

運用智慧型手機全球定位系統設計安全帽抬頭速度顯示裝置

赫弋翔、余政杰*

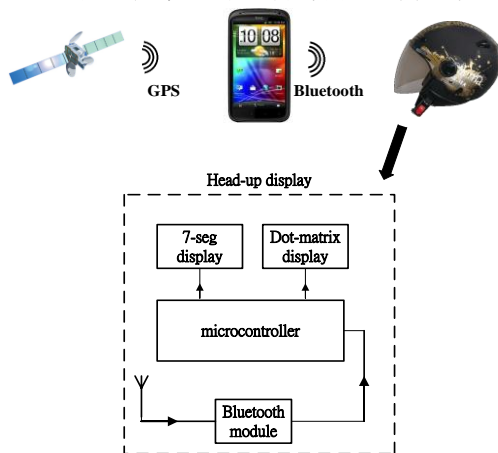
國立臺北科技大學電腦與通訊研究所

摘要 本論文主要目的為建立機車使用的抬頭顯示安全帽，整體的架構是利用持有率日趨普及的智慧型手機全球定位系統 GPS (Global Positioning System) 進行測速、指示方向以減低裝置的成本，將資料傳送到利用微控制器整合的抬頭顯示裝置上，兩者之間的溝通方式為藍牙。研究主要分為兩個部份，一是 Android 手機應用程式的撰寫，主要為 GPS 和 Bluetooth API 的應用，並探討影響 GPS 測速準確性的各種因素。二是微控制器整合的抬頭顯示裝置部分，使用低階 8 位元 PIC16F887 微控制器作接收處理。
關鍵字—抬頭顯示器、全球定位系統、藍牙。

一、簡介

抬頭顯示器(Head-up Display)是早期為了航空機所發明的飛行輔助儀器[1]，而近幾年抬頭顯示器也開始在汽車上出現，現今市面上的新車都普遍配有抬頭顯示器，儼然已經是汽車的基本配備。然而現在在台灣的市面上還沒有看到專門為機車族群來設計的抬頭顯示安全帽產品，有句話說汽車是鐵包肉，機車是肉包鐵，對於增加駕駛安全的重要性，機車遠比汽車來的高許多，所以國外已經有不少家安全帽公司有致力於抬頭顯示安全帽的開發，並且非常看好他未來的市場銷售潛力。近年來政府放寬對 250 cc 以上重型機車可以在快速道路上行駛，如此更提升了抬頭顯示器對機車駕駛安全的重要性。

現今智慧型手機普及率越來越高，且 Android 採取開放源碼的策略大大降低了手機應用程式開發的門檻，所以本文的發想架構如圖一，利用智慧型手機內含的 GPS 功能，撰寫一個測速與方向的應用程式，將資料透過藍牙傳送到一個使用 Microchip 公司的 PIC 微控制器整合的抬頭顯示裝置，這個裝置安裝於機車安全帽上，將測速的結果透過顯示器讓駕駛人能夠在行駛途中隨時查看。



圖一：抬頭顯示安全帽架構方塊圖

二、研究背景

國外有幾家安全帽公司發表了功能類似的安全帽產品，本文蒐集了兩款具代表性的產品與其作比較。一為 Sportvue 公司 2005 年 7 月所發表的 MC1 安全帽抬頭顯示器，分享於 webBike World 摩托車愛好者分享園地 [2]。MC1 這項產品雖在功能上與本論文提出的相似，但是在實現的手法上卻有很大的不同。此安全帽運作的原理是在機車本身加裝一個無線的訊號發送器，時速的訊號源則截取機車儀表板的訊號線，這項產品的缺點就是安裝十分不容易，使用者若想要自行安裝，需要對機車的結構相當熟悉，優點則是時速是與儀表板同步的，所以沒有加速度太大影響資料更新速率的問題，或是受到 GPS 使用環境的限制。

另一款則是 POPCI 公司於 2013 年六月 19 發表的一款具有時速抬頭顯示和地圖導航功能的新式安全帽 Livemap，這款安全帽是使用 GPS 來進行測速和地圖定位，使用的是最新的透明顯示器，顯示器平時顯示行車的速度、行駛方向箭頭和時間等不太會遮蔽視線的資訊畫面，功能相當先進且繁多詳細介紹如[3]，是目前可以找到最先進的一款抬頭顯示安全帽產品。表一將本論文所提出之抬頭顯示器與上述兩款抬頭顯示器做比較。

表 I
安全帽抬頭顯示器比較

	本文	Sportvue MC1	POPCI LiveMap
發表日期	2013 年 7 月	2005 年	2013 年 6 月
測速方式	GPS	儀表板訊號	GPS
顯示方式	七段顯示器、 點陣顯示器	七段顯示器	透明顯示器
售價	NT 400 (硬體成本)	USD 329	USD 2000 (含安全帽)
安裝複雜	低	高	低

表一中可以看到本文使用智慧型手機達成的安全帽抬頭顯示裝置成本比起其他兩家公司的產品，在售價低上非常多，主要是因為大部分較為昂貴的硬體設備都使用智慧型手機內部原有的硬體，所以在裝置本身的硬體成本是很低廉的，且使用者不需要懂得機車儀表板的結構便可以輕鬆安裝。但相較起來，缺點就是使用的時間過長，會影響了手機電量的使用，這項缺點會隨著智慧型手機逐步往高電量電池發展的趨勢而降低其影響力。

三、 手機應用程式

3.1 Andriod 簡介

Android 是一個由 Google 基於 Linux 核心(Kernel)開發的手持設備作業系統，以佔有率 32.9 % 在全球手持裝置市場中排行第一。Andriod 系統架構可以分為五層，由上到下依序是應用程式層、應用程式框架層、Android 運行環境、系統函式庫層、Linux 核心層。

Android 應用程式的開發主要包含了使用者界面的布置與功能 API (Application Programming Interface) 的使用，圖二為本文應用程式的使用者介面，所使用到的功能為 GPS 定位功能與短距離的藍芽傳輸，分別為 LocationManager API 與 Bluetooth API 兩項。



圖二：應用程式使用者介面

3.2 手機GPS 測速功能

3.2.1 GPS 簡介

全球定位系統為美國國防部建置及維護，整個系統由 24 顆以上的訊號衛星、地面上 1 個主控站、4 個數據注入站和 6 個監測站及作為用戶端的 GPS 接收機所組成，分佈在離地面 20,200 公里各與赤道呈 55 度夾角的六個軌道，並不間斷的發出頻段為 1.575GHz 帶有衛星訊息的訊號。

3.2.2 LocationManager API

在主程式中匯入 GPS 必要的函式庫「LocationListener」，接著宣告 Activity 繼承這個類別的功能，接著 Eclipse 開發環境會直接建立「LocationListener」介面所需的四個方法：「onLocationChanged」、「onProviderDisable」、「onProviderEnable」、「onStatusChanged」。其中每當地點改變的時候，都會呼叫這個「onLocationChanged」的副程式，在 GPS 測速應用程式中主要修改的對象就是他。達成 GPS 測速首先在生命週期中的「onStart」方法中添加程式碼：

```
「mgr = (LocationManager) getSystemService
(Context.LOCATION_SERVICE)」
```

旨在向系統取得定位服務，接者設定理想的更新時間與最短更新距離：

```
「mgr.requestLocationUpdates("gps",500,0,this);」
```

因為目的是在測速，所以更新速率的要求比較高，設定 500 ms 為其最短更新時間，且將最短更新距離設為 0 m，確保有位移產生就讓 GPS 更新移動速度的資訊。

接著在「onLocationChanged」的部分當中，添加程式碼「location.getSpeed()」用來取得移動速度，單位是 meter/sec，乘上 3.6 換算成台灣所習慣的單位 km/hr，和程式碼「location.getBearing()」取得方向，資料的單位為度，以正北方為 0 度順時針到 360 度又回到北方的格式，接著將度數資料判定為方向，依序分為北方、東北、東、東南、南、西南、西、西北。

3.3 手機藍芽傳輸

3.3.1 藍芽簡介

藍芽是由 5 家公司聯合制定的近距離無線通信技術標準，其目的是實現最高資料傳輸速率 1 Mbps、最大傳輸距離為 10 m 的無線通信。工作在全球通用的 2.4 GHz ~ 2.48 GHz ISM 頻段的一個低耗電技術在行動電話及其附件之間探求一種新的低功耗、低成本的介面，建立一種新的連接方式，希望能去除連接行動電話與耳機、筆記本電腦及其他設備之間繁雜的線纜。設計協定和通訊協定堆疊的主要原則是盡可能的利用現有的各種高層協定，保證現有協定與藍芽技術的融合以及各種應用之間的互通性，充分利用相容藍芽技術標準的軟硬體系統。

3.3.2 Bluetooth API

藍芽設備分為兩個端點，一個為主動發起連線請求的 Client 端，一為利用 BluetoothServerSocket 監聽外來配對的處於被動的 Server 端。在生命週期中的「onCreate()」建立一個 BluetoothAdapter，在 Google 官方文件中[4],[5]，描述它是一個區域藍芽接口，也就是所有藍芽交互活動的一個起始點，用來偵測區域內的藍芽裝置、查詢配對過的藍芽列表、用 MAC 地址建立一個 BluetoothDevice 實體或是建立一個 BluetoothServerSocket 來監聽是否有其它藍芽裝置傳送訊息到本機，程式內有三個執行緒。

1. AcceptThread

此執行緒的工作為 Server 端監聽外來的連線要求，本文使用的藍芽模組內建這樣的功能，所以讓手機扮演發起配對的腳色，省略了這個等待配對的執行緒，以減少應用程式的複雜度。

2. ConnectThread

這個執行緒就是 Client 端的發起連線工作，在使用配對裝置按鈕向 mBluetoothAdapter 取得了過去以配對藍芽裝置的 MAC 位置後，點開下拉式選單來進行選取，接著按下連接按鈕執行 ConnectThread 建立連接取得 BluetoothSocket。

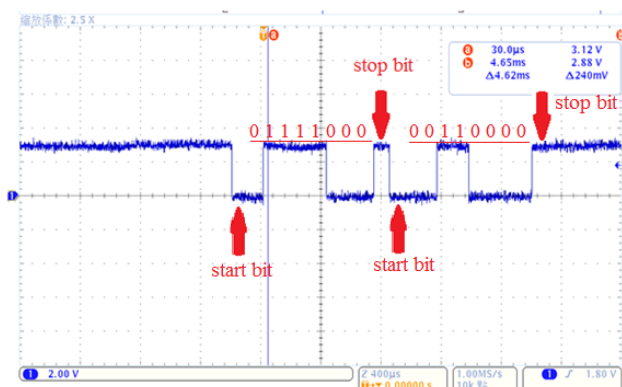
3. ConnectedThread

在監聽連線建立後 BluetoothSocket 透過 InputStream 和 OutputStream 進行資料交換，就是程式當中的 Write() 和 Read() 副程式。

3.3.3 傳送速度與方向封包

當藍芽連線建立並且 GPS 定位完畢後，圖三中的程式碼

宣告兩筆資料的長度為 8 位元的 byte，並且放到 setCmmd 矩陣封包中，每當 GPS 取得新的速度與方向就更新矩陣資料，再利用 Bluetoothsocket 將封包傳送出去，所以封包長度為兩筆 8 Bits 資料加上各 2 個 Start 和 Stop 位元總共 20 個位元長度的封包，如圖三。



圖三：速度與方向資料封包

四、 抬頭顯示裝置

抬頭顯示裝置處理資料的部分為一個 PIC16F887 微控制器，負責接收資料的是型號為 HC-05 的藍牙模組，顯示的部分則為七段顯示器與點陣顯示器。透過藍牙模組接收手機傳送的資料封包在透過微控制器做處理在由兩個顯示器做顯示。

4.1 裝置安裝位址

抬頭顯示裝置實體如圖四， [6]-[10]參考文獻中提到大部分的汽車駕駛者判定路況的視線大多集中於擋風玻璃的中間，而判定行車狀態的視線則往下延伸到儀表板上，所以較佳的抬頭顯示器安裝位置為此兩區域的中間也就是安全帽擋風鏡的正下方。

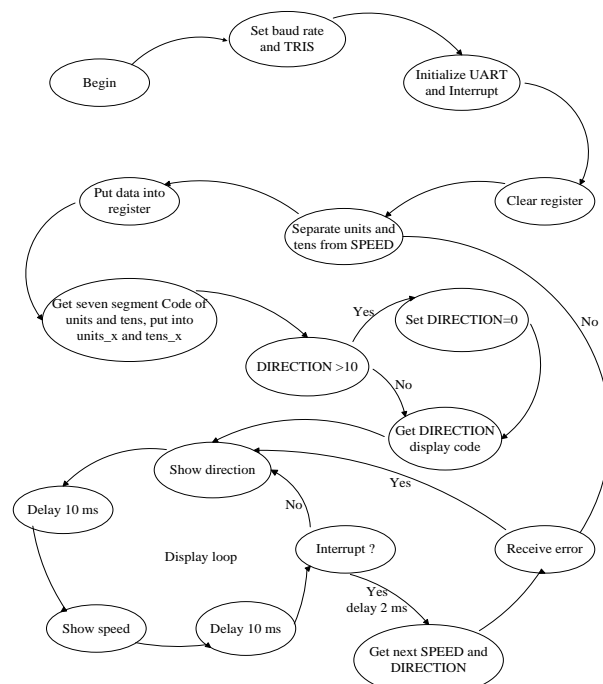


圖四：抬頭顯示裝置與安裝位置

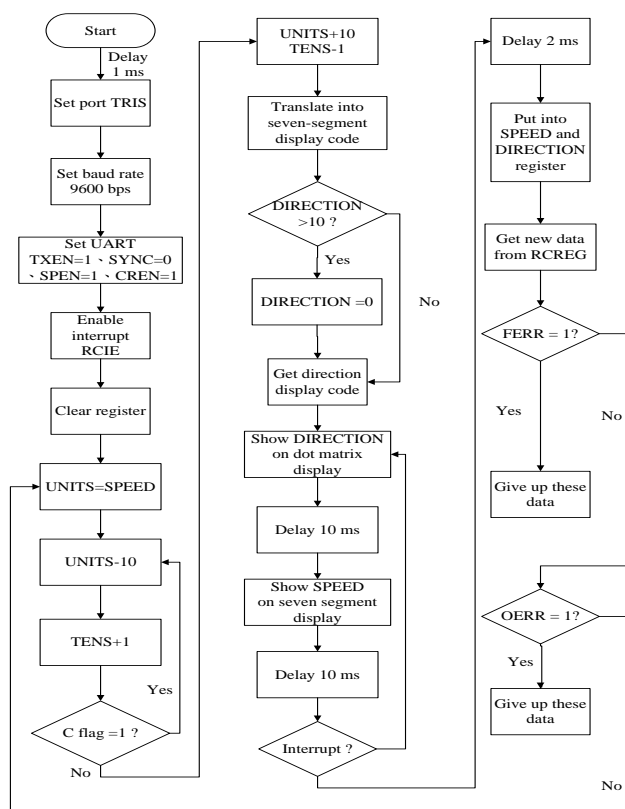
4.2 微控制器程式

微控制器使用的功能為 UART 與中斷，設定 TXSTA 與 RCSTA 中的四個位元 TXEN = 1、SYNC = 0、SPEN = 1、CREN = 1，致能 TXEN 位元是為了起動傳送，啟動接收須設定 CREN = 1，令 SYNC=0 是讓 UART 進入非同步模式，SPEN = 1 則是為了驅動 PIC 的 RX/DT 和 TX/CK 腳位，設定 INTCON 暫存器內的全域中斷致能、

周邊中斷致能和 PIE 中的接收中斷完成設定 [11]，圖五為程式狀態轉換圖，圖六為程式流程圖。



圖五：接收顯示程式狀態轉換圖



圖六：接收顯示程式流程圖

4.3 封包接收探討

當中斷一產生進入中斷向量便呼叫了一個 2 ms 的延遲副程式，再進行接收的動作，以避免產生明明傳送

了兩筆不同的資料，可是存放接收資料的「SPEED」、
「DIRECTION」暫存器都是存放第二筆資料的錯誤。

是由於 PIC 內部振盪器為 4 MHz，而四個振盪週期為一個指令週期，所以一個指令週期的長度為 1 μ s，比起藍牙傳送一個 Bit 需要花費約 104 μ s 要來的小很多，而一個 Stop Bit 的長度就非常足夠讓程式運行到超過存取資料副程式的範圍，所以程式接收到了第一筆資料以後，尚未處理完第二筆資料就進來引發了中斷，所以才造成了兩個顯示器都顯示了代表第二筆資料的錯誤，也就是說對 PIC 來講，手機是很快的連續丟了一筆資料兩次，而不是一次丟了兩筆資料。

所以要加入一個至少大於傳送資料所花時間的延遲，等到兩筆資料都放入 RCREG 中，再執行往後的程式，但希望延遲時間越短越好，若傳送速度為 9600 bps，完整傳送一筆資料所花的時間約為 1.042 ms，在延遲結束之前，要讓第二筆資料開始放入 RCREG 中，所以至少要大於 1.042 ms 的時間延遲才能符合需求然而為了保險起見，延遲時間設定為 2 ms。這個延遲大小會隨著提高傳輸速率而縮短。

五、測速準確性探討

5.1 GPS 定位原理簡介

衛星定位系統的定位採用的是「空間後方交會法」，其原理如下。GPS 衛星上裝置有非常精準的原子鐘，衛星會不斷的將位置、時間等資料藉由無線電波，以光速傳送到地面，手機在接收到衛星訊號的同時，將此訊號產生的時間點與自身的時間裝置加以比對，得到兩者的時間差，而此段時間差即為 GPS 衛星無線電波傳達到地面所需的時間，再將此時間差與光速相乘，即為 GPS 衛星與手機間的距離[12]。

5.2 測速準確度矯正

進行道路測試比較 GPS 測速與汽機車儀表板上所顯示間的差異，汽機車的測速原理都是利用輪胎轉速與輪胎的尺寸來計算移動速度，理論上汽機車取得時速的方法，造成誤差的因素比較少，主要誤差來自於輪胎的磨損與汽機車廠商的設定。

測試所使用的手機為 HTC Sensation，汽車為 2012 年底出廠輪胎尺寸沒有變更的 yaris，測試的地點為台中環中路與國道三號清水路段。使用汽車來作測試工具是為了安全起見，因為測試的過程需要操作多樣儀器。使用錄影的方式記錄手機測速與汽車儀表板所顯示的速度

表 II
道路測試誤差結果

儀表板速度 (km/hr)	GPS 速度 (km/hr)	誤差 (km/hr)
10	9	1
20	19	1
30	27	3
40	36	3
50	46	4
60	55	5
70	64	6
80	74	6
90	83	7
100	92	8

在程式內部作軟體的修正，因為測速準確度會根據手機硬體規格的不同而有差異，所以選擇在手機端的程式做修正。針對表 II 中的測試結果，得到的一個概略的修正公式 $y = x + \frac{x}{10}$ ，其中 y 為修正後的數據，x 為 GPS 測到的時速，又在 3.2.2 節中有提到，需要對手機程式得到的最原始測速數據單位 meter/sec 作轉換，轉換公式為 1 km/hr = 3.6 meter/sec，將上述兩式合併後，可以得到一個較為精簡的修正公式為 $y = 3.96 \times z$ ，z 為利用「location.getSpeed()」得到的速度資料。

這樣的修正結果會因不同廠牌手機所使用的不同 GPS 晶片，其第一次定位所花的時間與定位的精準度都有所不同，所以要應用程式需要依據各家手機來調整。

結論

本文為微控制器結合手機的整合應用，目的在改善機車使用人的駕駛經驗，減少超速的情況發生，還能具有導航功能的雛形，且倘若真發展成商品，整個裝置都使用低階的硬體，量產成本低廉且體積還能再縮小。若能在顯示器的部分，採用最新研發之透明顯示器，且該顯示器的單色穿透率可達 40% 以上，如此則可以解決觀察視角的問題，還能大大的增加顯示的區域和圖文量，便能增加更多的顯示功能，甚至設計圖形化和文字化的導航功能也是可行的。

參考文獻

- [1] Ma Chunxia, Zhuang Damin, Wanyan Xiaoru, Deng Fan, "Development of Flight Simulation and Head-up Display System for Ergonomics Evaluation," in *4th International Congress on Image and Signal Processing*, Oct. 2011.
- [2] webBikeWorld, <http://www.webbikeworld.com/head-up-display/>, 2002.
- [3] POPSCI, <http://www.popsci.com/cars/article/2013-06/motorcycle-helmet-head-up-display>.
- [4] Google Android Developers—Location API Guides, <http://developer.android.com/guide/components/>, 2013.
- [5] Google Android Developers—Location API Guides, <http://developer.android.com/guide/components/>, 2013.
- [6] Krittiya Tangmanee, Sakol Teeravarunyou, "Effects of Guided Arrows on Head-Up Display Towards the Vehicle Windshield," in *Southeast Asian Network of Ergonomics Societies Conf. (SEANES)*, July. 2012.
- [7] John R. Troxell, arie I. Harrington, Robert A. Perisho Jr., "Reconfigurable Head-Up Display for Enhanced Vehicle-To-Driver Communication", in *Intelligent Transportation System Conf.*, Oct. 1997, pp. 308-313
- [8] Park Min Hee, Kim Hyun Jeong, Lee Kwang Hyun, "Development of Circumstance-based Variable Information Structure for Head Up Display in a Car", in *Data Mining and Intelligent Information Technology Applications Conf.*, Jun. 2011, pp.127-130.
- [9] Tsuyoshi Todoriki, Junichi Fukano, Shigeru Okabayashi, Masao Sakata and Hiroshi Tsuda, "Application of head-up displays for in-vehicle navigation/route guidance," in *Vehicle Navigation and Information Systems Conference*, Yokohama, Aug. 1994, pp. 479 – 484.
- [10] Paul L Wisely B.Sc, M.Phil, C.Phys, M.Inst.P. C.Enq, M.I.E.E., "The design of wide angle head up displays for synthetic vision," in *Digital Avionics Systems Conference*, Phoenix, AZ, Oct. 1994, pp. 608 – 619.
- [11] Microchip Inc., "PIC16F87XA Data Sheet," available from Microchip's homepage, <http://www.microchip.com/>, 2003.
- [12] Author, Ahmed El-Rabbany (2002), *Introduction to GPS: The Global Positioning System*, Artech House.