

吳鳳科技大學

光機電暨材料研究所

碩士學位論文

Android 系統車牌辨識之研製
Design of License-Plate Recognition
System for Android

研 究 生：傅建誠

指導教授：何漢彰 博士

共同指導教授：蕭紋旭 博士

中 華 民 國 101 年 07 月

吳鳳科技大學
研究所碩士班
論文口試委員會審定書

本校 光機電暨材料研究所 傅建誠 君

所提論文 Android 系統車牌辨識之研製

Design of License-Plate Recognition Systems for Android

合於碩士資格水準、業經本委員會評審認可。

口試委員: 劉榮佑 何漢勳
蕭紋旭 _____

指導教授: 何漢勳 _____

所長: 張如華

中華民國 101 年 7 月 8 日

吳鳳科技大學光機電暨材料研究所

碩士論文著作財產權同意書

立書人： 傅建誠 之碩士畢業論文

中文題目：Android 系統車牌辨識之研製

英文題目：Design of License-Plate Recognition System for Android

指導教授：何漢彰 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

☒ 共同享有著作權

☐ 共同享有著作權，學生願「拋棄」著作財產權

☐ 學生獨自享有著作財產權

學 生： 傅建誠 (請親自簽名)

指導老師： 何漢彰 (請親自簽名)

中 華 民 國 101 年 7 月 23 日

中文摘要

隨著智慧型手機與平板電腦的發展，移動裝置的應用與發展日趨多元化，移動裝置具有攜帶方便之特點，本研究是以Android 作業系統作為開發平台，利用智慧型手機連接網路的便利性，透過校園裡的無線網路的建置(Wi-Fi)來連結至另一端的WeOCR系統來辨識車牌文字，並定時的更新手機內之資料庫資料，以確保資料為最新之資料。在系統運作流程上，是先透過手機上鏡頭拍攝車牌畫面，影像在透過手機本身來裁切，將裁切過後影像透過無線網路傳輸至WeOCR上做辨識，辨識完後把結果在回傳至手機系統上，最後透過資料庫的查詢，將查詢完畢後回傳至手機上顯示該車牌車主。

透過本研究之成果可減少校方檢查是否購買通行證的檢查時間與準確度，透過本系統可達到既快速與準確度查詢以及攜帶方便等特點。

關鍵詞：Android系統、校園無線網路、車牌辨識、資料庫、WeOCR

Abstract

The application and the development of mobile devices have become more diversified with the development of smart phones and tablet PCs. Mobile device has the characteristics of easy to carry. This study of the development platform is based on the Android operating system. Through the convenience of using the smart phones to connect the network, we can build the campus of Wi-Fi wireless network to connect to the WeOCR system to recognize the license plate text. In addition, we will regularly update the database data within the mobile phone to ensure that the data for the latest information. In the system operation process, we first shot the license plate screen through on the phone lens. And then the image is cropped through the smart phone itself. The cropped image would be transferred to WeOCR on recognition through the wireless network. The results will be back to the phone system after recognition. Finally we can query through the database and it will be back to the smart phone to display the license plate owners. The research results indicate that we can reduce the check time and accuracy whether the check purchases pass or not. We can achieve both speed and accuracy of query, and easy to carry through the system.

Keywords: Android system, the campus wireless network, license plate recognition, database, WeOCR

誌謝

在讀研究時間要感謝很多人，首先我想感謝共同指導教授蕭紋旭老師，在這段時間研究與論文內容的指導、相關儀器的借助，再來感謝指導教授何漢彰老師，在這當中研究的叮嚀注意事項與論文內容指導，在這期間內認識不少其他學校老師與在其他領域上有成就的人，再來感謝本校資訊工程系主任與系上助理在相關書籍與儀器借閱，再來感謝所上所有老師課堂在知識的教導與在書報討論中或是課台上另外請來的學者與專家們，讓我知道除了我所不知道知識以外還有以後在職場會面臨的問題外還有解決方式，以上各位老師學者與專家們都在百忙當中抽空來指導與教學。另外感謝的是所上的同學們，有大家的陪伴，在讀研究所當中也有許多樂趣。另外還有感謝學校各位老師與處室人員或系辦助理與所上助理與其他系上與所上助理，給我不少鼓勵與方便，讓我可以在这段期間內順利完成。

目錄

中文摘要	I
Abstract	II
誌謝	III
表目錄	V
圖目錄	VI
第一章 緒論	1
1.1 研究動機	1
1.2 研究目的	1
1.3 論文架構	2
第二章 相關研究	3
2.1 Android 介紹	3
2.2 Android 手機與開發資源	3
2.3 App Inventor 技術	8
2.4 雲端運算服務	13
2.5 辨識之技術	14
第三章 系統設計	17
3.1 系統流程	17
3.2 影像之處理	20
3.3 資料庫之查詢	22
第四章 系統實作	23
4.1 建立 App Inventor 開發環境	23
4.2 App Inventor 開發環境介紹	24
4.2.1 系統展示之規劃	24
4.2.2 系統的程式設計與規畫	26
第五章 實驗與功能驗證	28
5.1 測試樣本環境	28
5.2 結果	28
第六章 結論與未來展望	31
參考文獻	33
附錄	36
A. 系統安裝與操作說明	36
B. 系統測試	49
C. 資料庫程式	56

表目錄

表 1.1：Android 與 iSO 比較.....	2
表 2.1： HTC Wildfire 規格表	7
表 3.1：WeOCR 私服器規格.....	21
表 4.1：使用區塊	24
表 5.1：以時間進行比例項目	28

圖目錄

圖 1 Android 架構(此圖來源：Linuxconfig.org 的 Android system architecture).....	5
圖 2 HTC Wildfire 手機圖	7
圖 3 App Inventor 介面	8
圖 4 App Inventor 介面	9
圖 5 Camera 元件	9
圖 6 啟動 Android 裝置上的相機並進行拍照	10
圖 7 AfterPicture(text image) 事件	10
圖 8 Imagepicker 元件	10
圖 9 使用者所選擇的圖片，以字串回傳該圖片的位置	11
圖 10 元件高度 (y 軸像素)	11
圖 11 元件寬度 (x 軸像素)	11
圖 12 使用者點選 ImagePicker 中某項目完成後呼叫本事件	11
圖 13 當指頭移到 ImagePicker 之上，代表現在可以點選本元件時呼叫本事件 ...	12
圖 14 使用者點選 ImagePicker，但還沒點選某項目時呼叫本事件	12
圖 15 當指頭移出 ImagePicker，代表不能點選本元件時呼叫本事件。	12
圖 16 開啟 Imagepicker，效果同使用者親自點選一樣	12
圖 17 架構邏輯圖(來源:維基百科).....	13
圖 18 系統之架構圖	17
圖 19 系統之流程圖	19
圖 20 OCR 流程圖	20
圖 21 車牌搜尋方式	22
圖 22 搜尋結果	22
圖 23 App inventor 登入方式	24
圖 24 手機面板	25
圖 25 內部程式設計	26
圖 26 連接相機	26
圖 27 連接至網路上	27
圖 28 圖片顯示	27
圖 29 辨識中畫面	29
圖 30 申請 Gmail	36
圖 31 申請 Gmail	37
圖 32 登入 App inventor	37
圖 33 java 安裝	38
圖 34 java 測試是否有安裝及版本	39

圖 35 下載完成後	39
圖 36 執行安裝	40
圖 37 點選”Next>”，下一步安裝	40
圖 38 安裝條約	41
圖 39 確認安裝路徑	41
圖 40 相關目錄位置	42
圖 41 安裝開始	42
圖 42 AppInventor Setup 目錄	43
圖 43 App Inventor My Project	44
圖 44 專案匯入與匯出	44
圖 45 App Inventor Designer 作業	45
圖 46 開啟拼塊編輯	47
圖 47 打包應用程式選項	47
圖 48 QR Code 方式呈現	48
圖 49 設計之畫面	50
圖 50 設計之元件	51
圖 51 模擬器開機	52
圖 52 為載入前	52
圖 53 載入中	53
圖 54 載入完成	53
圖 55 點選拍照	54
圖 56 拍照中但無影像支援	54
圖 57 用手機實際拍攝	55
圖 58 拍攝顯是點選 WeOCR	55
圖 59 辨識結果	56

第一章 緒論

1.1 研究動機

近幾年由於台灣人車輛之普及化與全球石油上漲開車人明顯變少，再加上汽機車排放廢氣之問題，影響生活品質與加數了全球暖化，所以機車成為國人第一首選的交通工具，根據統計100年1月-11月新汽機車掛牌量34.7萬輛[1]，導致校園裡機車停放車輛增加，在加上網路環境的盛行促使下，使用無線網路上網人數狂升[2]，因電信業者也推出上網吃到飽的方案，使得越來越多人使用可以上網的手機來上網[3]，以台灣人來說有許多人使用手機是Android的 HTC或是Apple的iPhone，甚至也有其他家的手機，所以電信業者也紛紛推出了很多搭配方案其中最多以平板電腦方案，造成了路上很多人手1隻手機或是人有1台平板電腦，邊走邊聊天或是邊走邊打卡，打卡是指臉書，造成資訊技術的發展越來越快，也有雲端這個名詞出現，所謂雲端是指網路上另外許多台大型伺服器提供服務，但這些大型伺服器是看不到也找不到的，接合起來形狀像是雲那樣，所以稱雲端運算服務。以雲端服務來說會讓人聯想Google，所以Android開發是以Google App Inventor 設計工具，來建置車牌辨識系統，為了方便校方人員查詢，我們以市面上最多人創作開發Android 系統的HTC 來做為開發之環境，透過目前盛行的無線網路來達到快速查詢與管理機制。

1.2 研究目的

總結我研究動機敘述，本研究目的提出Android車牌辨識系統，使用目前攜帶方面HTC Android phone 作為平台。為什麼會選擇HTC Android phone而不用iPhone(iSO)? 是因為Android核心為Linux再加上開發使受限制比較少及開發時不受作業系統的限制，所以選擇HTC Android phone作為平台，為了比較一下Android

與iSO所以其比較如下表 1.1。

表 1.1：Android 與 iSO 比較

	Android	iSO
開發公司	Google	蘋果(Apple)
系統核心	Linux	Mac Os
系統程式開發	受限制比較少	受限制比較多
支援設備	HTC、三星、宏碁、摩托摩托、華碩等相關產品	iPhone、iPad 相關產品
開發環境	Windows	Mac
開發語言	Java	Objective-C 或 C 語言
Apps 附檔	.apk	.ipa
原開發工具	Google Android SDK、 APP Inventor	Objective-C
上傳與下載審查	Android Market 沒有	iPhone SDK 有

每間學校都規劃有U化的架構，我們想要利用U化的架構來建置車牌辨識系統。透過此架構與Android手機的結合下，產生出許許多多之應用，我們為了充份使用學校無線網路資源，隨時隨地準確調查我們想要的車牌訊息與是否違規，藉此應用在Android任何系統裡達到攜帶方便，可提供校園人員之使用，達到校園無線網路查詢車牌之快速與減少時間之需求。

1.3 論文架構

本論文共分六章。第二章介紹本論文之相關研究主要是介紹Android系統與雲端技術之相關技術說明。第三章介紹系統之架構設計將系統運作流程作詳細的說明，包括:系統環境、系統架構以及系統流程圖作詳細的說明與介紹。第四章介紹系統之實作，此部分的實作室說明系統架構如何在APP Inventor 環境下如何開發與建置。第五章為功能驗證及實驗結果討論，除驗證系統之結果外並加入實測結果，分析在不同網路環境下，系統運作與處理之時間。第六章為結論及未來發展。

第二章 相關研究

2.1 Android 介紹

Android是以Linux為基礎的開放原始碼作業系統，主要用於移動設備，是由安迪·魯賓（Andy Rubin）開發[4]，最初功能是為了支援手機，美國Google在2005年收購下成立22個月的高科技公司Android，於2007年公布Android手機系統平台，同時也並拉攏多家製造商組成開放手機聯盟（Open Handset Alliance）開發改良，逐漸擴充功能到平板電腦及其他領域上，它採用了軟體堆層（software stack，又名以軟體疊層）的架構，主要分為三部分。低層以Linux核心工作為基礎，只提供基本功能，其他的應用軟體則由名公司自行開發，以java作為編寫程式的一部分。

Android在正式發行之前，最開始擁有兩個內部測試版本，並且以著名的機器人名稱來對其進行命名，它們分別是：阿童木（Android Beta），發條機器人（Android 1.0）。後來由於涉及到版權問題，Google將其命名規則變更為用甜點作為它們系統版本的代號的命名方法。甜點命名法開始於Android 1.5發布的時候。作為每個版本代表的甜點的尺寸越變越大，然後按照26個字母數序：紙杯蛋糕（Android 1.5 Cupcake），甜甜圈（Android 1.6 Donut），鬆餅（Android 2.0/2.1 Eclair），凍酸奶（Android 2.2 Froyo），薑餅（Android 2.3 Gingerbread），蜂巢（Android 3.0/3.1 Honeycomb），冰激凌三明治（Android 4.0 Ice Cream Sandwich），根據最新消息新一代Android版本將命名為果凍豆（Jelly Bean Jelly Bean）。

2.2 Android 手機與開發資源

以資訊家電為代表的網際網路時代，手機產品是不可或缺的產品，在市場佔有一席之地，以近年來智慧型手機的發展，造就2大龍頭HTC的Android與Apple的iPhone等，是目前市場上激烈的目標，也因為這樣導致Apple告HTC侵權[8]，以HTC為國產商品又因Google的Android作為系統，導致HTC本身軟體為公開授權，所以

許多開發者首選HTC手機。

本文採用Android做為平台，因Google買下Android公司後所推出之平台，Android透過開放原始碼的Linux做為基礎所建置平台，所以系統包含C語言的標準函數庫，但又因使用程式為Java語言在Dalvik VM虛擬機器執行，應用程式開發環境以Eclipse為主，加上 Android Development Tools (ADT) Plugin[5]此外Android功能介紹如下:[6]

- 1、 應用程式框架： 可重複使用和更換的元件。
- 2、 達爾維克虛擬機器針對移動設備進行了優化。
- 3、 基於開放原始程式碼 WebKit 引擎集成的瀏覽器。
- 4、 優化圖形由一個自訂的 2D 圖形庫；基於 OpenGL ES 1.0 規範(可選的硬體加速) 3D 圖形。
- 5、 SQLite 的結構化的資料儲存。
- 6、 媒體支援常見的音訊、視頻和仍然(MPEG4、H.264、MP3、 AAC、 AMR、 JPG、 PNG、 GIF)圖像格式。
- 7、 GSM 電話服務 (取決於硬體)。
- 8、 藍牙、 邊緣、 3G 和 WiFi (取決於硬體)。
- 9、 攝影機、 GPS、 指南針和加速度計(取決於硬體)。

Eclipse IDE 工具調適、 記憶體和性能評測和外掛程式豐富的開發環境，包括設備模擬程式。Android的系統架構圖(如圖 1，此圖來源:Linuxconfig.org的**Android system architecture**^[7])。

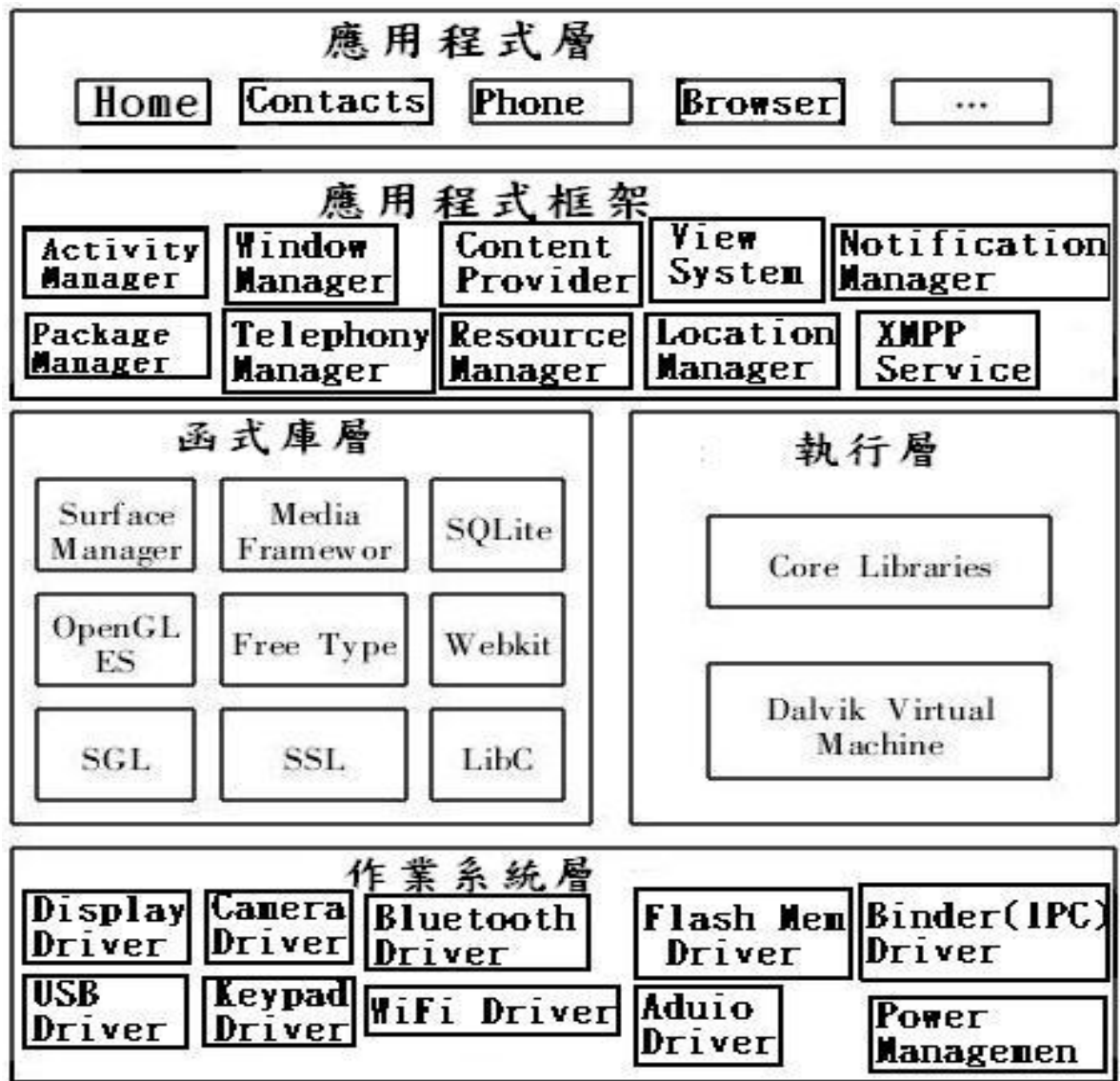


圖 1 Android 架構(此圖來源：Linuxconfig.org 的 Android system architecture)

Android架構(如圖 1)，可分為:作業系統層(Linux Kernel)、執行層(Android Runtime)、函式庫層(Libraries)、應用程式框架層(Application Framework)、應用程式層(Applications)^[8]。

1、 作業系統層(Linux Kernel)：

Android平台的作業系統用的是Linux，其核心版本為2.6版，Android所用的Linux核心，其包含的功效包括：安全(Security)、記憶體管理(Memory Management)、行程管理(Process Management)、網路堆疊(Network Stack，大陸方面稱為：網絡堆棧)、驅動程式模型(Driver Model)等，另外也在前述的軟體堆疊與硬體間建立起一

個抽象層(Abstraction Layer)。

2、執行層(Android Runtime)：

Android 雖然用Java程式語言來開發、撰寫應用程式，但卻不使用一般大家目前所用的Java Runtime(J2ME)版本來執行Java程式，而是用Android自有的Android Runtime來執行。Android Runtime包含下面兩個核心。

3、函式庫層(Libraries)：

Android所有豐富多樣的元件皆是由C/C++函式庫所組成，但Android所有的應用程式皆必須經由API來使用這些功能。

4、應用程式框架層(Application Framework)

提供了用來建立Android類別。

5、應用程式層(Applications)：

包含本機和第三方的所有應用程式。

系統實作部分我們以Android 手機 HTC Wildfire 作為實作，手機表 2.1：
HTC Wildfire規格表為Wildfire之規格，圖 2 HTC Wildfire手機。

表 2.1： HTC Wildfire 規格表

處理器	Qualcomm® MSM7225，528MHz
作業系統	Android 2.1 Eclair (目前可升級 Android 2.2 Froyo)
記憶體	唯讀記憶體 ROM: 512MB 隨機存取記憶體 RAM: 384MB
尺寸	106.75mm(長) x 60.4mm(寬) x 12.19 mm (厚)
重量	118 公克 (4.16 盎司)(含電池)
螢幕	3.2 吋電容式觸控感應式螢幕，解析度可達 240 x 320 QVGA
網路	HSDPA/WCDMA： 歐洲/亞洲：900/2100 MHz 上傳速率最高 384 kbps，下載速率最高 7.2 Mbps 四頻 GSM/GPRS/EDGE： 850/900/1800/1900 MHz (頻寬頻率及資料傳輸速度取決於電信業者。)
畫面導覽控制	光學感應器
GPS	內建 GPS / AGPS 天線
感應器	G-Sensor、數位羅盤、距離感應器、環境光源感應器
連線能力	可傳輸檔案的 Bluetooth® 2.1、支援 A2DP 無線立體聲耳機 Wi-Fi®：IEEE 802.11 b/g 3.5 mm 立體聲耳機插孔 標準 micro-USB (5 針腳 micro-USB 2.0)
相機	500 萬畫素相機，支援自動對焦和閃光燈
支援的音訊格式	播放：.aac、.amr、.ogg、.m4a、.mid、.mp3、.wav、.wma 錄音：.amr
支援的影片格式	播放：.3gp、.3g2、.mp4、.wmv 錄影：.3gp



圖 2 HTC Wildfire 手機圖

2.3 App Inventor 技術

App Inventor 是在去年底(2010年底)推出的，當然，這個 Project 在正式推出之前早已在許多高中、大學校園當中做了許多教學測試^[9]。可見得 Google 選擇發展這樣的開發路徑，瞄準的就是教育市場Google 在這個部分特別引入在教育系統當中頗為成功的 Scratch 來作為拼圖式程式寫作的基礎^[10]，實際上 Google App Inventor 負責教授即來自 MIT 與 Scratch 技術發展有相當深且廣的關聯，標榜可以讓使用者完全不需寫任何程式碼即可完成應用程式開發^[11]。驚訝的是，已經支援 NXT 樂高機器人程式開發，使得 Android 與 NXT 的結合應用，更添了許多想像空間^[12]以下圖片App Inventor做的本系統(圖 3、圖 4)。

App inventor不需要有高深的語言能力，在美國國小學生也會做的程式，現在台灣都以開發成樂高機器人程式^[13]，台灣也為此舉辦不少機器人比賽^[14]。

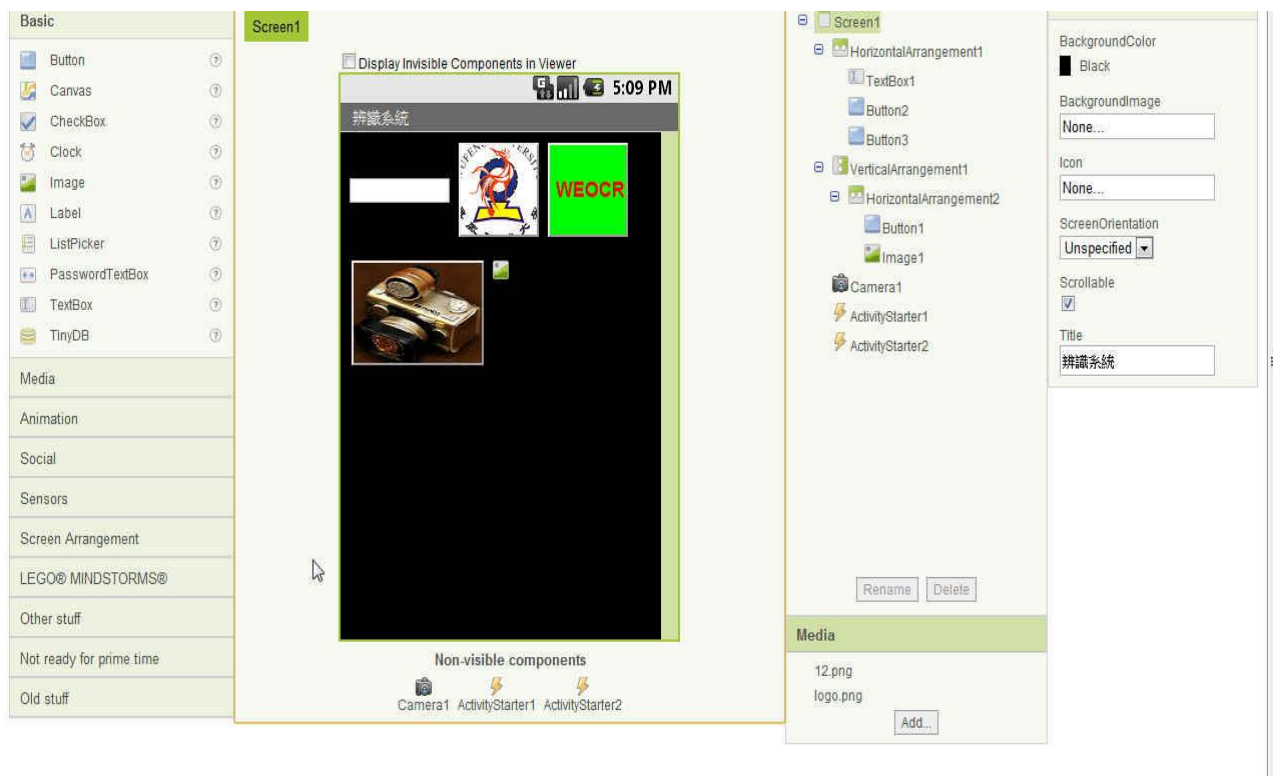


圖 3 App Inventor 介面

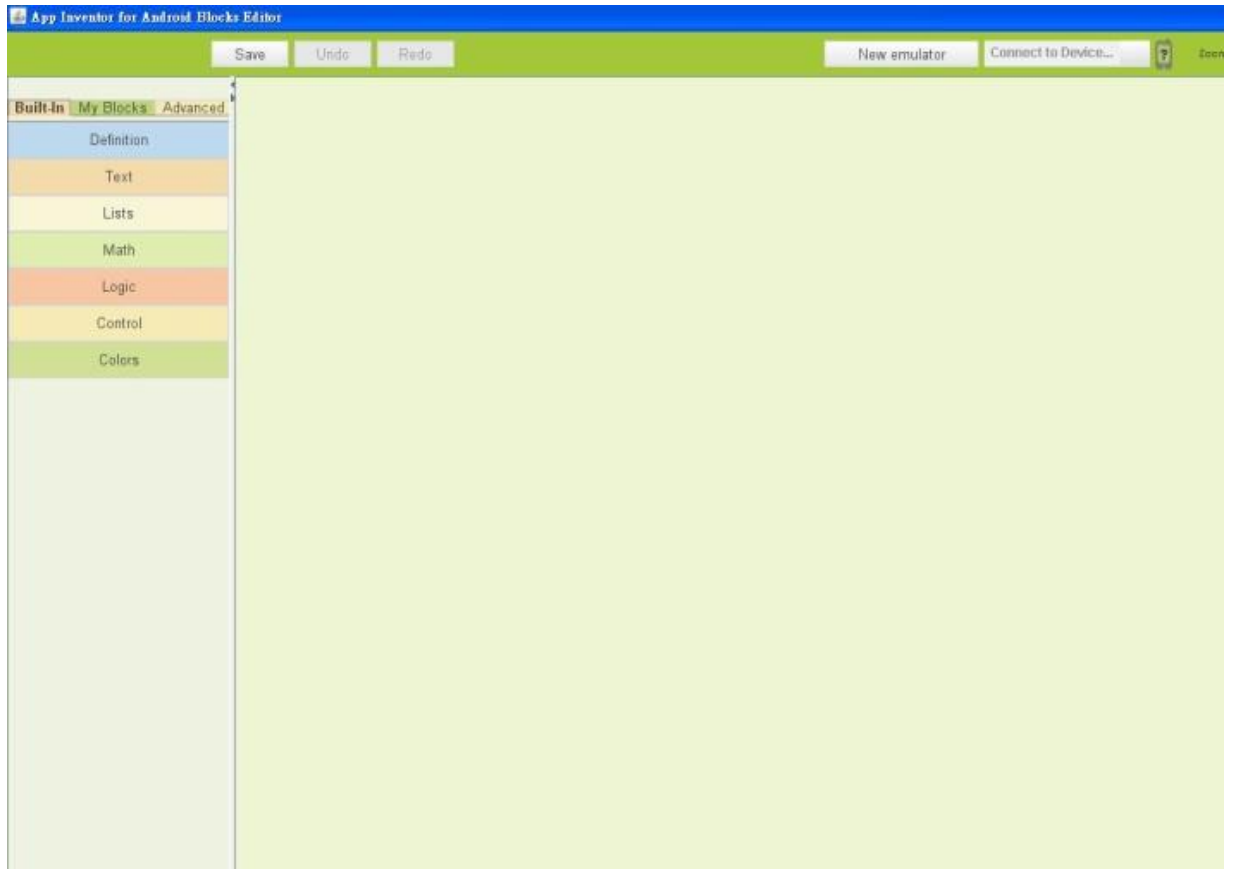


圖 4 App Inventor 介面

以App inventor相機物件來說明：^[15]

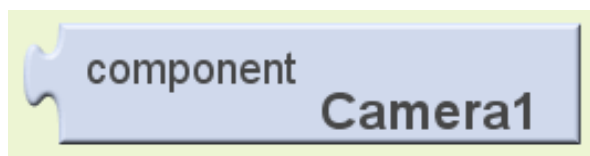


圖 5 Camera 元件

Camera元件(如圖 5 Camera元件)可呼叫Android裝置上的相機進行拍照。Camera元件為一非可視元件，它可呼叫Android裝置上的相機進行拍照。拍完照之後，您可從AfterPicture事件中的參數找到剛剛所拍照片的檔案位置。您可將這個檔案位置用於Image元件的Picture屬性，將這個Image的圖片指定為剛剛所拍的照片。

- 方法：TakePicture()

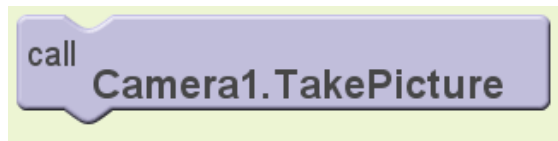


圖 6 啟動 Android 裝置上的相機並進行拍照

- 事件：AfterPicture(text image)

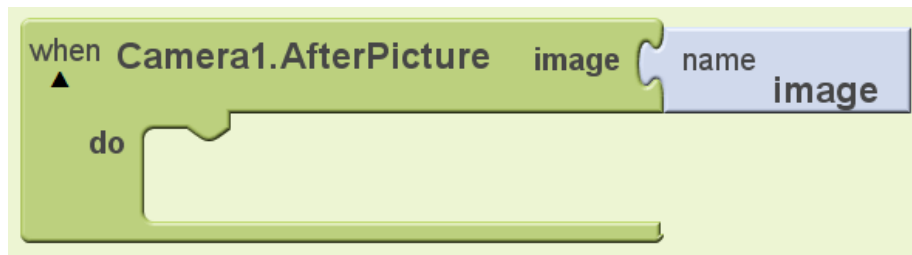


圖 7 AfterPicture(text image) 事件

圖 7 AfterPicture(text image) 事件為拍照完成後呼叫本事件，image這個字串參數代表剛剛所拍照片儲存於Android裝置中的位置，可用來呼叫這張照片。

- 屬性：Imagepicker圖片選取器

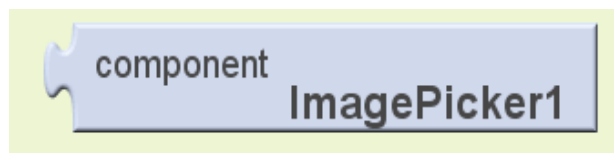


圖 8 Imagepicker 元件

圖 8 Imagepicker元件為可從您的圖片庫中選取圖片。Image picker是一種按鈕。當您點選它之後，會跳到Android裝置上的圖片庫，請接著選擇您所需要的圖片。當您選擇好圖片之後，Image Path這個屬性是用一個字串來代表該圖片的路徑。您可使用該參數來設定按鈕的背景圖片。

■ 屬性：ImagePath

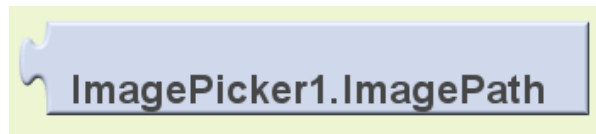


圖 9 使用者所選擇的圖片，以字串回傳該圖片的位置

■ Height



圖 10 元件高度 (y 軸像素)

■ Width

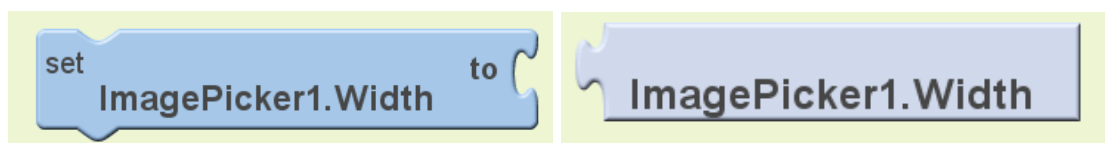


圖 11 元件寬度 (x 軸像素)

■ 事件：AfterPicking



圖 12 使用者點選 ImagePicker 中某項目完成後呼叫本事件

■ GotFocus

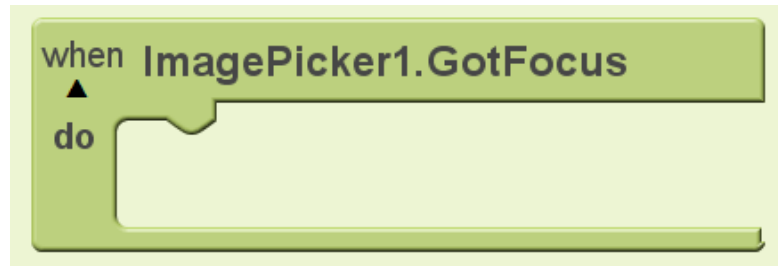


圖 13 當指頭移到 ImagePicker 之上，代表現在可以點選本元件時呼叫本事件

■ BeforePicking

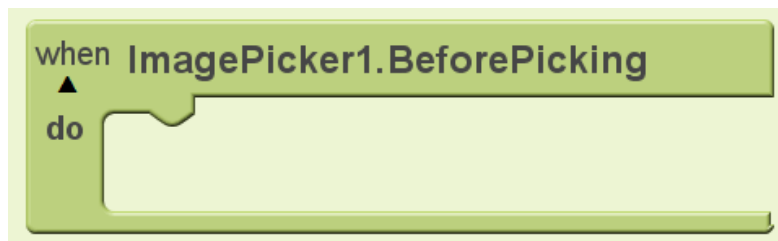


圖 14 使用者點選 ImagePicker，但還沒點選某項目時呼叫本事件

■ LostFocus

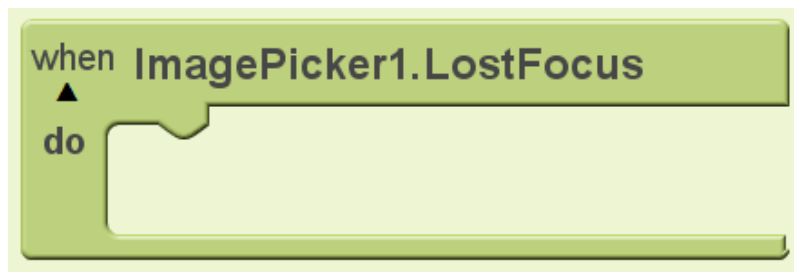


圖 15 當指頭移出 ImagePicker，代表不能點選本元件時呼叫本事件。

■ Open



圖 16 開啟 Imagepicker，效果同使用者親自點選一樣

2.4 雲端運算服務

根據維基百科整理資料，雲端運算發展最早是由1983年的昇陽電腦提出網路是電腦而在2006年3月亞馬遜推出彈性運算雲端同一年8月9日Google行政總裁Eric Schmidt在搜索引擎大會首次提出雲端計算，而後2007年10月Google與IBM開始在美國知名大學與工學院內推廣雲端運算計畫，此計畫降低分散運算技術在學術研究的成本，2008年1月30日Google宣布在台灣起動雲端運算學術計畫^[16]。當前雲端所提供服務分為三類：

- 1、上層軟體(Software as a Server)簡稱:Saas。
- 2、中層設備(Infrastructure as a Server)簡稱:Iaas。
- 3、下層平台(Plateform as a Server)簡稱:Paas。

以上三類，其架構邏輯圖(如圖 17架構邏輯圖(來源:維基百科))。

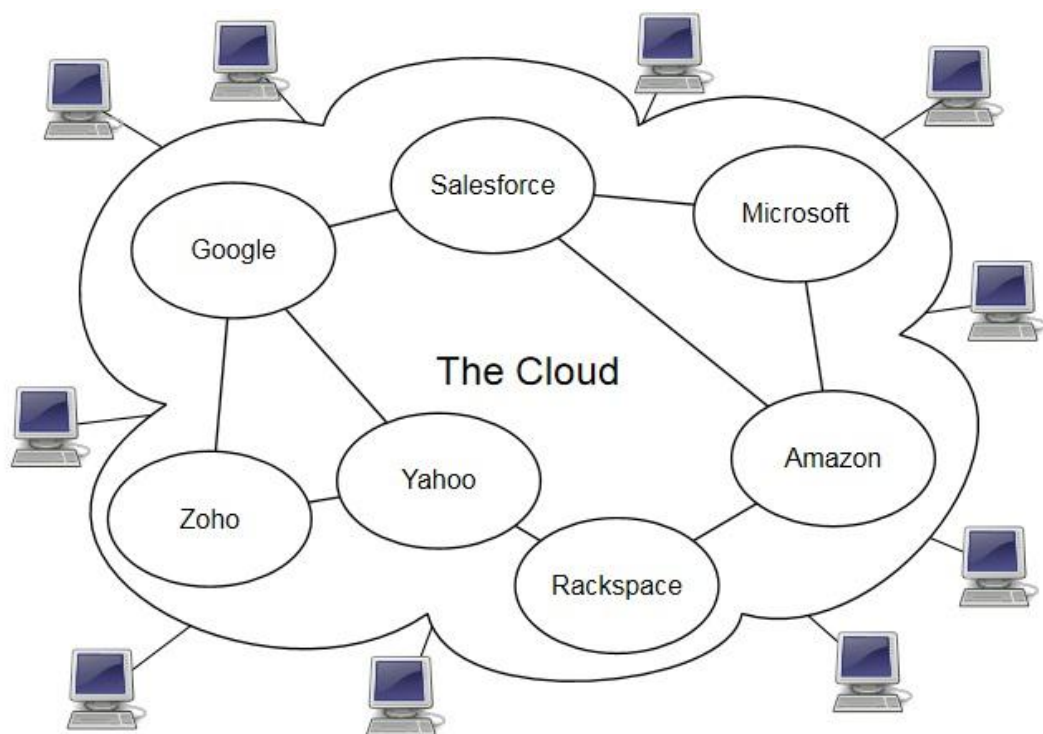


圖 17 架構邏輯圖(來源:維基百科)

此外，臺灣大學資訊工程系提出一篇對雲端看法雲端等於網路運算，雲端本身是不新的技術，雲端是結合電腦與網路間的合作關係^[6]。雲端成熟得三大關鍵是：

- 1、虛擬的技術的普及化。
- 2、連接網路速度加快。
- 3、硬體設備的處理數率。

另外，Openfind 產品行銷經理在2010的1月，提到雲端被是為繼Web2.0以後的重要商機，根據美國某證券公司估計，往後的五年內站市場上九百五十億美金^[7]。

2.5 辨識之技術

OCR的概念是在1929年由德國科學家Tausheck最先提出來的，OCR技術(Optical Character Recognition)，是通過掃描等光學輸入方式將各種文字轉化為圖像信息，再利用文字識別技術將圖像信息轉化為可以使用的文字的計算機輸入技術。近年來，隨著檔案實體數字化應用的發展，有一些檔案部門也引進了OCR技術，在數字檔案PDF文件中設立文本層，從而實現全文搜索功能。

1、目前 OCR 技術存在識別率不高的問題

OCR識別率一直是OCR技術上的一個薄弱環節，為研究OCR技術對數字化檔案的文字識別效果，我們對掃描後的各類檔案進行了OCR識別測試，並對結果進行分析，發現不同條件下的OCR識別率差異很大：

- (1). 清晰的打印字體：識別率在 90% 以上。
- (2). 夾雜表格的頁面：識別率在 80% 以上，主要是表格影響內部文字識別。
- (3). 頁面中較小或較大的字體：識別率在 30% 以下。
- (4). 存在塗改、汙漬等、深底色的頁面：識別率在 30% 以下。
- (5). 頁面中的非漢語語體和繁體字、公式、化學方程式：識別率非常低。

(6). 手寫字體、特殊字體、印章、圖紙和圖片：基本無法識別。

從上可以看出，OCR識別針對清晰文字效果最好，但也不可能達到100%的識別率。

2、OCR 技術在檔案數字化應用中有很多製約因素

(1). 最終數據的文本部分過於簡單

OCR識別的原理，是通過將含有文字的圖像按字切割成可獨立識別的單元，然後運用各種算法分析每個圖像單元中文字的形態特徵，通過比對標準特徵庫中的數據，判斷出該文字在計算機中的標準編碼，並按通用格式輸出保存在文本文件中，對非打印文字或不清晰的文字等內容一律隻能識別為亂碼。從檔案數字化工作來講，OCR識別出文字基本都存放在雙層PDF中的文本層，文字在該文本層是以無格式文字出現，除文字和標點符號外，各種文本格式、非文本符號、圖像等等都無法顯示，隻能作為關鍵詞(字)定位查詢用。

(2). 進行數字化的檔案種類太雜

從目前整理的情況看，進行檔案數字化的檔案中有公文、發文稿、草稿、文本、書本、圖紙、繳費單、證書、流程圖、報紙、營業執照、各類複印件、清單、證明、許可證、圖紙目錄、填寫表、報告單、會議紀要、介紹信、工作證、工作計劃、工作總結、員工名冊、試卷、成績單、幻燈片、協議書、勞動情況、合同書、體檢表、代碼證、登記表、登記證、營業執照、工作聯繫單、通知書、傳遞單、簽證書、清單、圖紙、許可證、審批表、收據、填寫表、發票、承諾書以及會計檔案等等各種類型，上面包含手寫、表格、複印、汙漬、印章、圖表、圖片、線條、繁體、字母以及特大特小號字符等等各種OCR識別不了的因素，每頁的識別率都在90%以下，甚至整頁無法識別，體現不出OCR的技術優勢。

(3). OCR 識別軟件之間存在不兼容性

在實際的操作過程中，我們碰到過一個情況，某個單位數字化的檔案文件，在用Adobe Acrobat察看識別的文本層時，某些頁面的文本層中文字顯示

為空白，但這些頁面在實施轉化的電腦上能顯示文字，經過分析，認為可能是該公司使用的OCR轉換軟件，與Adobe Acrobat存在一定的不兼容性，因此在轉換電腦中能正常顯示的某些頁面文字，到另一台電腦中就無法正常顯示，解決方法是統一識別與察看軟件，但不可能統一每個單位的使用軟件，會對使用單位的檔案利用造成一定的麻煩。

第三章 系統設計

本文提出以Android系統車牌辨識系統，主要利用於手機上原有的鏡頭拍照取得車牌相片後，透過3.5G或WiFi 方式傳送至網路上辨識系統WeOCR上做辨識，辨識完後進一步利用資料庫搜尋車牌知車主，其架構圖如錯誤! 找不到參照來源。[18]。



圖 18 系統之架構圖

3.1 系統流程

本系統流程分為三個部分圖19、第一部分為啟動拍照模式，拍照模式中要避免光感問題、曝光、閃光燈、距離、手晃、模糊等以上問題就可以增加辨識的成

功率。

第二部分是上傳至WeOCR辨識是將第一部份的照片結果上傳至WeOCR伺服器辨識，辨識結果回傳至手機上，此部分辨識結果有誤要回至第一部份，此章節3.2有說明OCR處理之流程。

第三部分是資料庫比對，是將第二部分結果連接至資料庫比對，比對完後就可以知道這車牌之車主是誰，是否有違規。

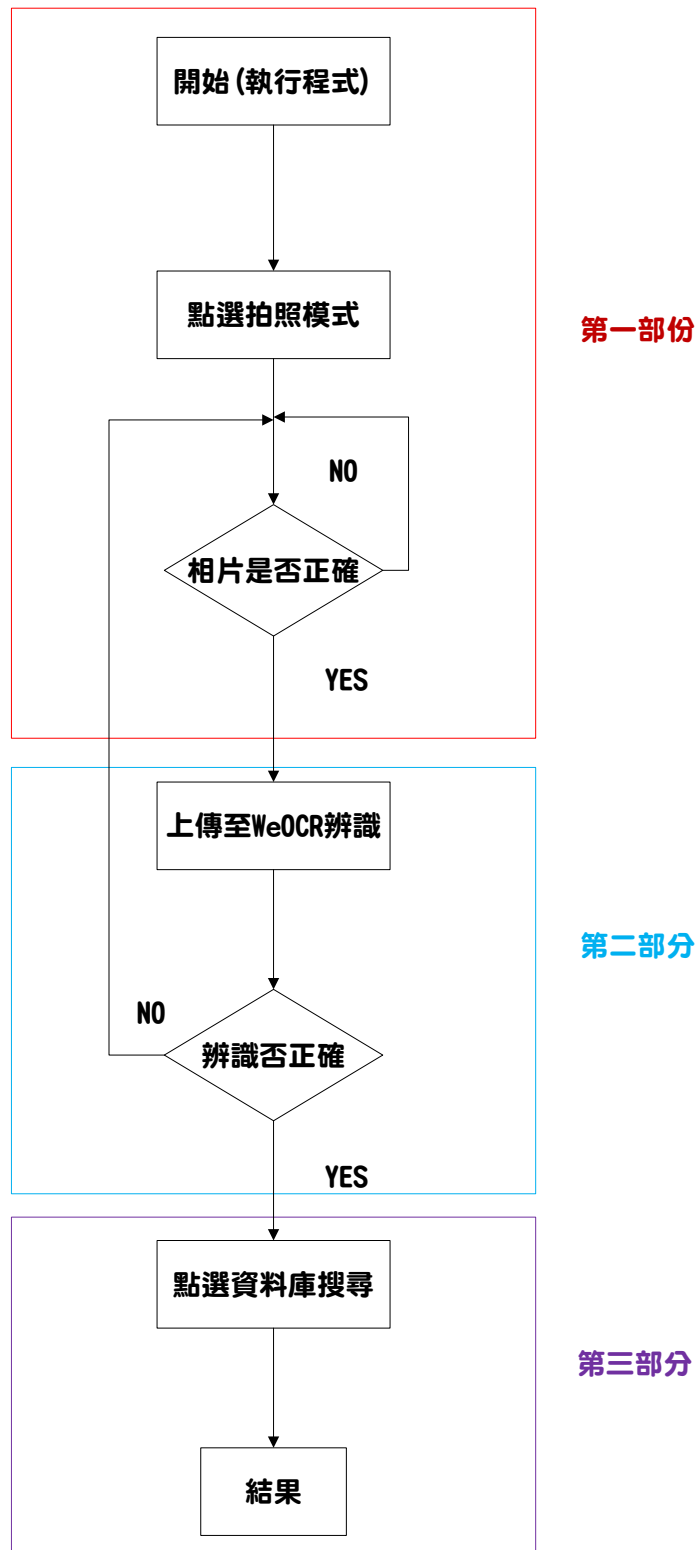


圖 19 系統之流程圖

3.2 影像之處理

我們以OCR來做辨識(Optical Character Recognition)，OCR是光學文字辨識，用途在已存在的文件做文字辨識，轉換成電腦所能認識的電子訊號^[19]。

■ OCR 流程圖如圖 20^[20]：

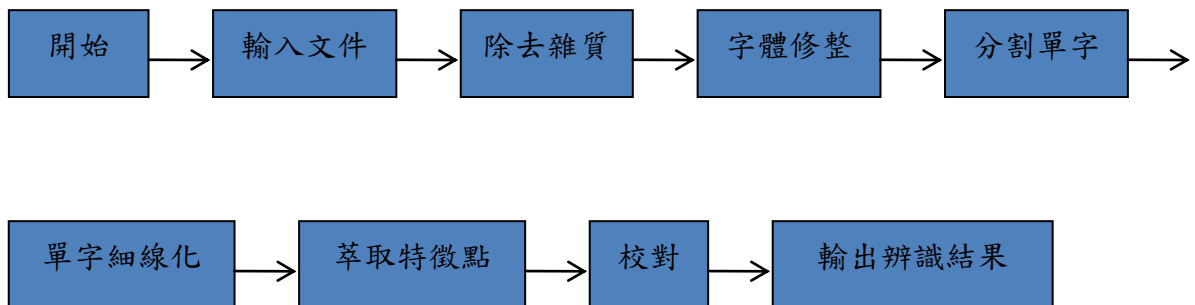


圖 20 OCR 流程圖

- (1). 開始：準備。
- (2). 輸入文件：OCR 的資料輸入是利用平台型掃描，將卻識別的文章，掃描成圖形格式檔。
- (3). 除去雜質：由於輸入文件或圖形表面不乾淨，或掃描器本身造成失真現象，將使輸入的影像存有一些汙點，如此將造成文字辨識之困擾。
- (4). 字體修整：為了改變掃描器造成的失真或是解析度解析之問題，固於文字辨識前，修整字體。
- (5). 分割單字：在進行辨識前，需先將文字或符號獨立分割出來
- (6). 單字細線化：單字細線化是讓字體只剩下骨架。
- (7). 萃取特徵點：單字細線化後，就要用個種方法來表示或描述法來針對骨架體做編碼的工作。
- (8). 校對：當第一個圖形或文字被辨識與編碼後，必需執行一次比對動作，以便找出相近的文字
- (9). 輸出辨識結果：把辨識之結果輸出到文字檔。

等以上之流程來辨識出圖像，我們把拍好的圖檔傳至網頁OCR(WeOCR)上做便是將結果傳回至手機上^[21]，WeOCR網路上有列出幾個公開的私服器，表3.1與表3.2為某一台私服器之規格^[22]。

表 3.1：WeOCR 私服器規格

調整	110907.01
服務器引擎	WeOCR (0.10)
服務器的 URL	http://michelle.ocrgid.org/e1/~V
spec 文件	http://michelle.ocrgid.org/e1/srvspec.xml
服務器位置	日本仙台東北大學
聯繫信息。	http://www.sc.isc.tohoku.ac.jp/~hgot/
服務器類型	單
服務水平	實驗
支持的文本方向	橫
支持的元素類	page text_block text_line word
支持的字符類型	machine_printed
最大圖像尺寸	5000 * 5000
硬件	Athlon X2 5050e (2600MHz)，核心，2048MB

表 3.2：支援的文件類型

Type	Description	Depth
BMP	Windows bitmap, OS/2 bitmap	1,8,24
JPEG	JPEG/JFIF	1,8,24
PBM	Portable bitmap	1
PGM	Portable graymap	8
PPM	Portable pixmap	24

3.3 資料庫之查詢

車牌辨識後的結果會自動存取至手機的剪貼簿，最後，我們來做資料庫的搜尋，在搜尋地方(如圖 21 車牌搜尋方式)貼上辨識完後的車牌號碼在經過資料庫搜尋比對，搜索車牌的車主之結果如圖 22 搜尋結果，就可得之車牌車主與是否違規。

車牌查詢系統

FA8067 車牌號碼 搜尋

圖 21 車牌搜尋方式

找到1個與 FA8067 相關結果。 [下載查詢結果用Excel打開](#)

車牌號碼	班級	姓名	學號
FA8067	資訊工程系	羅恆	19001011

圖 22 搜尋結果

第四章 系統實作

Android開發環境支援各種的作業系統，本研究開發環境作業系統為Windows XP 與 Windows 7 (32位元版本)，以下是詳細說明。

4.1 建立 App Inventor 開發環境

在建立App inventor前需要幾樣工具安裝，首先，電腦需下載安裝JDK1.6，請前往http://www.java.com/zh_TW/download/chrome.jsp?locale=zh_TW下載安裝安裝完成後，請下載安裝App Inventor Setup Software(Android Emulator模擬器)，接下請至App inventor網站，請前往<http://www.appinventor.mit.edu/>，選擇如圖 23 App inventor登入方式方式登入^[23]。其安裝過程如附錄一所示，App Inventor的開發環境提供一個簡易使用的開發環境，拋去以往的複雜程式以樂高方式堆疊，給程式開發者來說是一大福音，其次是在雲端執行模擬，這項創舉也只有App inventor，如果在自己電腦上寫上好幾百行的程式您想想這樣您的電腦跑起來快嗎?透過App inventor在雲端上寫程式就不用怕這樣問題，也可以順便幫您也好程式儲存在雲端伺服器上您也不用在自己電腦內占空間，您這樣也不用帶著隨身碟到處走，只要有網路情況下可以走到哪都能寫程式，這樣在開發上真大了許多的方便性。



圖 23 App inventor 登入方式

4.2 App Inventor 開發環境介紹

本系統是以App Inventor為開發環境，整體的設計環境分為兩個部分，一為系統展示之規劃設計，另一個部分為系統的程式設計之規劃。以下詳細說明系統設計之內容。

4.2.1 系統展示之規劃

本系統規劃如表4.1(圖 49)，有詳細說明我們所設計手機面板所需要的物件，有了這些物件不代表可以直接用要透過程式設計的把這些物件串連起來。

表 4.1：使用區塊

分類	物件	備註
	Screen	此物件是您手機面板，一開始我們先改面板背景為黑色 (BackgroundColor=Black)，接下來改面板名稱『吳鳳科技大學車牌辨識系統』。

分類	物件	備註
Screen Arrangement	HorizontalArrangement	此物件為橫排用，讓您在裡面不會跑掉與執行時大小不依。
Basic	TextBox	此物件可輸入文字用，本論文是為了能複製回傳著的文字用。FontSize：14、Width：95 pixels、Height：Automatic。
	Button1	此物件觸碰會載入相機或是連接至其他地方用。 FontSize：40、Image：您自己的代表圖、Width：Fill parent、Height：Fill parent。
	Button2~3	此物件觸碰會載入相機或是連接至其他地方用。 FontSize：20、Image：您自己的代表圖、Width：80 pixels、Height：80 pixels。
	Image	此物件是為了顯示拍照之結果。 Width：Fill parent、Height：Fill parent。
Media	Camera	此物件是為了載入手機原有的拍照模式。
Other stuff	ActivityStarter1~2	此物件是為了連接至網路上所指定地方。 Action：android.intent.action.VIEW、DataUri 連接的網頁

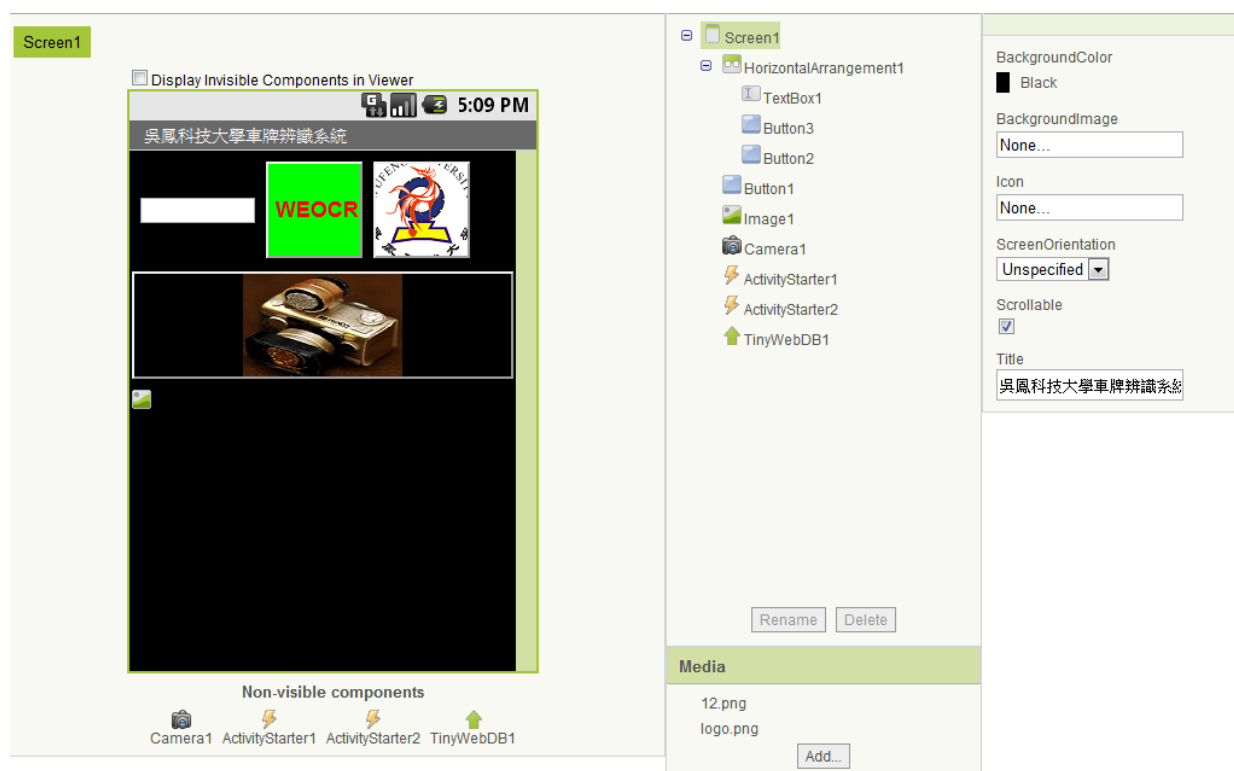


圖 24 手機面板

4.2.2 系統的程式設計與規畫

透過以上面板設計後，接下來說明面板設計後需要的串連部分，有了著些您的系統才算完成，首先，先開啟拼塊編輯後，接著繼續來完成車牌辨識系統的專案設定(如圖 25)。

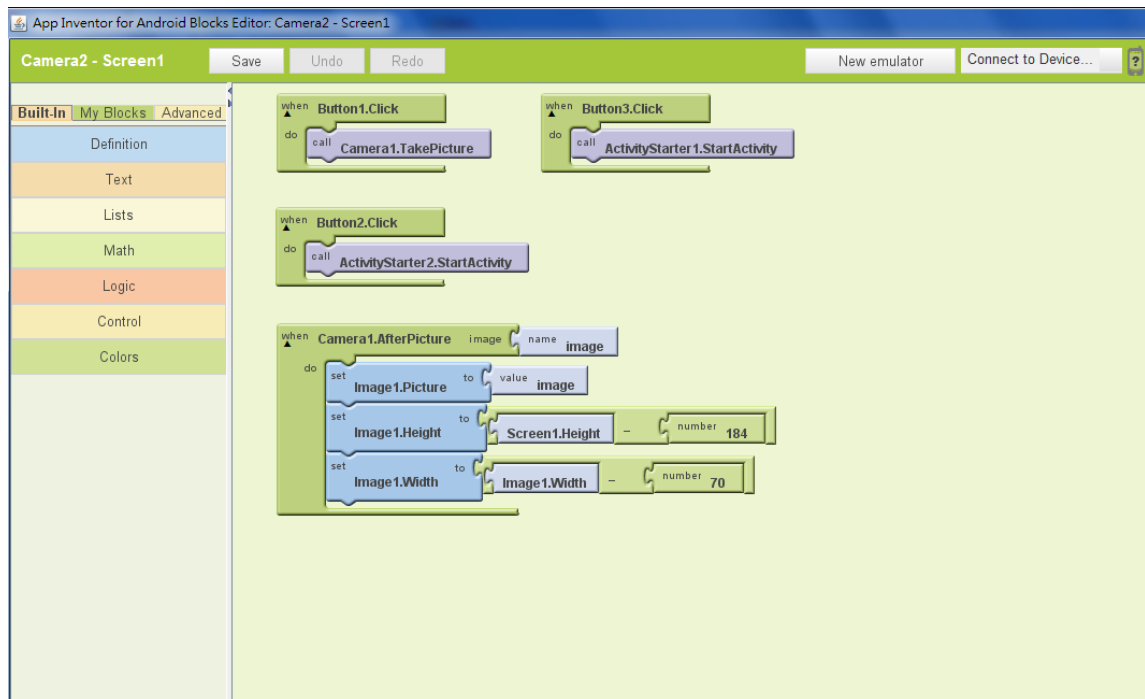


圖 25 內部程式設計

接著說明這些使用方式圖 26 Bulton1.Click do，是為了點選這個物件後(Call Camera1.TakePicture)呼叫手機內的相機模組，開啟相機模式。

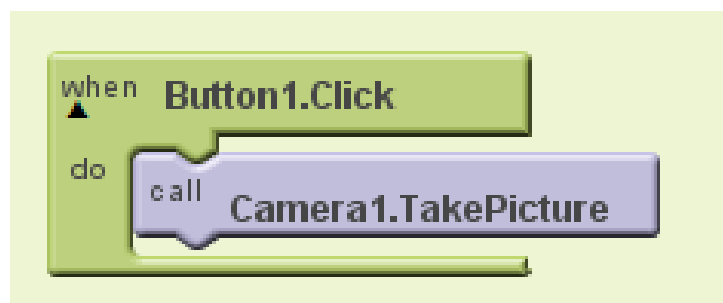


圖 26 連接相機

圖 27 的Bulton2.Click、Bulton3.Click do，是為了點選這個物件後(Call

ActivityStarter1.StartActivity 與Call ActivityStarter2.StartActivity)連接至我們所設定的網站上所使用的模組。

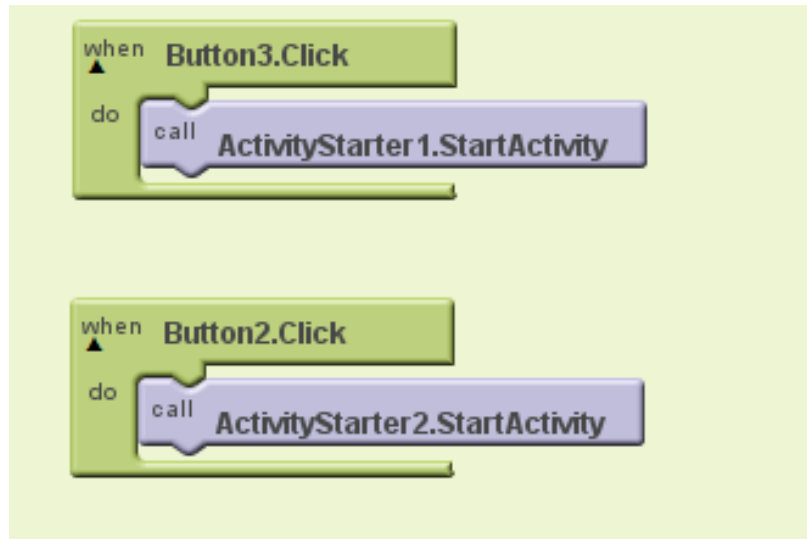


圖 27 連接至網路上

錯誤! 找不到參照來源。 的Camera1.AfterPicture do，是為了回傳相片顯示在手機上，為了顯示的結果所以設定 Set Image1.Height to Screen1.Height- 184 與Set Image1.Width to Inage1. Width-70以上為了圖片顯示的大小，接下如附錄一有詳細說明，執行結果。

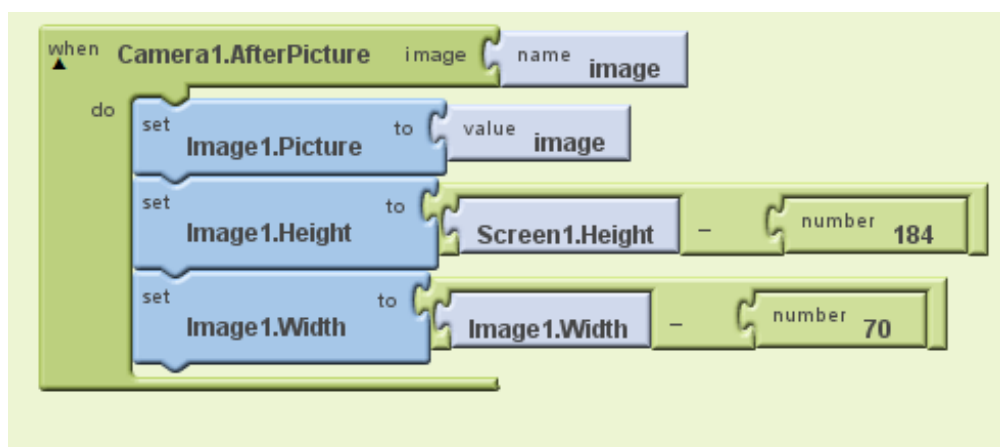


圖 28 圖片顯示

第五章 實驗與功能驗證

本文在上章節說明實驗環境之配置，接下來部分針對車牌辨識各步驟進行分析與討論。實驗設備包含HTC手機、NOVO7極光平板二部分。HTC手機與NOVO7平板透過上述章節方式安裝，安裝完後實際到學校測試結果如下。

5.1 測試樣本環境

測試與比對樣板之車牌影像樣本，取自於本校內停駐之靜止汽機車。試驗標的物為白底黑字之六字車牌。以時間對該樣本集合進行分組區隔，白天所測試辨識為1/4，夜晚所測試辨識為1/2。表 5.1以時間進行比例項目為以時間進行分組區隔的比例。

表 5.1 以時間進行比例項目

時間區隔	測試辨識數量	成功	失敗	比例
白天	10	8	2	1/4
夜間	10	5	5	1/2

車牌影像樣本大小設定為175×80 pixels。車輛與鏡頭在無縮放的狀況下，以距離範圍1至8公尺內拍攝，以上夜晚測試結果是因為燈光問題所至建議在有燈光時要調整手機內的設定即可。

5.2 結果

為了測試結果，我們以本校停車場上，任一台做測試(如圖 57)所示，將圖片傳至WeOCR上辨識(如圖 29)所示，辨識後回傳至手機上(如圖 59)所示，就可以至資料庫搜尋車牌訊息。



圖 29:辨識中畫面

透過本文以上敘述我們再以此系統做一些測試。

- (1). 測試一：在同一情況下用校園無線網路與電信用的網路比較 WeOCR 辨識之速度如下表 5.2。

表 5.2 網路環境比較速率

網路環境	速率	張數
校園無線網路	30 秒	1
電信用無線網路	45 秒	1

- (2). 測試二：在正光與背光測試，如下表 5.3。

表 5.3 正背光之測試

光的方向	成功	失敗
正光	4	6
背光	7	3

- (3). 測試三：辨識之距離遠近，如下表 5.4。

表 5.4 辨識之距離

距離	成功	失敗
1~8 公尺	8	2
9~17 公尺	2	8

由測試結果得知，在本校的無線網路在上傳至WeOCR做辨識，辨識速度是比較快，在沒本校無線網路雖比較慢還是可以接受，因為會讓您沒感覺到差異性，所以也可以使用電信業者無線網路，在表5.1的時間比較是為了測試白天跟晚上的

差異性，因為本校也有夜校生所以做比較，其結果會因為手機設定與畫素有關因為您的設定沒有變更會因為當時光線與陰影造成以上之結果，所以建議要因為當下早晚變更您手機設定。於測試二中的正與背光中會呈現表5.3之原因是您要考慮正光會出現曝光的情況，背光則是會碰到光線不足所出現之陰影部分，在辨識時就會出現錯誤與便是很久結果都是什麼多沒有的情況。於測試三中拍照距離部分如表5.4但這表是在同一設定上沒改變情況下拍攝的，要避免著失敗是可以的拍照時調整您手機的距離，把距離拉近即可避免便是失敗的發生。所以透過以上測試可以得知辨識跟自己人為疏失(搖晃到或是其他東西跑進去或是鏡頭有汙漬)有關、也跟您所設定有關，避掉以上因素就可以達到百分之90%以上的辨識率。

第六章 結論與未來展望

本研究提出以Android車牌系統之研製，以現在最方便也是最好攜帶的Android手機做結合，透過此結合能減少紙張的浪費，在輸入方面別於以往的查詢方式，本研究的提出，則是利用HTC的Android系統手機結合照相功能加上其連接手機提供WiFi的功能，提供快速獲得辨識與查詢之結果，如此可以省去在查詢時所使用的時間，並且更方便不受限於以往只能用在以往的紙張或是以肉眼去查看，而是可以在手機上方便的操作，在寫程式的方面是用App Inventor的雲端方式存放，在以往撰寫程式與偵錯部分靠的是您的電腦性能也取決的速度，但透過雲端方式的App Inventor可做到不靠電腦本身的硬體的強弱，有了這項技術可以提供一個很好的環境，可達到以前人們想不到的在其他地方也可以工作，只要有電腦與網路的情況下就可成功了，在其次儲存地方在雲端您也可以不用怕您的電腦壞掉或是資料遺失，有了以上這些功能加上資料庫的支持所以本文提出以Android車牌系統，在校園管理人員透過這項技術達到節省時間與人力之目的，也不會因漏看或鎖錯車輛的情況發生，也可快速查詢車牌，對車輛更有效管理，也可減少錯誤可能性發生。

本研究在照片攝取部分，因拍照時會造成晃動或是在設定的問題與車牌畫面不是車牌文字，如背景、車牌附近的貼紙或是車牌上掛了東西在辨識中會出現辨識錯誤，如字元不正確、產生多於字元或是亂碼等，在未來研究上可以考慮在手機上採取影像處理的方式來做車牌英文與數字的定位，使辨識效果更好，辨識率提升，在系統上方面不必在做修改。

本系統採用無線網路服務來提升速度，所以要注意要有連接網路情況才能，而目前許多地方皆有建置WiFi或是您手機的電信業者無線網路(靠訊號強弱)，但不見得可以很順暢連接網路如電信業者是靠訊號部分，不見得一定收得到或是很快在其次有些地方沒有3G的基地台，因此傳輸速度會比較慢，也會造成等待結果延遲，所以無線網路與3G的環境對這系統很重要。

本研究用的開發程式用App Inventor 雖說在程式儲存很快,也比較簡單易懂,但在開發程式時會因是堆疊而成,而造成程式沒辦法修改,而在資料庫部分目前App Inventor也是靠網路才能成立,在撰寫資料庫只能另外方式處理,原本論文想法是可以不透過網路情況下做辨識與建置資料庫,但因App Inventor的受限,只能另外處理這方面問題,所以建議要將系統改成不透網路方式,因為不透網路就可更便利也可接近人性化,如手機的導航程式,雖說在撰寫程式要用其他方式,如Eclipse在撰寫時會先需要安裝與下載SDK套件,在下載SDK套件時會需要許多時間,也需要許多設定,在程式撰寫、除錯、模擬器部分要跟您電腦有關,比較不受限,但在這之中您會發現資料庫可能會很龐大佔據您手機內記憶體或是手機外記憶卡的容量,如導航程式那樣,不過不用怕因為現在科技的發達記憶體與記憶卡的容量一直在上升您可購買更高的記憶體與記憶卡。

透過以上說明後可以更有用的利用手機將辨識系統變成在手機內辨識與資料庫也在手機上就不受限於網路,可以使用手機網路上資料庫下載更新,如台鐵或是其他離線查詢,也可透過網路在每幾天線上更新資料庫內容達到準確度百分百情況,也可結合其他內容進入,如警政署的車輛竊盜、車牌失竊(含計程車)資料查詢與環保署的排氣檢查資料庫,不只可以在校園也可在其他地方使用,根據報導「柯南大學生,查贓車70萬獎金入袋」^[25],透過此系統就可以達到百萬以上。

參考文獻

- [1] 行政院主計處，「國情統計通報」1-11 月汽車新增掛牌數34.7 萬輛，第241 號,<http://www.stat.gov.tw/public/Data/1121916103871.pdf> ,2011,12,19.
- [2] 陳啟川，「台灣寬頻網路使用調查，3.5G為主要行動上網方式」，RUN!PC,<http://www.runpc.com.tw/news.aspx?id=100201>,2009,0,17.
- [3] 侯嘉威，「3.5G 一網打盡」數位狂潮電腦雜誌，42 期，<http://event.asus.com/DigiTrend/42html/M-blog-094203.htm>,2009.
- [4] Wuzhimin，「《程式員》雜誌」Android 之父 Andy Rubin：生而 Geek，第9 期,<http://www.programmer.com.cn/3970/>,2010
- [5] 賴婷婷，「蘋果與宏達電之專利戰」，交通大學科技法律研究所，http://cdnet.stpi.org.tw/techroom/pclass/casefocus/2011/pclass_casefocus_11_015.htm ,2011,07, 19.
- [6] Reto Meier、鐘政欣 譯，「Android 2.X 應用程式開發經典」，基峰資訊股份有限公司, 2011,06.
- [7] Linuxconfig.org，「Android system architecture」，<http://linuxconfig.org/android-system-architecture>，最後修訂於 2011,01, 17.
- [8] developer，「What is Android? 」，<http://developer.android.com/guide/basics/what-is-android.html>，最後修訂於 2012,04, 26.
- [9] 郭長祐，「Google Android 裡頭有什麼？」，<http://kit168.blogspot.com/2008/10/google-android.html> 2007,12.
- [10] 謝良奇，「Android 程式開發不求人 Google 將推視覺化開發工具」，http://www.openfoundry.org/index.php?option=com_content&task=view&id=8097&Itemid=4;isletter=1 , 2010,07.
- [11] 蓋索林，「App Inventor 讓你用拼圖來寫 Android 應用程式」，網路資訊雜誌,2010,09,23.
- [12] 探奇自然科學教室 LEGO NXT 樂高機器人 DNA 自然科學實驗，「Android 手機控制 NXT 樂高機器

- 人」, http://tw.myblog.yahoo.com/touch_classroom/article?mid=4900, 2010, 08, 04.
- [13] 曾吉弘、李卿澄, 「Google Map 路徑追蹤機器人」, CAVE 教育團隊
<http://pansci.tw/archives/10072>, 2011, 12.
- [14] 優勝家機器人教室-LEGO NXT 樂高機器人教學, 「2012 WRO 國際奧林匹克機器人大賽 鳴槍起跑了!」,
<http://blog.sina.com.tw/win940/article.php?pbgid=87179&entryid=634819>, 2012, 01, 15.
- [15] 黃重憲, 「淺談雲端運算 (Cloud Computing)」, 臺灣大學電機資訊學院資訊工程系, http://www.cc.ntu.edu.tw/chinese/epaper/0008/20090320_8008.htm, 技術論壇, 第 0008 期, 2009, 03, 20.
- [16] 維基百科, 「雲端運算」,
<http://zh.wikipedia.org/zh-tw/%E9%9B%B2%E7%AB%AF%E8%A8%88%E7%AE%97>, 最後修訂於 2012, 04, 24.
- [17] 林育竹, 「2010 雲端運算 Cloud Computing internet radio (803) 美林證券九百五十億美元」, <http://blog.udn.com/tsuoscott/3916233>, 2010, 0.
- [18] App Inventor 中文學習網, 「指令中文化」,
<http://dl.dropbox.com/u/11288673/AppInventorRef.rar>, 2011
- [19] 鄭吉辰, 「Android 平台手機之雲端車牌系統建置」, 2010 年。
- [20] 技術服務小百科, 「光學字元識別」,
<http://techserviceslibrary.blogspot.com/2011/04/ocr-optical-character-recognition.html>, 2011, 04, 27.
- [21] Searchable PDF, 「OCR 的主要辨識流程」,
<http://searchable-pdf.com/content.php?lang=tc&c=16>, 2010.
- [22] wordsnap-ocr, 「Mobile OCR input: “Fully automatic” and reality」,
<http://www.bitquill.net/blog/?p=119>, 2009.
- [23] weocr, 「WeOCR service : Server spec」,
<http://michelle.ocrgrid.org/e2/srvspec.html>, 2006.
- [24] 王培坤, 「Google App Inventor 開發手冊不會寫程式也能設計你的 App」, 上奇資訊股份有限公司, 2011, 12.

[25] 彭慈然，「柯南大學生 查贓車 70 萬獎金入袋」，30 雜誌，
<http://www.30.com.tw/Board/show.aspx?go=976>, 第 041 期 2008,1.

附錄

A. 系統安裝與操作說明

Android開發環境支援各種的作業系統，本研究開發環境作業系統為Windows XP 與 Windows 7 (32位元版本)，以下是詳細說明。

a. 建立 App Inventor 開發環境

在建立App inventor前需要幾樣工具安裝，首先App inventor網站，請前往<http://www.appinventor.mit.edu/>，選擇如下圖 23 App inventor登入方式方式登入。

進入完後，請申辦Google帳號(如有Gmail即可,也可以用其他Email帳號申請)如圖 30 申請Gmail、圖 31 申請Gmail，申請完後，初次登入後會要求同意使用版權說明，後就可以進入首頁畫面如圖 32 登入App inventor。



圖 30 申請 Gmail

登

建立新的 Google 帳戶

1
2
3



除了搜尋，您的 Google 帳戶還能帶您體驗其他更多服務。

聊天、即時通訊、分享、排程、儲存、管理、協作、探索與創作，各項功能應有盡有！只要一組使用者名稱和密碼，您就可以登入 Gmail、Google+、YouTube 等 Google 產品，還可以瀏覽自己的搜尋記錄，而且永遠不必擔心備份問題，因為 Google.com 就是您最方便的檔案庫，隨時為您儲存重要資料！



盡享所有的服務。

舉凡您的 Gmail 或線上相簿等各種內容，全都透過 Google 帳戶跨裝置同步存取。還有更豐富的搜尋方法，拍個照或用語音輸入都能搜尋。除了免費的路線導航功能，您還能自動上載剛拍好的相片。不久之後，用手機上的 Google 電子錢包就能購物了。



分享多少，由您作主。

想與好友、家人甚至老闆分享生活大小事？Google+ 的社交圈分類功能，幫您輕鬆選好分享對象。您可以跟三五好友一同進行視訊聚會、傳送群組訊息，或是追蹤有趣的圈外使用者。豐富的互動功能，完全隨心所欲。

名稱

選擇您的使用者名稱

@gmail.com

建立密碼

確認密碼

生日

性別

我是...
▼

行動電話


+886

其他電子郵件地址

協助我們排除自動程式


LIMITED.

圖 31 申請 Gmail



MIT App Inventor

BETA

[My Projects](#) [Design](#) [Learn](#) [\(Debugging\)](#)

Welcome to the App Inventor beta preview release. Be sure to check the list of [known issues](#).

New

Delete

Download All Projects

More Actions ▾

Projects		
	Name ▲	Date Created
<input type="checkbox"/>	App_1	2012 Mar 21 13:30:10
<input type="checkbox"/>	Camera	2012 Mar 12 09:57:12
<input type="checkbox"/>	Camera1	2012 Apr 11 05:44:15
<input type="checkbox"/>	Camera2	2012 Apr 11 06:11:15
<input type="checkbox"/>	Project14	2012 Mar 20 01:10:18
<input type="checkbox"/>	Project3	2012 Mar 21 14:25:29
<input type="checkbox"/>	scoreInput	2012 Mar 22 13:53:31
<input type="checkbox"/>	test	2012 Mar 21 13:29:05
<input type="checkbox"/>	tinyDB	2012 Mar 12 09:58:27

圖 32 登入 App inventor

完成以上申請後請至網路上下載安裝 java，請前往 http://www.java.com/zh_TW/download/chrome.jsp?locale=zh_TW，點選我同意並開始免費下載，下載完後執行此安裝檔，接受授權憑證後，選下一步即可自動安裝(如圖 33 java 安裝)。如果安裝過或者不知道是否有安裝可以前往此網址測試 <http://www.java.com/en/download/testjava.jsp>，測試如果沒問題即可不需再安裝(如圖 34 java 測試是否有安裝及版本)。



圖 33 java 安裝



圖 34 java 測試是否有安裝及版本

接下來App Inventor Setup Software(Android Emulator 模擬器) 的安裝下載 http://dl.google.com/dl/appinventor/installers/windows/appinventor_setup_installer_v_1_2.exe，檔案大大約有90MB，執行安裝步驟如下：

(a). 下載完成後，找出指定路徑即可開始安裝(如圖 35 下載完成後)。

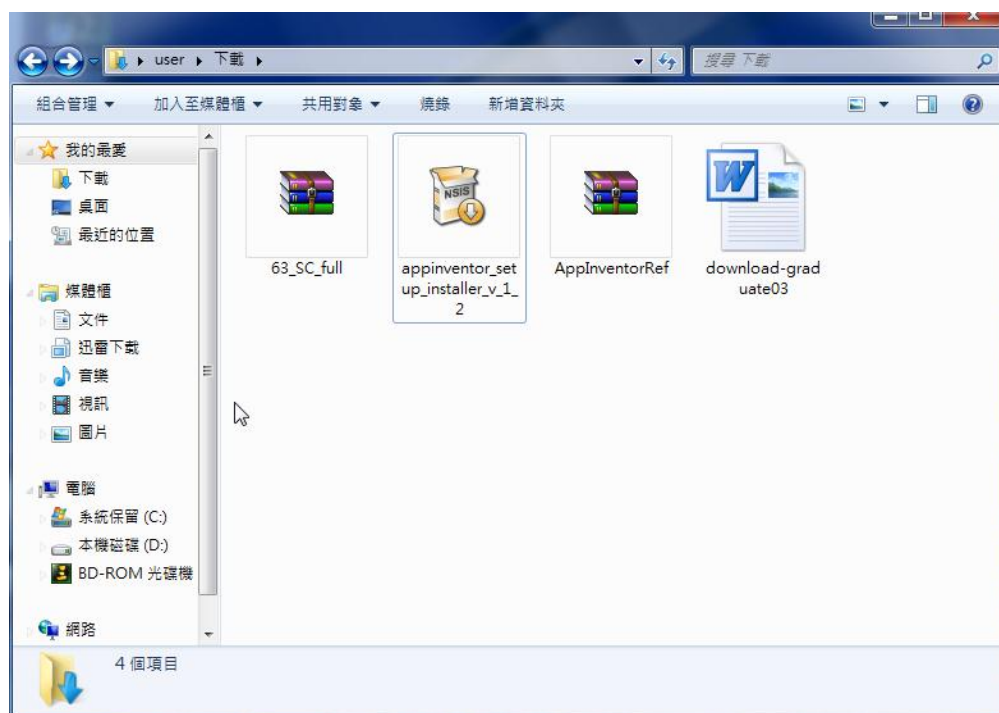


圖 35 下載完成後

(b). 點擊”執行”安裝，圖 36 執行安裝。



圖 36 執行安裝

(c). 點選”Next>”，圖 37 點選”Next>”，下一步安裝。

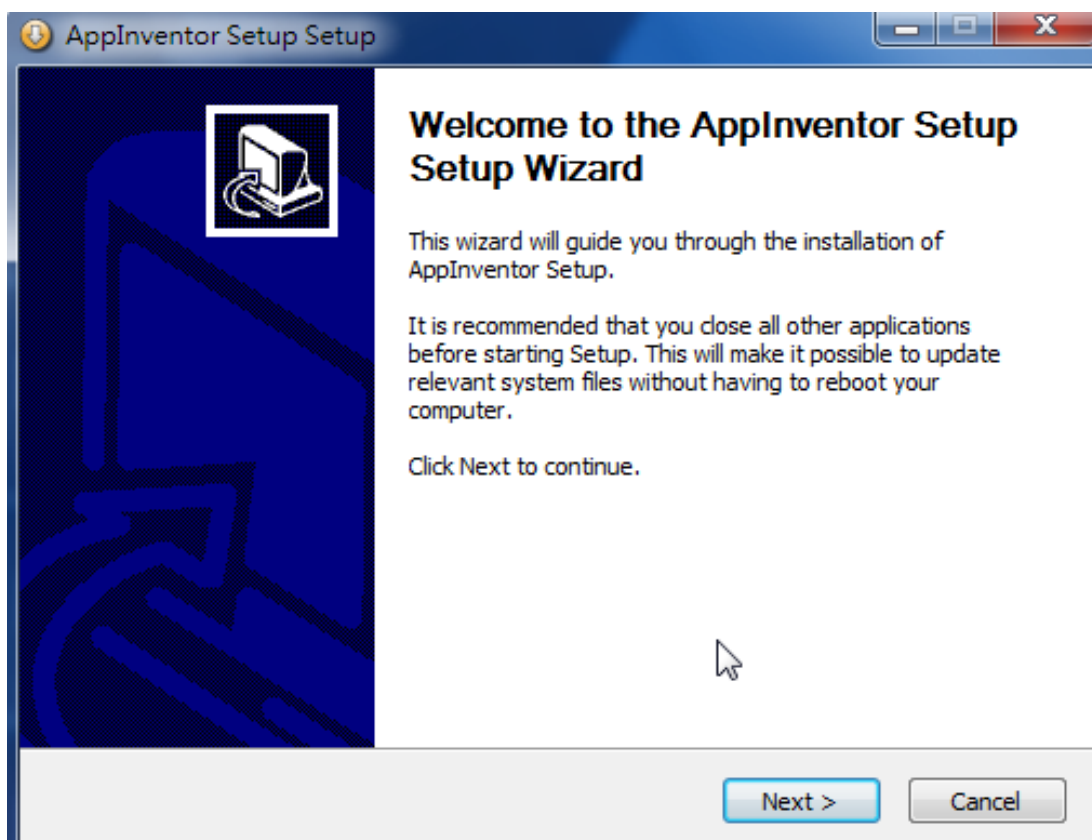


圖 37 點選”Next>”，下一步安裝

(d). 請同意下一步驟安裝條約，若不同意無法安裝，圖 38。

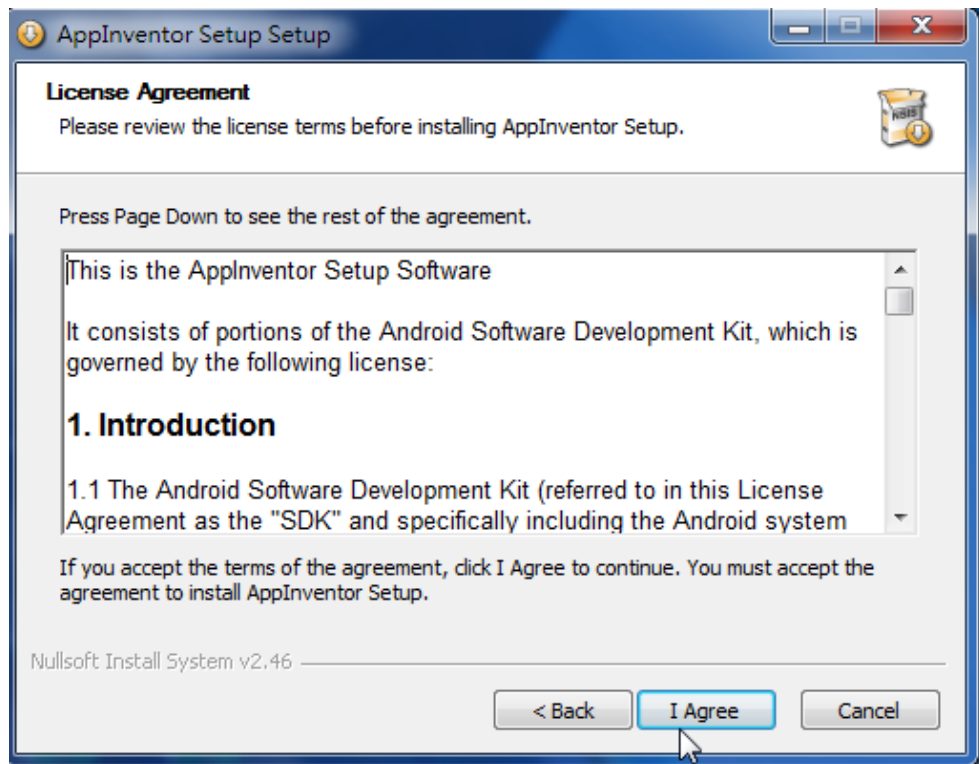


圖 38 安裝條約

(e). 請確認安裝路徑，確認完後按下”Next>”，圖 39。

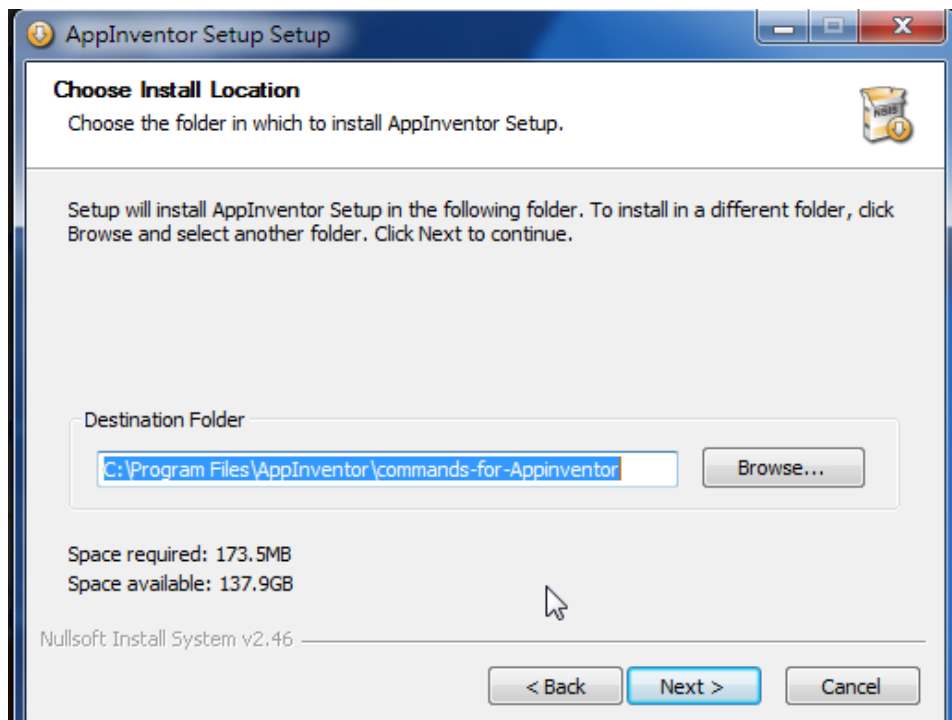


圖 39 確認安裝路徑

(f). 按下 Next 後會出相關目錄位置，可直接按下”Install”，圖 40。

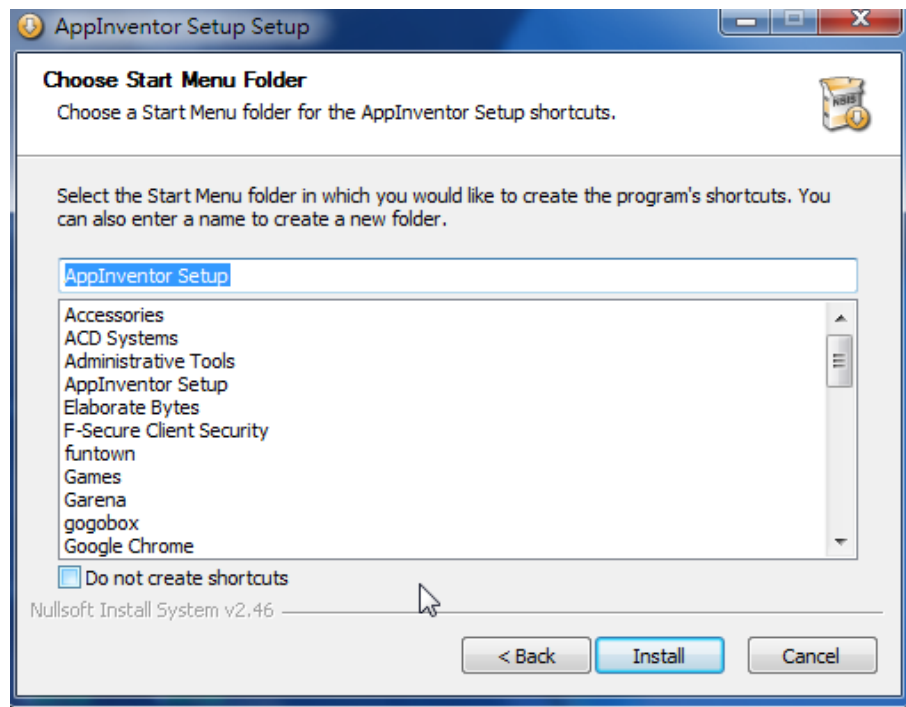


圖 40 相關目錄位置

(g). 安裝開始，接下來請等待，圖 41 安裝開始。

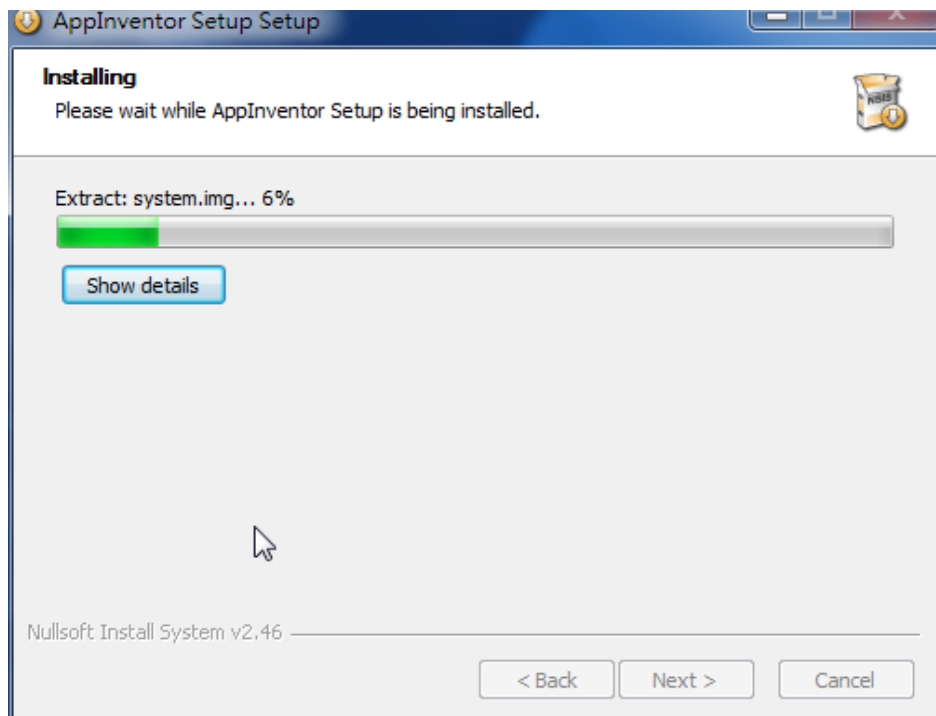


圖 41 安裝開始

(h). 安裝完成後您可以在程式集目錄中找到 AppInventor Setup 目錄，圖 42。

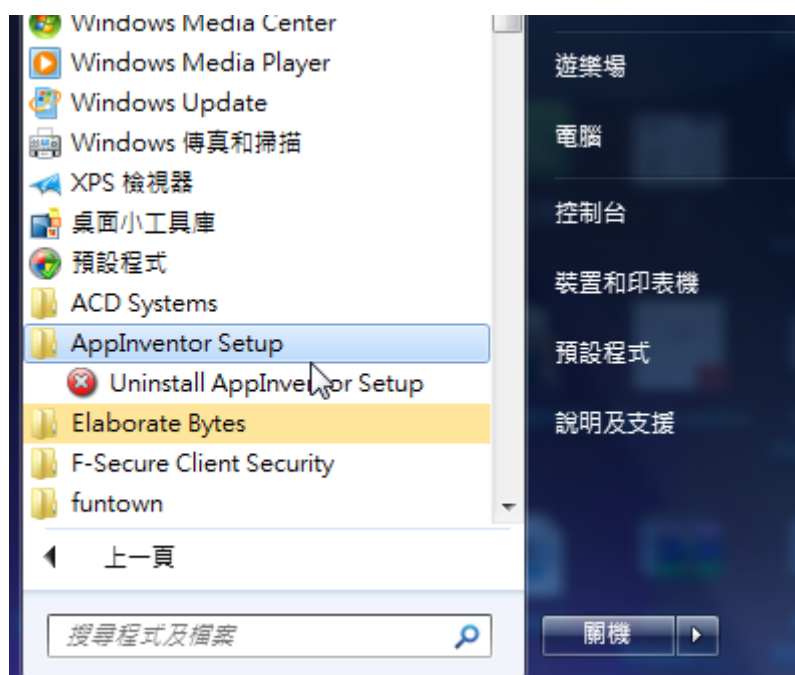


圖 42 AppInventor Setup 目錄

(i). 安裝完後，請下載 Google 瀏覽器

<https://www.google.com/chrome/index.html?hl=zh-TW&brand=CHMI&installdataindex=homepagepromo>，方發部分在IE會有一些問題發生，所以使用Google 瀏覽器。

b. App Inventor 開發環境介紹

完成以上安裝後，接下來介紹一下App Inventor 環境，因所有環境都是環環相扣，所以在建置專案前先了解一下。

(a). My Project 我的專案

首先介紹 My Project 作業，以圖 43 App Inventor My Project 中所標示來逐一說明。

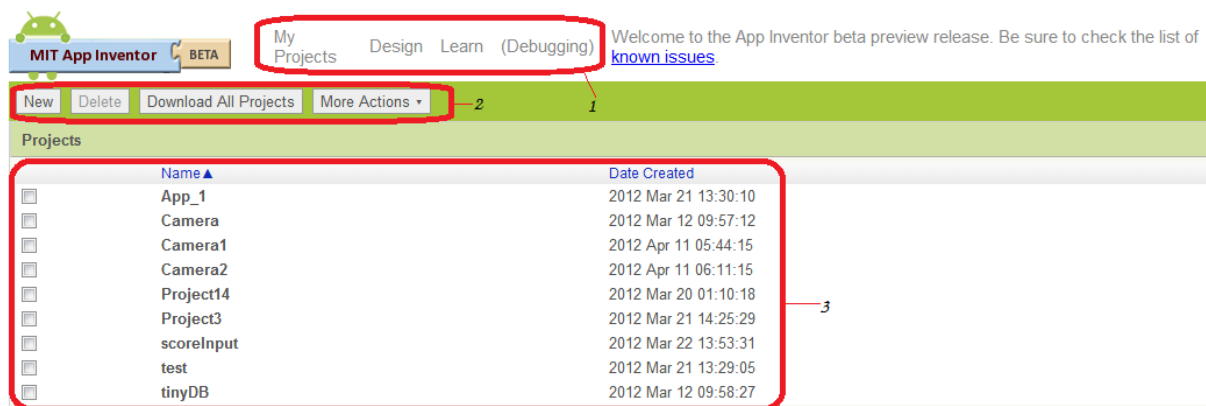


圖 43 App Inventor My Project

1 我的專案、設計與學習

- (1) 點選”My Project”，若沒有專案是不會有動作。
- (2) 點選”Design”，為開啟專案設計，但在點選”Design”前先在呈現專案前單選。
- (3) 點選”Learn”，為開啟 App Inventor 學習視窗，內有詳細教學。

2 新增、刪除與匯入專案即匯出專案

- (1) 若您想新增專案時，請點選”New”鈕，填入專案名稱，但名稱不能使用中文只能英文，填入名稱後即可新增。
- (2) 若您想刪除專案時，請點選”Delete”鈕，即可刪除專案。
- (3) 專案匯入與匯出，專案匯入(Upload Source)功能時，請尋找分享的 App Inventor 專案為壓縮檔，專案匯出(Download Source)功能時，系統會產生”專案名稱”+”ZIP”壓縮檔。如圖 44

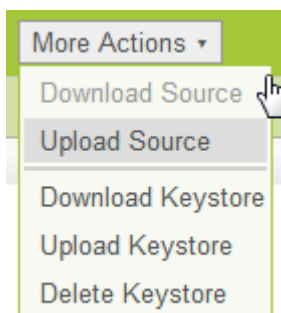


圖 44 專案匯入與匯出

3 我的現有專案

所有完成或沒完成專案都在此。

(b). App Inventor Designer 設計作業

接著介紹 App Inventor Designer 作業，此目的為布置相關元件與設計屬性。(如圖 45)

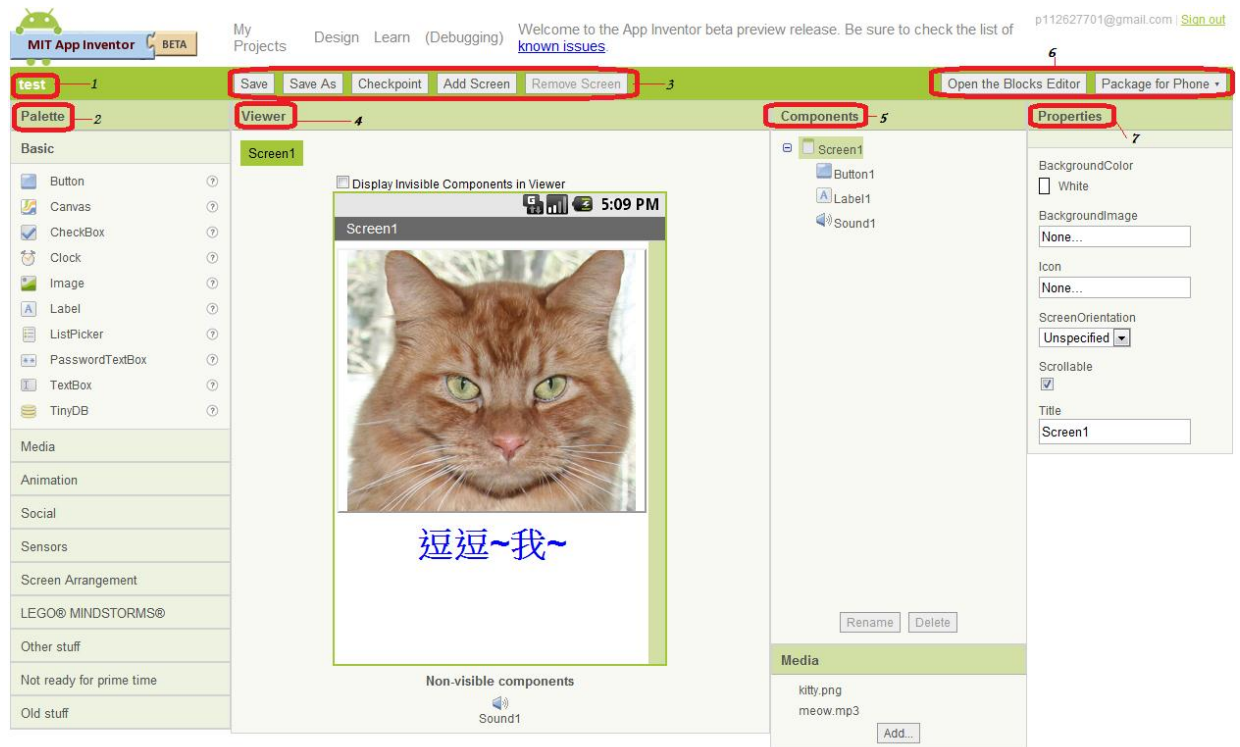


圖 45 App Inventor Designer 作業

1 為專案名稱。

2 Palette 開發元件庫

此元件庫為多種元件，詳細元件說明請至”Learn”。

3 保存、另存新檔、另存為檢查點、添加畫面、刪除畫面

- (1) 保存專案(Sava)，點選可以直接儲存專案。
- (2) 另存新檔(Sava As)，點選完後專案前面出現 Copy 字樣，當另存新專案名稱。
- (3) 另存為檢查點(Checkpoint)，開發大型專案時比較好用，在開發大型專案時往往會遇到瓶頸需耗回頭檢查或另尋找開發方向。
- (4) 添加畫面(Add Screen)，可以再添加一個畫面。
- (5) 刪除畫面(Remove Screen)，刪除添加畫面。

4 Viewer 設計開發區

配置相關元件，當您開始配置時，系統會自動產生 Android Phone 畫面。

5 Components 元件應用區

此區域為您所選取之開發元件庫物件，您可以在此區域改變元件名稱或刪除。

6 開啟拼塊編輯(Open the Blocks Editor)、打包應用程式(Package for Phone)

開啟拼塊編輯按鈕按下去後，將會開啟檔案下載期開始格式 Java，開啟後會出現另一視窗，視窗內容就是拼塊編輯器。如圖 46 開啟拼塊編輯。

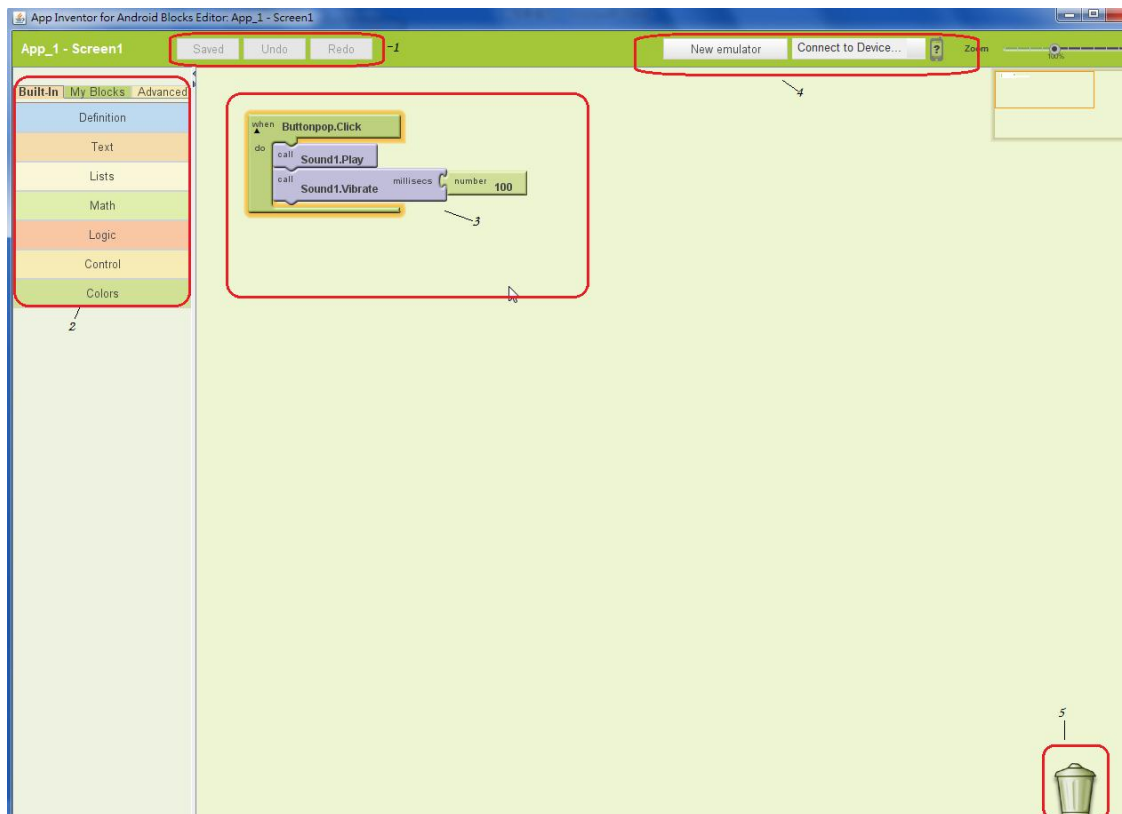


圖 46 開啟拼塊編輯

打包應用程式三個選項，如圖 47。

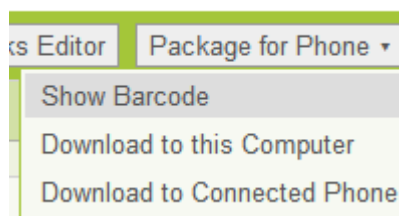


圖 47 打包應用程式選項

- (1) 此選項是把設計好的程式以 QR Code 方式呈現，可提供 QR Code 方式讀取。
如圖 48

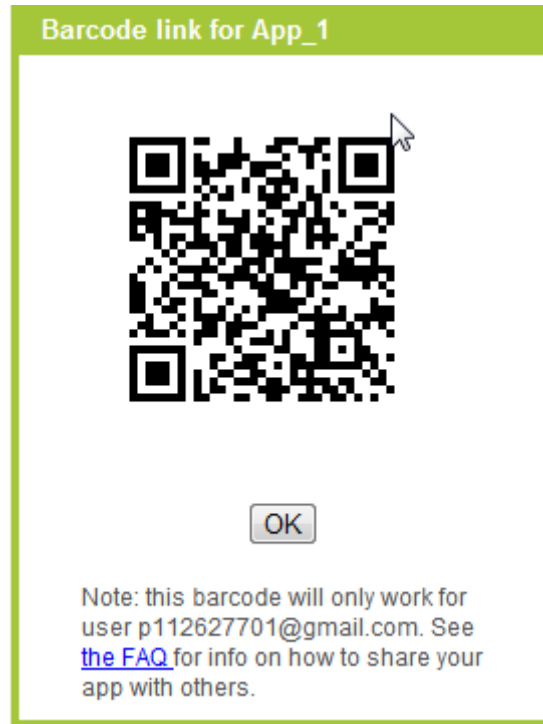


圖 48 QR Code 方式呈現

- (2) 此選項是把設計好的程式下載至電腦中。其格式是”專案名稱.apk”此 apk 可以分享至網路上，供人測試。
- (3) 此選項是把設計好的程式下載至手機中，進行安裝。

7 Properties 元件屬性區

此區會針對您在元件應用區所選取之元件顯示相關屬性。

(c). 開啟拼塊編輯器

當您在樣板設計好後，開啟拼塊編輯器開能，可以在這製作您 APP 所要之功能，如圖 46。

1 為儲存、返回上一步、重作。

2 邏輯控制庫

此區是將您所設計好的元件在這裡作什麼動作。

3 設計控制區塊

此區是佈置與設計邏輯控制件的地方。

4 開啟模擬器與執行專案

(1) 開啟模擬器按下”New emulator”，會出現 Android Phone 模擬器。

(2) 執行專案當您模擬器打開始在執行”Connect to Device”可將您程式下載至模擬器內。

5 垃圾桶

可將您設計錯的元件丟進去，進行刪除。

(d). Android 手機設定

為了讓您手機跟電腦同步須以下設定：

- (1) 先將您的手機進入到”設定”頁面下，開啟後點”應用程式”，接著在”未知的來源打勾”即可。
- (2) 進入”開發”選單後，勾選”USB 偵錯”與”保持喚醒”。

B. 系統測試

介紹完以上後，接下來解說我們開發的”Android車牌辨識系統”，以下作簡介。

- a. 我們所開發框架如圖 49 與圖 50，先將以下元件先拉好後。



圖 49 設計之畫面

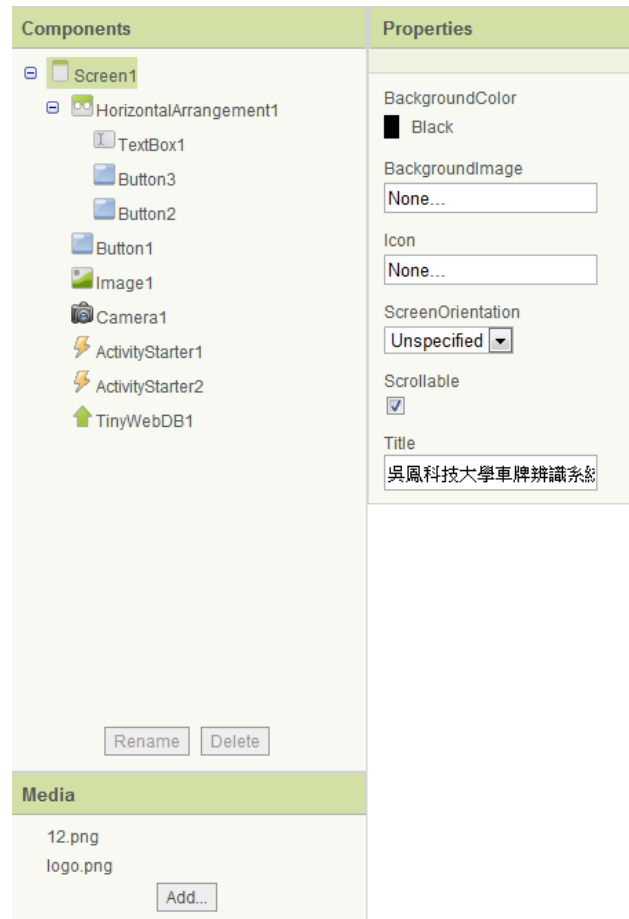


圖 50 設計之元件

- b. 其元件有 TextBox1、Bulton1~3、Image1、Camera1、TinyWebDB1 等，以 TextBox1 來說是用來接收 WeOCR 辨識後的文字或數字，Bulton1~3 是用來作觸發元件像是點選完後會發生拍照或辨識功能等，Image 是用來想是拍照完後的照片，Camera 是相機的觸發元件，DB 是用作為資料庫使用。
- c. Bulton1 用來觸發手機裡的相機模式，此模式為點選後可拍照，如，拍好後選擇是否是妳要的。
- d. Image1 用來顯示您拍好的照片。
- e. Bulton2 用來觸發進入 WeOCR 辨識。
- f. Bulton3 用來觸發進入我們資料庫進行檢查車牌。
- g. 結果以下圖 51、圖 52、圖 53、圖 54、圖 55、圖 56、圖 57、圖 58、圖 59。

圖 51 為用模擬器測試首先一開始會出現如下圖，需先解鎖後圖 52 才能載入您寫好的程式。



圖 51 模擬器開機

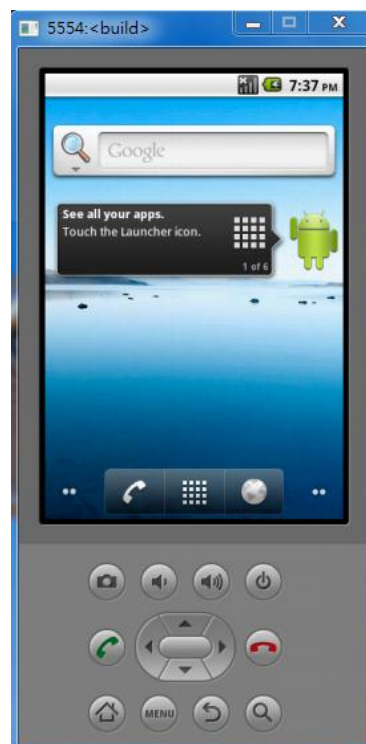


圖 52 為載入前

圖 53是載入中等待的圖載入完後就可以像圖 54一樣，可以清楚看到自己一開始的面板設計的那樣。



圖 53 載入中



圖 54 載入完成

圖 55是為了測試我們所寫的程式連接是否有問題，點完發現如圖 56拍照中

但無影像支援所以改以實際手機拍攝。

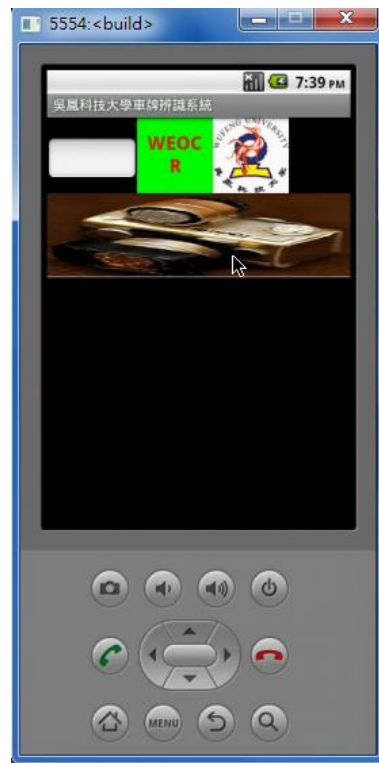


圖 55 點選拍照



圖 56 拍照中但無影像支援

圖 57 用手機實際至本校車棚測試拍攝，圖 58是拍完結果後要點選上傳至 WeOCR辨識，辨識完成後如圖 59，回傳至手機上後就可以點選至本校資料庫內做搜尋，搜尋結果如上第三章所述。



圖 57 用手機實際拍攝



圖 58 拍攝顯是點選 WEOCR



圖 59 辨識結果

C. 資料庫程式

以下是我們設計資料庫所寫的網頁的程式碼其結果如圖 21、圖 22。

```
<?php
$y=date(Y);
$m=date(m);
$d=date(d);
$h=date(H);
$n=date(i);
$t = ".$y.$m.$d.$h.$n.";
$n=mt_rand(111,999);
$q = trim($_GET['q']); //獲取查詢關鍵詞。
$w = trim($_GET['w']); //獲取查詢類別關鍵詞。
$page = intval($_GET['p']); //頁數。
$title = '車牌號碼,班級,姓名，學號'; //分項,用於生成 EXCEL 表格標題和網頁關鍵字。
if($page==0) $page=1;
$r_num = 0; //結果個數。
$p_num = 10; //每頁條數。
$result = "";
```

```

$shengpy = array('key1');
$sheng = array('車牌號碼');
if($q){
    switch ($_GET['w']){
        case "key1":
            $keydb = "k/".$t.$n.$w.$q.".csv";
            break;
        default:
            $keydb = "k/".$t.$n.$q.".csv";
            break;
    }

    if (!@file_exists($keydb)){
        $dreamdb=file("d/p.dat");//讀取區號文件，也是資料的地方。
        $count=count($dreamdb);//計算行數。

        for($i=0; $i<$count; $i++) {
            $keyword=explode(" ",$q);//拆分關鍵字。
            $dreamcount=count($keyword);//關鍵字個數。
            $detail=explode("\t",$dreamdb[$i]);

            for ($ai=0; $ai<$dreamcount; $ai++){
                switch ($_GET['w']){
                    case "key1":
                        @eval("\$found =
ereg(\\"$keyword[$ai]\\.\\\\"$detail[0]\\.\\");");
                        $qw = "車牌號碼: ";
                        break;
                }

                if(($found)){
                    $r_num++;
                    if(fmod($r_num, $p_num)==0) $s .= "\n";
                    $s .= ".$detail[0].'. '$detail[1].'. '$detail[2].'. '$detail[3].";
                    if($r_num>=$p_num*($page-1)+1 &&
                    $r_num<=$p_num*$page){
                        $result .= '<tr
height="24"><td>'.$detail[0].</td><td>'.$detail[1].</td><td>'.$detail[2].</td><td>'.$d

```

```

etail[3].</td></tr>';
        }
        break;
    }
}
$sp = ceil($r_num/$p_num); //結果實際頁數
}
//將數據緩存下來
$fp = @fopen($keydb,"a");
@fwrite($fp,$head.$q."\n".$title."\n".$s);
@fclose($fp);
}else{
    $dreamdb=file($keydb);
    $r_num = trim($dreamdb[0],"\n\r");
    $p = ceil($r_num/$p_num); //結果實際頁數
    if($page>$p) $page=$p;
    $result = $dreamdb[$page];
}

for($i=1; $i<=$p; $i++){
    $post_1 .= '<a href="?q='.urlencode($q).'&p='.$i;
    if($_GET['w']) $post_1 .= '&act='.$_GET['w'];
    if($i==$page){
        $post_1 .= "><font color='red'>['.$i.']/</font></a> ";
    }else{
        $post_1 .= ">['.$i.']/</a> ";
    }
}

$post_1 = '<tr><td align="center" style="font-size:14px;padding:10px;"
bgcolor="#EDF7FF">分頁：'.$post_1.' (共計'.$r_num.'個，每頁'.$p_num.'
個)</td></tr>';

```

```

$result = '<table width="800" cellpadding="2" cellspacing="0" style="border:1px
solid #B2D0EA;"><tr><td style="background:#EDF7FF;padding:0
5px;color:#014198;" height="26" valign="middle"><b>找到'.$r_num.'個與 <font
color="#c60a00">'.$q.'</font> 相關結果。</b> <a href=".'.$keydb.'"
target="_blank"><font color="#c60a00">下載查詢結果用 Excel 打開
</font></a></td></tr><tr><td><table cellpadding="4" cellspacing="4" width="100%"

```

```

style="text-align:center"><tr style="text-align:center;font-weight:bold;" height="26"
bgcolor="#efefef"><td width="80">車牌號碼</td><td width="80">班級</td><td
width="80">姓名</td><td width="80">學號
</td></tr>'. $result.'</table></td></tr>'. $post_1.'</table>';
}
?>
<html xmlns="http://www.wfu.edu.tw">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Big5" />
<?
if($q){
    echo "<title>".$qw.$q." - ".$head."</title>";
    echo '<meta name="keywords" content="'. $q.'.'. $title.'" />';
    echo '<meta name="description" content="'. $q.'.'. $title.'" />';
}else{
    echo "<title>".$head." aiyaha.cn</title>";
    echo '<meta name="keywords" content="'. $title.'" />';
    echo '<meta name="description" content="'. $title.'" />';
}
?>
<link href="i/common.css" rel="stylesheet" type="text/css" />
</head><body background=bg.gif><center><br><br>
<body>
<div align="center">
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<body bgcolor="#FFFFFF">
<DIV id=Head>
<table cellpadding="4" cellspacing="0"
style="background-color:#f7f7f7;border-bottom:1px solid #dfdfff;" width="800">
<tr><td align="left">
<? echo ".$head.";
if($q) echo ' ( &gt; <strong>'. $q.'</strong> - 查詢結果) ';?>
<br>
<style type="text/css">
h3{font-size:24px;padding:15px 10px 5px 10px;color:#014198;}
p{padding: 10px;}
</style>
<table width="800" cellpadding="2" cellspacing="0" style="border:1px solid

```

```

#B2D0EA;" id="top"><tr><td style="background:#EDF7FF;padding:0
5px;color:#014198;" height="26" valign="middle" colspan="5"><b>車牌查詢系統
</b></td></tr><tr><td align="center" valign="middle"
style="padding:20px;"><?=$re?><form action="index.php" method="get"
name="f1"><input name="q" id="q" type="text" size="18" delay="0"
value="<?=$q?>" style="width:200px;height:22px;font-size:16px;font-family: Geneva,
Arial, Helvetica, sans-serif;" onmouseover="this.select()" /> <select name="w"
id="w">
<?
$count = count($sheng);
for($i=0;$i<$count;$i++){
    echo '<option value="'. $shengpy[$i]."'";
    if($_GET['w']=="sheng" && $sheng[$i]==$q) echo ' selected';
    echo '>'. $sheng[$i]. '</option>';
}
?>
</select> <input type="submit" value=" 搜尋 " /></form></td></tr></table>
</td></tr></table><br />
<?
if($q!=""){
    echo $result;
}else{
    echo '<table width="800" cellpadding="2" cellspacing="0" style="border:1px solid
#B2D0EA;"><tr><td style="background:#EDF7FF;padding:0 5px;color:#014198;"
height="26" valign="middle" colspan="5"><b>說明</b></td></tr><tr><td><p
style="line-height:150%">輸入：車牌號碼查詢，如有問題至:吳鳳科技大學查詢。
</p></td></tr></table><br>';
    echo $result;
}
?>
</div>
<a href="mailto:<?=$email?><?=$school?></a>&copy;2012<br></font>
<a href="http://www.wfu.edu.tw" target=_blank>吳鳳科技大學</a></font></p>
<?
}
?>
</div>
</body>
</html>

```