

大安高工電子科
藍芽遙控自拍器
專題製作報告書

組長：郭致佑

組員：李祥宇

組員：李是頡

組員：張丞禮

指導老師：簡靖哲

中文摘要

旅行可以增廣見聞。透過相機紀錄下點點滴滴，但有時候，你的身邊只有你孤獨一人，但自拍技術又不太好或想要拍下背後的整片美景時，搖桿自拍神器是你最好的選擇。

所謂：「科技的進步是為了使人類的生活更加便利。」為了拍下一張美景，你必須反反覆覆的調整你的相機。是否太過麻煩呢？如果這時候有一個道具可以让你預覽你的照片並可以在遠處無線調整的話就方便許多了。

英文摘要

Travel can be augmented. Record the bits and pieces through the camera, but sometimes you are alone, but the self-timer technology is not very good or you want to take the whole scene behind you. The joystick self-timer is your best choice.

As the saying goes "The advancement of technology is to make human life more convenient." Isn't it too trouble to adjust your camera repeatedly in order to take a good scene? It can be a lot easier if there is a prop that allows you to preview your photo and you can adjust your camera wirelessly in the distance.

目錄

中文摘要	I
英文摘要	II
目錄	III
表目錄	IV
圖目錄	V
第一章 前言	1
1-1 製作背景及目的	1
1-2 專題製作方法、步驟與進度	1
1-3 預期成果	1
第二章 理論探討	3
2-1 結構與外觀	3
2-2 軟體與硬體	4
第三章 專題設計	7
3-1 架構圖	7
3-2 流程圖	8
3-3 甘特圖	9
3-4 電路設計	9
3-5 軟體設計	11
第四章 專題成果	15
4-1	15
4-2	16
第五章 結論與建議	17
5-1 建議	17
5-2 結論	17
參考文獻	1
附錄	1

表目錄

表 1	主端腳位及變數宣告.....	11
表 2	主端主要程式區段.....	13
表 3	從端腳位及變數宣告.....	13
表 4	從端主要程式區段.....	14

圖目錄

圖 1	預想圖.....	2
圖 2	電路板位置.....	3
圖 3	Keyes_SJoys.....	4
圖 4	伺服馬達.....	5
圖 5	藍芽元件.....	5
圖 6	Arduino Pro Mini.....	6
圖 7	架構圖.....	7
圖 8	流程圖.....	8
圖 9	甘特圖.....	9
圖 10	主端電路圖.....	10
圖 11	從端電路圖.....	10
圖 12	搖桿讀取方位示意圖.....	12
圖 13	專題成果圖.....	15

第一章前言

1-1 製作背景及目的

(一) 背景：現代人時常用照相機來記錄、分享自己的生活，從黑白相機到現在大部份的人都用手機拍攝，為了在拍照時有更好的畫面而有了周邊的配備自拍棒、腳架等等。

(二) 目的：因為平常拍照時使用的自拍器具要用手調整很麻煩，在拍全家照時還要麻煩路人幫忙容易造成別人的困擾，遠距拍攝如果角度不對還要走回去調整，所以我們設計出能利用手上的搖桿遙控遠處的架子，按下手上的搖桿來使遠端的手機按下快門，就能順利完成拍照。

1-2 專題製作方法、步驟與進度

1. 製作計劃書
2. 蒐集資料
3. 電路設計
4. 機構規劃、軟體設計、硬體設計
5. 整合測試
6. 作品完成
7. 撰寫報告

1-3 預期成果

我們預期可以將手機架設在一塊平台上，平台上有兩顆馬達分別可以控制水平、垂直轉動，透過手上的遙控器讓我們平台上的手機可以調整成適當的角度。最後，完成調整之後，就可以藉由遙控器上的快門鍵進行拍照。

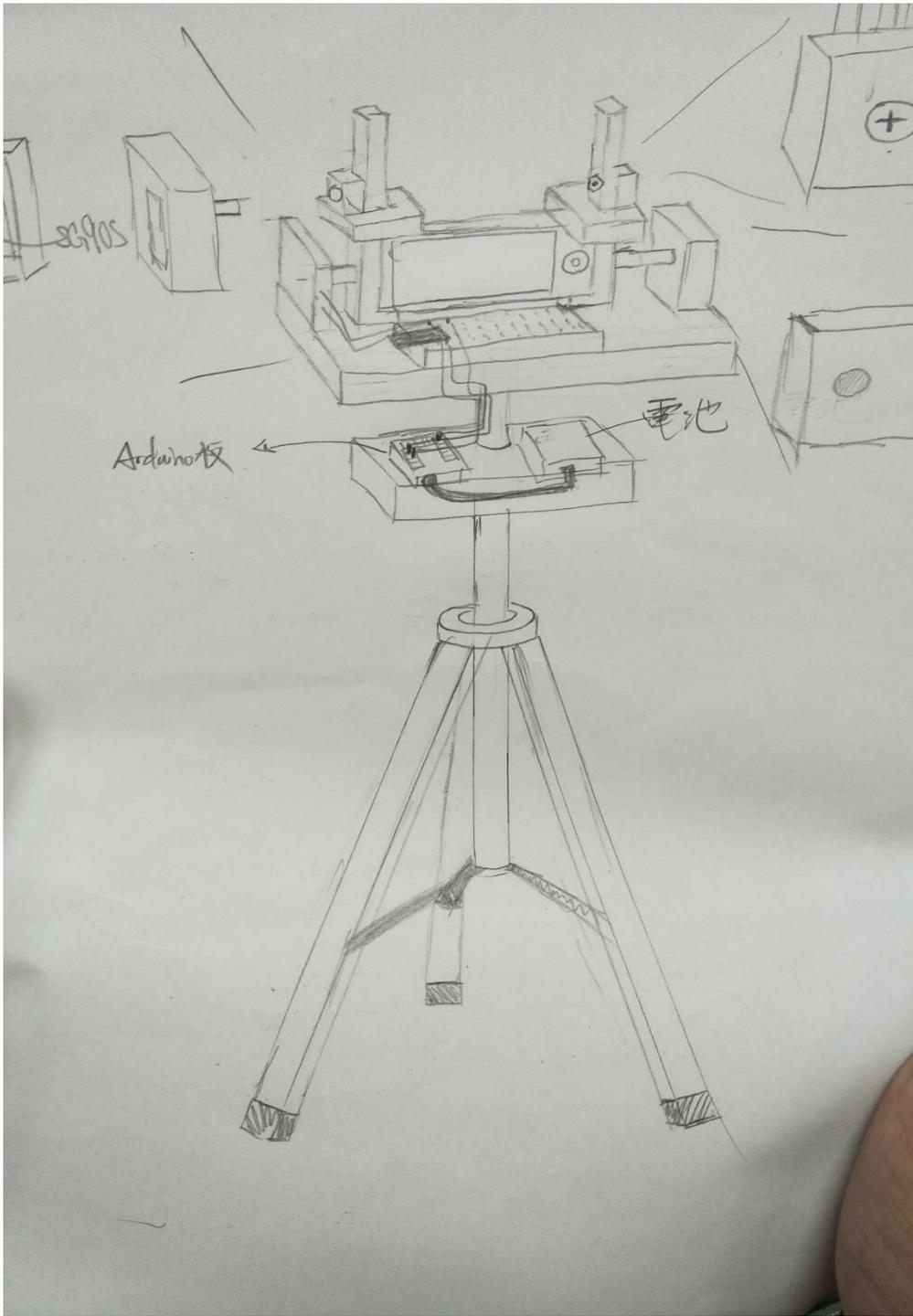


圖 1 預想圖

第二章 理論探討

2-1 結構與外觀

我們將整體系統架構分成主端跟從端。所謂主端就是操作者透過遠端控制相機水平垂直的電路。從端就是控制相機的伺服馬達的電路。

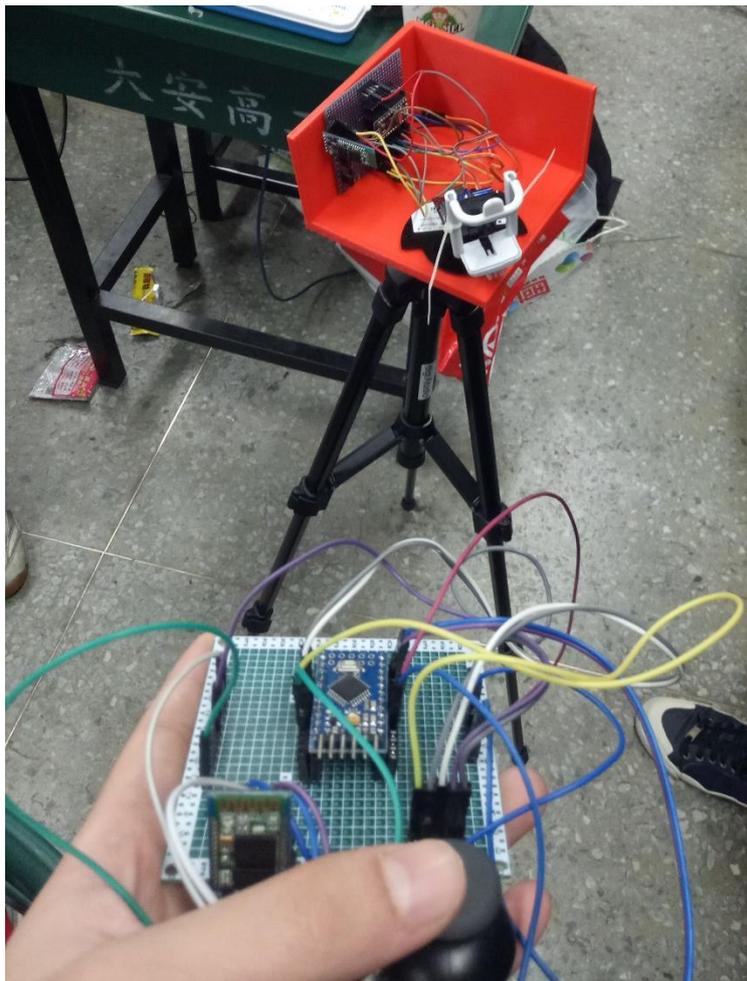


圖 2 電路板位置

2-2 硬體

2-2-1 搖桿

Keyes_SJoys 這類搖桿有五隻接腳，分別是 VCC、GND、VRX、VRY 和 SW，VCC 與 GND 輸入電源，VRX 與 VRY 控制上下左右，剩下的 SW 為按鈕。

其中搖桿的上下左右是透過類比電壓的輸出，來表示目前搖桿的方向狀態，完全沒動在正中央的時候，輸出 2.5 伏特，X 軸的部分為往左下降，一直到 0 伏特，往右為上升，一直到 5 伏特。Y 軸的部分為往上下下降，一直到 0 伏特，往上為上升，一直到 5 伏特。而它在 Arduino 的類比輸出值，X 軸的部分在不動的時候為 508，往左為減，一直到 0，往右為加，一直到 1023，Y 軸的部分不動的時候為 488，往下為減，一直到 0，往上為加，一直到 1023。按鈕的部分是按下輸出為 5V，不按的情況下為 0V。



圖 3 Keyes_SJoys

2-2-2 伺服馬達

伺服馬達(servo motor)分直流、交流兩種，一是用電壓、另一個是輸送特定工作週期的交流訊號；馬達外部來看有三條線接至馬達內部。分別是 Vcc、Gnd 和控制線；內部構造又分為四個部分：減速齒輪、直流馬達、電位器及控制電路，減速齒輪：調整、定位、轉動馬達的主軸；當我們送直流或交流訊號時，控制電路會判斷目標角度與電位器之間有差異而轉動馬達，此時馬達轉動會同時帶動電位器變動，當控制電路判斷電位器回饋的電壓與目標角度位置相同時，即停止轉動，如此即能達到角度控制。



圖 4 伺服馬達

2-2-3 藍芽(HC-05)

這次使用的元件有五隻接腳，分別是 VCC、GND、RXD、TXD 和 KEY，VCC 與 GND 輸入電源，RXD 是接收端，TXD 是傳送端，而 KEY 腳是用來開啟 AT 模式時使用的。AT 模式也就是用來對藍芽設定一些參數或是尋找參數，例如連線速率、回復原廠模式、定為主端或從端、尋找每顆 HC-05 的位置等。

而進入 AT 模式的方法就是將 KEY 腳輸入高電位，再配合 Arduino 的 Serial Monitor 輸入指令。欲使藍芽連線需要兩個藍芽元件分別設定為主端和從端，然後從主端透過 AT 模式找尋 Address，像是尋找 IP 的感覺，接著再從從端的 AT 模式輸入指令，其中最重要的是必須把兩個藍芽的連線速率設定為一致，否則怎麼輸入指令都只是徒勞無功。若成功連線，將來只要一通電兩藍芽就會自動連上了。



圖 5 藍芽元件

2-2-4 Arduino Pro Mini

Arduino 是一個開放原始碼的單晶片微控制器，使用了 Atmel AVR 單晶片，採用了開放原始碼的軟硬體平台，建構於簡易輸出/輸入介面，並且具有使用類似 Java、Processing 的開發環境，也因為其簡單操控以及體積較小的特性，我們選擇使用 Pro Mini 這顆晶片來做為主要的控制器。

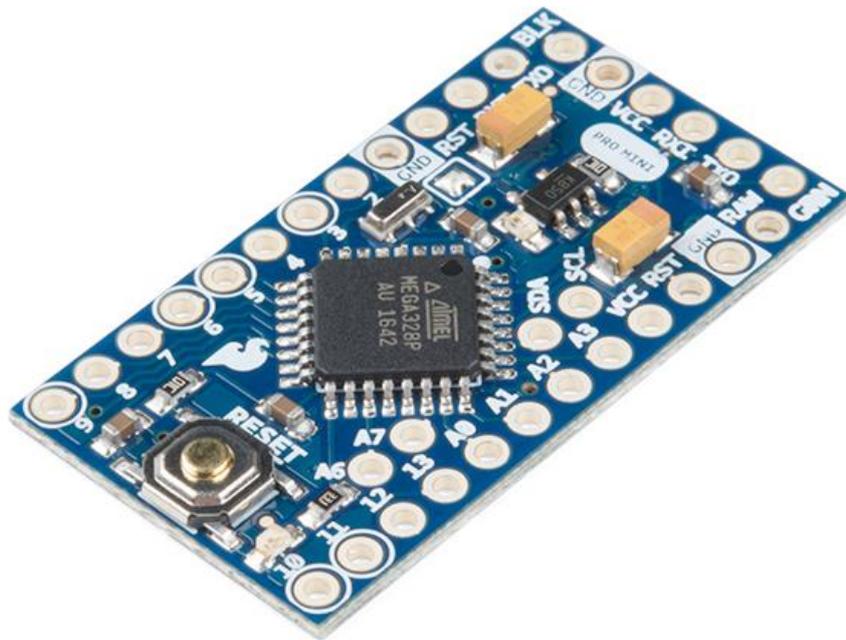


圖 6 Arduino Pro Mini

第三章 專題設計

3-1 架構圖

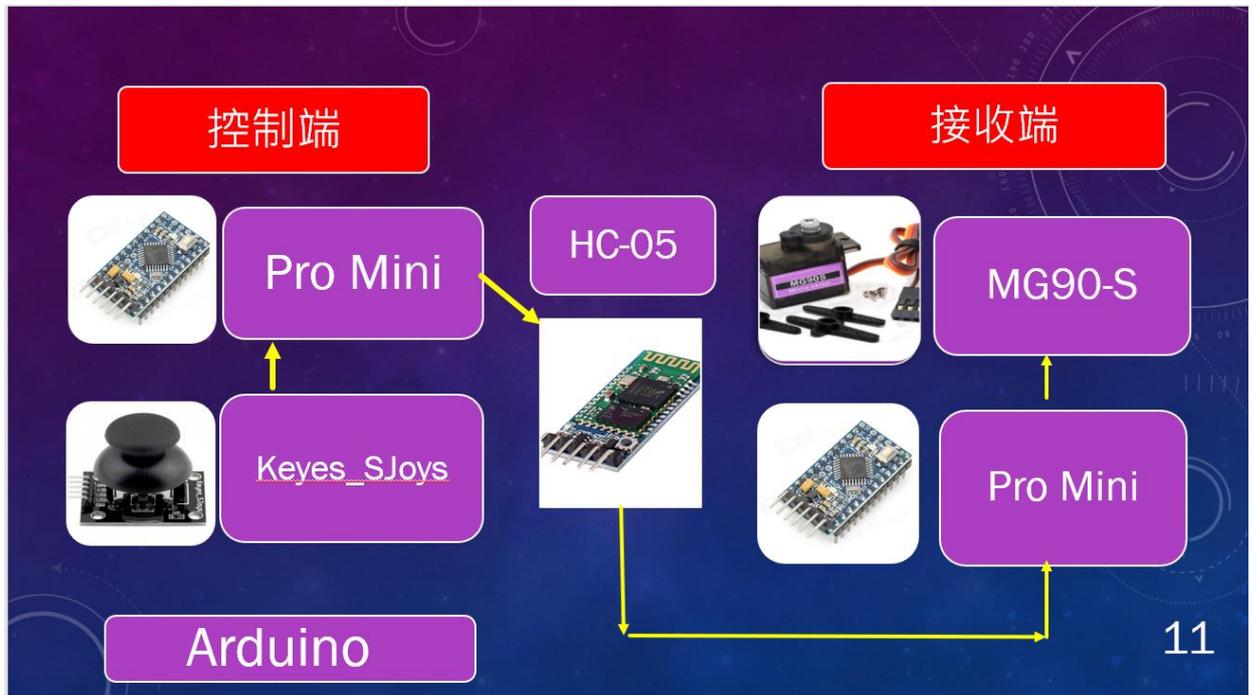


圖 7 架構圖

3-2 流程圖

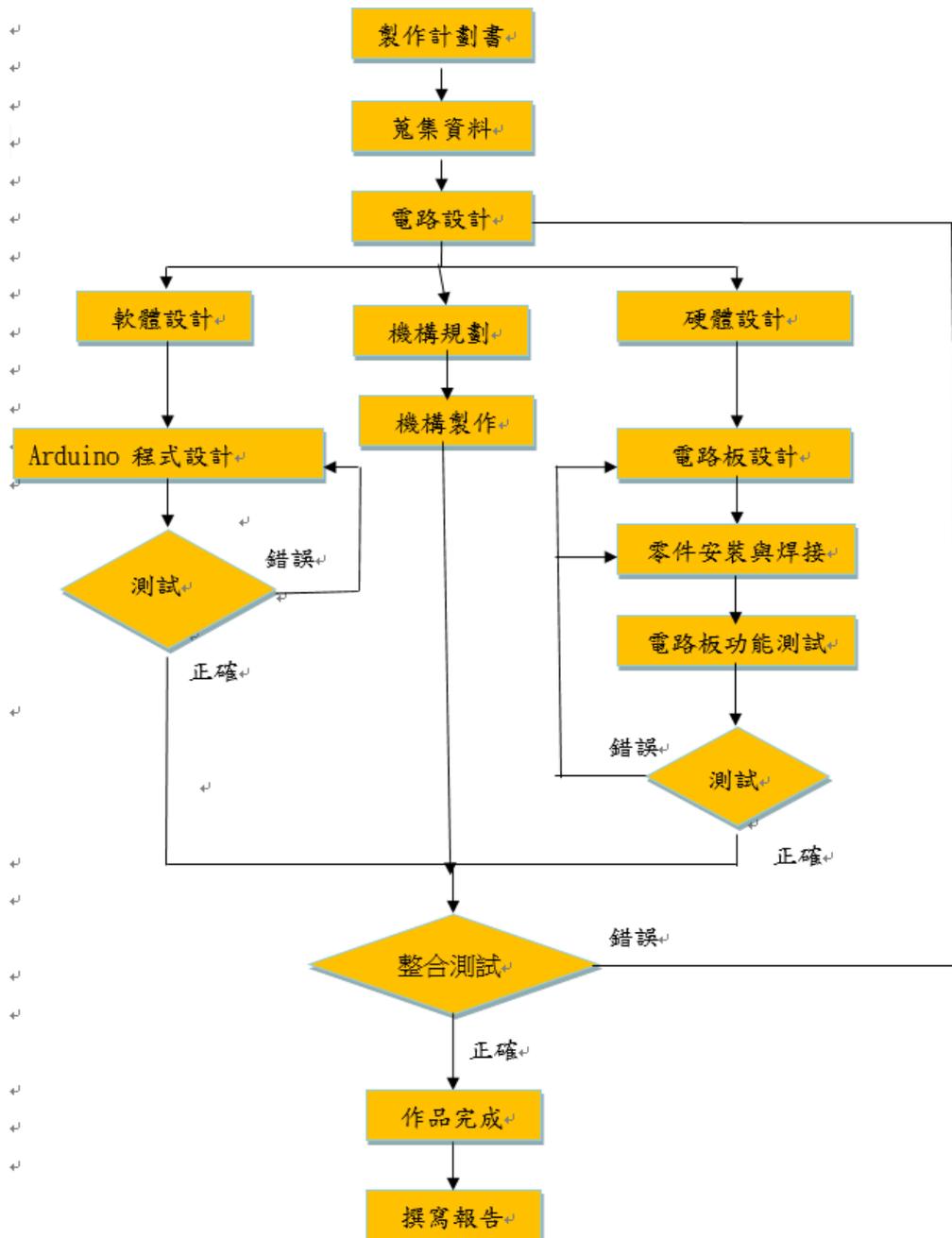


圖 8 流程圖

3-3 甘特圖

工作項目	週次																	負責成員	
	9 /3 ~9	9 /10 ~9	9 /17 ~9	9 /24 ~9	10 /1 ~10	10 /15 ~10	10 /29 ~10	11 /5 ~11	11 /12 ~11	11 /19 ~11	11 /26 ~11	11 /18 ~11	12 /3 ~12	12 /10 ~12	12 /17 ~12	12 /24 ~12	12 /31 ~12		1 /7 ~1
寫計劃書					●	●													全體成員
資料蒐集	●	●	●	●	●														全體成員
資料整理	●	●	●	●	●														全體成員
程式碼設計、測試	●	●	●	●	●	●	●	●	●	●	●								郭、顏、宇
電路板設計、測試									●	●	●	●	●	●	●				郭
軟體、硬體整合														●	●				全體成員
撰寫報告													●	●	●	●	●		張、宇
簡報製作													●	●	●	●	●	●	張、郭
焊接								●	●	●	●								郭
口頭報告					●		●		●			●		●	●	●	●	●	全組成員
預定進度	5	10	15	20	25	30	35	40	45	55	65	70	75	80	85	90	95	100	累積百分比%

圖 9 甘特圖

3-4 電路設計

我們專題系統主要分為主端與從端，主端接有 Pro mini、HC-05 以及 Keyes-Sjoys 三大元件。而從端主要有 Pro mini、HC-05 以及 SG-90s 三大元件。由於元件數相對較少，因此我們直接用焊接的方式解決接線的問題。

電源供應部分，由於我們是採用 Pro mini 為我們整個系統的核心，所以勢必只能藉由電池，或者透過燒錄器進行供電，但是因為本專題旨在能夠隨身攜帶，因此我們選擇將燒錄器接上 Pro mini，將燒錄器上的 Micro USB 插孔，透過隨身的行動電源進行供電，不必額外需要電池，或者需要擔心電池是否有電的問題。

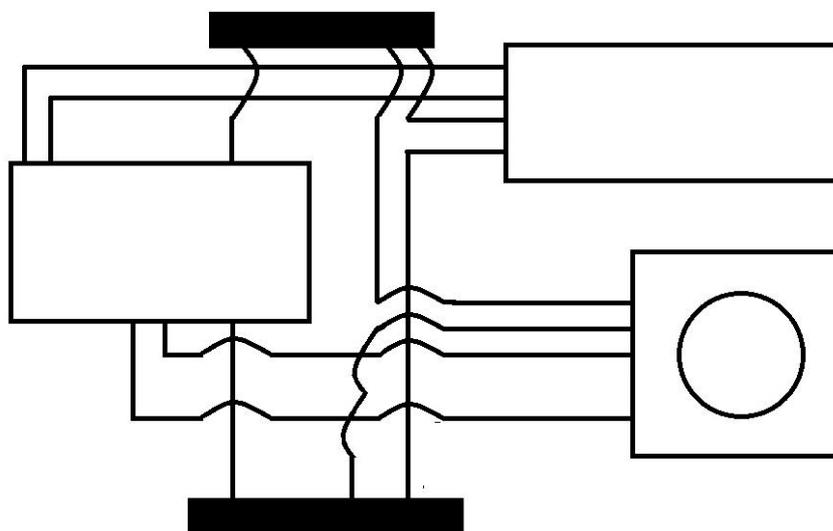


圖 10 主端電路圖

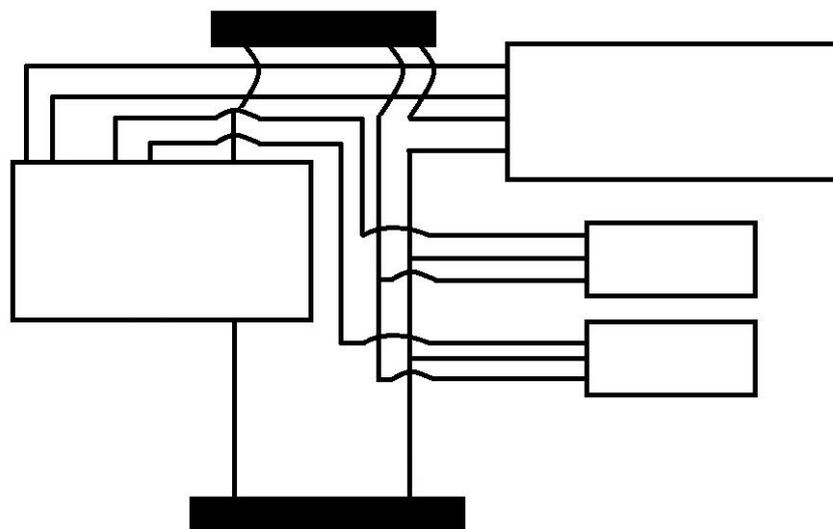


圖 11 從端電路圖

3-5 軟體設計

軟體設計的部分，我們採用 Arduino IDE 進行撰寫，主要分成主端以及從端進行轉寫。將主端讀取到的數值，透過藍芽傳送到從端，進行相應的動作。

3-5-1 主端

簡單的說，主端要做的事情就是：傳送動作訊號。我們外部透過轉動搖桿讓 Pro mini 讀到我們希望的動作後，藉由內部的程式轉換成相應的動作信號後，傳送給從端。

<pre>#include "Wire.h" #include <SoftwareSerial.h> SoftwareSerial BT(8,9); int val; int odd=0; int GG; int SW = 3; int pit; const int pinX = A0; const int pinY = A1; int valX, valY ; void setup() { Serial.begin(57600); pinMode(SW, INPUT); BT.begin(57600); }</pre>	<p>引用匯流排函式庫。</p> <p>引入可模擬 Txd、Rxd 腳位的函式庫。</p> <p>宣告 Txd、Rxd 的腳位(前為 Txd，後為 Rxd)。</p> <p>儲存接收資料的變數。</p> <p>傳送動作訊號的變數。</p> <p>傳送快門訊號的變數。</p> <p>宣告按鈕的腳位。</p> <p>將讀入的腳位信號指定給此變數。</p> <p>宣告搖桿輸入的腳位。</p> <p>將讀入的搖桿類比訊號指定給這兩個變數。</p> <p>設定序列埠預設連線速率。</p> <p>將按鈕的腳位預設為輸入。</p> <p>設定藍芽預設的連線速率。</p>
---	---

表 1 主端腳位及變數宣告

完成主端的腳位宣告後，接著就是主要動作判斷的部分了。在此我們先插入一張示意圖，由下圖可見，我們將搖桿的信號分為八個區塊，由正上方順時針依序編號1至8，正中央不動為0。在 Pro mini 讀取到搖桿的信號後，要將它所讀取到的信號轉變成指定區域的動作值，透過藍芽傳送給從端。

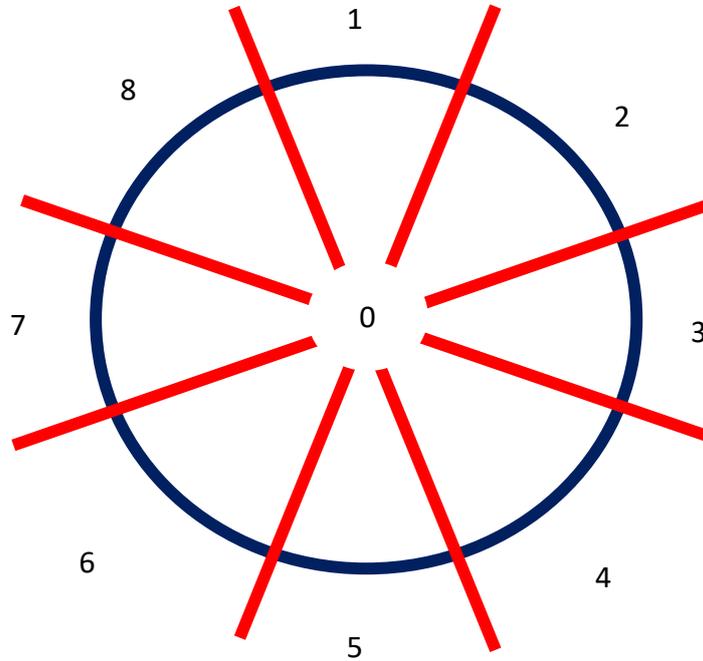


圖 12 搖桿讀取方位示意圖

```
void loop()
{
  pit=digitalRead(SW);
  if(pit==HIGH)GG = 1;
  else GG = 0;
  valX = analogRead(pinX);
  valY = analogRead(pinY);
  if(valX >= 510 & valY < 510 & valY > 490)
  odd = 1;
  else if (valX >= 510 & valY >= 510)
  odd = 2;
  else if(valX < 510 & valX > 490 & valY >= 510)
  odd = 3;
  else if(valX <= 490 & valY >= 510)
  odd = 4;
  else if(valX < 490 & valY >= 490 & valY <= 510)
  odd = 5;
}
```

讀取快門的讀值。
 如果按下按鈕，快門鍵會輸出高準位。
 讀取搖桿 X 軸讀值。
 讀取搖桿 Y 軸讀值。
 上推。
 右上推。
 右推。
 右下推。
 下推。

<pre> else if(valX <= 490 & valY <= 490) odd = 6; else if(valY < 490 & valX >= 490 & valX <= 510) odd = 7; else if(valY <= 490 & valX >= 510) odd = 8; else odd = 0; delay(100); Serial.println(odd); BT.print(odd); BT.print(GG); } </pre>	<p>左下推。</p> <p>左推。</p> <p>左上推。</p> <p>靜止。</p> <p>在序列阜上顯示動作值。</p> <p>將動作值及快門讀值傳給藍芽。</p>
---	--

表 2 主端主要程式區段

3-5-2 從端

在完成主端的設定之後，接下來看到從端。從端需要做的事情就是：將透過藍芽接收到的數值轉換成相應的方向(如：上轉、下轉等動作)。藉由 Pro mini 送給馬達訊號讓馬達偏轉。

<pre> #include <SoftwareSerial.h> #include <Servo.h> #include "Wire.h" int cam = 5; int GG; int odd; SoftwareSerial BT(8, 9); void setup() { BT.begin(57600); servo1.attach(2); servo2.attach(3); Serial.begin(57600); pinMode(cam, OUTPUT); } </pre>	<p>引入藍芽連接函式庫。</p> <p>引入馬達函式庫。</p> <p>引入匯流排函式庫。</p> <p>快門用腳位。</p> <p>傳輸快門訊號的變數。</p> <p>傳輸動作訊號的變數。</p> <p>宣告 Txd、Rxd 的腳位(前為 Txd，後為 Rxd)。</p> <p>設定藍芽預設的連線速率。</p> <p>設定馬達輸出腳位。</p> <p>設定序列阜預設連線速率。</p> <p>將快門的腳位預設為輸出。</p>
---	---

表 3 從端腳位及變數宣告

完成從端的設定後，剩下主要動作判斷的部分了。利用” parseInt” 語法，持續將讀取到的數值指定給” odd” 變數，直到接收到第一個非整數值為止。再利用相同手法擷取第二段數值指定給” GG” 變數。即完成同時傳送兩段數值的程式區段。

將數值讀取進來後，接下來就是要判讀我們所讀取到的數值進行相應的動作。

<pre> void loop() { if (BT.available()) { odd = BT.parseInt(); GG = BT.parseInt(); } if(odd==1)posX+=5; else if(odd==2) { posX+=5; posY+=5;} else if(odd==3) { posY += 5;} else if(odd==4) { posX-=5; posY+=5;} else if(odd==5) { posX-=5;} else if(odd==6) { posX-=5; posY-=5;} else if(odd==7) { posY-=5;} else if(odd==8) { posX+=5; posY-=5;} if(GG == 1) { digitalWrite(cam, HIGH);} else { digitalWrite(cam, LOW);} if(posX >= 175)posX = 170; if(posY >= 175)posY = 170; if(posX <= 15)posX = 15; if(posY <= 15)posY = 15; </pre>	<p>讀取主端藍芽傳送過來的資料。 “odd” 為方向的判斷變數。 “GG” 為快門是否按下的判斷變數。</p> <p>「上轉」運算。 「右上轉」運算。</p> <p>「右轉」運算。</p> <p>「右下轉」運算。</p> <p>「下轉」運算。</p> <p>「左下轉」運算。</p> <p>「左轉」運算。</p> <p>「左上轉」運算。</p> <p>如果接收到按下快門，則「快門」輸出高準位。</p> <p>反之則「快門」輸出低準位。 限制馬達最大/最小值</p>
---	--

表 4 從端主要程式區段

第四章 專題成果

4-1 成果展示

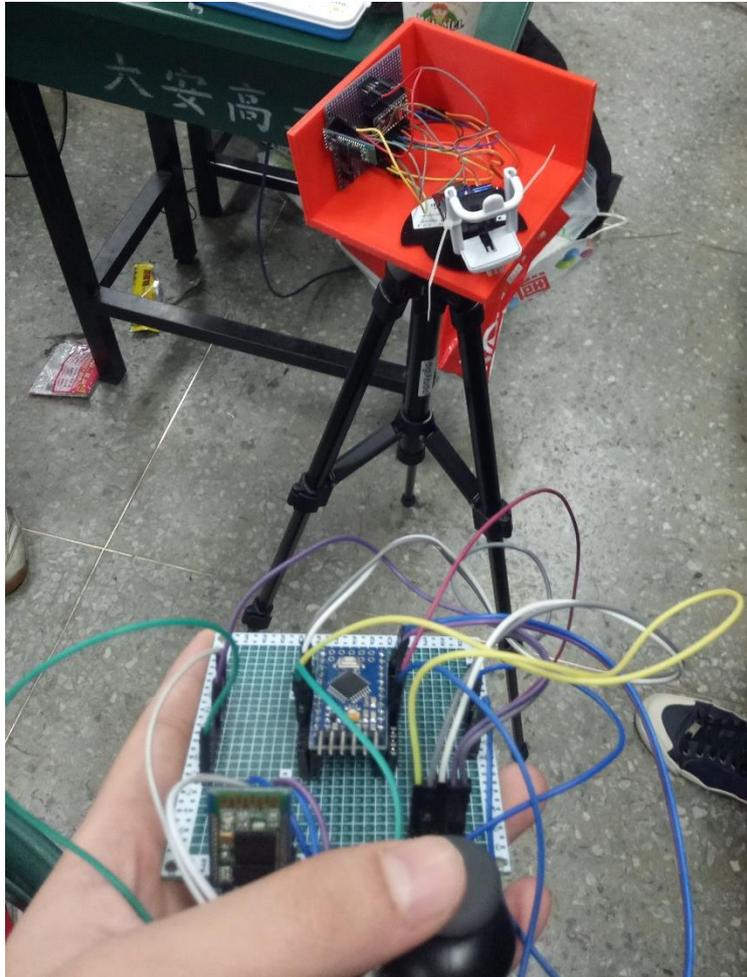


圖 13 專題成果圖

基本上我們的專題本來就是屬於未完成的作品，所以專題成果上我們也只能端出最終的模樣。也就是上圖這樣兩塊電路板分為遙控端和相機端還有一個平台放置在腳架上的模樣。

4-2 問題與解決

我們一開始遭遇的問題就是藍芽的連線不穩定，一開始的時候有可能成功但下次卻也有可能失敗，其中的原因之後發現是鮑率沒設定好，還有 TX 與 RX 接反的問題，解決之後又發現還有設定 AT 模式時 KEY 腳的程式寫法不能一直給電，不然會出現問題。之後還有用快門時程式撰寫有問題和硬體線路有斷路的問題。

第五章 結論與建議

5-1 建議

因為本專題的主要部件比較簡單，所以可以再額外擴充其他功能(例如:預覽畫面等等)，另外要注意在設定藍芽 AT 模式的時候，主從端的連線報率是否有相同

5-2 結論

這個專題是為了讓生活更方便的其中一個想法而已，如果再做得更好是一定可以行的通並且有實用性質的，這次專題雖然沒有完成但過程中還是有學到許多東西。

參考文獻

一. 楊仁元、張顯盛、林家德 (2014)。專題製作理論與呈現技巧。台北市：台科大圖書股份有限公司。

二. HC-05 藍芽設定 AT 方法。2018 年 10 月 20 日取自：

<https://swf.com.tw/?p=712>

三. 搖桿資訊。2018 年 11 月 25 日取自：

<http://www.codedata.com.tw/social-coding/mblock-arduino-13-joystick/>

四. 伺服馬達相關資訊。2018 年 11 月 15 日取自：

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

附錄

附錄一 材料清單

類別	設備、軟體名稱	應用說明
伺服馬達	SG90S	控制相機架
電池	鋰電池	電力來源。
藍芽	HC-05	無線傳輸
搖桿	Keyes_SJoys	遙控
控制元件	Arduino Pro Mini	寫程式的原件
開關	按鈕開關	相機快門。
耗材	電路板	製作電路用。
耗材	杜邦線	偵測器接至電路板。
耗材	ABS 線料	3D 列印材料。
耗材	焊錫	製作電路用。

附錄二 成員簡歷

姓名	李是頡	班級	電子108甲
曾修習 專業科 目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD邏輯實習 7. Visual Basic程式設計實習 8. Altium Designer電路圖與印刷電 路板設計 		
參與專 題工作 項目	<ol style="list-style-type: none"> 1. 整體計畫擬定 2. 專題旁觀與建議 3. 電路發想 4. 程式設計 5. 研究與討論 6. 藍芽連線 		
經歷 簡介	<ol style="list-style-type: none"> 1. 擔任105學年度上學期班級副班長 2. 擔任105學年度下學期班級風紀股長 3. 工業電子丙級技術士合格 		



姓名	郭志佑	班級	電子108甲	
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD邏輯實習 7. Visual Basic程式設計實習 8. Altium Designer電路圖與印刷電路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 整體計畫擬定 2. 計畫書撰寫 3. 電路設計與配置 4. 3D列印外殼與設計 5. 研究與討論 6. 資料彙整 			
經歷簡介	<ol style="list-style-type: none"> 1. 擔任105學藝股長 2. 擔任106學藝股長 3. 擔任大安高工連三年度數學小老師 4. 擔任107學年度衛生股長 5. 擔任107學年度廠長 6. 擔任106學年度安全與衛生 7. 工業電子丙級技術士合格 			

姓名	李祥宇	班級	電子108甲	
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD邏輯實習 7. Visual Basic程式設計實習 8. Altium Designer電路圖與印刷電路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 整體計畫擬定 2. 整體程式設計 3. 電路設計與配置 4. 研究與討論 5. 鑽研伺服馬達原理 			
經歷簡介	<ol style="list-style-type: none"> 1. 擔任105級任班長 2. 擔任107學年度風紀股長 3. 擔任106學年度數位邏輯小老師 4. 擔任107學年度衛生股長 5. 擔任106學年度廠長 6. 工業電子丙級技術士合格 			

姓名	張丞禮	班級	電子108甲	
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD邏輯實習 7. Visual Basic程式設計實習 8. Altium Designer電路圖與印刷電 路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 整體計畫擬定 2. 計畫書撰寫 3. 電路設計與配置 4. 電路焊接 5. 研究與討論 6. 3D列印外殼與設計 			
經歷簡介	<ol style="list-style-type: none"> 1. 連續三年度擔任班級餐膳 2. 工業電子丙級技術士合格 			