資工系實務專題研究計畫成果報告
Android Car

專題編號：100-CSIE-S009
執行期限：99 年 1 學期至 100 年 1 學期
指導教授：梁文耀
專題參與人員：陳昱光 陳至圓 賴柏翰

一、 摘要

本計畫的目的是藉由 Android 開放手機平台作業系統，因此使用者能夠以 **Wi-Fi 為傳輸模式**來達成遠端操控遙控車，進一步的在車上**加裝 Camera**，使得遙控車能夠拍攝前方影像，並**即時地將畫面回傳至操作端螢幕上**，讓使用者在操控方面增添了不少的真實感，而控制方面則有 **G-sensor 與一般搖桿**兩種控制方法，讓使用者能夠擁有不同的方式來操作遙控車，同時也增加了圖片儲存功能，讓使用者能夠擷取有趣的畫面存到 SD Card 中，也可以**分享到 Facebook、Twitter 等社群網站**。

關鍵詞：Android、Camera、Wi-Fi、G-sensor

二、 緣由與目的

遙控車(Radio-controlled cars，或稱 R/C Cars)，傳統是以無線電或紅外線等傳輸方式操控玩具車。但由於無線電的方式來控制，則會有**距離上的限制**，因此我們希望能讓遙控車到更遠的地方也能控制。

然而在我們日常生活中常見的傳輸方式莫過於網路，況且無線網路逐年盛行，因此遙控車如果能以 **Wi-Fi 來傳輸**，在訊號良好的環境下，便能實現**無距離限制**傳輸，因此畫面變是不可或缺的功能之一，所以我們在遙控車上**加裝一台 Camera 接收畫面**，因此遙控車上必須要能夠有 **Wi-Fi** 與 **USB host** 功能的開發板，如此一來我們就可以透過 Android 手機與遙控車上的開發板傳輸影像與控制訊號。

然而我們之所以選用 **OpenMoko** 當開發板，是因為它可以嵌入 **Android 系統**，加上有內置的 GPS、**Wi-Fi**、Bluetooth、**G-Sensor**、**可切換的模式的 USB 模組**、方便的程式開發環境，跨平台的優越性等...，對於未來的發展有較大的空間。

三、 軟體與系統架構

由於遙控車與控制端都透過 Wi-Fi 作傳輸溝通，因此兩者皆必須位於具有無線網路的環境下操作。

在本架構中控制訊號傳輸流程為 **Phone→ Openmoko→ 8051→ 遙控車**，手機到 Openmoko 之間則是透過 **Socket** 來傳輸控制訊號，而當 Openmoko 接收到指令後便可以經由 RS232 傳輸至遙控車上的 8051 UART 上，接著經由 8051 直接透過外部電路(H 橋等...)驅動遙控車馬達，藉此方可達成遠端操控的效果。

而傳輸遙控車影像的流程則是 **USB Camera→ Openmoko→ Phone→ Cloud** 的方式，主要由 Openmoko 從 USB Camera 中擷取影像，緊接著**解碼**，再透過 **Socket** 將每一個 Frame 傳輸至 Android 手機上作即時畫面顯示，之後手機可以透過相關的網路服務，分享手機所取得的影像。

四、 實作方式

基於架構流程下可知 Openmoko 平台上必須擁有支援 USB to RS232 的驅動程式(**Prolific 2303**)，而且 **USB 模式必須為 Host 模式且可供電**，才能驅動外部裝置。然而要做好 USB 模式切換，則 Openmoko 上必須要擁有 **Root 權限**。因此我們選擇在 **Android Source 中修改 su 執行權限**，並將 **Kernel** 中加入 **Prolific 2303 與 V4L、UVC 驅動支援**。

處理完驅動上的問題後，我們透過 Java 寫出能控制 Serial port 的程式了。利用 **JNI 來調用 C 程式**是最快速且有效率的方式，經由 C 程式來幫助我們**設定與開啟 Serial port**，然後再透過 **JNI (Java Native Interface)**的方式回傳 **FileDescriptor** 給 Java，如此一來 Java 便能**以檔案讀寫檔案的方式來控制 RS232** 了。

基於上述方式應該可以完成遙控車遠端操控了，但最大的問題則在於影像部分，如何將 Android 程式能透過原有的 Camera API，直接控制我們剛嵌入的相機。

因此我們需將遵照 **V4l2(Vedio for linux version 2)流程**將 USB Camera 的操作方式實作於 **Android Camera HAL (Hardware Abstract Layer)**中。在 Android Camera 架構流程中提供了相機開發的**雛形**，方便我們能夠依據自己需求，將硬資源完美的契合上 **Android Framework** 之中，便是所謂的 HAL 硬體抽象層，在擁有 Linux driver 之後，便可以在 HAL 中實作與 Driver 操作的程式碼，當我們將 Camera 遵照流程將相機**初始、預覽、拍照、錄影與解碼實作在 HAL 中**，在頂層的 Android 程式便可以直接透過一般的 Camera 使用方式，來使用我們新嵌入的 Camera 了。

Android 手機端的應用程式我們除了利用 Socket 做基本傳送控制訊號到遙控車上及接收影像外，我們利用了 Wifi Manager 這項服務來做廣播，手機即能對在相同網域下的遙控車取得 Response，即能取得其 IP Address。我們也對 Sensor Manager 註冊事件，以實現利用擺動手機的方式來控制遙控車。

手機對 Cloud 的應用上，Android Car APP 利用了 Facebook 以及 Twitpic API，可直接分享圖片，Android Car 也會自動搜尋手機內建可分享圖片之 App 來做分享。最後，我們寫一個 HttpURLConnection，連上了自己架設的 Server 端，將我們遙控車的行徑過程給錄製了下來，做出簡單的網頁版行車紀錄器。

五、 結論

目前所有的流程都已經完全實做出來了，包括手機控制遙控車、獲得遙控車當前影像、與遙控車建立連線，所有的核心功能都已建構完畢，此外我們還提供了畫圖片上傳的功能，使用者可以依據自己喜好將喜好的畫面上傳至社群網站中，與好友做分享。

基於我們專案的架構下我們更可以進一步的遠端遙控任何的東西，例如：無人駕駛交通工具、資訊家電、戰爭上的殺人武器等…。

六、 參考文獻

[1] **Android Camera 架構介紹**
http://www.cnmsdn.com/html/201003/1268929336ID2257_10.html

[2] **Android Developers**
http://developer.android.com/index.html

[3] **Openmoko**
http://wiki.openmoko.org/wiki/Main_Page

[4] Max232
http://www.ti.com/lit/ds/symlink/max232.pdf