

高雄市高英高級工商職業學校
Kao Ying Industrial Commercial Vocational High School

專題製作報告



智慧型停車場

學生姓名：洪得億

黃意寶

王柏翔

翁聖凱

指導老師：葉忠賢老師

中華民國 105 年 05 月

誌 謝

這項專題的完成，首先要感謝我們的指導老師葉忠賢老師。感謝老師為學生在專題指導上所花費的精力與時間，讓學生能夠從無到有的順利完成專題。若沒有葉忠賢老師的幫助和監督，我想我們可能再過十年也生不出這項專題。而且除了專題的指導外，老師平常對我們人格上的循循善誘，使我們對自己性格上的缺點以及如何去克服有更深刻的認識，老師不但是我們的授業老師，更是一位重要的人生導師。

組長：洪得億、組員：王柏翔、黃意寶、翁聖凱等謹上 2016/05

智慧型停車場

摘要

我們身處於地窄人稠的台灣，隨著高樓大廈的落成—尤其是像台中這種都市的停車位置愈來愈難找了。汽車駕駛想找一個停車位，都會消耗相當寶貴的時間，往往在同樣的地方來回數趟，卻遲遲找不到停車的位置。所以我們希望能利用 Arduino 和 App Inventor 設計出一套智慧型停車場車位引導系統，給予駕駛最明確的指示，並以最短時間引導至距離最近的停車位置。

這次有用到的是 Arduino，相較於學長姐的 8051，我們是用新的課程來上，所以有些功能還是第一次的用到，但實驗過了裡面的一些教材內容，覺得相較於之前的 8051，Arduino 把它包含在裡面還增加了一些功能，雖然有點複雜，但實際上用了還覺得更方便了，所以是一個不錯的軟硬體。

智慧型停車場的動作流程是由車子經過入口，七段顯示器增一位，手機的 App 顯示剩餘的車輛，當車子出去出口時，則七段顯示器則退一位，當下一位的使用者可以下載此停車場的 App，到停車場附近就可以偵測到目前的停車場的剩餘車輛，更方便的停車。

關鍵詞：Arduino、App Inventor、藍牙等等

目 錄

誌謝.....	I
摘要.....	II
目錄.....	III
表目錄.....	IV
圖目錄.....	V
壹、前言.....	1
一、製作動機.....	1
二、製作目的.....	1
三、製作架構.....	1
四、製作預期成效.....	3
貳、理論探討.....	4
參、專題製作.....	15
一、設備及器材.....	15
二、製作方法與步驟.....	16
三、專題製作.....	17
肆、製作成果.....	22
伍、結論與建議.....	25
一、結論.....	25
二、建議.....	25
參考文獻.....	26
附錄一 智慧型停車場程式碼.....	27

表目錄

表 2-1-1 藍芽功能	4
表 2-3-1 輸出腳位	11
表 2-3-2 Arduino 腳位功能	11
表 2-3-3 Arduino 輸出腳位	12
表 2-3-4 Arduino 輸入腳位	12
表 3-1-1 專題使用設備軟體	15
表 3-3-1 專題製作計畫書	17
表 3-3-2 智慧型停車場材料表	20

圖目錄

圖 1-3-1 專題製作流程圖	2
圖 2-1-1 藍芽模組 HC-05.....	5
圖 2-1-2 藍芽模組 HC-06.....	6
圖 2-2-1 App Inventor 2 主畫面	7
圖 2-2-2 App Inventor 2 撰寫程式介面	7
圖 2-2-3 App Inventor 2 專案選單	8
圖 2-2-4 App Inventor 2 連接選單	8
圖 2-2-5 App Inventor 2 打包選單	9
圖 2-2-6 App Inventor 2 輔助選單	9
圖 2-3-1 Arduino 主控版	10
圖 2-3-2 Arduino 下載頁面	12
圖 2-3-3 Arduino 開發環境的介面	13
圖 2-3-4 Arduino 功能介紹	14
圖 3-2-1 製作方法及步驟	16
圖 3-3-1 智慧型停車場電路圖	18
圖 3-3-2 運作流程圖	19
圖 4-1-1 電路板焊接過程	22
圖 4-1-2 檢測電路板	22
圖 4-1-3 電路板製作成品	22
圖 4-1-4 智慧停車場製作成品	22
圖 4-1-5 智慧停車場專題報告撰寫	22
圖 4-1-6 智慧停車場專題簡報撰寫	22
圖 4-1-7 智慧停車場成品圖	23
圖 4-1-8 手機 App 剩餘車位圖.....	24

壹、前言

一、製作動機

隨著時代越來越進步，現今擁有車的家庭越來越多了，人們出門都使用機車、機車作為代步工具，尤其是以高度開發的城市，如：台北、高雄等。停車的問題也隨之到來，民眾們在全家出遊時，常常會選擇開車出門，但是停車方面卻面臨了很多的阻礙，例如：在沒有規劃的停車場裡不容易找到停車位、以及找停車位會花費許多的時間、甚至是隨意在路邊停車而使道路狹窄，造成人民的不方便，而我們設計的專題是希望能夠幫助民眾在找停車位方面能更方便一些。

二、製作目的

如今停車方面的問題越來越嚴重，汽機車的密度越來越高，在台灣每平方公里汽車數量就有 190 幾輛之多，機車數量更高達 610 幾輛，再加上台灣土地面積較小相對而言停車這方面的問題更日趨重要，有許多民眾在出門遊玩或是觀光…等，因違規停車而被罰錢，造成路上行人通行不便情況也日趨增加，甚至許多人光是找停車位就需要花上 20 到 40 分鐘找停車格停車，有時候可能找了兩三個停車場也找不到停車格，使排放的廢氣汙染物大量增加。

所以我們希望藉由 Arduino 以及七段計數器來計算停車場剩餘的停車格並連接至我們所製作的 APP 中，使駕駛人能在遠處就清楚知道那些停車場還有剩餘位子來節省駕駛人停車所花費的時間，以及廢氣汙染量。

三、製作架構

(一) 專題製作流程

我們小組先一起討論專題方向，再去上網搜尋此專題的相關料，商量好題目並確定之後購買相關材料，才在麵包板上面進行模擬試，測試成功後再去製作成品，並書寫相關報告與資料，成品完成後檢查是否有錯誤，如有錯誤的地方在與老師進行商量如何改進。我們會再多準備一組備用零件，以防不時之需，若是失敗的話還有備用零件可以使用，還可以控管專題製作成本，不會花太多多餘的錢去購買料。

(二) 製作流程圖

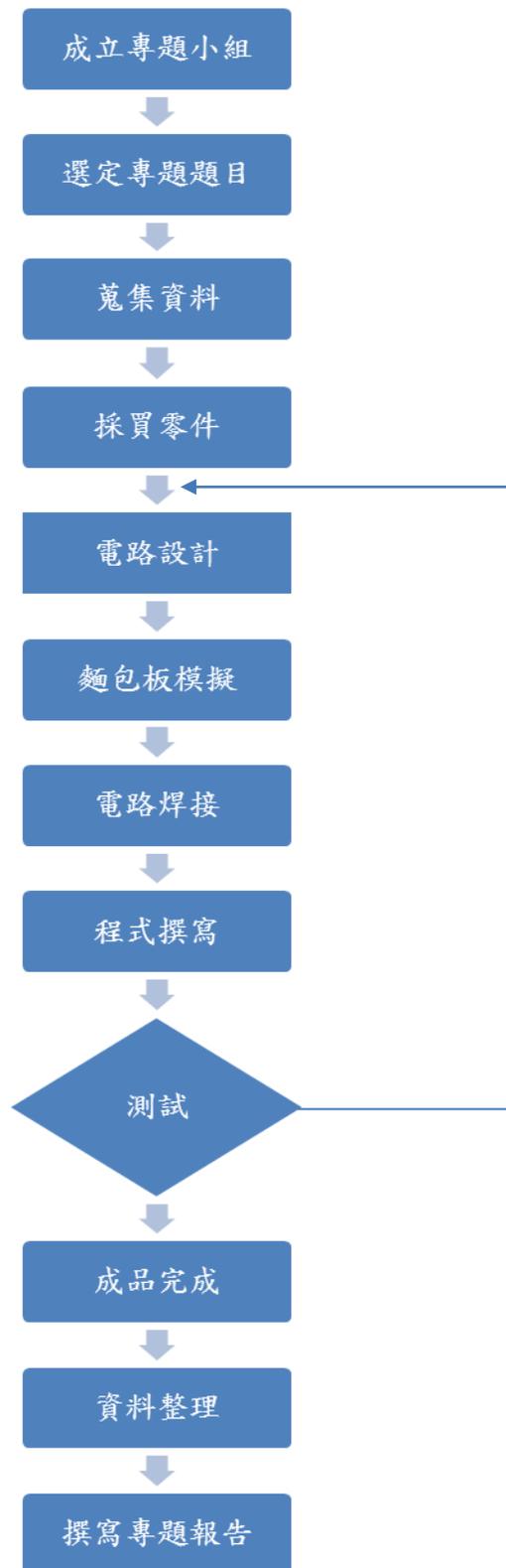


圖 1-3-1 智慧型停車場專題製作流程圖

四、製作預期成效

我們這一個小組雖然是第一次一起進行專題製作...。為此為這專題初步定義成效如下...

1. 藉由 Arduino 控制七段顯示器顯示停車場共有 9 個空的停車位，在設置一個顯示器來做計數，當車子進入停車場時，增加一個車位使用者。
2. 駕駛人下載此停車場所搭配的 APP，當登錄了 APP，Wi-Fi 會做連線，則 APP 會顯示此停車場剩餘多少停車位，當車輛進出時，APP 則會顯示增加或者減少，讓停車的人能快速地尋找車位。
3. 希望藉由此項設計來使駕駛人知道停車場的狀況，進而減少大家的時間，也不會造成佔車位的原因

貳、理論探討

接下來的是要介紹專題裡面有用到的軟體及硬體，裡面有包含藍芽、Arduino、App Inventor 2，藉此向大家介紹及更加認識以下的東西，以下是各個的功能說明及介紹。

一、藍牙

關於藍牙 (Bluetooth) 這個名字，倒是有個有趣的起源：西元十世紀時，挪威有個維京國王 Harald(哈拉德藍芽;Blatand 為丹麥文字，相當英文的 bluetooth) 將其新的無線電介面的 Project 命名為「藍牙」(Bluetooth)。

藍芽技術是一種小範圍的無線電頻率技術，裝置間透過晶片可互相溝通，不必再透過纜線傳輸。另外與藍芽對比的紅外線，其傳輸的距離短、且受方向限制，用過筆記型電腦的紅外線埠和印表機連線的人可能有過這樣的經驗：努力地調整筆記型電腦和印表機的角度（幾乎是面對面的情況）和位置（一公尺內的範圍）後，發現其慢無比，最後還是乖乖的找條線路連接上電腦。是不是資料的傳遞一定得要經過銅線呢？「藍芽」就是著眼於此，而開發出的無線傳輸技術，它的主要特點如下表 2-2-1 所示：

表 2-1-1 藍芽功能

1.操作頻段 2.4GHz 工業、科學、醫療頻段。可及範圍十公分到一百公尺。
2.使用跳頻展頻技術 (Frequency Hopping Spread Spectrum ; FHSS)。
3.最多可支援八個連接埠 (包含八個以上)。
4.傳輸方向不受限制，資訊由本身四面八方傳輸。
5.支援同步和非同步傳輸模式，易與 TCP/IP 整合成網路。

藍芽運作原理：

藍芽的運作原理是在 2.45GHz 的頻帶上傳輸作業，除了資料外，也可以傳送聲音。每個藍芽技術連接裝置都具有根據 IEEE 802 標準所制定的 48-bit 地址；可以一對一或一對多來連接，藍芽的傳輸範圍在 10 公尺(0dBm)到 100 公尺(20dBm)左右，採用每秒 1600 次跳頻展頻技術。在資料傳輸上，藍芽以 ACL(Asynchronous

Connection-Less) 的連線方式，提供最高下傳資料 723.2kbps 及上傳資料 57.6kbps 的非對稱性質傳輸速率或 433.9kbps 的對稱性質傳輸速率。在語音部分，以 SCO (Synchronous Connection-Oriented) 的連線方式，提供 64kbps 的音訊傳輸速率。此外，跳頻展頻在同步的情況下，發射與接收兩端以特定型式的窄頻電波來傳送訊號，為了避免在一特定頻段受其他雜訊干擾，收發兩端傳送資料經過一段極短的時間後，便同時切換到另一個頻段，由於不斷的切換頻段，因此較能減少在一個特定頻道受到的干擾，也不容易被竊聽或盜取。跳頻展頻所展開的訊號，可依特別設計來規避雜訊或重覆的頻道，並且跳頻訊號必須遵守 FCC (Federal Communications Commission) 的要求，使用 75 個以上的跳頻訊號，且跳頻至下一個頻率的最高時間間隔為 400ms，在 IEEE 802.11 中最大時間間隔通常定為 250ms，也就是每秒跳頻 4 次。而且，藍芽所發出的信號需低於 10 pico 瓦，而 GSM 手機啟動時會產生一至三瓦不等的發射功率，相較於手機，藍芽所造成的傷害降低許多。

藍芽模組：

基本的藍牙序列通訊模組沒有引出接腳，而是在印刷電路板的四周留下郵票般的齒孔，方便焊接。若不想動手焊接，可以選購附帶底板的模組。這個底板附帶直流電壓轉換 IC，方便連接 3.6V~6V 的電源。附帶底板的 HC-05 藍牙模組外觀與接腳如下：

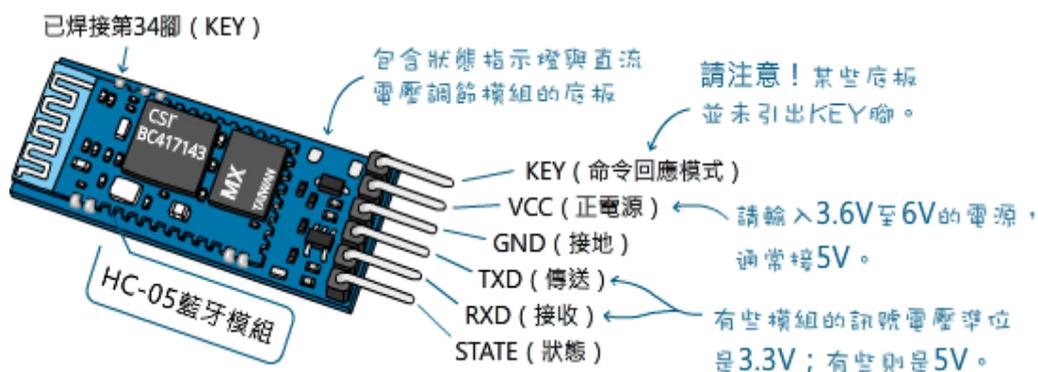


圖 2-1-1 藍牙模組 HC-05

「命令回應模式」接腳（通常標示成“KEY”或“EN”），用於啟動 AT 命令模

式，調整藍牙模組的設定（例如，調整序列埠的傳輸速率、修改模組的名稱、修改配對密碼...等等）。附帶底板的 HC-06 藍牙模組的外觀與接腳：

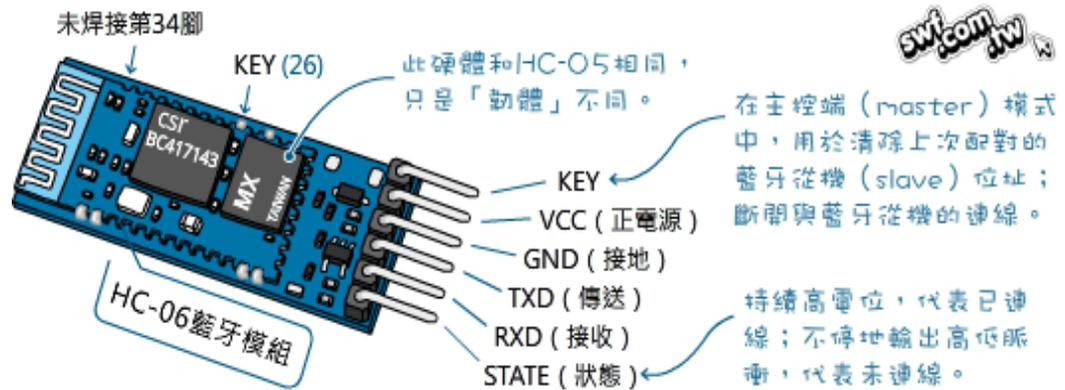


圖 2-1-2 藍牙模組 HC-06

實際上，因為模組生產廠商不同，並非所有附帶底板的 HC-05 和 HC-06 模組都像上圖一樣。有些模組只有 4 支引腳，在個人的應用和實驗場合，4 支引腳足數使用，只是不方便修改模組的參數，比較不建議購買。

二、APP Inventor 2

MIT 行動學習中心已發表 App Inventor 2 (本文後簡稱 AI2)，省略了需要使用 Java 才能開啟的 Blocks Editor，將其整合在網頁中即可使用，可說是與 Scratch 愈來愈像了。在操作上也將各指令藉由下拉式選單大幅簡化，使用者應該可以更快找到所需的指令。要差異如下：

1. 原有的 App Inventor 將更名為 App Inventor Classic，已於 2015 年 7 月關閉。
2. 程式設計概念不變，一樣是在 Designer 頁面初步決定好畫面元件配置之後，再到 Blocks Editor 新增指令來決定程式的行為。
3. App Inventor Classic 的原始檔格式為 .zip，AI2 則是 .aia，兩者不通用。您可使用 MIT 開發團隊的轉換網頁將 .zip 轉為 .aia，但可能會有一些轉換上的錯誤，但絕大多數狀況下都可以轉換。
4. Blocks 頁面可輸入正體中文，不會再出現可怕的框框了。
5. 請先安裝 MIT A12 Companion 才能進行無線同步。

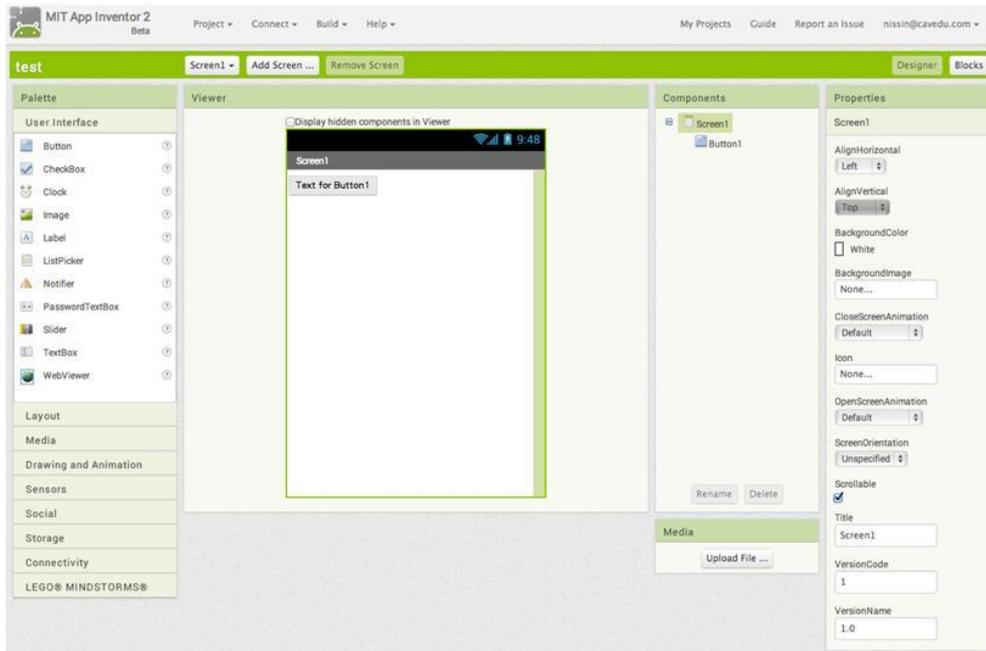


圖 2-2-1 App Inventor 2 主畫面

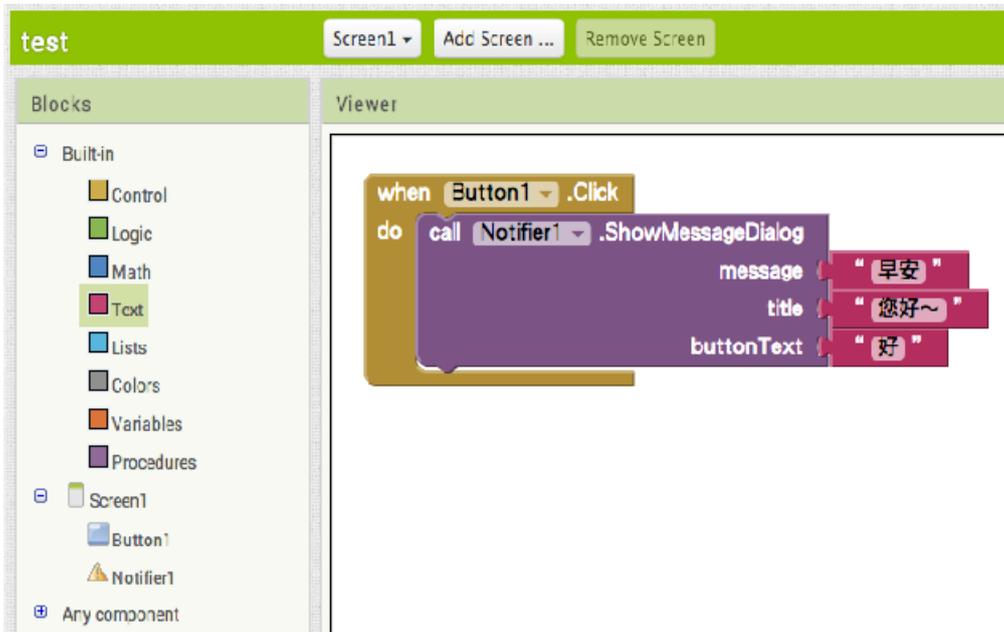


圖 2-2-2 App Inventor 2 撰寫程式介面

選單介紹

A. Projects 專案選單，在此新增、匯入專案。還有存檔與 KeyStore 相關功能。

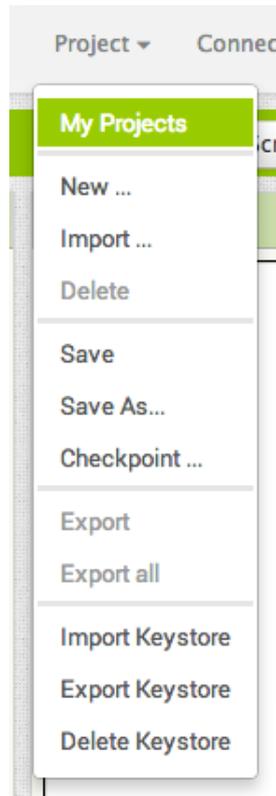


圖 2-2-3 App Inventor 2 專案選單

B. Connect 連結選單，在此您可選擇 AI Companion 將程式經由 WiFi 無線同步到您的手機；Emulator 模擬器以及 USB 將程式經由 USB 同步到您的手機（需安裝您手機專屬的 Android driver 或 sync 程式）。請注意在此都是同步，並非真的安裝程式，如果需要將程式安裝到實體手機中，請使用 Build 選單。

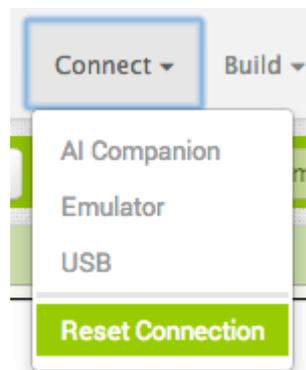


圖 2-2-4 App Inventor 2 連接選單

C. Build 打包選單，provide QR code 會在畫面上出現一個 QR code，您的手機掃描之後即可下載 .apk 安裝檔。或使用 save to my computer 將 .apk 安裝檔下載到您的電腦。

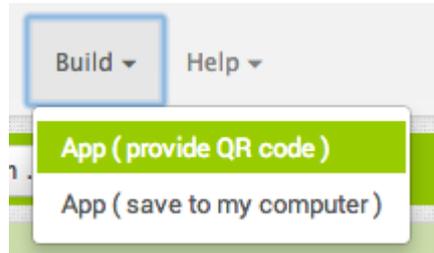


圖 2-2-5 App Inventor 2 打包選單

D. Help 輔助選單，相關教學資源都可在此找到。Library 指令說明，Get Started 新手上路、Tutorials 範例教學、Troubleshooting 問題排除以及 Forums 論壇等等，請多多利用唷。

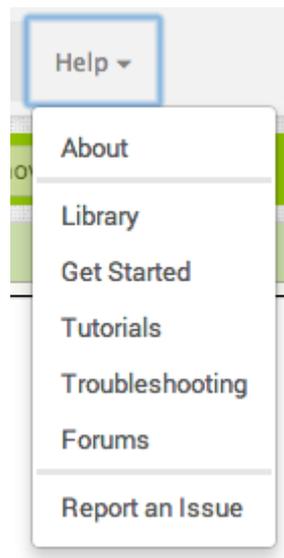


圖 2-2-6 App Inventor 2 輔助選單

三、Arduino



圖 2-3-1 Arduino 主控版

這是個智慧型手機流行的年代，大家都聽過 Android，但對於 Anduinot 卻充滿了陌生感。和 Android 相同的地方在於，兩者皆是使用開放原始碼的軟硬體平台。不同的是，Arduino 提供了簡單好用的 I/O 介面，並具可使用類似於 Java、C 語言等高階語言的開發環境。

Arduino 是一種開放授權的互動環境開發技術，互動裝置其實無所不在，像是冷氣的恆溫裝置，便是使用感測器偵測環境溫度，進行室內溫度的自動調節；還有汽車使用的倒車雷達，過於靠近物體便會發出聲音警告駕駛者。這些裝置為生活增加不少安全及便利，互動能帶給使用者驚喜，有時會以藝術品的形式呈現在生活當中。

主要可區分為三大類，分別是 Arduino 硬體、Arduino 軟體及 Arduino 擴充原件。硬體部分可購買現成的板子或是依據網路下載的電路圖自行組裝，出學者建議購買現成的，根據 Arduino 官方網站所寫，台灣地區具有官方合法授權的代理商為藝科資訊 Arobot Studio、PlayRobot 颯機器人/普特企業有限公司、iCshop (台灣)電子零件，電子材料，當然，還有別的商家，到網路上搜尋一下可找到不少，拍賣網站上也有，甚至也可以到國外網站去買。目前官方網站提供了許多不同的版子供開發者購買。

表 2-3-1 Arduino 輸出腳位

名稱	規格
微控制器	ATMEGA328
工作電壓	5V
輸入電壓 (推薦)	7-12V
輸入電壓 (限制)	6-20V
數字 I/O 接腳	14 (6 個提供 PWM 輸出)
模擬輸入接腳	6 支
EEPROM	1 KB (ATMEGA328)
震盪速度	16 MHz

1. 功能說明：

數位 I/O 接腳:14 支數位 I/O 接腳可以當作 input 使用，也可以當作 output 使用，使用方法是透過 pinMode(), digitalWrite(), and digitalRead() 這幾個函式。這 14 支數位 I/O 接腳，其中幾支腳有特殊的功能：

表 2-3-2 Arduino 腳位功能

Serial 通訊	0(RX) 和 1 (TX) 這兩支腳。用來接收(RX)與傳輸(TX) TTL 訊號的序列資料。這兩支腳也連接到 USB Converter 晶片中。
外部中斷	2 和 3 這兩支腳。這兩支腳可以利用外部事件觸發中斷。詳細內容請參考 attachInterrupt() 函式。
PWM	3, 5, 6, 9, 10 和 11 共六支腳。透過 analogWrite() 函式可以提供 8-bit 的 PWM 輸出。
SPI	10 (SS), 11 (MOSI), 12 (MISO) 和 13 (SCK) 這四支腳。這四支腳搭配 SPI Library 可提供 SPI 序列通訊。
LED	13。內建一顆 LED，當 pin 腳為 HIGH 時，LED 打開，當 pin 腳為 LOW 時，LED 關閉。

類比輸入腳位：

Arduino Uno 有 6 支類比輸入腳，標記為 A0 到 A5，每支腳都可提供 10 位元的解析 (即 1024 種不同的數值)。這些腳位所用的參考電壓預設為 0 到

5V，不過參考電壓也是可以更改的，方法是透過 AREF 腳和 analogReference()

表 2-3-3 Arduino 輸入腳位

I2C	4 (SDA) 和 5 (SCL) 這兩支腳。透過 Wire library 可以提供 I2C 通訊。
-----	---

其它：

表 2-3-4 Arduino 輸入腳位

AREF	類比輸入的參考電壓，搭配 analogReference() 函式一起使用。
Reset	當 Reset 腳為 LOW 時，微控制器會重置。

軟體介紹：

軟體取得：需要開發環境、編譯器、連結器、開發工具鏈、燒錄程式、板子與電腦連接所需要的驅動程式、等等等等，這些通通都已經準備好、打包好了，到 Arduino 官方網站的下載頁面 <http://arduino.cc/en/Main/Software> 即可下載，可在 Windows、Mac OS X、Linux 上運行。

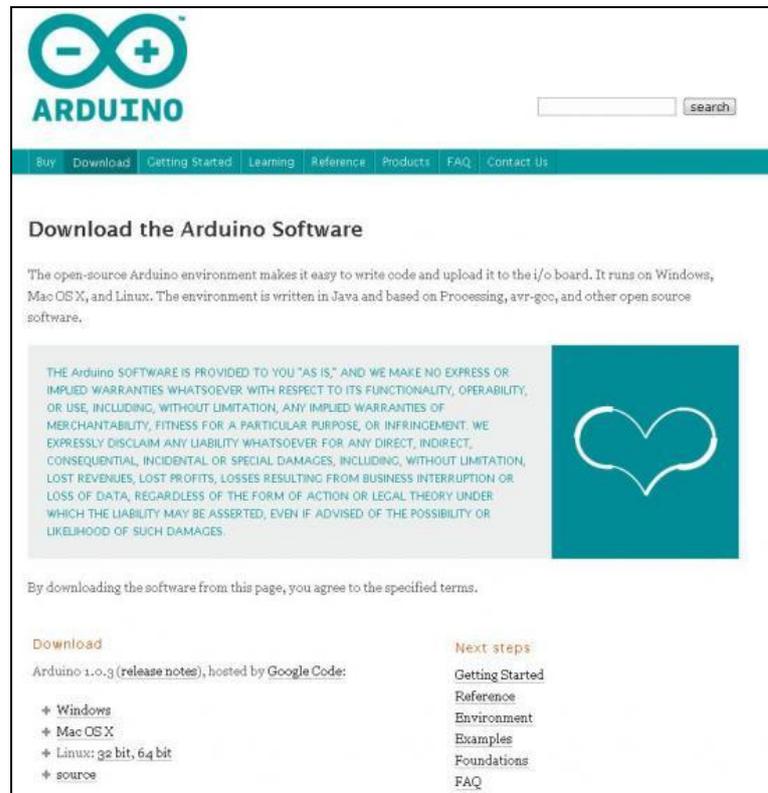


圖 2-3-2 Arduino 下載頁面

2. 開發介紹：

Arduino 的軟體開發環境是開放源碼，可以在 Arduino 官網免費下載，它所用的程式語言語法類似於 C/C++，具備文字編輯介面、常用工具欄、圖形化控制介面及錯誤編輯器，運用這個工具進行城市的上傳，及各項感應器的溝通。

3. 編譯介紹：

使用的 Arduino 編寫的軟件被稱為 Sketch(腳本)。這些的腳本都寫在本文的編輯器。腳本名稱就是檔案名稱。上排圖形提供了新增、上傳、除錯及腳板切換介面，而 console 介面紀錄了完整的執行訊息，更可藉由此介面監視 Arduino I/O 數值。

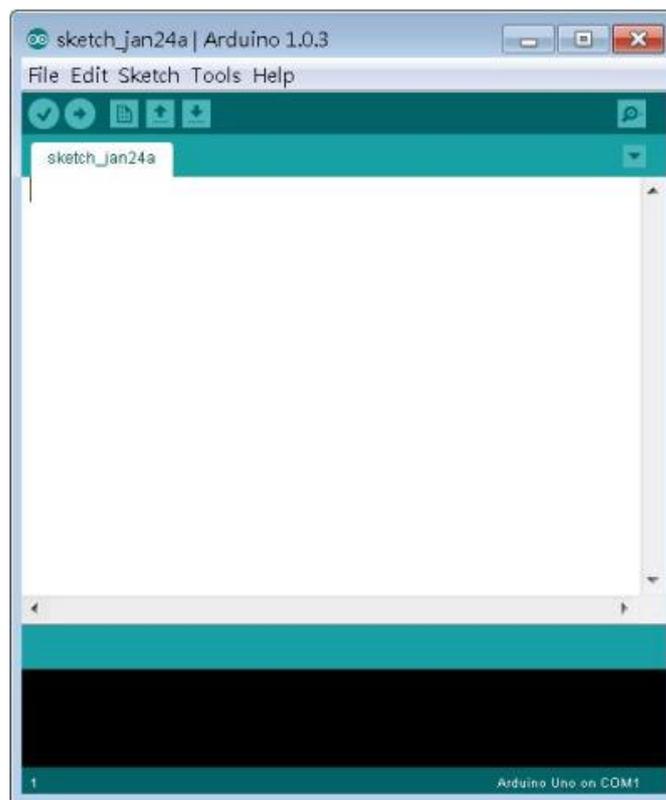


圖 2-3-3 Arduino 開發環境的介面



圖 2-3-4 Arduino 功能介紹

參、專題製作

一、設備及材料

表 3-1-1 專題使用設備軟體

儀器(軟體)設備名稱	應用說明
個人電腦	專題報告電路圖製作及進行專題成品電路測試
App Inventor	撰寫用於控制藍芽的程式
MicroSoft Office Word	專題報告 製作過程的撰寫
MicroSoft Office Power Point	進行口頭報告製作及專題成品告報呈現
Arduino Uno板	寫在Arduino Uno板對電路板做連接達成專題成果
手機	專題製作過程的拍攝

二、製作方法及步驟

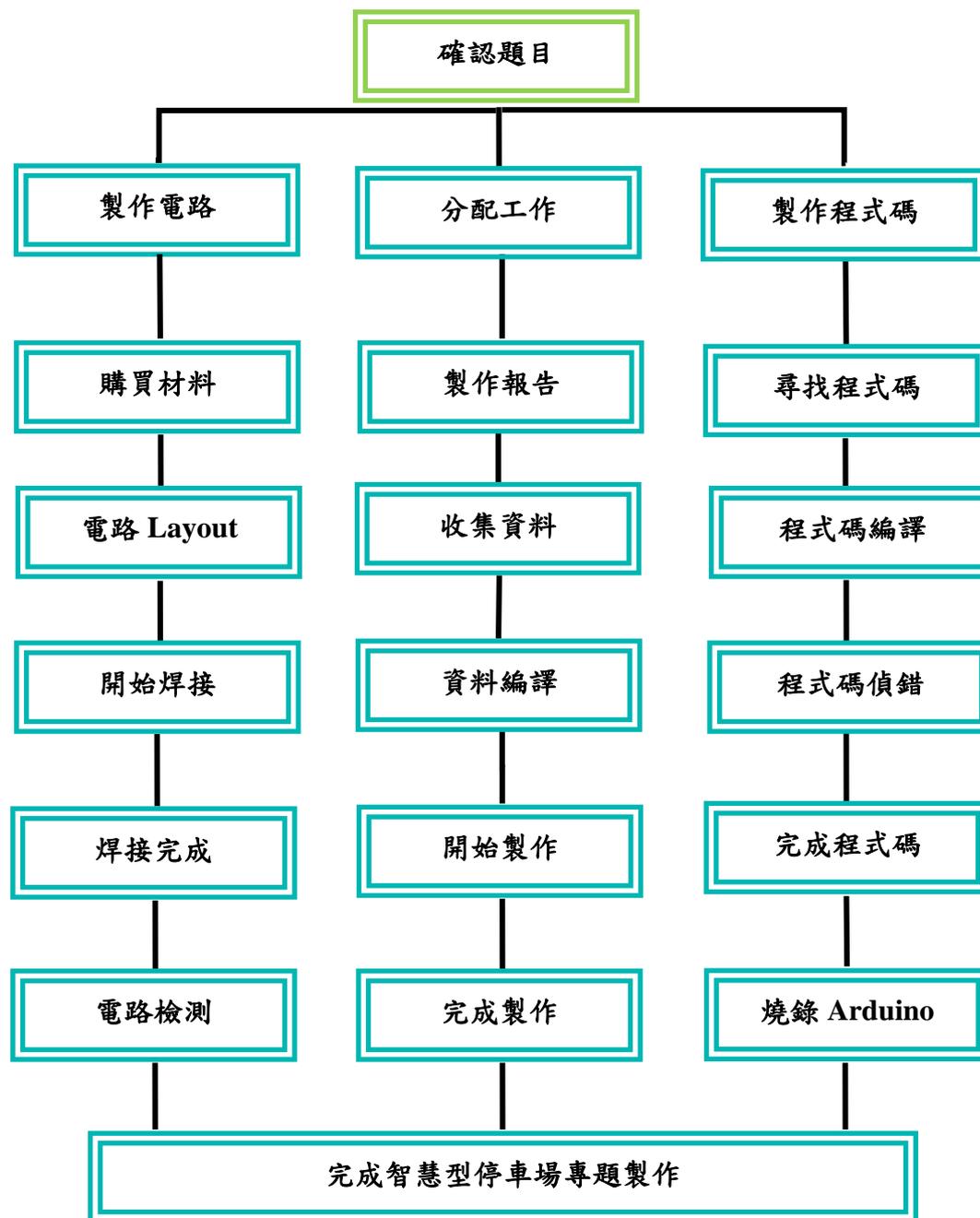


圖3-2-1 製作方法及步驟

三、專題製作

表 3-3-1 專題製作計畫書

專題型別		<input type="checkbox"/> 個人型專題	<input checked="" type="checkbox"/> 團隊型專題
專題性質		利用8051單晶片來驅動升降台及感測	
科別/年級		資訊科三年級	
專題名稱	中文名稱	智慧型停車場	
	英文名稱	Smart Parking	
專題內容簡述		<p>相較於之前學長姐用的 8051，我們選擇 Arduino 來製</p> <p>做智慧型停車場，裡面的程式用了Arduino、App inventor</p> <p>，硬體則是Arduino Uno，用App inventor寫程式給藍芽作</p> <p>連線，搭配著Arduino程式對電路版的操作，並且以手機</p> <p>的App來確認，此停車位的車位輛，才能快速的完成，便</p> <p>利的停車。</p>	
指導老師姓名		葉忠賢 老師	
參與同學姓名		洪得億 (資訊科3-2)	王柏翔 (資訊科3-2)
		黃意寶 (資訊科3-2)	翁聖凱 (資訊科3-2)
專題執行日期		104年 09 月 01 日至 105 年 05 月 31 日	

(一)硬體電路圖

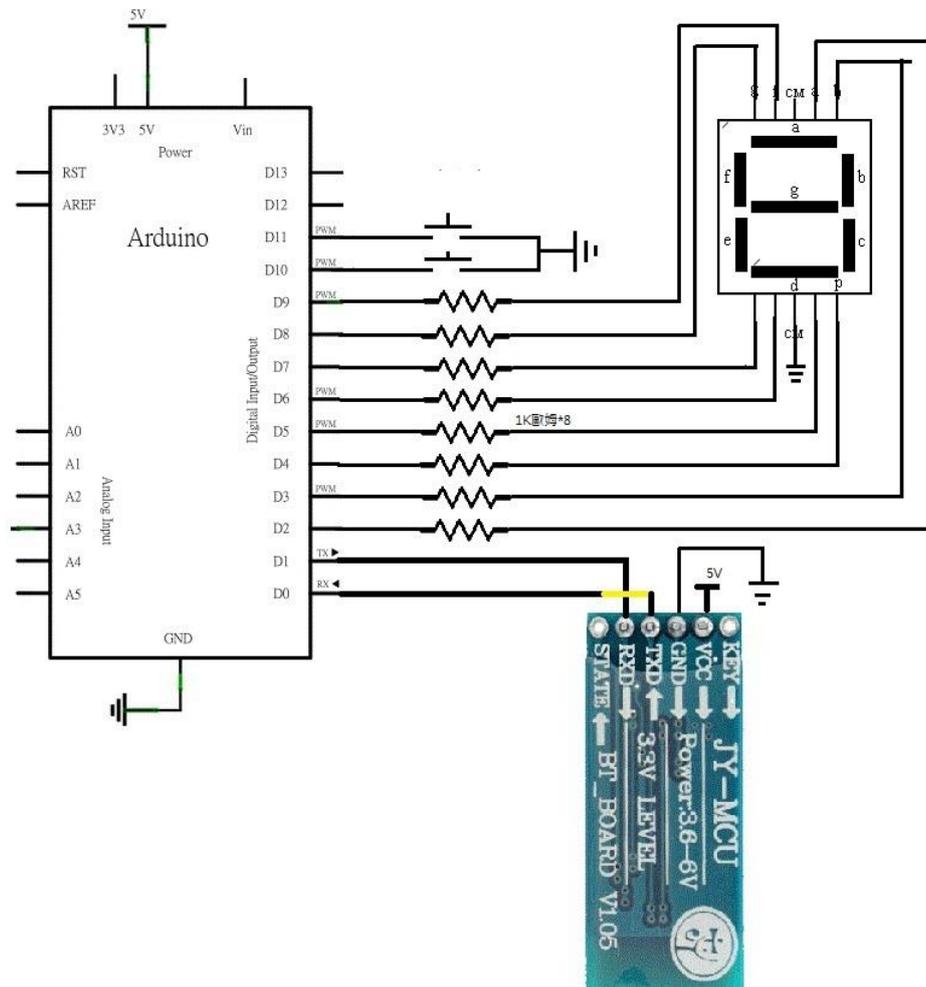


圖 3-3-1 智慧型停車場電路圖

(二)智慧停車場動作之流程圖

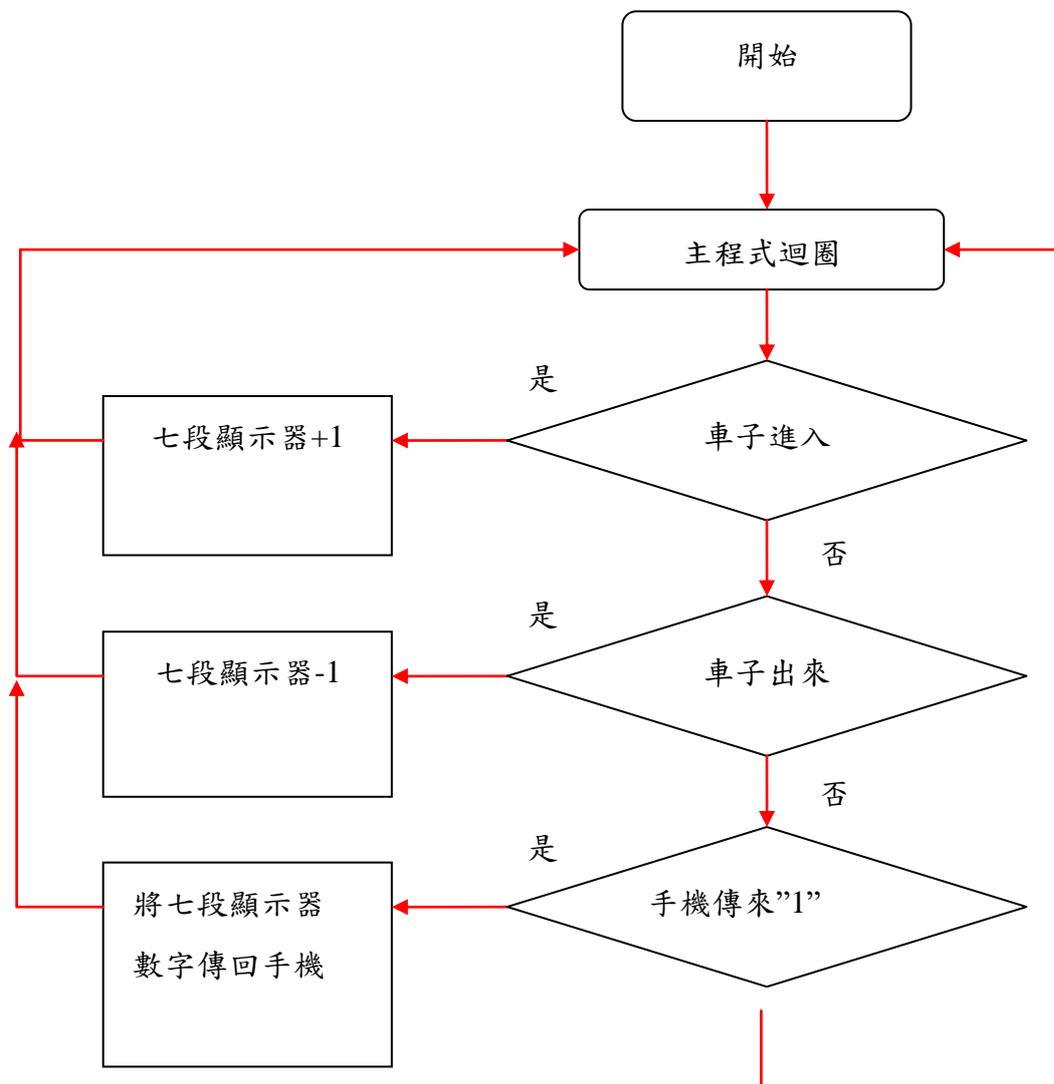


圖 3-3-2 運作流程圖

表3-3-2 智慧型停車場材料表

項目	材料名稱	規格	數量
1	按鈕	無段開關	2
2	圓孔針腳	40PIN	1
3	Arduino	UNO	1
4	杜邦線	公、母	13
5	藍芽	HC-06	1
6	七段顯示器	一位數共陽極	1
7	電阻	1kΩ	7

(三) 專題製作成員工作分配及進度安排

專題製作成員： 洪得億、翁聖凱、王柏翔、黃意寶

前置作業進度： 資料收集，分析討論，決定題目，共約六週

預計完成再需時間：約四個月(16 週)

進度 1 → 硬體製作 (王柏翔、洪得億)

1. 研究電路圖： 二週
2. 搜尋，購買材料： 二週
3. 焊接電路板： 二週
4. 檢查焊線及零件焊接是否正確： 二週
5. 整合軟體 Debug： 五週
6. 整合測試： 三週

進度 2 → 軟體製作 (王柏翔)

1. 研究程式資料： 三週
2. Key in 程式碼： 三週
3. 程式碼加註解： 三週
4. 檢查程式碼及註解是否正確： 一週
5. 編譯及燒錄 Arduino 程式： 一週
6. 整合硬體 Debug： 二週

7.整合測試：三週

進度3 → 報告書編撰(翁聖凱、黃意寶)

1.研究電路圖及程式資料：四週

2.搜尋參考技術文件：二週

3.報告書相關內容打字：四週

4.整合編輯相關文件：三週

5.檢查報告書內容是否正確：二週

6.報告書列印編冊：一週

肆、製作成果

我們小組先決定了題目，麵包板模擬電路、測試與預期的結果相符、進而焊製電路板成品、撰寫專題報告的紀錄，我們將這些過程用手機照相來完成對專題的整個流程，以下是我們的呈現：



圖 4-1-1 電路板焊接過程



圖 4-1-2 檢測電路板

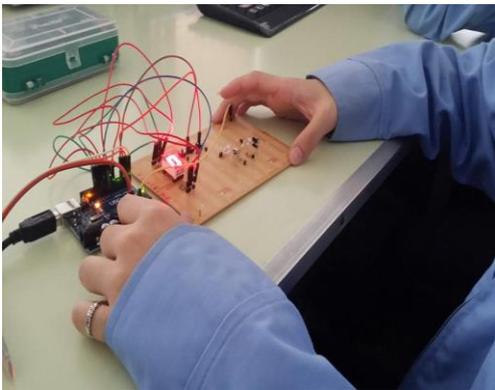


圖 4-1-3 電路板製作成品

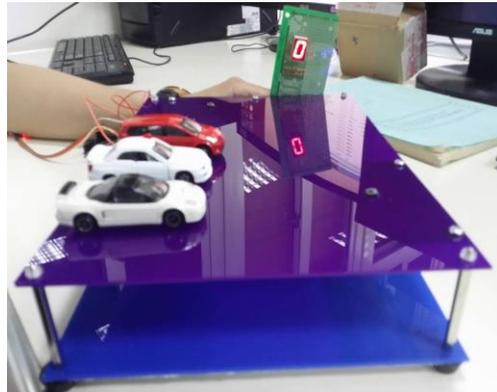


圖 4-1-4 智慧停車場製作成品



圖 4-1-5 智慧停車場專題報告撰寫



圖 4-1-6 智慧停車場專題簡報撰寫

操作方法:



圖 4-1-7 智慧停車場成品圖

入口

出口

車位剩餘車輛

當遠方車子要在此停車場停車時，可以下載停車場的 App，當車子進入入口時七段顯示器近位+1，手機的 App 則顯示剩餘車位為 8，到出口時，七段顯示器則退位-1，手機的 App 顯示剩餘車位為 9。

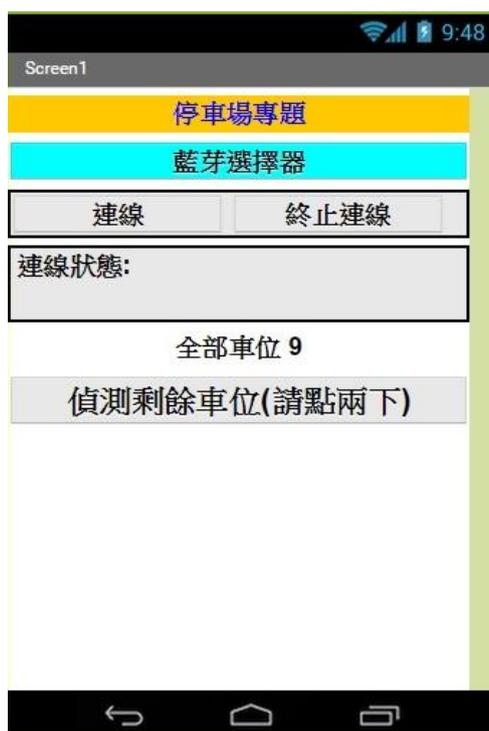


圖 4-1-8 手機 App 剩餘車位圖

這是圖 4-1-2 手機停車場的剩餘車位的 App，下載之後要先選擇藍芽作連線，連線完成後連線狀態顯示 Connction Sun Ceed，全部車位會在感應車子後作進位或退位，偵測剩餘車位時，需要點兩次按鈕顯示剩餘車位，讓駕駛人更清楚的知道車位的剩餘量。

伍、結論與建議

一、結論

專題的製作到了後面，我們這一組統整了一些對自己或對其他人有所幫助的一些成果與過程中的發現，加以告訴大家。

1. 這個專題可以讓我們不是只會照著課本上的電路來做出老師所需要的結果，而是由組長與他的組員，共同來創造、思考、研究，然後做出理想的結果。
2. 專題上的原理、應用，也可以在所學的知識上加以使用，並不是白學。
3. 透過組隊的小組也能增進同儕間的團隊合作，以不同人的思考，提出個人的意見，加以總和做出所要的專題。
4. 製作的過程也會有不能理解的地方，可以向其他師長、同儕請問過後，了解為何不能動作的原因，增加問題的解決方法，而不是不明白就放棄。
5. 電路的製作不同以往，小組構想的專題，Laout 所需要的電路，進而提升對電路的思考與焊接的功力。
6. 專題的成果在發表後，可以介紹功能給學弟妹參考，以便於三年級發表，一個能思考的地方，而不是毫無頭緒的尋找。
7. 藉由上述這些，我們可以在上大學之後比較知道，自己對於這個專題的實用度與可看性，提早對專題的認識，而不至於沒有方向。

二、建議

我們可以透過 Wi-Fi 在遠處的地方就可以用手機來確認停車場的車位，因為藍牙的距離有限，所以我們可以利用這個方法來達到最大效益。

也可以利用 QR CODE 來確認自己的車位在哪裡，不會說忘記了就找了半天還是找不到，這樣一來也可以明確的知道車位的放置處。

車位更精準的偵測，我們可以利用 Double Cheak，在入口與出口的地方設置橫桿，另外再裡面得停車格在設置一個感測器，這樣一來就不會有其他的事物而導致於誤判結果。

參考文獻

- (1) 王安邦，中文版 MIT App Inventor 2，上奇資訊
- (2) 施士文，ARDUINO 微電腦應用實習，台科大圖書
- (3) 黃宏彥、余文俊、楊國輝編著，感測器原理與應用電路實習，高立圖書有限公司，1999
- (4) 蔡朝陽，電工實習（四），全華科技圖書股份有限公司，中華民國八十三年五月
- (5) 林明德，2014，專題製作-電子電路篇，台灣：台科大
- (6) 張駿榮、盧聰勇，最簡單的互動設計 Arduino 一試就上手，基峰資訊
- (7) 趙英傑，超圖解 Arduino 互動設計入門，旗標出版社
- (8) 曾吉弘、CAVE 教育團隊，LabVIEW for Arduino：控制與應用的完美結合，馥林文化
- (9) 王冠勛、陳錦輝，Arduino 完全實戰手冊，博碩
- (10) 小林茂，Prototyping Lab，馥林文化

附錄一 智慧型停車場程式碼

1. App Inventor2:

```
initialize global vbluedevice to ""

when 藍芽選擇器 .BeforePicking
do call 藍牙客戶端1 .Disconnect
set 藍芽選擇器 . Elements to 藍牙客戶端1 . AddressesAndNames

when 藍芽選擇器 .AfterPicking
do set global vbluedevice to 藍芽選擇器 . Selection
set 藍芽選擇器 . Text to get global vbluedevice
set 標籤3 . Text to "press connect button to connection"

when 連線 .Click
do set 計時器1 . TimerEnabled to false
if length get global vbluedevice > 0
then if call 藍牙客戶端1 .Connect
address get global vbluedevice
then set 藍芽選擇器 . Text to get global vbluedevice
set 計時器1 . TimerEnabled to true
call 對話框1 .ShowAlert
notice "connection sun ceed !"
else call 對話框1 .ShowAlert
notice "NO device connect ted !"
```

```
when 終止連線 - .Click
do
  set 計時器1 - .TimerEnabled to false
  call 對話框1 - .ShowAlert
    notice "client stop connection!"
  call 藍牙客戶端1 - .SendText
    text "client stop connection!"
  call 藍牙客戶端1 - .Disconnect
  set 標籤3 - .Text to "client stop connection!"
```

```
when abc - .Click
do
  call 藍牙客戶端1 - .SendText
    text "a"
```

```
when 計時器1 - .Timer
do
  if 藍牙客戶端1 - .IsConnected
  then
    if call 藍牙客戶端1 - .BytesAvailableToReceive > 0
    then
      set 標籤6 - .Text to call 藍牙客戶端1 - .ReceiveText
        numberOfBytes: call 藍牙客戶端1 - .BytesAvailableToReceive
      if compare texts 標籤6 - .Text = " sever STOP "
      then
        set 標籤6 - .Text to " sever STOP "
    else
      call 對話框1 - .ShowAlert
        notice "沒收到"
      set 標籤6 - .Text to "沒收到"
      set 計時器1 - .TimerEnabled to false
```

2. Arduino:

```
#include <SoftwareSerial.h>
SoftwareSerial I2CBT(11,10);
//定義變數
int BluetoothData;
int a=2;           //seg_a接到D2
int b=3;           //seg_b接到D3
int c=4;           //seg_c接到D4
int d=5;           //seg_d接到D5
int e=6;           //seg_e接到D6
int f=7;           //seg_f接到D7
int g=8;           //seg_g接到D8
int buttonpin = 11 ;
int buttonstatus ;
int but = 10 ;
int butt ;
byte seven_seg[10][7] = { { 1,1,1,1,1,0 }, // = 0
                           { 0,1,1,0,0,0,0 }, // = 1
                           { 1,1,0,1,1,0,1 }, // = 2
                           { 1,1,1,1,0,0,1 }, // = 3
                           { 0,1,1,0,0,1,1 }, // = 4
                           { 1,0,1,1,0,1,1 }, // = 5
                           { 1,0,1,1,1,1,1 }, // = 6
                           { 1,1,1,0,0,0,0 }, // = 7
                           { 1,1,1,1,1,1,1 }, // = 8
                           { 1,1,1,1,0,1,1 } // = 9
                           };
```

```

void setup() {
    Serial.begin(9600);
    buttonstatus =digitalRead(buttonpin);
    butt =digitalRead(but);
    pinMode(a, OUTPUT);
    pinMode(b, OUTPUT);
    pinMode(c, OUTPUT);
    pinMode(d, OUTPUT);
    pinMode(e, OUTPUT);
    pinMode(f, OUTPUT);
    pinMode(g, OUTPUT);
    byte digit = 0;
    CodeWrite(digit);
}
byte digit = 0;

void loop() {
    buttonstatus =digitalRead(buttonpin);
    butt =digitalRead(but);
    if (Serial.available()){
        if(BluetoothData=='a' && digit <1 ){
            Serial.println("9 ");
        }
        if(BluetoothData=='a' && digit > 0 && digit < 2 ){
            Serial.println("8");
        }
        if(BluetoothData=='a' && digit > 1 && digit < 3){
            Serial.println("7");
        }
    }
}

```

```

    if(BluetoothData=='a' && digit > 2 && digit < 4 ){
        Serial.println("6");
    }
    if(BluetoothData=='a' && digit > 3 && digit < 5 ){
        Serial.println("5");
    }
    if(BluetoothData=='a' && digit > 4 && digit < 6 ){
        Serial.println("4");
    }
    if(BluetoothData=='a' && digit > 5 && digit < 7 ){
        Serial.println("3");
    }
    if(BluetoothData=='a' && digit > 6 && digit < 8 ){
        Serial.println("2");
    }
    if(BluetoothData=='a' && digit > 7 && digit < 9 ){
        Serial.println("1");
    }
    if(BluetoothData=='a' && digit > 8 && digit < 10 ){
        Serial.println("0");
    }
    delay(100);
}
    BluetoothData=Serial.read();
if (buttonstatus == HIGH && digit <9.1) {
    digit++;
    CodeWrite( digit);
}
if (butt == HIGH && digit >0){
    digit-- ;
}

```

```
CodeWrite(digit);  
}  
// 暫停 2 秒鐘  
delay(1000);  
  
// 在七段顯示器上顯示指定的一個數字  
}  
  
void CodeWrite(byte digit) {  
byte pin = 2;  
for (byte seg = 0; seg < 7; ++seg) {  
digitalWrite(pin, seven_seg[digit][seg]);  
++pin;  
}  
}
```

End