&(cm>60)){aa=1;}
else if((cm<60)&&(cm>70)){aa=2;}
else if((cm<70)&&(cm>80)){aa=3;}
else if((cm<80)&&(cm>90)) {aa=4;}
else if((cm<90)&&(cm>100)) {aa=5;}
else {aa=6;}
cc=aa-bb;
if(cc>0)
{ for(bb=0;bb<cc;bb++)          //放大
{digitalWrite(fuction1, LOW);
digitalWrite(fuction2, HIGH);
digitalWrite(fuction3, HIGH);
delay(25);
digitalWrite(fuction1, LOW);
digitalWrite(fuction2, HIGH);
digitalWrite(fuction3, LOW);
delay(25);}
}
else if(cc<0)
{ ee=abs(cc);
for(bb=0;bb<ee;bb++)          //縮小
{ digitalWrite(fuction1, HIGH);
digitalWrite(fuction2, HIGH);
digitalWrite(fuction3, HIGH);
delay(25);
digitalWrite(fuction1, LOW);
digitalWrite(fuction2, HIGH);
digitalWrite(fuction3, LOW);
delay(25);}
}
digitalWrite(fuction1, HIGH);
digitalWrite(fuction2, HIGH);
digitalWrite(fuction3, LOW);
```

如果被偵測物體沒有移動，掃描裝置將被鎖定，此時如果有其他人進入掃描範圍，也不會被干擾，被偵測物體移動後，掃描裝置則會繼續掃描。超音波角度控制馬達反轉的程式與正轉時大致相同，因此不特別介紹。

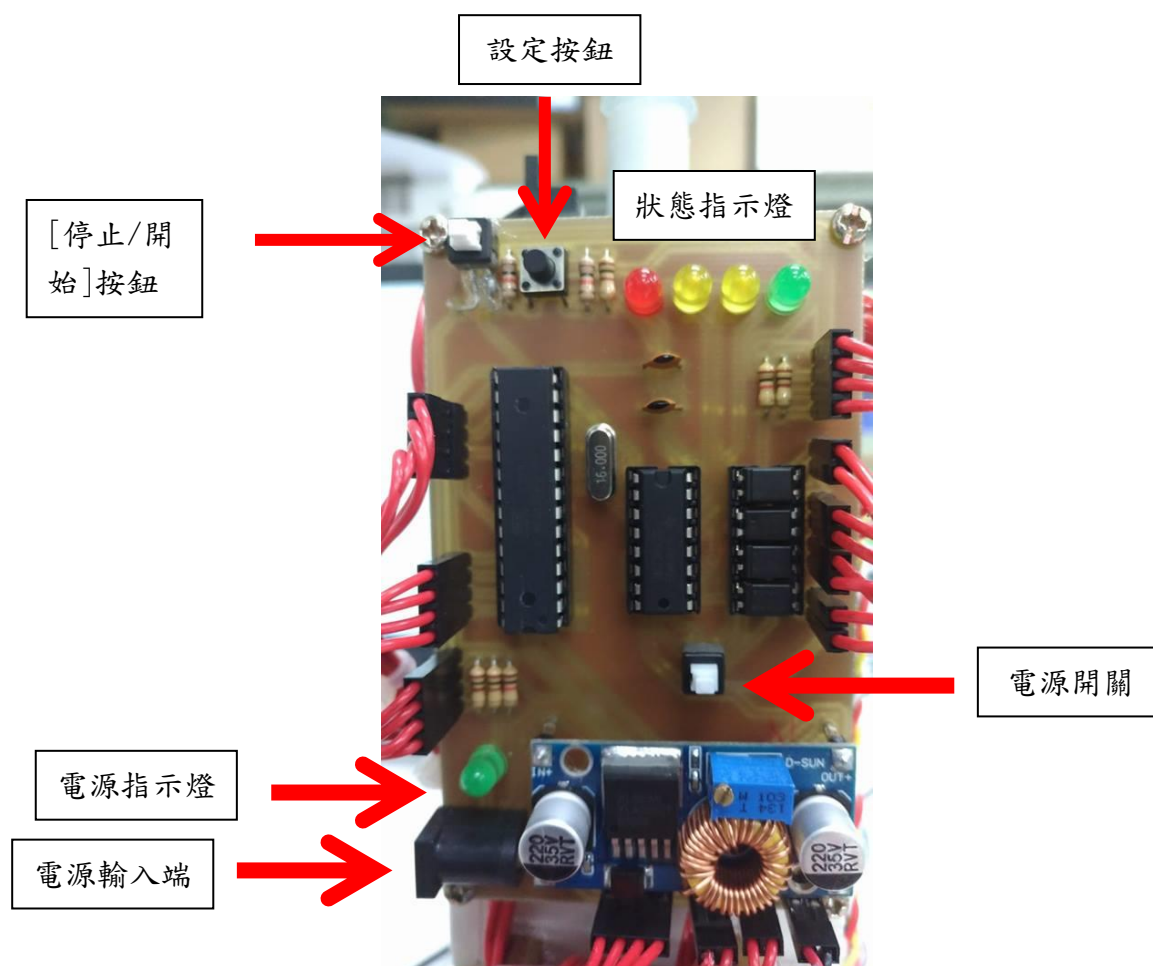
4. [停止/開始]中斷副程式

當系統在運作時，當按下[停止/開始]按鈕，可以任意的暫停，系統將自動暫停錄影，但仍保留停止運作前的參數，因此不需要重設，只要放開此按鈕，系統即可繼續運作，此功能讓使用者方便的控制錄製影片的時間。

```
//[停止/開始]中斷副程式
void swISR()
{
  for(int x=0;x<30000;x++)
  {
    digitalWrite(fuction1,HIGH); //暫停錄影
    digitalWrite(fuction2, LOW);
    digitalWrite(fuction3,HIGH); }
  digitalWrite(fuction1,LOW); //點亮待機指示燈
  digitalWrite(fuction2, HIGH);
  digitalWrite(fuction3,LOW);
  swStop = digitalRead(sss); // [停止/開始]按鈕未放開之前先待機
  while(swStop==LOW)
  {
    swStop = digitalRead(sss);}
  for(int x=0;x<30000;x++)
  {
    digitalWrite(fuction1,HIGH); //開始錄影
    digitalWrite(fuction2, LOW);
    digitalWrite(fuction3,HIGH);
  }
  digitalWrite(fuction1,HIGH); //點亮運轉指示燈
  digitalWrite(fuction2, HIGH);
  digitalWrite(fuction3,LOW);
}
```

(三)操作介面

此系統的操作介面如下圖(32)。

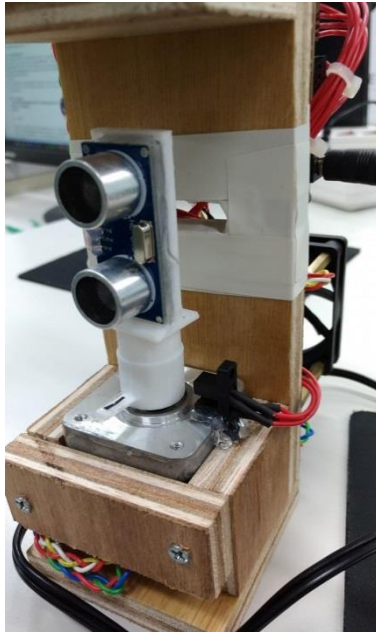


圖(32)操作介面電路板

肆、研究結果

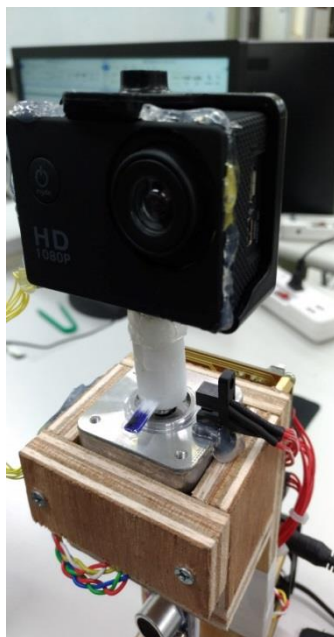
專題的成品分為兩個主要部分，為掃描裝置及攝影機角度控制裝置，分別介紹如下：

一、超音波掃描裝置



圖(33)超音波掃描裝置圖

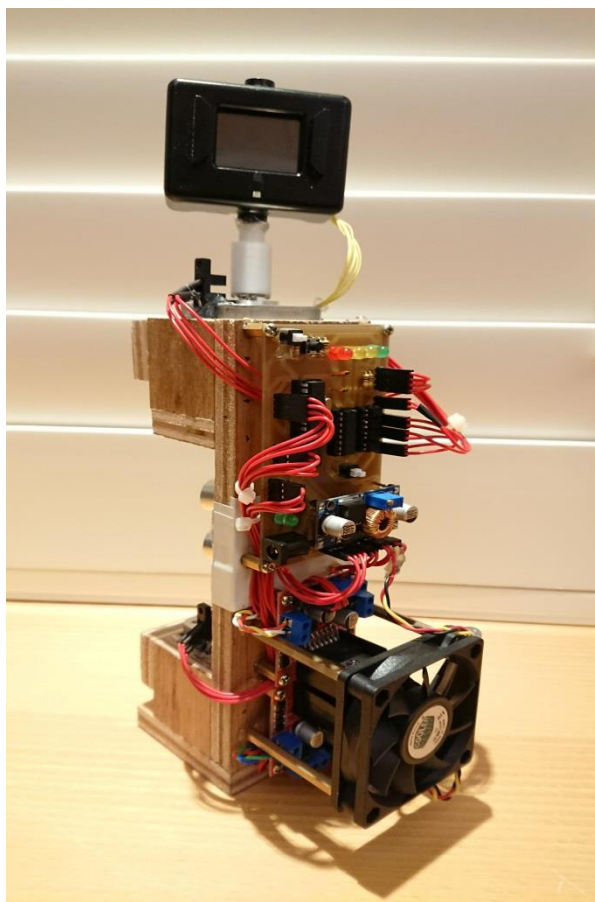
二、攝影機角度控制裝置



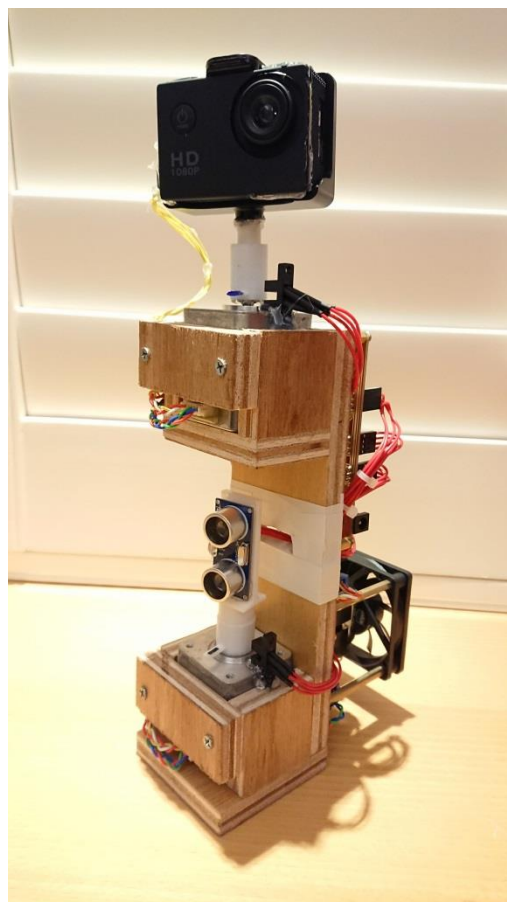
圖(34) 攝影機角度控制裝置

三、作品展示

完成品如下圖(35)、(36)



圖(35) 專題成品背面



圖(36) 專題成品正面

伍、討論

本專題特別著重在速度、精確度與穩定性上，所以致力於改善這三個方面的問題。

一、畫面延遲

掃描速度過慢導致畫面延遲，跟不上被追蹤的物體。

解決方法：

為了加快掃描速度，我們重新設計超音波感測器與馬達之間的連接部件，使用 3D 列印的輕量化材料，並刻意縮短迴轉半徑，使旋轉造成的慣性最小化，因此能提升掃描速度五倍以上。

二、超音波容易被干擾

由於超音波是透過空氣傳導，所以容易受到氣流的干擾，如果放置在風扇旁邊可能無法正確測量，此外，超音波也容易受到被偵測物體的材質、形狀的影響，尤其是在掃描動態物體，超音波不正常反射的情形也會加重，超音波感測器還會受到感應頭的構造與殘響震盪的影響，超音波感測器的指向特性也不夠理想，過寬的偵測範圍導致角度定位不精準。

解決方法：

為了不降低掃描速度，我們採用軟體的方式消除雜訊，我們設計了許多錯誤修正程式，程式中也設計了消除感測器震盪的防雜訊程式，還有一連串為了消除干擾而設計的開機程式。

三、機械延遲

馬達的反應速度受到機械慣性的影響，因此動作會逐漸產生誤差，最終導致動作錯誤，甚至會損壞機體。

解決方法：

在硬體的方面，使用更堅固的木製結構支撐所有機件，以減少振動與噪音進而增加穩定性，而轉部元件也盡量輕量化。

軟體方面，機器運轉時，控制器會刻意等待硬體的延遲，經過時測之後，幾乎不會發生誤動作的情況。

陸、 結論

雖然在專題製作的過程中歷經了多次的失敗，但是最終我們成功讓攝影機拍出快速且精準的完美畫面，甚至比人工調整的畫面還要準確，這個專題擁有極大的實用性，由於穩定性極高，因此面對大部分的場合都能應對，小巧的體積與堅固的結構讓機體能夠方便攜帶，人性化的操作介面讓使用者能輕鬆操作。

未來我們還希望加上高度的感測的功能，能夠執行全方位的感測讓拍攝，揚角也能自動調整，使自動追蹤攝影機能夠應用在更多方向。

柒、 參考資料及其他

1.超音波感測器程式參考(Ming's Blogger)。2018 年 2 月 6 日

取自：<http://ming-shian.blogspot.tw/2013/09/arduino-hc-sr04.html>

2.L298N 程式、接線參考(Cooper Maa)。2018 年 2 月 6 日

取自：<http://coopermaa2nd.blogspot.tw/2012/09/l298n.html>

3.趙英傑（2017 年 1 月）超圖解 Arduino 互動設計入門

臺北市：旗標科技股份有限公司。

4.張義和(2016)。新例說 ALTIUM DESIGNER。台灣：新文京

5.3D 繪圖使用程式(Tinkercad)。2018 年 2 月 6 日

取自：<https://www.tinkercad.com/>