

臺北市立大安高級工業職業學校

電子科

專題報告

自動書報翻拍機

Automatic book retrace machine

學生 組長：羅昱婷
組員：林妤婕
組員：王子熙
組員：蘇少睿

指導老師：張顯盛老師

中華民國 107 年 5 月

摘要

有感於充分利用時間是越來越重要的事，越來越繁忙的都市；越來越緊促的生活步調；越來越令人窒息的環境，每一刻都像是在戰場上作戰一樣，俗話說：Time is money，把握住時間就等於掌握了金錢，一天只有 24 小時，卻要像有著 37 小時般的利用。

正所謂：腹有詩書氣自華、三日不讀書，便面目可憎。閱讀，是我們生活中很重要的一環，不管生活的節奏是多麼的緊促，總是要抽空來感受一下書的魅力，不過在通勤時間，人擠人的環境下，拿著一本書，倘若上下車產生推擠而導致書受損的話那有多傷腦筋，所以我們專題研究出可以使用手機搭配藍牙（Bluetooth）來控制我們的翻拍機，因為我們現在這時代人手一機，通勤時間手機絕對是我們的好夥伴，不妨利用這瑣碎時間藉由書的幫助昇華一下自己。透過自動翻拍書報機各部分的完成，學習到許多課本所沒有的知識，獲益良多。

Abstract

Feeling that it is more and more important to make full use of time, an increasingly busy city; an increasingly tight pace of life; a more and more suffocating environment, every moment is like fighting on the battlefield.

As the saying goes, “Time is money”, to grasp the time equivalent to master the money, only 24 hours a day, but like to have 37 hours of use.

A proverb says, “Wisdom in hold, elegance in mold”, “After three days without reading, talk becomes flavorless.” Reading is a very important part of our lives, no matter how tight the pace of life, we still should take the time to feel the charm of books.

However, when holding a book in crowded condition at commuting time, may cause the book damaged, and it will be annoying. Therefore, we specially researched out a machine that can use cell phone with Bluetooth to control it. Cell phones are definitely our good partner at commuting time because everyone has it, may wish to use this trivial time to improve yourself by the help of books.

After completing the completion of each part of the automatic book retrace machine, we have learned a great deal of the knowledge that many textbooks do not have.

目錄

目錄.....	III
表目錄.....	V
圖目錄.....	VI
第 1 章 前言.....	1
1-1 專題製作背景及目的.....	1
1-2 預期成果.....	1
第 2 章 理論探討.....	2
2-1 機構部分.....	2
2-1-1 機構設計.....	2
2-1-2 馬達位置.....	3
2-2 元件及硬體.....	3
2-2-1 伺服馬達.....	3
2-2-2 風扇.....	5
2-2-3 DC12V 電路.....	6
2-2-4 Leap Motion.....	7
2-2-5 Arduino UNO.....	8
2-2-6 HC-05 藍牙模組.....	9
2-3 軟體部分.....	10
2-3-1 Arduino IDE.....	10
2-3-2 Processing.....	10
2-3-3 Android Studio.....	11
第 3 章 專題準備.....	13

3-1	架構圖.....	13
3-2	流程圖.....	14
3-3	甘特圖.....	15
第 4 章	專題成果.....	16
4-1	專題過程與問題.....	16
4-2	專題成果.....	17
第 5 章	結論與建議.....	18
5-1	結論.....	18
5-2	建議.....	18
	參考文獻.....	19
	附錄.....	20
附-1	設備清單.....	20
附-2	材料清單.....	21
附-3	程式碼.....	22
附-3-1	Processing 程式碼.....	22
附-3-2	Arduino 程式碼.....	24
附-3-3	Android Studio 設計介面—藍牙.....	29
附-3-4	Android Studio 設計介面—主介面.....	32
附-3-5	Android Studio 藍牙連接.....	36
附-3-6	Android Studio 主要動作.....	43
	成員簡歷.....	62

表目錄

表 1 設備清單	20
表 2 材料清單	22

圖目錄

圖 1 機構設計基底	2
圖 2 機構的伸縮	3
圖 3 完整機構圖	3
圖 4 伺服馬達 — MG90S	5
圖 5 伺服馬達 — MG996R	5
圖 6 風扇	6
圖 7 DC 12V 變壓器	6
圖 8 雙開繼電器電路	7
圖 9 LEAP MOTION	7
圖 10 ARDUINO UNO 板與腳位標示	9
圖 11 HC-05 藍芽模組的接腳	9
圖 12 ARDUINO IDE 介面	10
圖 13 PROCESSING 操作介面	11
圖 14 ANDROID STUDIO 程式設計介面	12
圖 15 手機程式的介面設計	12
圖 16 結構圖	13
圖 17 流程圖	14
圖 18 甘特圖	15
圖 19 手機藍牙配對介面	32
圖 20 手機操作主介面	36

第1章 前言

1-1 專題製作背景及目的

隨著科技越來越發達，人類的需求也變得更加多樣化，為了增加生活的便利性，我們善用所學，將高一至高三所學習的知識及專業技巧濃縮起來，於是嘗試做出一台可以利用手勢辨別的自動書報翻拍機，不僅能使手部不方便的人能夠輕鬆做到翻書的動作，也能使一般人在忙碌時透過手機藍牙操控，來自動翻拍書中的資料，並且將照片儲存在手機之中，輕鬆將資料帶著走，希望未來能夠廣泛運用在生活之中。

1-2 預期成果

本專題預期完成一台自動書報翻拍機，能利用 Leap Motion 偵測手勢，並依照手勢完成往左或往右翻書的動作，同時也能透過開關的切換，利用手機控制翻頁的頁數並進行翻拍動作。

專題製作，除了學習程式的撰寫，其中包括 Arduino、Processing 以及手機的 Android Studio，也學習到同學之間的相互合作、分工以及自己解決問題的能力，獲益良多。

第2章 理論探討

2-1 機構部分

2-1-1 機構設計

基底架構是採用壓克力板作為材料，邊框則是採用木頭，如圖 1，側邊設計成可伸縮式的邊框，是利用 L 型固定夾搭配鐵尺製作，用來適應各種不同大小的書，如圖 2。



圖 1 機構設計基底



圖 2 機構的伸縮

2-1-2 馬達位置

馬達主要控制翻書、固定頁面以及控制風扇吸起頁面，翻書馬達置於下方，將固定頁面馬達放置於兩側上方，最後將控制風扇馬達放置於兩側下方，如圖 3。

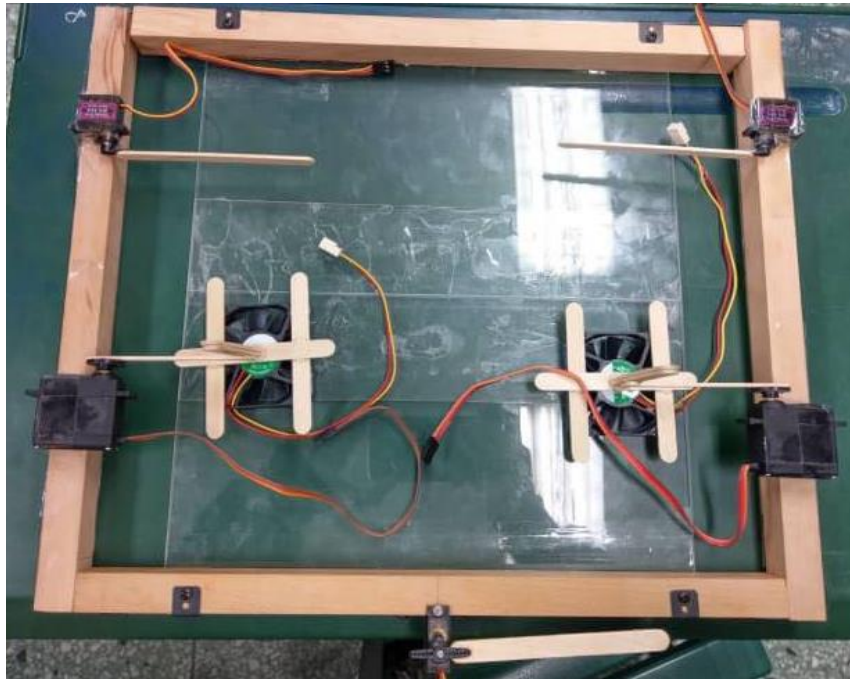


圖 3 完整機構圖

2-2 元件及硬體

2-2-1 伺服馬達

伺服馬達（Servomotor）是對用於使用伺服機構的馬達總稱，指依照命令動作的意義。

所謂伺服系統，就是依照指示命令動作所構成的控制裝置，應用於馬達的伺服控制，將感測器裝在馬達與控制對象機器上，偵測結果會返回伺服放大器與指令值做比較，伺服馬達有機械結構上的限制，旋轉的角度是 180 度，但並非完全的 180 度，可能是 1 度到 180 度，或是 -2 度到 177 度都有可能

伺服馬達的裝置由下列三者構成：

1. 發出動作指令的“指示裝置”（控制器，Controller）

2. 依照指示裝置的指示訊號與回饋訊號下等指令使馬達動作的“控制裝置”（伺服放大器），

3. 以及將由伺服放大器而來的電力供給驅動控制對象和偵測其狀態的“驅動、感測裝置”。

伺服馬達的動作特性是進行位置定位控制和動作速度控制，其主要特點是轉速可以精確控制，速度控制範圍廣，可以安定平順等速運轉之外，還可以根據需求隨時變更速度。在極低速度也可以穩定轉動。能迅速做出正轉與逆轉，也能迅速加減速。在由靜態改為動態運作或由動態改為靜態運作所需費時極短，而且即便有外力附加仍可以保持位置。並在額定容量範圍內瞬間產生大轉矩，輸出功率大且效率也高。

直流伺服馬達機體較細長，具有線性反應佳與簡單易於控制特性，因為直流伺服馬達因為操作容易，也就是旋轉方向由電流決定，並且旋轉速度由改變施加的電壓來控制，控制簡單所以廣泛使用因此現在直流伺服馬達是使用最多的馬達。

伺服馬達具有三條電線。紅色的為正電，深咖啡色是接地 GND，橘色的則是訊號線，電壓通常介於 4V~6V 之間，一般為 5V，該電源應盡可能與處理系統的電源隔離（因為伺服馬達會產生噪音）。

本專題分別使用了 MG90S（圖 4）及 MG996R（圖 5）兩種型號，其規格如下：

產品型號：MG90S

使用電壓：4.8V

工作扭矩：2KG/cm@4.8V

轉動角度：最大 180 度

反應轉速：0.11 秒/60 度(4.8V)

死區設定：5 微秒



圖 4 伺服馬達 — MG90S

(http://www.playrobot.com/2937-thickbox_default/mg90s-metal-gearred-micro-servo.jpg)

產品型號：MG996R

使用電壓：4.8~7.2V

工作扭矩：13KG/cm@4.8V

轉動角度：最大 180 度

反應轉速：0.20 秒/60 度 (4.8V)

死區設定：5 微秒



圖 5 伺服馬達 — MG996R

(<https://www.terasic.com.tw/attachment/archive/1028/image/MG996R.png>)

2-2-2 風扇

方型風扇 (圖 6)

型號：MS.G.(9).M

額定電壓：DC 12V

額定電流：0.23A

額定功率：2.76W

尺寸：6 x 6 x 1 cm

連接：3 線 3 針連接器

(紅線：電源線、黑線：地線、黃線：測速線)



圖 6 風扇

來源：<https://world.taobao.com/item/522011111927.htm>

2-2-3 DC12V 電路

為解決供應風扇 DC 12V 的電源，我們採用 DC 12V 變壓器 (圖 7) 轉接頭連上麵包板的電源，接上雙開繼電器電路 (圖 8)，這個電路是用來 Arduino 方便控制風扇開關所增設的。



圖 7 DC 12V 變壓器

<http://goods.ruten.com.tw/item/show?21707289813165>

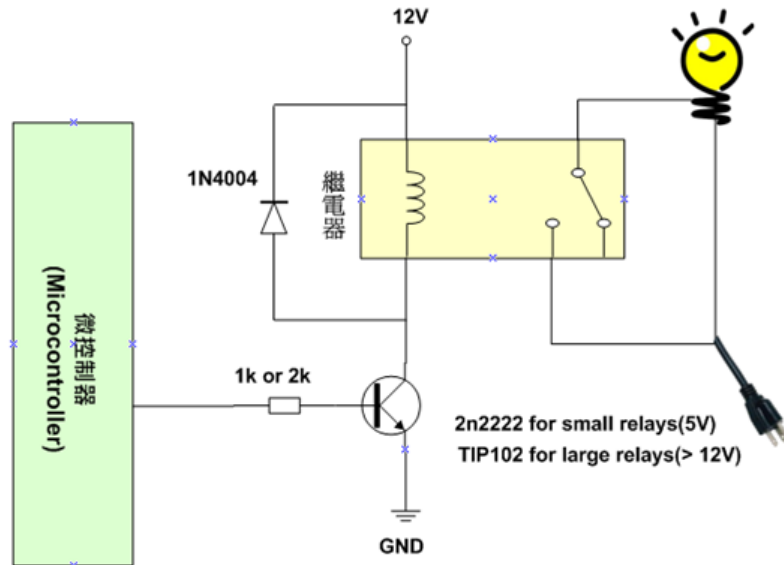


圖 8 雙開繼電器電路

<http://coopermaa2nd.blogspot.tw/2011/03/lab21-12v.html>

2-2-4 Leap Motion

Leap Motion (圖 9) 是由美國 Leap Motion 公司製作的一種體感控制器，類似於滑鼠，其裝置無須經由手接觸，就可按照手部動作進行操作。



圖 9 Leap Motion

<https://store-us.leapmotion.com/products/leap-motion-controller>

Leap Motion 只要由 USB 連接至電腦並且安裝驅動程式，就可以開始操控。其內部結構由三顆紅外線 LED 及兩個紅外線鏡頭組成，分別用來發射和接收光源，再利用鏡頭感測深度，以定位手與感測器之間的距離，並且精準的掌握細微的手部動作。

2-2-5 Arduino UNO

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，也因為其簡單操控的特性，我們選擇使用 UNO 板來配合專題，既可以達到我們的需求，也不會需要太多困難的操作。

Arduino Uno (圖 10) 的運作直流電壓為 5V，可由 USB 連接埠、電源輸入插座及 Vin 接腳提供電源。輸出接腳腳位則分成數位輸出以及類比輸出。

數位腳位編號由 D0 至 D13，D 則代表 Digital。其中 D3、D5、D6、D9、D10、D11 的數字編號旁，有波浪符號 (~)，表示這些腳位可以使用 PWM (Pulse Width Modulation) 的方法來模擬出類比訊號。

類比腳位編號則由 A0 至 A5，可用來接受類比電壓輸入，但不能輸出類比電壓，如前所述，類比電壓必須透過數位腳位 D3、D5、D6、D9、D10、D11，以 PWM 模擬。

ATmega328 內建類比數位轉換器 (Analog-to-digital converter，簡稱 ADC)，預設會將 0V 到 5V 轉換為 0 至 1023 的數值。

實際上，A0 至 A5 也可作為數位輸出、輸入腳位使用，此時 A0 至 A5 分別可視為 D14 至 D19。

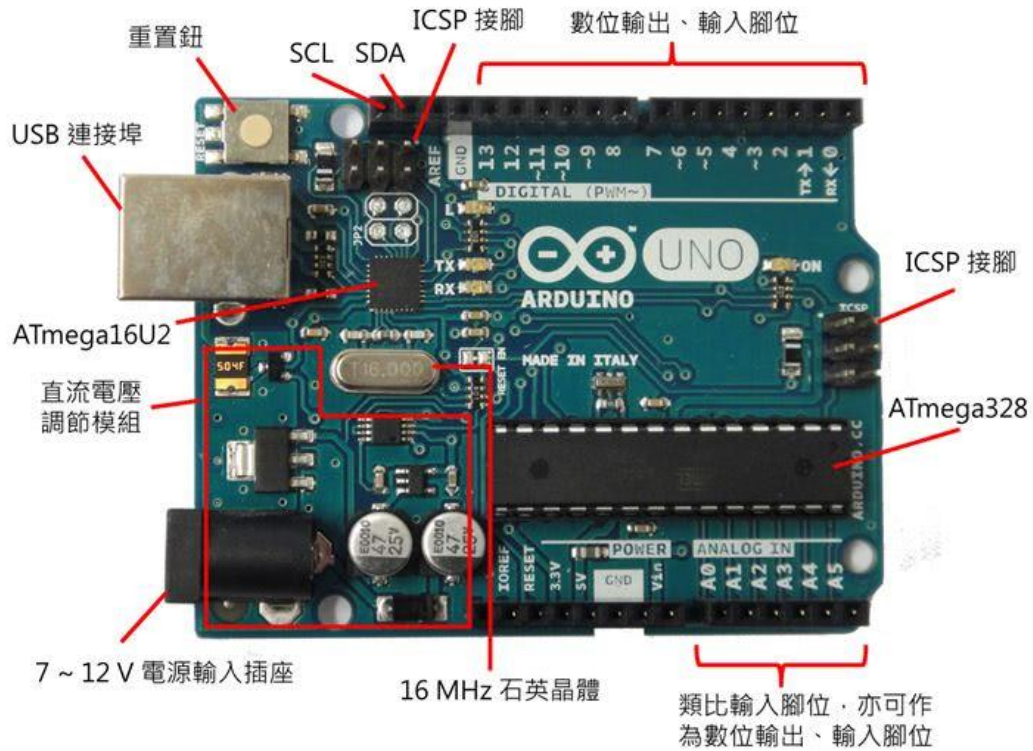


圖 10 Arduino UNO 板與腳位標示

<https://openhome.cc/Gossip/Books/mBlockArduino1-3and1-4.html>

2-2-6 HC-05 藍牙模組

HC-05 採用英國劍橋的 CSR (Cambridge Silicon Radio) 公司的 BC417143 晶片，支援藍牙 2.1+EDR 規範，是主 / 從 (master/slave) 一體型的模板，出廠預設通常是「從端」模式，但是能自行透過 AT 命令修改。HC-05 藍牙模組外觀與接腳如圖 11

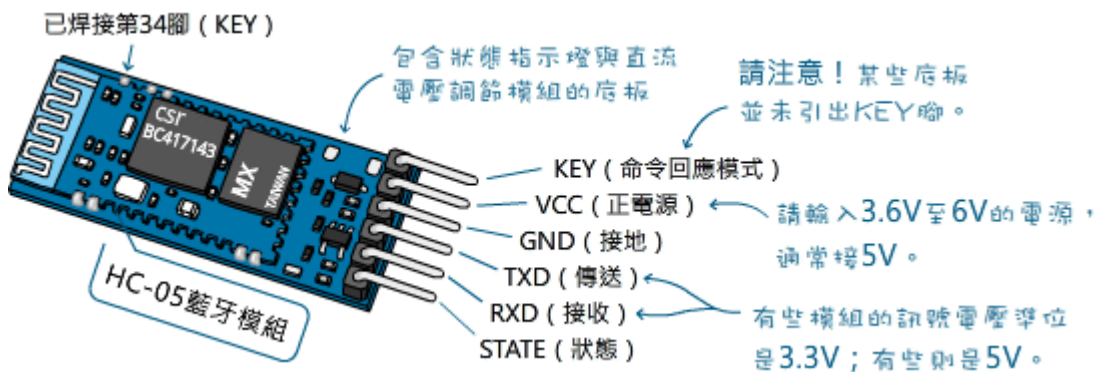


圖 11 HC-05 藍芽模組的接腳

2-3 軟體部分

2-3-1 Arduino IDE

Arduino 是以 AVR-GCC 和其他一些開源軟體為基礎，以 Java 編寫的跨平台應用軟體，軟體無需安裝，只要到官方網站下載並完成解壓縮後即可開始使用。

Arduino Software IDE 使用與 C 語言和 C++相仿的程式語言，並且提供了包含常見的輸入/輸出函式的函式庫。程式會包含兩個函式：

setup()：在程式執行開始時會執行一次，用於初始化設定。

loop()：直到 Arduino 硬體關閉前會重複執行函式放的程式碼。

此專題所使用的 Arduino 版本為 1.8.5，其介面如圖 12，白色部分為程式撰寫部分，黑色部分為偵錯顯示內容。

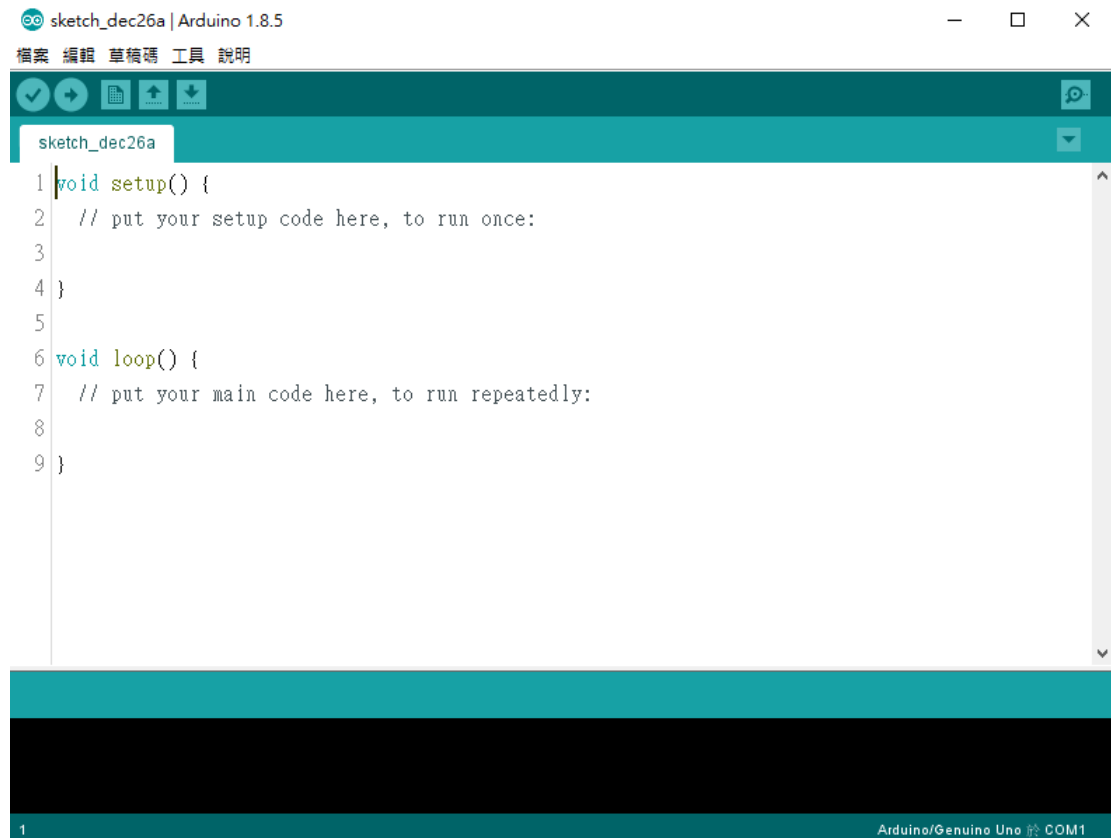


圖 12 Arduino IDE 介面

2-3-2 Processing

Processing 是一個開放原始碼的程式語言及開發環境，建立在 JAVA 語言之

上，在 Windows、OS X 和 Linux 平台上均可執行。

開發這套軟體的目的，是為了教導學習者一些以視覺呈現為主的電腦程式基礎，並且將這套軟體看作是一個軟體的描繪本，以及專業的製作工具。

其操作介面（圖 13）與 Arduino 很相像，程式結構通常會包含兩部分：

setup()：放置程式啟動要做的事情，如設定背景顏色或螢幕大小

draw()：每次畫面更新要做什麼事情，像是狀態的改變

在使用 Processing 的時候，可以把它想像成影格的播放，每一格的影格要播放怎麼樣的內容，我們都必須要在 draw 中定義，而本專題則使用 3.3.6 的版本。

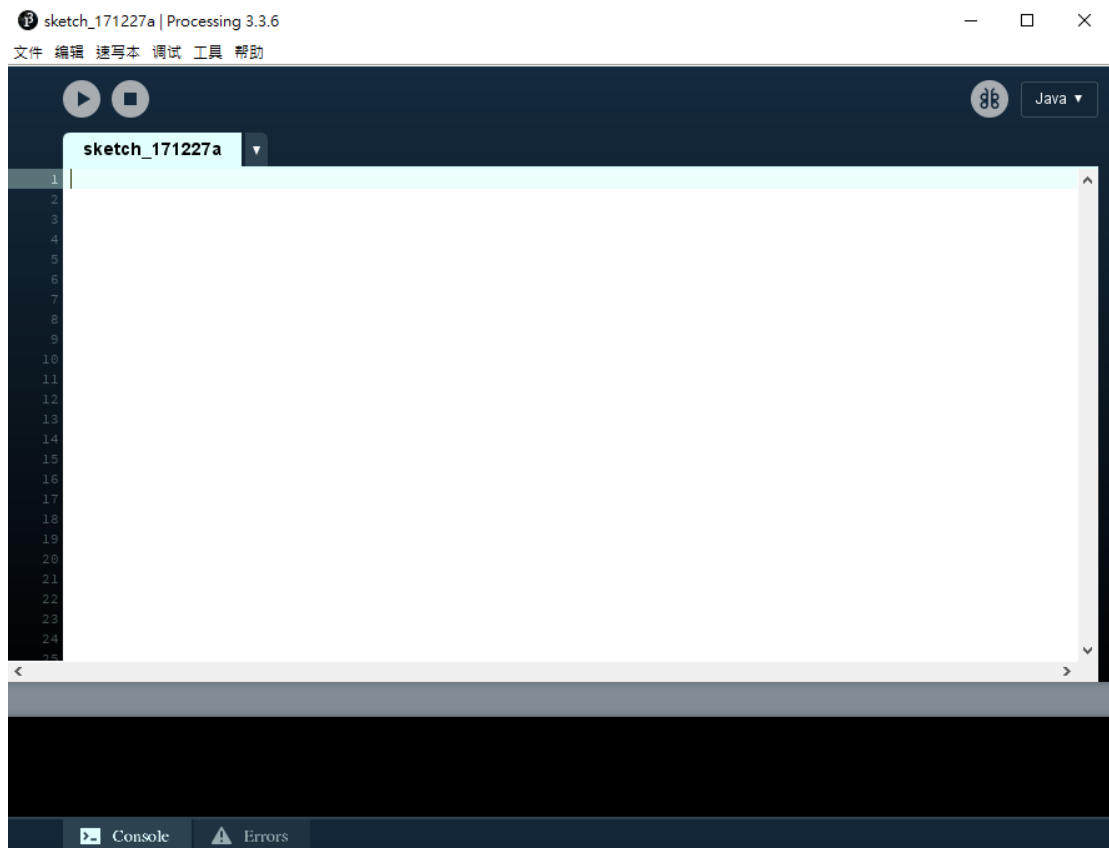


圖 13 Processing 操作介面

2-3-3 Android Studio

Android Studio 是 Google 公司開發的軟體，為 Android 平台開發程式的整合式開發環境，使用 JAVA 語言撰寫，供開發者免費使用，在 Windows、OS X 和 Linux 平台上均可執行。

智能化的程式碼編輯功能（圖 14、圖 15）是 Android Studio 核心特色之一，

當開發者編寫程式碼時，會自動提供相關的語法提示，並進一步協助開發者重組、完整化與分析程式碼。此外，Android Studio 也提供程式碼範本，來協助開發者建置應用程式的基本功能、並支援豐富的版面編輯工具。

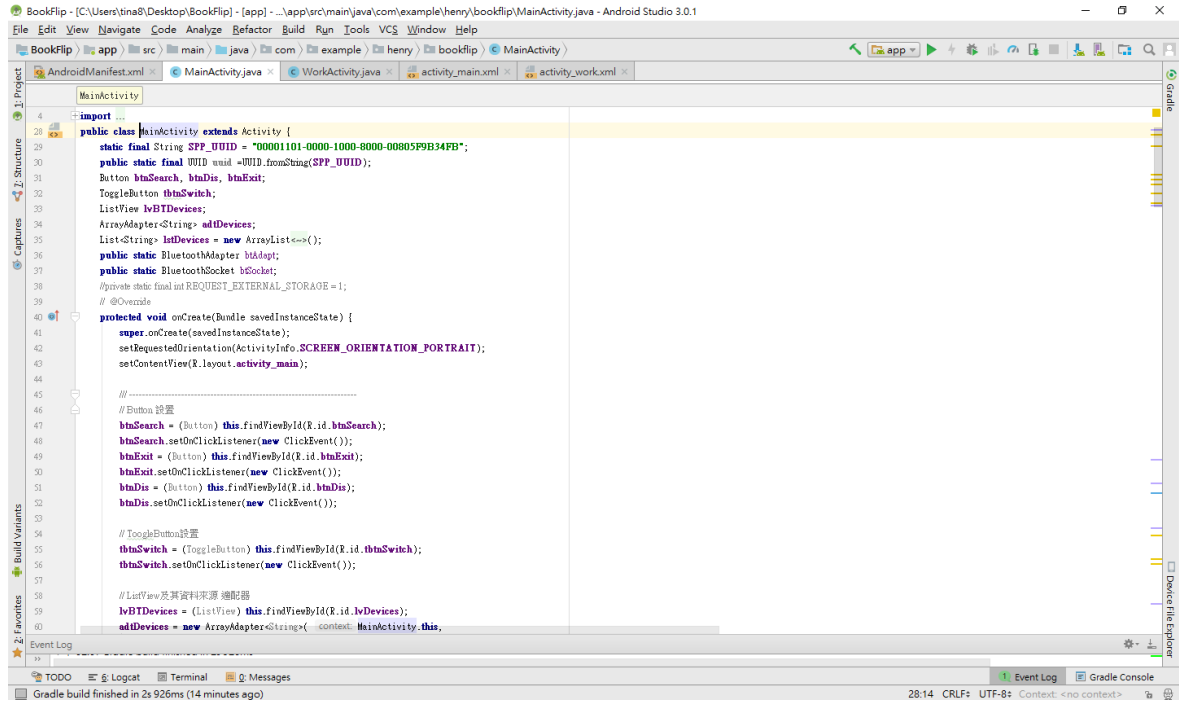


圖 14 Android Studio 程式設計介面

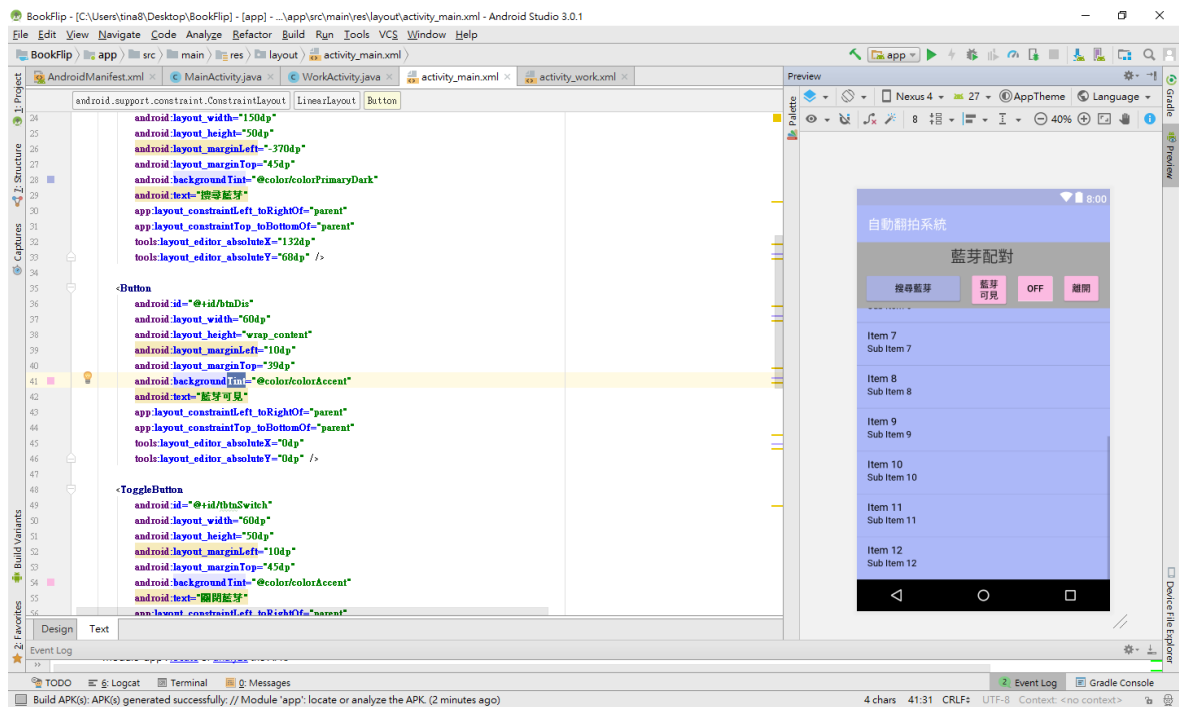


圖 15 手機程式的介面設計

第3章 專題準備

3-1 架構圖

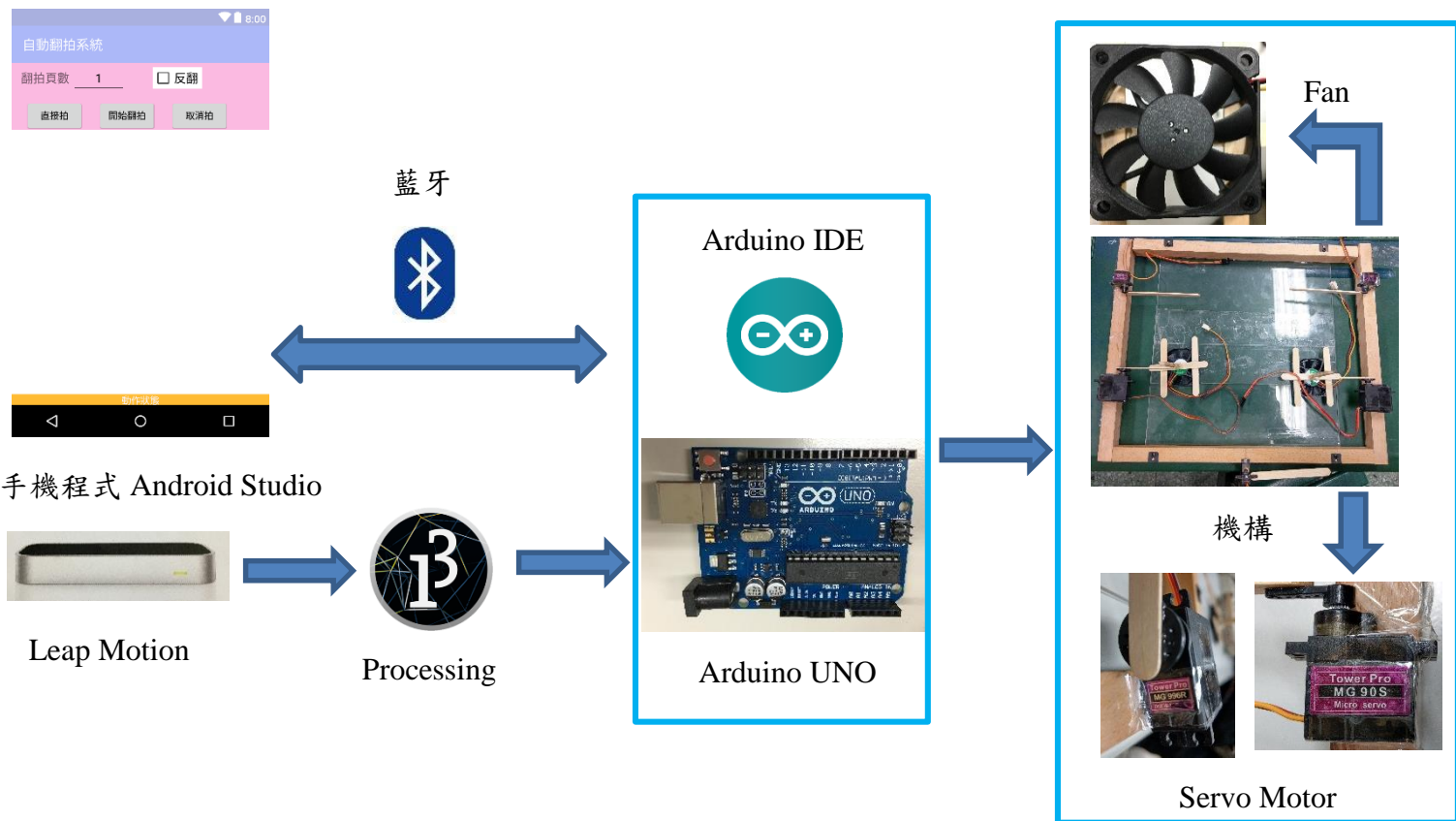


圖 16 結構圖

3-2 流程圖

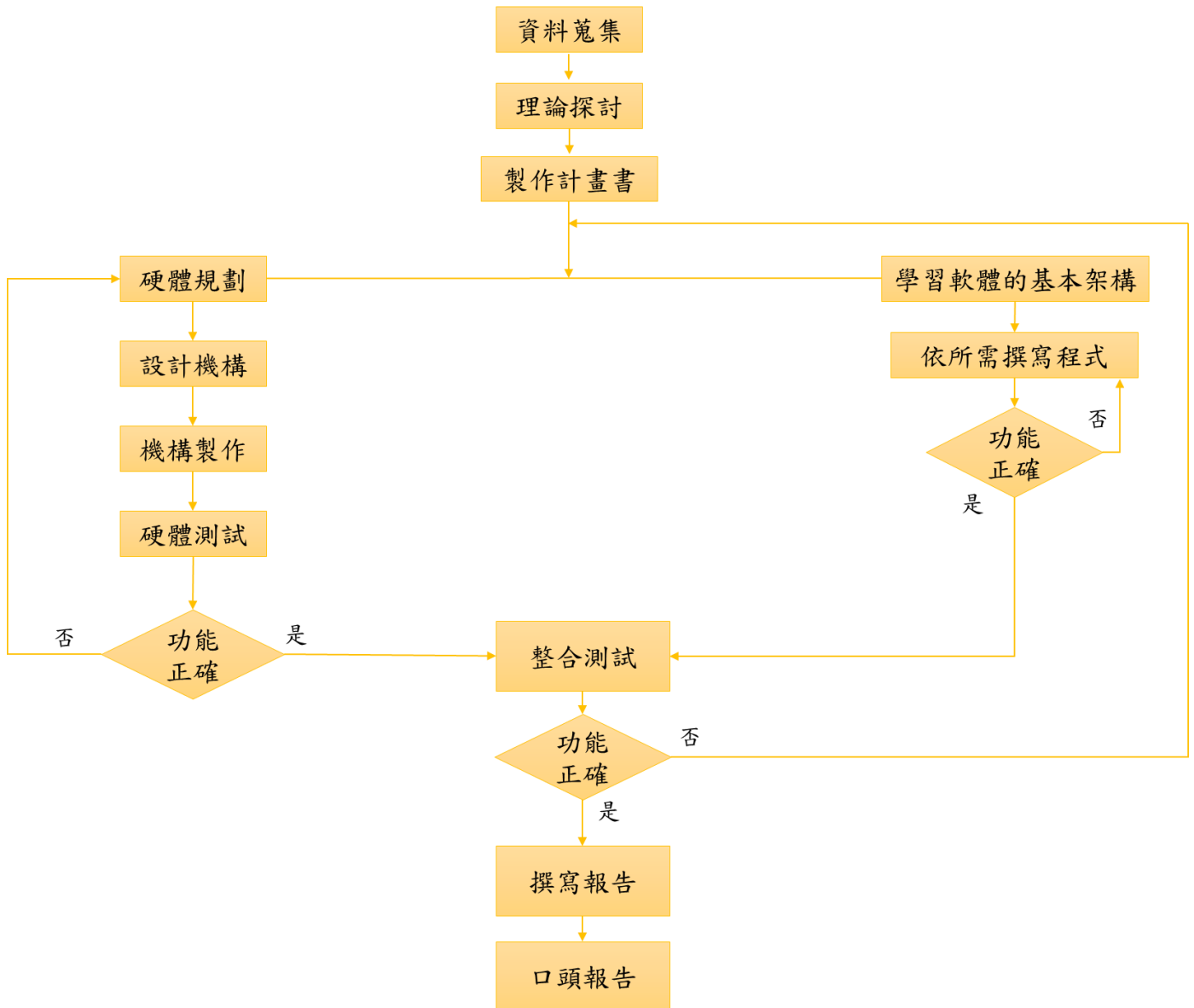


圖 17 流程圖

3-3 甘特圖

工作項目	週次 (日期)																		負責成員
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
資料蒐集	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				全
理論探討	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				全
擬定計畫書	■	■																	全
硬體規劃		■																	王、蘇
設計機構		■	■																王、蘇
機構製作				■	■	■	■	■						■	■	■			王、蘇
硬體測試									■	■	■								王、蘇
學習軟體基本架構		■	■	■	■	■	■	■	■	■									林、羅
依所需撰寫程式						■	■	■	■	■	■	■	■	■	■	■			林、羅
撰寫手機程式			■	■	■	■	■	■	■	■	■	■	■	■					羅、蘇
整合測試												■	■	■	■	■			全
撰寫報告															■	■	■		羅、王、蘇
影片拍攝																	■		林
口頭報告																		■	全
預定進度	5	10	15	20	30	35	40	45	55	60	65	80	85	90	95	98	100		累 積 百分比%

圖 18 甘特圖

第4章 專題成果

4-1 專題過程與問題

我們一開始是毫無頭緒的，之後上網查詢後發現了有人也有做過這專題，便先仿造他們的樣式，用珍珠板製作出了一份小模型，然後再依照小模型的樣式，使用木頭、壓克力板製作出了機構，製作過程並非一帆風順，因為小模型不需要考慮到材料的材質問題，也沒有所謂的方向大小，以及鎖定的問題，像我們再鎖定翻書馬達的時候，因為馬達的鎖孔限定使用原廠附贈的螺絲，而且那螺絲非常的小，光是要成功鎖緊就費了一大番努力才完成，之後各個馬達全定位後，帶起頁面的用具也讓我們傷透腦筋，使用風扇是我們最先想到的，但是真正使用時，我們發現若風扇沒有準確貼合頁面或是供給電壓沒有滿額的話，是無法成功帶起來的。在這製作過程我們有苦有樂，組員們偶爾也會發生爭執，不過大家都有著共同的目標，所以不管過程是怎樣的坎坷，就算這是一條天堂路，我們也能攜手互相勉勵地走完。

專題製作過程中，我們遇到了許多問題：

- 一、 基底製作：因為我們沒有製作過這種專題的經驗，對於還是初生之犢的我們可是措手不及，光是做出機構的小模型就花了許多時間。做完小模型之後隨即著要做真實的機構，不過這也讓我們遭遇到更多的問題，因為小模型不會有物件大小不合或是材質不合的問題，所以製作機構也是一個很頭痛的項目，需要一直思考著甚麼材質才不會無法搭配或是重量太重。
- 二、 伺服馬達：因為馬達的鎖孔是原廠大小，所以只能使用他原廠附贈的螺絲，不過原廠附贈的螺絲太小了很難使用螺絲起子鎖入機構上，也花了一些時間。
- 三、 翻頁：我們原本要採用真空幫浦來進行翻頁動作，無奈的是我們找了很多店家都找不到，後來決定網購這項產品，不過這產品的價格

高昂，只好打消這念頭。而後我們決定採用第二方案，也就是 DC12V 小風扇，這東西方便取得而且價格便宜，而且操作不難，只不過因為沒有使用到機械手臂的關係，而且風扇接觸頁面的角度設定問題影響到頁面吸起的動作，所以翻頁動作不大流暢，我們嘗試了許多角度去進行測試，最後發現必須要完全垂直於頁面，增加吸頁面積才能吸起，並製造兩頁之間的空隙進行翻書。

- 四、 Leap Motion：Leap Motion 的更新速率太快，導致位置的取樣資料太相似，我們想到的是多新增一個存取方向的變數，讓 if 的條件式變多，使程式正常執行。
- 五、 Android Studio：設計介面的 Radio Group 排列問題，於是上網找發現有其他屬性可以設定。其次為相片無法順利儲存，而我們加入讀取 SD 卡的權限後就可以了。
- 六、 Arduino UNO 板：有時候會碰到無法燒錄的問題，即使重新插拔連接線、重開啟程式、重新安裝驅動程式等等方式都無法解決，發生問題的原因也不明，所以只能重新開機以解決問題。

4-2 專題成果

我們經歷了許多波折後，大家一同討論並解決問題，終於將每個人所負責的部分逐一整合起來，合併成一個完整的專題：自動翻拍書報機，可以做到翻頁動作，由各個伺服馬達的配合以及動作順序，加上 DC12V 小風扇的吸頁功能，將頁面帶起後，由下方的伺服馬達所控制的翻書棒移動進行翻頁動作，而左右皆設有風扇，所以這是一個可以左右翻的翻書機，除此之外，也能透過手機端程式，利用藍牙輸入欲翻拍的頁數，透過 Arduino 外接的藍牙接收資料，並且按照指示動作。

第5章 結論與建議

5-1 結論

在未來的世界裡，無論工業界還是生活圈，都是朝向自動化、電子化來發展，可以解決時間的問題，自動書報翻拍機可以在你沒空閱讀時幫你將書的畫面拍照起來，正在翻拍的過程即可去做私事，抑或是稍作休息之類的，等到你都忙完了，準備出門，就可以去將翻拍完的手機隨身攜帶。

現在是個凡事講求速度的年代，一分一秒都很寶貴，為了讓時間運用達到極致，因此我們製作了這個專題。在這專題實施過程中，讓我們學到許多課本所沒有的知識，收穫豐碩，也讓我們知道分工合作是多麼的重要，團隊中絕對不能允許有米蟲的存在，因為每個人都需要負責一部份，最後在整合起來，倘若今天有人沒做好自己的部分，那最後整合的過程就不會順利了。

5-2 建議

我們認為使用“DC12V 小風扇”不是很好的做法，而使用小風扇不好的原因有兩個：

- 一、風扇對頁面角度問題：如果你風扇降下時，沒有貼平頁面，會導致吸合面不足而造成頁面無法順利帶起。
- 二、供應電壓問題：若風扇供應電壓不到 12V 的話，對於吸頁這功能會不太正常，造成頁面無法順利被帶起。

因為我們沒有使用機械手臂來控制風扇的起落，而是使用單顆伺服馬達來控制，雖然這樣也是可以達到我們要的目的，不過這還不是最好的辦法，我們有發現有人使用真空幫浦搭配吸盤，同時使用伺服馬達控制起落，那個效果很好，不過由於真空幫浦的價格不菲而且找尋實體店面賣家不易，所以我們沒有採用這方法，不過相信到了科技大學後，有比較多的知識和資金，可以做出更好的改善。

此外，利用 Leap Motion 控制翻頁時需要透過電腦的 Processing 程式才能夠操控，使機構無法獨立運作，在未來可能還要改善如何不透過電腦就能夠完成所有動作。

參考文獻

1. Cubie。(2014年11月01日)。HC-05與HC-06藍牙模組補充說明(一)。
2017年12月20日擷取自 <https://swf.com.tw/?p=693>。
2. gitbook.net。(2015)。Android登錄實例。2017年12月7日擷取自 gitbook.net:
http://tw.gitbook.net/android/android_login_screen.html。
3. MaaCooper。(2011年3月27日)。Arduino筆記-Lab21用繼電器控制12V
風扇。2017年12月15日擷取自
<http://coopermaa2nd.blogspot.tw/2011/03/lab21-12v.html>。
4. 李家祥。(2007年7月14日)。Processing介紹與教學。2017年9月20日擷
取自 所以然實驗室：<http://www.soiizen.com>。
5. 林信良。(2015年9月24日)。Arduino Uno功能簡介。2017年12月13日
擷取自 openhome.cc：<https://openhome.cc/Gossip/Books/mBlockArduino1-3and1-4.html>。
6. 施威銘。(2016)。Android App程式設計教本。臺北市：旗標出版股份有限公
司。
7. 梅克2工作室。(2014)。Arduino微電腦控制實習。新北市：台科大圖書股份
有限公司。
8. 葉正聖。(2016)。2016教學軟體設計 Week 08 04 Leap Motion for Processing。
2017年7月25日擷取自 <https://watchyt.com/videos/2016jiao-xue-ruan-ti-she-ji-week-08-04-leap-motion-for-processing?v=IpmzSDfHgXA>。
9. 蔣大偉。(2012)。Processing入門互動圖形實作介紹。臺北市：基峰資訊股份
有限公司。
10. 環游天下 mey。(2014年12月11日)。Android如何排版RadioGroup的
RadioButton?。2017年11月22日擷取自 百度經驗：
<https://jingyan.baidu.com/article/925f8cb81c5350c0dde056fd.html>。

附錄

附-1 設備清單

類別	設備、軟體名稱	應用說明
硬體	個人筆記型電腦	1. 蒐集資料 2. 撰寫程式 3. 測試功能 4. 影片剪輯
硬體	個人行動通訊裝置	1. 蒐集資料 2. 照片、影片拍攝 3. 過程記錄
硬體	HTC_ONE_M9PLUS	1. 翻拍使用
硬體	Leap Motion	1. 偵測手勢與辨別
硬體	Arduino UNO	1. 控制伺服馬達
硬體	HC-05 藍牙模組	1. 作為手機及 Arduino UNO 版的溝通
軟體	Arduino	1. 控制伺服馬達 2. 硬體與軟體之間的溝通
軟體	Processing	1. 控制 Leap Motion 2. 連接 Leap Motion 與 Arduino
軟體	Android Studio	1. 撰寫手機程式
軟體	MS Word 2016	1. 計畫書的撰寫 2. 製作個人報告及團體報告
軟體	MS Power Point 2016	1. 簡報製作 (發表與個人報告)

表 1 設備清單

附-2 材料清單

類別名稱	材 料 名 稱	單位	數量	應 用 說 明	備註
伺服馬達	MG90S	顆	3	1. 控制翻書棒 2. 控制壓書棒	
伺服馬達	MG996R	顆	2	控制風扇馬達	
繼電器	RY5W-K	顆	2	控制風扇的開關	
電阻	1k Ω	顆	2	繼電器電路用	
電阻	500 Ω	顆	1	指撥開關限流電阻	
電晶體	2N222A	顆	2	繼電器電路用	
二極體	1N4004	顆	2	繼電器電路用	
開關	指撥開關	個	1	切換 Leap Motion 操控 / 手機藍牙操控	
12V 風扇	MS.G.(9).M	個	2	吸起頁面	
耗材	手機夾	個	1	固定翻拍手機	
耗材	尺	支	4	伸縮用具	
耗材	冰棒棍	根	3	製作機構—翻書棒、壓書棒	
耗材	木頭	塊	n	製作機構	
耗材	單心線	條	n	連接伺服馬達及 Arduino 板	
耗材	排針	根	n	焊接單心線以連接電路板及 Arduino 版	
耗材	排針母座	個	n	連接排針	
耗材	電路板	片	1	製作繼電器及開關電路用	
耗材	壓克力板	片	2	製作機構	

耗材	直角夾板	片	10	1. 固定翻頁馬達 2. 固定伸縮尺	
耗材	熱熔膠、槍	組	1	固定各個馬達	

表 2 材料清單

附-3 程式碼

附-3-1 Processing 程式碼

```

import de.voidplus.leapmotion.*;
import processing.serial.*;
import cc.arduino.*;

LeapMotion leap;
Arduino arduino;
Serial myPort;

int dir=-1; // -1 沒有方向，11 右手向左，99 右手向右
float old_x=0;
int count;

void setup(){
  size(640,480,P3D);
  leap = new LeapMotion(this);
  myPort = new Serial(this,Arduino.list()[0],57600);
}

void draw(){
  background(255);
  for(Hand : leap.getHands()){
    PVector pos=hand.getPosition();
    delay(10);
    if(hand.isLeft()){ // 如果是左手
      ellipse(pos.x,pos.y,200,100);
      // 如果目前方向
      if(dir!=11 && pos.x>500) {old_x=pos.x;dir=11;} // 依據手位置在座標 500
      處，設為左移的起點
    }
  }
}

```

```

if(dir!=99 && pos.x<150) {old_x=pos.x;dir=99;}
switch(dir) {
  case 11: // 目前向左移動中
    if((pos.x-old_x)>10) dir=-1; // 如果發現 x 往回，方向取消
    if((old_x-pos.x)>250) { // 如果發現向左超過 250
      count++; // 你要進行的工作
      old_x=320;dir=-1; // 變數初始化
      myPort.write('B');
      println('B');
    }
    break;
  case 99: // 目前向右移動中
    if((old_x-pos.x)>10) dir=-1; // 如果發現 x 往回，方向取消
    if((pos.x-old_x)>250) { // 如果發現向右超過 250
      count--; // 你要進行的工作
      old_x=320;dir=-1; // 變數初始化
      myPort.write("F");
      println('F');
    }
    break;
}
}
else if(hand.isRight()){ // 如果是右手
  rect(pos.x,pos.y,10,100);
  // 如果目前方向
  if(dir!=11 && pos.x>500) {old_x=pos.x;dir=11;} // 依據右手位置在座標 500
  處，設為左移的起點
  if(dir!=99 && pos.x<150) {old_x=pos.x;dir=99;}
  switch(dir) {
    case 11: // 目前向左移動中
      if((pos.x-old_x)>10) dir=-1; // 如果發現 x 往回，方向取消
      if((old_x-pos.x)>250) { // 如果發現向左超過 250
        count++; // 你要進行的工作
        old_x=320;dir=-1; // 變數初始化
        myPort.write('B');
        println('B');
      }
      break;

```

```
case 99: // 目前向右移動中
    if((old_x-pos.x)>10) dir=-1; // 如果發現 x 往回，方向取消
    if((pos.x-old_x)>250) { // 如果發現向右超過 250
        count--; // 你要進行的工作
        old_x=320;dir=-1; // 變數初始化
        myPort.write("F");
        println("F");
    }
    break;
}
}
}
```

附-3-2 Arduino 程式碼

```
#include <Servo.h>
#include <SoftwareSerial.h>
Servo myservo; //翻書馬達
Servo myservo2; //壓書棒 1
Servo myservo3; //壓書棒 2
Servo fanservoright;
Servo fanservoleft;
//=====翻書機變數宣告=====
int pos=90; // 馬達初始角度
int pos2=45;
int pos3=130;
int fanservoposright=180;
int fanservoposleft=0;
const int relayPin2 = 8; // 右繼電器
const int relayPin = 9; // 左繼電器
int relayState = 0; // 繼電器狀態
int relayState2 = 0;
const int switch1 = 4; // 切換功能開關
int State = 0; // 切換狀態
//=====藍牙相關變數宣告=====
int BTtx = 2; // TX
int BTrx = 3; // RX
char ReadBT; // 藍牙接收數值
```

```

char RBT;
SoftwareSerial bluetooth(BTtx, BTrx); // TX | RX

void hand(char RBT);
void setup() {
  pinMode(switch1, INPUT); // 把 switch 開關設置成 INPUT
  pinMode(relayPin, OUTPUT); // 把 relayPin 設置成 OUTPUT
  pinMode(relayPin2, OUTPUT);
  myservo.attach(7); // 翻書棒馬達輸出 pin
  myservo2.attach(10); // 右壓書棒輸出 pin
  myservo3.attach(11); // 左壓書棒輸出 pin
  fanservoright.attach(12); // 右風扇馬達輸出 pin
  fanservoleft.attach(13); // 左風扇馬達輸出 pin
  myservo.write(pos); // 定位馬達位置
  myservo2.write(pos2);
  myservo3.write(pos3);
  fanservoright.write(fanservoposright);
  fanservoleft.write(fanservoposleft);
  Serial.begin(57600); // 與電腦溝通之鮑率
  delay(100);
  bluetooth.begin(57600); // 藍芽溝通鮑率
  Serial.println("Bluetooth setup complete.");
}

void loop() {
  State = digitalRead(switch1); // 讀取 switch 高低準位
  if (State == HIGH) { // 利用 processing 控制翻書機
    if(Serial.available()){
      ReadBT=Serial.read();
      hand(ReadBT); // 如果從 processing 方收到數據 則開始下方的判斷
    }
  }
  else{ // 利用手機藍牙控制翻書機
    if (bluetooth.available()) { // 如果藍芽序列埠有資料
      ReadBT = bluetooth.read(); // 將讀取資料放入變數中
      if(ReadBT=='F' || ReadBT=='B' || ReadBT=='P') {
        hand(ReadBT);
      }
    }
  }
}

```



```

    bluetooth.write('K');
  }
  else Serial.println(ReadBT);
  }
}

//=====左風扇馬達上下控制=====
void leftservoup(){
  for(fanservoposleft = 70; fanservoposleft >= 0; fanservoposleft -= 1) {
    fanservoleft.write(fanservoposleft);
    delay(15);
  }
}

void leftservodown(){
  for(fanservoposleft = 0; fanservoposleft <= 90; fanservoposleft += 1) {
    fanservoleft.write(fanservoposleft);
    delay(15);
  }
  delay(1500);
  for(fanservoposleft = 90; fanservoposleft >= 70; fanservoposleft -= 1) {
    fanservoleft.write(fanservoposleft);
    delay(350);
  }
}

//=====右風扇馬達上下控制=====
void rightservoup(){
  for(fanservoposright = 105; fanservoposright <= 180; fanservoposright += 1) {
    fanservoright.write(fanservoposright);
    delay(15);
  }
}

void rightservodown(){
  for(fanservoposright = 180; fanservoposright >= 90; fanservoposright -= 1) {
    fanservoright.write(fanservoposright);
    delay(15);
  }
}

```

```

delay(1500);
for(fanservoposright = 90; fanservoposright <= 105; fanservoposright += 1) {
  fanservoright.write(fanservoposright);
  delay(350);
}
}

//=====左/右壓書棒副程式=====

void bkservo(){
  for(int pos2 = 45; pos2 < 150;pos2++ ) {
    myservo2.write(pos2);
  }
  for(int pos3 = 130; pos3 > 20 ; pos3--){
    myservo3.write(pos3);
  }
}

void bkservo2(){
  for(int pos2 = 150; pos2 > 45;pos2-- ) {
    myservo2.write(pos2);
  }
  for(int pos3 = 20; pos3 < 130 ; pos3++){
    myservo3.write(pos3);
  }
}

//=====繼電器副程式=====

void switchRelay(){
  if (relayState == 0)
    relayState = 1;          // 把繼電器狀態改為 ON
  else
    relayState = 0;          // 把繼電器狀態改為 OFF
  digitalWrite(relayPin, relayState); // 讓繼電器作動, 切換開關
}

void switchRelay2(){
  if (relayState2 == 0)
    relayState2 = 1;          // 把繼電器狀態改為 ON
}

```

```

else
  relayState2 = 0;           // 把繼電器狀態改為 OFF
  digitalWrite(relayPin2, relayState2); // 讓繼電器作動, 切換開關
}
void hand(char RBT){
  switch(RBT) {
    case 'F': // 手機傳遞左翻代碼;processing 傳送 F 代碼
      for (int pos = 0 ; pos < 175 ; pos++ ) {
        myservo.write(pos); // 翻書馬達從 0 度偏轉到 180 度
        delay(15);
      }
      delay(500);
      bkservo2(); // 開始壓書棒副程式
      delay(700);
      switchRelay();
      leftservodown();
      delay(0);
      for (int pos = 175 ; pos > 0 ; pos-- ) {
        myservo.write(pos); // 馬達從 180 度偏轉到 0 度
        delay(5);
      }
      leftservoup();
      switchRelay();
      delay(1000);
      bkservo(); // 開始壓書棒第二個副程式
      delay(100);
      break;
    case 'B': // 右翻傳遞右翻代碼
      for (int pos = 175 ; pos > 0 ; pos-- ) {
        myservo.write(pos);
        delay(15);
      }
      delay(500);
      bkservo2();
      delay(700);
      switchRelay2();
      rightservodown();
      for (int pos = 0; pos < 175; pos++ ) {

```

```

myservo.write(pos);
delay(5);
}
rightservoup();
switchRelay2();
delay(1000);
bkservo();
delay(100);
break;
default:
Serial.println(RBT);
break;
}
}

```

附-3-3 Android Studio 設計介面—藍牙

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent">
<LinearLayout
android:id="@+id/LinearLayout01"
android:layout_width="wrap_content"
android:layout_height="100dp"
android:background="@android:color/darker_gray">

<TextView
android:id="@+id/textView"
android:layout_width="380dp"
android:layout_height="40dp"
android:text="藍芽配對"
android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"
android:textSize="24sp" />

```

<Button

```
android:id="@+id/btnSearch"  
android:layout_width="150dp"  
android:layout_height="50dp"  
android:layout_marginLeft="-370dp"  
android:layout_marginTop="45dp"  
android:backgroundTint="@color/colorPrimaryDark"  
android:text="搜尋藍芽"  
app:layout_constraintLeft_toRightOf="parent"  
app:layout_constraintTop_toBottomOf="parent"  
tools:layout_editor_absoluteX="132dp"  
tools:layout_editor_absoluteY="68dp" />
```

<Button

```
android:id="@+id/btnDis"  
android:layout_width="60dp"  
android:layout_height="wrap_content"  
android:layout_marginLeft="10dp"  
android:layout_marginTop="39dp"  
android:backgroundTint="@color/colorAccent"  
android:text="藍芽可見"  
app:layout_constraintLeft_toRightOf="parent"  
app:layout_constraintTop_toBottomOf="parent"  
tools:layout_editor_absoluteX="0dp"  
tools:layout_editor_absoluteY="0dp" />
```

<ToggleButton

```
android:id="@+id/tbtnSwitch"  
android:layout_width="60dp"  
android:layout_height="50dp"  
android:layout_marginLeft="10dp"  
android:layout_marginTop="45dp"  
android:backgroundTint="@color/colorAccent"  
android:text="關閉藍芽"  
app:layout_constraintLeft_toRightOf="parent"  
app:layout_constraintTop_toBottomOf="parent"  
tools:layout_constraintBaseline_creator="0"  
tools:layout_editor_absoluteX="0dp"
```

```
tools:layout_editor_absoluteY="68dp" />
```

```
<Button
```

```
    android:id="@+id/btnExit"  
    android:layout_width="60dp"  
    android:layout_height="50dp"  
    android:layout_marginLeft="10dp"  
    android:layout_marginTop="45dp"  
    android:backgroundTint="@color/colorAccent"  
    android:text="離開"  
    app:layout_constraintLeft_toRightOf="parent"  
    app:layout_constraintTop_toBottomOf="parent"  
    tools:layout_constraintBaseline_creator="0"  
    tools:layout_editor_absoluteX="264dp"  
    tools:layout_editor_absoluteY="68dp" />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginLeft="0dp"  
    android:layout_marginTop="100dp"  
    android:background="@android:color/holo_purple"  
    android:orientation="horizontal">
```

```
<ListView
```

```
    android:id="@+id/lvDevices"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="@color/colorPrimary"  
    android:smoothScrollbar="true"  
    android:stackFromBottom="true" />
```

```
</LinearLayout>
```

```
</android.support.constraint.ConstraintLayout>
```



圖 19 手機藍牙配對介面

附-3-4 Android Studio 設計介面—主介面

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.henry.bookflip.WorkActivity">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@color/colorAccent"
    android:orientation="horizontal">
```

<TextView

```
android:id="@+id/textView2"  
android:layout_width="80dp"  
android:layout_height="wrap_content"  
android:layout_marginLeft="10dp"  
android:layout_marginTop="0dp"  
android:text="翻拍頁數"  
android:textAlignment="center"  
android:textSize="18sp" />
```

<EditText

```
android:id="@+id/txtNumber"  
android:layout_width="80dp"  
android:layout_height="wrap_content"  
android:layout_alignParentTop="true"  
android:layout_centerHorizontal="true"  
android:layout_marginLeft="0dp"  
android:layout_marginTop="0dp"  
android:ems="10"  
android:inputType="number"  
android:singleLine="true"  
android:text="1"  
android:textAlignment="center" />
```

<CheckBox

```
android:id="@+id/check3D"  
android:layout_width="74dp"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentTop="true"  
android:layout_marginLeft="40dp"  
android:layout_marginTop="0dp"  
android:layout_toEndOf="@+id/txtNumber"  
android:background="@android:color/background_light"  
android:onClick="oncheck3DClick"  
android:text="反翻"  
android:textSize="18sp" />
```



```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:id="@+id/linearLayout"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="50dp"  
    android:background="@color/colorAccent"  
    android:orientation="horizontal"  
    app:layout_constraintTop_toTopOf="parent">
```

```
<Button
```

```
    android:id="@+id/btnNow"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id/btnSend"  
    android:layout_alignBottom="@+id/btnSend"  
    android:layout_alignParentStart="true"  
    android:layout_marginLeft="20dp"  
    android:layout_marginTop="5dp"  
    android:onClick="onbtnSendClick"  
    android:text="直接拍" />
```

```
<Button
```

```
    android:id="@+id/btnSend"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id/btnCancel"  
    android:layout_alignBottom="@+id/btnCancel"  
    android:layout_centerHorizontal="true"  
    android:layout_marginLeft="20dp"  
    android:layout_marginTop="5dp"  
    android:onClick="onbtnSendClick"  
    android:text="開始翻拍" />
```

```
<Button
```

```
    android:id="@+id/btnCancel"  
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_below="@+id/check3D"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:onClick="onbtnSendClick"
    android:text="取消拍" />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="100dp"
    android:background="@android:color/holo_purple"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout">
```

```
<TextureView
```

```
    android:id="@+id/textureview"
    android:layout_width="match_parent"
    android:layout_height="495dp"
    android:layout_marginTop="0dp" />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="495dp"
    android:background="@android:color/holo_purple"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout">
```

```
<TextView
```

```
    android:id="@+id/msgText"
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:layout_marginTop="0dp"
    android:background="@android:color/holo_orange_light"
    android:focusable="false"
```

```

        android:text="動作狀態"
        android:textAlignment="center"

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#fbf8ff"
        android:textIsSelectable="true"
        android:textSize="14dp"
        app:layout_constraintBottom_toBottomOf="parent" />
    </LinearLayout>
</android.support.constraint.ConstraintLayout>

```

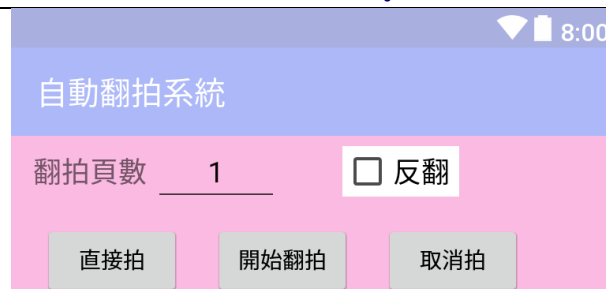


圖 20 手機操作主介面

附-3-5 Android Studio 藍牙連接

```

package com.example.henry.bookflip;

import java.io.IOException;
import java.util.ArrayList;

```

```

import java.util.List;
import java.util.UUID;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.ToggleButton;
import android.content.pm.ActivityInfo;
public class MainActivity extends Activity {
    static final String SPP_UUID = "00001101-0000-1000-8000-00805F9B34FB";
    public static final UUID uuid =UUID.fromString(SPP_UUID);
    Button btnSearch, btnDis, btnExit;
    ToggleButton tbtnSwitch;
    ListView lvBTDevices;
    ArrayAdapter<String> adtDevices;
    List<String> lstDevices = new ArrayList<String>();
    public static BluetoothAdapter btAdapt;
    public static BluetoothSocket btSocket;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT)
    ;

```

```

setContentViewById(R.layout.activity_main);

// -----
// Button 設置
btnSearch = (Button) this.findViewById(R.id.btnSearch);
btnSearch.setOnClickListener(new ClickEvent());
btnExit = (Button) this.findViewById(R.id.btnExit);
btnExit.setOnClickListener(new ClickEvent());
btnDis = (Button) this.findViewById(R.id.btnDis);
btnDis.setOnClickListener(new ClickEvent());

// ToggleButton 設置
tbtnSwitch = (ToggleButton) this.findViewById(R.id.tbtnSwitch);
tbtnSwitch.setOnClickListener(new ClickEvent());

// ListView 及其資料來源 適配器
lvBTDevices = (ListView) this.findViewById(R.id.lvDevices);
adtDevices = new ArrayAdapter<String>( MainActivity.this,
    android.R.layout.simple_list_item_1, lstDevices);
lvBTDevices.setAdapter(adtDevices);
lvBTDevices.setOnClickListener(new ItemClickEvent());
// =====
// Quick permission check
int permissionCheck =
this.checkSelfPermission("Manifest.permission.ACCESS_FINE_LOCATION");
    permissionCheck +=
this.checkSelfPermission("Manifest.permission.ACCESS_COARSE_LOCATION");
    if (permissionCheck != 0) {
        this.requestPermissions(new
String[] {android.Manifest.permission.ACCESS_FINE_LOCATION,
android.Manifest.permission.ACCESS_COARSE_LOCATION}, 1001); //Any
number
    }
// Check if we have write permission
try {
    // 取得寫 SD 卡權限

```

```

        int permission1 =
this.checkSelfPermission("WRITE_EXTERNAL_STORAGE");
        if (permission1 != PackageManager.PERMISSION_GRANTED)
    {
        // 無權限，向使用者請求
        this.requestPermissions(new
String[] { android.Manifest.permission.WRITE_EXTERNAL_STORAGE }, 1);
    }
        int permissionCheck2 =
this.checkSelfPermission("Manifest.permission.CAMERA");
        if (permissionCheck2 !=
PackageManager.PERMISSION_GRANTED) {
            this.requestPermissions(new
String[] { android.Manifest.permission.ACCESS_FINE_LOCATION,
android.Manifest.permission.ACCESS_COARSE_LOCATION }, 1001); //Any
number
                //java.lang.SecurityException: Lacking privileges to access
camera service
            }

        } catch(Exception e) {
            e.printStackTrace();
        }
        // =====
        btAdapt = BluetoothAdapter.getDefaultAdapter(); // 初始化本機藍牙
功能

        if(btAdapt==null) {
            //Show a mensag. that the device has no bluetooth adapter
            Toast.makeText(getApplicationContext(), "Bluetooth Device
Not Available", Toast.LENGTH_LONG).show();
            //finish apk
            finish();
        }

        if (btAdapt.getState() == BluetoothAdapter.STATE_OFF) // 讀取藍
牙狀態並顯示
            tbtnSwitch.setChecked(false);

```

```

else if (btAdapt.getState() == BluetoothAdapter.STATE_ON)
    tbtnSwitch.setChecked(true);

// 註冊 Receiver 來獲取藍牙設備相關的結果
IntentFilter intent = new IntentFilter();
intent.addAction(BluetoothDevice.ACTION_FOUND); // 用
BroadcastReceiver 來取得搜索結果

intent.addAction(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
this.registerReceiver(searchDevices, intent);
}

private final BroadcastReceiver searchDevices = new
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        // 搜索設備時，取得設備的 MAC 位址
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            BluetoothDevice device = intent
                .getParcelableExtra(BluetoothDevice.EXTRA_DE
VICE);

            String str = device.getName() + "|" + device.getAddress();
            if (lstDevices.indexOf(str) == -1) { // 防止重複添加
                lstDevices.add(str); // 獲取設備名稱和 mac 位址
                adtDevices.notifyDataSetChanged();
            }
        }
        if
        (BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
            if (lstDevices == null || lstDevices.isEmpty()) {
                Toast.makeText(getApplicationContext(), ">無藍芽裝
置<", Toast.LENGTH_LONG).show();
            }
            else {
                Toast.makeText(getApplicationContext(), "搜尋完畢",
                Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

    }

}

};

@Override
protected void onDestroy() {
    this.unregisterReceiver(searchDevices);
    super.onDestroy();
    android.os.Process.killProcess(android.os.Process.myPid());
}

class ItemClickEvent implements AdapterView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int
arg2, long arg3) {
        if (btAdapt.isDiscovering()) btAdapt.cancelDiscovery();

        String str = lstDevices.get(arg2);
        String[] values = str.split("\\|");
        String address=values[1];
        Log.e("address",values[1]);

        Toast.makeText(MainActivity.this, values[1],
Toast.LENGTH_LONG).show();

        UUID uuid = UUID.fromString(SPP_UUID);
        BluetoothDevice btDev = btAdapt.getRemoteDevice(address);

        try {
            btSocket =
btDev.createRfcommSocketToServiceRecord(uuid);
            btSocket.connect();

            Intent intent = new Intent();
            intent.setClass(MainActivity.this, WorkActivity.class);
            startActivity(intent);

```



```

    }
    catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

}

}

class ClickEvent implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        if (v == btnSearch) // 搜索藍牙設備，在 BroadcastReceiver 顯示結果
        {
            if (btAdapt.getState() == BluetoothAdapter.STATE_OFF)
            { // 如果藍牙還沒開啟
                Toast.makeText(MainActivity.this, "請先打開藍牙",
                    Toast.LENGTH_LONG).show();
                return;
            }
            setTitle("本機藍牙地址：" + btAdapt.getAddress());
            lstDevices.clear();
            if (btAdapt.isDiscovering()) {
                btAdapt.cancelDiscovery();
            }
            else {
                Toast.makeText(getApplicationContext(), "藍芽搜尋中 .....", Toast.LENGTH_SHORT).show();
                btAdapt.startDiscovery();
            }
        }
        else if (v == btnSwitch) { // 本機藍牙啟動/關閉
            if (btnSwitch.isChecked() == false)
                btAdapt.enable();
            else if (btnSwitch.isChecked() == true)
                btAdapt.disable();
        }
    }
}

```

```
    } else if (v == btnDis)// 本機可以被搜索
    {
        Intent discoverableIntent = new Intent(

BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
        discoverableIntent.putExtra(

BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300);
        startActivity(discoverableIntent);
    } else if (v == btnExit) {

        try {
            if (btSocket != null)
                btSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        MainActivity.this.finish();
    }
}
}
```

附-3-6 Android Studio 主要動作

```
package com.example.henry.bookflip;

import android.opengl.GLSurfaceView;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.Manifest;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.pm.ActivityInfo;
import android.os.Message;
import android.os.Handler;
import android.util.Log;
import android.view.SurfaceHolder;
```

```
import android.view.SurfaceView;
import android.view.View;
import android.view.ViewDebug;
import android.widget.CheckBox;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Context;
import android.graphics.ImageFormat;
import android.graphics.SurfaceTexture;
import android.content.pm.PackageManager;
import android.hardware.camera2.CameraCaptureSession;
import android.hardware.camera2.CameraCharacteristics;
import android.hardware.camera2.CameraDevice;
import android.hardware.camera2.CameraManager;
import android.hardware.camera2.CameraMetadata;
import android.hardware.camera2.CaptureRequest;
import android.hardware.camera2.TotalCaptureResult;
import android.hardware.camera2.params.StreamConfigurationMap;
import android.media.Image;
import android.media.ImageReader;
import android.os.Environment;
import android.os.Handler;
import android.os.HandlerThread;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.util.Size;
import android.util.SparseIntArray;
import android.view.Menu;
import android.view.MenuItem;
```

```

import android.view.Surface;
import android.view.TextureView;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import java.io.File;
import java.io.FileOutputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.logging.LogRecord;
import android.view.Menu;
import android.view.MenuItem;

//AppCompatActivity Activity
public class WorkActivity extends AppCompatActivity{
    TextView msgText,txtNumber;
    Button btnSend,btnCancel,btnNow;
    ScrollView scrollView;
    CheckBox check3D;

    static final String tag="BT_SPP";
    InputStream btIn=null; // 藍芽讀入資料
    OutputStream btOut; // 藍芽輸出資料
    private String msg = "";
    boolean sppConnected=false;
    SppServer sppServer;

    // 翻數執行緒使用;
    GoFlip goFlip; // 進行翻書與拍照動作
    boolean goFlip_working=false;

```

```

int page,workpg=1; // 翻書總頁數、正翻印頁數
boolean firstpage = true; // 是否為第一頁?(不翻、只拍)
boolean FlipWay=true; // 翻書方向，true 正翻、false 反翻
boolean FlipWait=true; // 等待翻書
String FlipRetCmd;
// #####

private Size previewsize;
private Size jpegSizes[]=null;

private TextureView textureView;
private CameraDevice cameraDevice;
private CaptureRequest.Builder previewBuilder;
private CameraCaptureSession previewSession;

private static final SparseIntArray ORIENTATIONS=new
SparseIntArray();
static
{
    ORIENTATIONS.append(Surface.ROTATION_0,90);
    ORIENTATIONS.append(Surface.ROTATION_90,0);
    ORIENTATIONS.append(Surface.ROTATION_180,270);
    ORIENTATIONS.append(Surface.ROTATION_270,180);
}
// #####

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_work);

    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT)
;

    msgText = (TextView)findViewById(R.id.msgText); // 動作狀態
    check3D = (CheckBox)findViewById(R.id.check3D); // 正、反翻
    txtNumber = (TextView)findViewById(R.id.txtNumber);
    btnSend=(Button)findViewById(R.id.btnSend);
    btnSend.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

```

```

        onbtnSendClick(v);
    }
});
btnCancel=(Button)findViewById(R.id.btnCancel);
btnCancel.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        onbtnCancelClick(v);
    }
});

sppServer = new SppServer();
sppServer.start();
// #####
textureView=(TextureView)findViewById(R.id.textureview);
textureView.setSurfaceTextureListener(surfaceTextureListener);

btnNow=(Button)findViewById(R.id.btnNow); // 直接拍照
btnNow.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ShowMessage("[拍照]","直接翻拍");
        getPicture();
    }
});
openCamera();
// #####
msgText.setText("準備動作完成!");
}

@Override
public void onDestroy() {
    super.onDestroy();

    try {
        MainActivity.btSocket.close();
    } catch(IOException e) {
        e.printStackTrace();
    }
}

```

```

    }
}

// 執行緒需要透過 Handler 才能送出訊息
private Handler btHandler =new Handler(){
    public void handleMessage(Message m) {
        FlipRetCmd=msg;
        Log.d("Handler", FlipRetCmd);
        switch(m.what) {
            case 1:msgText.setText(msg);break;
            default:msgText.setText(msg);break;
        }
    }
};

// =====
// 開始翻印動作
public void onbtnSendClick(View v) {
    if(!goFlip_working) {
        firstpage = true;
        FlipWait = true; // 預設等待翻書，直到翻拍完或 Cancel
        page = Integer.parseInt(txtNumber.getText().toString()); // 取得翻拍頁數

        workpg = 1;
        if (check3D.isChecked()) FlipWay = false;
        else FlipWay = true;
        FlipRetCmd = "";
        goFlip_working = true;
        goFlip = new GoFlip();
        goFlip.start();
    } else {
        Toast.makeText(WorkActivity.this, "【翻拍】 正在翻拍第
"+ workpg + "/" + page + " 頁 .....", Toast.LENGTH_SHORT).show();
    }
}

private class GoFlip extends Thread {
    public GoFlip() {

```

```

    }

    // 注意：執行緒終須透過 Handler' 才能送出訊息
    public void run() {
        synchronized (WorkActivity.this) {
            while (FlipWait && workpg <= page &&
FlipRetCmd.compareTo("[E]") != 0) {
                msg= "【翻拍】 正在翻拍第 " + workpg + " / "
+ page + " 頁 .....";
                btHandler.sendMessage(0);
                if (firstpage) {
                    btCommand("P"); // 第一頁壓
書、不翻書

                    firstpage = false;
                } else {
                    // 翻書後、壓
                    if (FlipWay) btCommand("F"); // 正翻
                    else btCommand("B"); // 反翻
                }

                // 等待翻書或壓書動作完成
                FlipRetCmd = "";
                while (FlipWait &&
FlipRetCmd.compareTo("[K]") != 0 && FlipRetCmd.compareTo("[E]") !=
0) {
                    // 等待翻書完成，畫面可 K 以顯示狀態
                    Log.d("Handler", FlipRetCmd);
                }
                // 拍書
                if (FlipRetCmd.compareTo("[K]") == 0) {
                    getPicture();
                    workpg++;
                } else if (FlipRetCmd.compareTo("[E]") == 0) {
                }
            } // 全部頁數翻拍完畢或發生錯誤
            if (FlipWait) {
                btCommand("Z");
                if (FlipRetCmd.compareTo("[K]") == 0) {

```



```

        msg= "【翻拍】 完成翻拍工作，共 "+ page
+ " 頁";

        btHandler.sendMessage(0);
    } else {
        msg= "【中斷】 發生錯誤，中斷工作。";
        btHandler.sendMessage(0);
    }
}
}
goFlip_working=false; // 表示翻拍工作結束
}

}; // End of class GoFilp

// =====
// 取消翻印動作
public void onbtnCancelClick(View v) {
    ShowMessage("【翻拍】","取消翻拍工作!");
    FlipWait=false; // 不再等待翻書動作，取消翻書複製
}
// =====

public void oncheck3DClick(View v) {
    if(check3D.isChecked()) {
        FlipWay=false;
    } else {
        FlipWay=true;
    }
}

@Override
protected void onResume() {
    super.onResume();
}

@Override
protected void onPause() {
    super.onPause();
}
// #####

```

```

        if(cameraDevice!=null)
        {
            cameraDevice.close();
        }

// #####

    }

    // 顯示在畫面下方的訊息
    private void ShowMessage(String header,String msg) {
        msgText.setText(header + msg);
    }

    // 透過藍芽送出命令
    private void btCommand(String cmd)
    {
        try {
            btOut.write((cmd).toString().getBytes());
        } catch(IOException e) {
            e.printStackTrace();
        }
    }

    private class SppServer extends Thread {
        public SppServer() {
        }
        public void run() {
            if(MainActivity.btSocket==null) {
                Log.d(tag,"ServerSocket null");
                return;
            }
            try {
                synchronized (WorkActivity.this) {
                    if(sppConnected) return;
                    Log.i(tag, "Device Name: " +
MainActivity.btSocket.getRemoteDevice().getName());
                    btIn=MainActivity.btSocket.getInputStream();
                    btOut = MainActivity.btSocket.getOutputStream();
                    connected();
                }
            }
        }
    }

```

```

        }
    } catch(IOException e) {
        e.printStackTrace();
        Log.e(tag,"serverSocket accept fail!");
    }
    Log.i(tag,"End Bluetooth SPP Server!");
}

public void cancel() {
    if(MainActivity.btSocket ==null) return;
    try {
        MainActivity.btSocket.close();
    } catch(IOException e) {
        e.printStackTrace();
        Log.e(tag,"ServerSocket failed");
    }
}

}

private void connected() {
    sppConnected=true;
    new SppReceiver(btIn).start();
    Log.e(tag,"connect");
}

private void disconnect() {
    sppConnected=false;
    btIn=null;
    btOut=null;
    sppServer= new SppServer();
    sppServer.start();
    Log.e(tag,"disconnect");
}

private class SppReceiver extends Thread {
    private InputStream input = null;

    public SppReceiver(InputStream in) {
        input = in;
    }
}

```

```

        Log.i(tag, "SppReceiver!");
    }

    public void run() {

        byte[] data = new byte[1024];
        int length = 0;
        if (input == null) {
            Log.e(tag, "Spp InputStream null");
            return;
        }

        while (true) {
            try {
                length = input.read(data);

                if (length > 0) {
                    msg = new String(data, 0, length, "ASCII");
                    FlipRetCmd="[" + msg + "];
                    Log.d(tag,FlipRetCmd );
                }
            } catch (IOException e) {
                Log.e(tag, "SPPReceiver Disconnect!");
                disconnect();
            }
        }

    }

    // =====
    // 相機功能程式碼
    void getPicture()
    {
        if(cameraDevice==null) return;
        CameraManager
manager=(CameraManager)getSystemService(Context.CAMERA_SERVICE);

        try

```

```

    {
        CameraCharacteristics
characteristics=manager.getCameraCharacteristics(cameraDevice.getId());
        if(characteristics!=null)
        {

jpegSizes=characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURATION_MAP).getOutputSizes(ImageFormat.JPEG);
        }
        int width=640,height=480;
        if(jpegSizes!=null && jpegSizes.length>0)
        {
            width=jpegSizes[0].getWidth();
            height=jpegSizes[0].getHeight();
        }
        ImageReader
reader=ImageReader.newInstance(width,height,ImageFormat.JPEG,1);
        List<Surface> outputSurfaces=new ArrayList<Surface>(2);
        outputSurfaces.add(reader.getSurface());
        outputSurfaces.add(new
Surface(textureView.getSurfaceTexture()));

        final CaptureRequest.Builder
capturebuilder=cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_STILL_CAPTURE);
        capturebuilder.addTarget(reader.getSurface());
        capturebuilder.set(CaptureRequest.CONTROL_MODE,
CameraMetadata.CONTROL_MODE_AUTO);

        int
rotation=getWindowManager().getDefaultDisplay().getRotation();

capturebuilder.set(CaptureRequest.JPEG_ORIENTATION,ORIENTATIONS.get(rotation));

        ImageReader.OnImageAvailableListener
imageAvailableListener=new ImageReader.OnImageAvailableListener() {
            @Override

```

```

public void onImageAvailable(ImageReader reader) {
    Image image = null;
    try {
        image = reader.acquireLatestImage();
        ByteBuffer buffer =
image.getPlanes()[0].getBuffer();
        byte[] bytes = new byte[buffer.capacity()];
        buffer.get(bytes);
        save(bytes);
    } catch (Exception ee) {

    }
    finally {
        if(image!=null)
            image.close();
    }
}
void save(byte[] bytes)
{
    File file12=getOutputMediaFile();
    OutputStream outputStream=null;
    try
    {
        outputStream=new FileOutputStream(file12);
        outputStream.write(bytes);

        Toast.makeText(WorkActivity.this, file12 + " 儲
存完畢!", Toast.LENGTH_LONG).show();
    } catch (Exception e)
    {
        e.printStackTrace();
    } finally {
        try {
            if (outputStream != null)
                outputStream.close();
        } catch (Exception e){ }
    }
}
}

```

```

};

        HandlerThread handlerThread=new
HandlerThread("takepicture");
        handlerThread.start();

        final Handler handler=new Handler(handlerThread.getLooper());

reader.setOnImageAvailableListener(imageAvailableListener,handler);

        final CameraCaptureSession.CaptureCallback
previewSSession=new CameraCaptureSession.CaptureCallback() {
            @Override
            public void onCaptureStarted(CameraCaptureSession
session, CaptureRequest request, long timestamp, long frameNumber) {
                super.onCaptureStarted(session, request, timestamp,
frameNumber);
            }

            @Override
            public void onCaptureCompleted(CameraCaptureSession
session, CaptureRequest request, TotalCaptureResult result) {
                super.onCaptureCompleted(session, request, result);
                startCamera();
            }
        };

        cameraDevice.createCaptureSession(outputSurfaces, new
CameraCaptureSession.StateCallback() {
            @Override
            public void onConfigured(CameraCaptureSession session) {

                try
                {

session.capture(capturebuilder.build(),previewSSession,handler);

                }catch (Exception e)

```

```

        {

        }

    }

    @Override
    public void onConfigureFailed(CameraCaptureSession
session) {

        }

    },handler);
}
catch (Exception e)
{

}
}

public void openCamera()
{
    CameraManager
manager=(CameraManager)getSystemService(Context.CAMERA_SERVICE);
    try
    {
        String camerId=manager.getCameraIdList()[0];
        CameraCharacteristics
characteristics=manager.getCameraCharacteristics(camerId);
        StreamConfigurationMap
map=characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGU
RATION_MAP);
        previewsize=map.getOutputSizes(SurfaceTexture.class)[0];
        int permissionCheck2 =
this.checkSelfPermission("Manifest.permission.CAMERA");
        if (permissionCheck2 !=
PackageManager.PERMISSION_GRANTED) {
            this.requestPermissions(new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION,
android.Manifest.permission.ACCESS_COARSE_LOCATION}, 1001); //Any

```


number

```
        }
        manager.openCamera(camerId,stateCallback,null);
    } catch (Exception e)
    {

    }
}

private TextureView.SurfaceTextureListener surfaceTextureListener=new
TextureView.SurfaceTextureListener() {
    @Override
    public void onSurfaceTextureAvailable(SurfaceTexture surface, int
width, int height) {
        openCamera();
    }

    @Override
    public void onSurfaceTextureSizeChanged(SurfaceTexture surface,
int width, int height) {

    }

    @Override
    public boolean onSurfaceTextureDestroyed(SurfaceTexture surface) {
        return false;
    }

    @Override
    public void onSurfaceTextureUpdated(SurfaceTexture surface) {

    }
};

private CameraDevice.StateCallback stateCallback=new
CameraDevice.StateCallback() {
    @Override
    public void onOpened(CameraDevice camera) {
        cameraDevice=camera;
        startCamera();
    }
};
```

```

    }

    @Override
    public void onDisconnected(CameraDevice camera) {

    }

    @Override
    public void onError(CameraDevice camera, int error) {

    }
};

void startCamera()
{
    if(cameraDevice==null||!textureView.isAvailable()||
previewsize==null) return;

    SurfaceTexture texture=textureView.getSurfaceTexture();
    if(texture==null)
    {
        return;
    }

    texture.setDefaultBufferSize(previewsize.getWidth(),previewsize.getHeight());
    Surface surface=new Surface(texture);

    try
    {

    previewBuilder=cameraDevice.createCaptureRequest(CameraDevice.TEMPLA
TE_PREVIEW);
        }catch (Exception e)
        {
        }
    previewBuilder.addTarget(surface);

```

```

        try
        {
            cameraDevice.createCaptureSession(Arrays.asList(surface), new
CameraCaptureSession.StateCallback() {
                @Override
                public void onConfigured(CameraCaptureSession session) {
                    previewSession=session;
                    getChangedPreview();
                }

                @Override
                public void onConfigureFailed(CameraCaptureSession
session) {

                    }
                },null);
            } catch (Exception e)
            {

            }
        }
        void getChangedPreview()
        {
            if(cameraDevice==null)
            {
                return;
            }
            previewBuilder.set(CaptureRequest.CONTROL_MODE,
CameraMetadata.CONTROL_MODE_AUTO);
            HandlerThread thread=new HandlerThread("changed Preview");
            thread.start();
            Handler handler=new Handler(thread.getLooper());
            try
            {
                previewSession.setRepeatingRequest(previewBuilder.build(),
null, handler);
            } catch (Exception e){ }
        }
    }

```

```

private static File getOutputMediaFile() {
    File mediaStorageDir = new File(
        Environment
            .getExternalStorageDirectory(),
        "MyCameraApp");
    if (!mediaStorageDir.exists()) {
        if (!mediaStorageDir.mkdirs()) {
            Log.d("MyCameraApp", "failed to create directory");
            return null;
        }
    }
    // Create a media file name
    String timeStamp = new
SimpleDateFormat("yyyyMMdd_HH:mm:ss")
        .format(new Date());
    File mediaFile;
    mediaFile = new File(mediaStorageDir.getPath() + File.separator
        + "IMG_" + timeStamp + ".jpg");

    return mediaFile;
}

// =====
}

```

成員簡歷

姓名	羅昱婷	班級	107 電子三乙	
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. Visual Basic 程式設計 5. 電腦輔助設計繪圖 6. Arduino 程式設計 7. 微處理機與實習 8. 電子電路與實習 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 資料蒐集 2. 理論探討 3. 擬定計畫書 4. 學習軟體基本架構 5. 依所需撰寫程式 6. 撰寫手機程式 7. 撰寫報告 8. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 擔任 104 學年度輔導小老師 2. 榮獲 105 學年度班級優良學生 3. 擔任 105 學年度化學小老師 4. 擔任 105 學年度班級幹部—班長 5. 擔任 105 學年度工場幹部—安全衛生 6. 擔任 106 學年度微處理機小老師 			
姓名	林妤婕	班級	107 電子三乙	
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. Visual Basic 程式設計 5. 電腦輔助設計繪圖 6. Arduino 程式設計 7. 微處理機與實習 8. 電子電路與實習 			

參與專題 工作項目	<ol style="list-style-type: none"> 1. 資料蒐集 2. 理論探討 3. 擬定計畫書 4. 學習軟體基本架構 5. 依所需撰寫程式 6. 影片拍攝 7. 口頭報告 		
經歷簡介	<ol style="list-style-type: none"> 1. 擔任 104 學年度英文小老師 2. 擔任 104 學年度工場幹部—衛生安全 3. 擔任 105 學年度數學小老師 4. 擔任 105 學年度班級幹部—輔導股長 5. 擔任 105 學年度社團幹部—吉他社社長 6. 設計班級網站及網頁規劃與管理 		
姓 名	王子熙	班 級	107 電子三乙
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. Visual Basic 程式設計 5. 電腦輔助設計繪圖 6. Arduino 程式設計 7. 微處理機與實習 8. 電子電路與實習 		
參與專題 工作項目	<ol style="list-style-type: none"> 1. 資料蒐集 2. 理論探討 3. 擬定計畫書 4. 硬體規劃 5. 設計機構 6. 機構製作 7. 硬體測試 8. 整合測試 9. 撰寫報告 10. 口頭報告 		
經歷簡介	<ol style="list-style-type: none"> 1. 擔任 104 學年度全民國防小老師 2. 擔任 104 學年度班級幹部—體育股長 3. 擔任 104 學年度工場幹部—工具管理 4. 擔任 105 學年度工場幹部—領班 		

姓名	蘇少睿	班級	107 電子三乙	
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. Visual Basic 程式設計 5. 電腦輔助設計繪圖 6. Arduino 程式設計 7. 微處理機與實習 8. 電子電路與實習 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 資料蒐集 2. 理論探討 3. 擬定計畫書 4. 硬體規劃 5. 設計機構 6. 機構製作 7. 硬體測試 8. 整合測試 9. 撰寫手機程式 10. 撰寫報告 11. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 榮獲 104 學年度跆拳道教育盃第三名 2. 擔任 105 學年度班級幹部—環保股長 3. 擔任 106 學年度班級幹部—圖書股長 4. 擔任 106 學年度工場幹部—衛生安全 5. 擔任 106 學年度國文小老師 			