

專題名稱:在iOS系統上實作UPnP以及Internet-Radio.

專題成員:李沅迪,陳廷碩,連奕盛

專題編號:NTPUCSIE-103-007

執行期間:2014年9月至2015年5月

一、簡介

隨著行動裝置以及無線設備的普及,許多影音設備可用APP去操控,因此我們做了使檔案可共享的UPnP-Controller、可以把Internet-Radio播放至非本機裝置的Controller以及可以在音訊檔播放至揚聲器之前就做好左右聲道分離,以下分別對我們實作的背景以及所使用的技術、系統架構、實作內容、成果和未來目標做說明。

二、背景、動機及目的

在此這個章節中,我們會介紹這個作品的相關背景,還有因為背景而延伸出來的動機以及我們的目的。

A. 背景

A.1 雖然許多影音設備可無線化使用,但大多數未具備檔案共享的功能。

A.2 隨著行動裝置的普及,許多Controller可以被整合至同一APP中,比如UPnP整合Internet-Radio。

A.3 目前大多的Internet-Radio只能播放至本機端。

A.4 目前實作左右聲道的方法大多都是直接在揚聲器上實作。

B. 動機

B.1 透過UPnP共享行動裝置上的音樂檔案。

B.2 利用我們的APP使Internet-Radio可以在非本機端播放。

B.3 在音訊傳送至揚聲器之前即執行左右聲道分離。

C. 目的

C.1 開發iOS系統上的

UPnP Controller。

C.2 開發iOS系統上的Internet Radio Controller。

C.3 利用ALSA音效卡分離左右聲道。

三、系統架構

依Figure 1.所示,紅色虛線為連結至Wifi AP。綠色實線為UPnP-Mode時,Control point從Server選取檔案。而黑色實線在UPnP Mode代表傳送檔案給Renderer播放,在Radio Mode代表傳送控制data給樹莓派,以此控制Internet radio(包含設定左右聲道)。

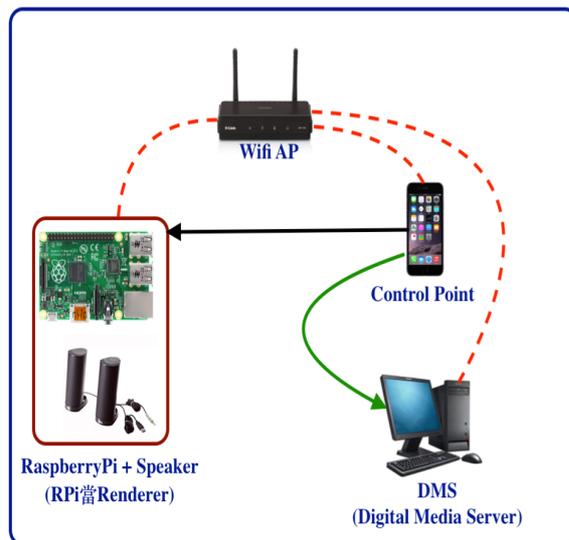


Figure 1. 系統整體架構,包含UPnP以及Internet Radio

四、技術說明

在這個章節中,我們將說明在實作時所用到的技術。

A.UPnP(Universal Plug and Play)

在iOS APP的部分我們利用了Platinum UPnP函式庫來實作,而UPnP定義了以下六個步驟:

-Addressing

此步驟讓同一WifiAP下的裝置取得IP位址。

-Discovery

這個動作讓Control-Point主動去搜尋使用UPnP協定的設備。

-Description

UPnP下的裝置利用此步驟宣告自己的功能以及其他資訊。

-Control

Control-Point利用此步驟操做其他裝置。

-Eventing

Control-Point利用此步驟得知其他裝置的狀態變化。

-Presentation

UPnP下的裝置若有URL,使用者可從其網頁得知設備狀態或控制設備。

B.Internet Radio

在Internet Radio的實作過程中,我們在R-Pi中安裝了兩個套件,並且以UDP Broadcast的方式讓R-Pi與APP溝通,以下做說明:

B.1 R-Pi安裝之Internet-Radio套件

-MPD

MPD 是“Music Player Daemon”的縮寫。MPD這個套件主要的功能是将一個輸入流(input stream)轉成音頻輸出(audio output)。

需要注意的是是一個輸入流並不精確的代表一個數據流(steam of data),輸入流可以來自一個資料夾或是一個網路位置。

因為MPD只是默默的在背景裡執行,所以他沒有使用使用者介面而且需要另一個客戶端軟體跟使用者溝通,然而這些都可以被MPC這個套件提供

-MPC

MPC是“Music Player Client”的縮寫。MPC可以用來管理播放清單,清單內容也包含了電台。MPC是MPD的使用者介面。

B.2 R-Pi與APP溝通方法

-UDP broadcast

Broadcast會將資料送給同一個Wifi下所有想接收到此資料的裝置,這裡不需要在socket程式中綁定一組固定的IP。

B.3 暫停功能

按下暫停鍵後,iOS裝置將會送一個特定的資料給樹莓派,而樹莓派接收到該資料後會下指令“mpc stop”來暫停播放。

五、流程說明

在這個章節中,我們將對我們的運作流程做說明,首先針對UPnP做說明:

如Figure 2.所示,圖左為R-Pi端,圖右為iOS-Device端,在UPnP的運作狀態中我們先啟動R-Pi中的GMediaRenderer,讓Renderer在UPnP環境中啟動,然後再啟動iOS APP,啟動APP之後,點選UPnP Mode,此時iOS-Device為一在UPnP環境中運作的Control point,可透過此Control point取得DMS中的音樂檔案,然後再將選取的音樂檔送至Renderer播放。

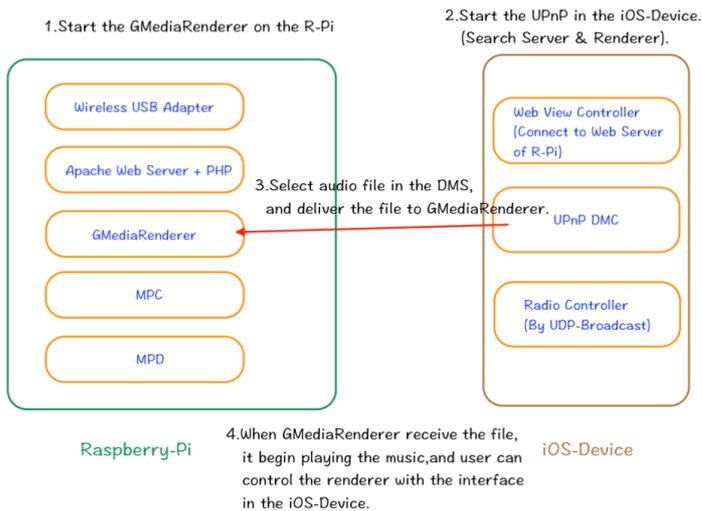


Figure 2. UPnP運作流程圖,左端為R-Pi,右端為iOS-Device,在UPnP Mode中,R-Pi角色為Renderer,iOS-Device的角色為Control Point

接下來說明Internet Radio的部分,如Figure 3.所示,圖左為R-Pi端,圖右為iOS-Device端,在Internet Radio的運作流程中,我們先開啟R-Pi中的Internet Radio套件MPC、MPD,開啟之後,我們點選iOS APP中的Internet Radio,此時iOS-Device為Radio Controller,可以利用UDP Broadcast的方式傳送data給R-Pi,再利用此data去對Internet做控制(包含左右聲道)。

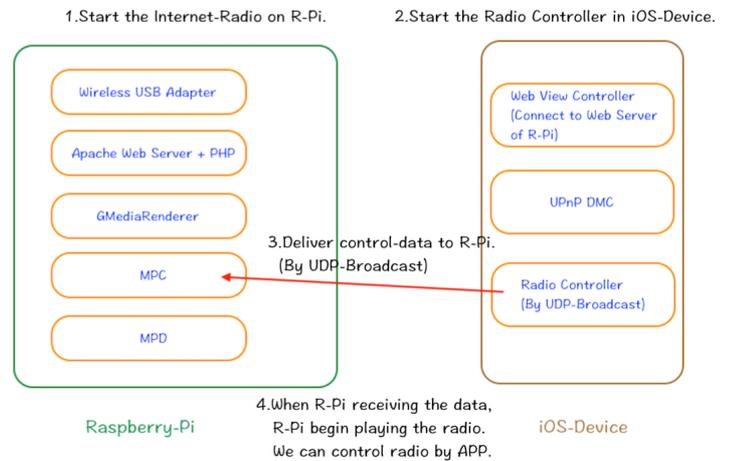


Figure 3. Internet Radio運作流程圖,左端為R-Pi,右端為iOS-Device,R-Pi的角色為Internet Radio,iOS-Device的角色為Radio的Controller

最後說明我們從如何從APP端更改R-Pi的網路設定,如Figure 4.所示,我們先啟動圖左R-Pi的Web Server,此時R-Pi為一AP(Access Point),在R-Pi的頁面中,我們用PHP寫好了無線網路如何設定,然後iOS-Device經由load R-Pi的頁面去更改R-Pi的網路設定。

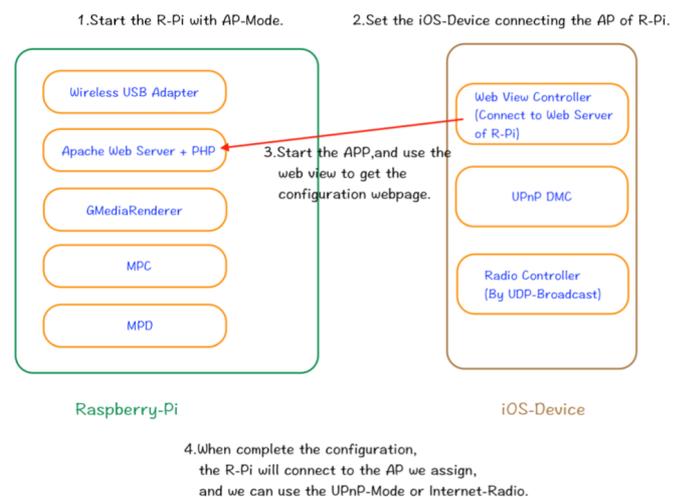


Figure 4. Device Setup運作流程圖,圖左R-Pi的角色為AP,圖右的iOS-Device可load R-Pi的頁面去更改網路設定

六、系統實作

在以上的架構以及技術之下,我們的實作成果如下:

Figure 5. 為APP之首頁,其中有3個Button,分別為UPnP Mode、Internet Radio、Device Setup,UPnP Mode的功能是開啟UPnP,Internet Radio的功能為開啟Internet Radio,Device Setup的功能使我們可以從APP去更改樹莓派的網路設定。

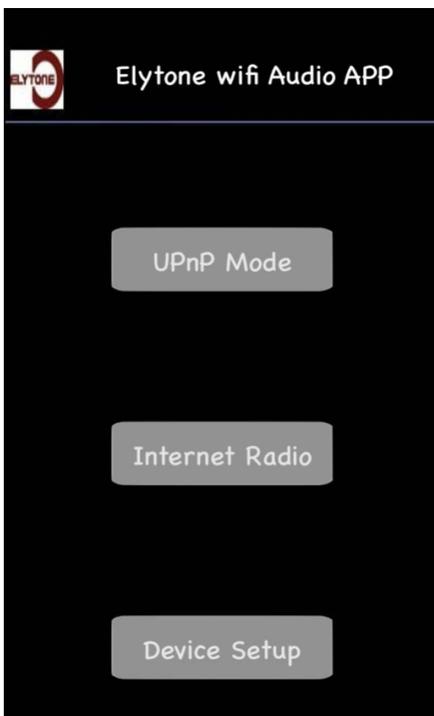


Figure 5. APP首頁,上面的Button各表示點選之後的功能

A. UPnP Controller

在首頁點選UPnP Mode之後會進入DMS(Digital Media Server)頁面(如Figure 6.所示),我們可以從DMS選取音樂檔案,選取完之後,我們進入Renderer頁面(如Figure 7.所示),在這個頁面,我們可以選擇要播放音樂檔案的Renderer,選取完Renderer之後,我們進入控制頁面(如Figure 8.所示),藉由此頁面,我們可以控制Renderer之播放狀態(音量,播放/暫停)。

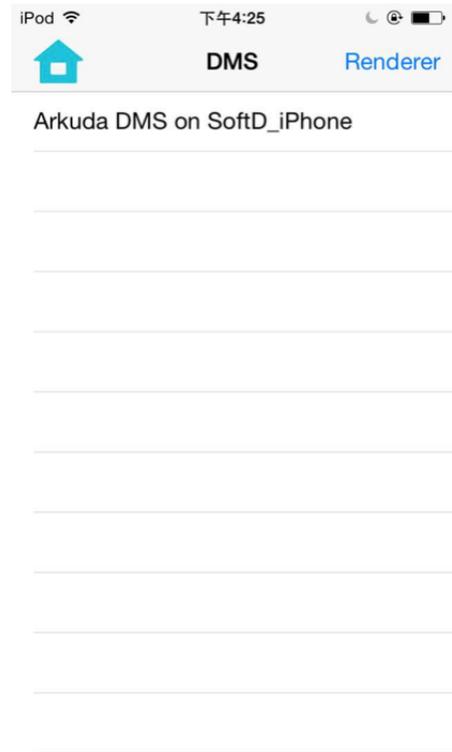


Figure 6. DMS頁面,列出周遭的DMS

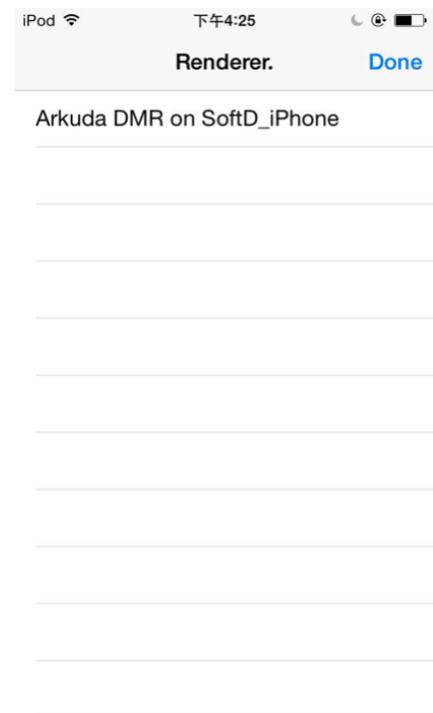


Figure 7. Renderer頁面,列出周遭的Renderer

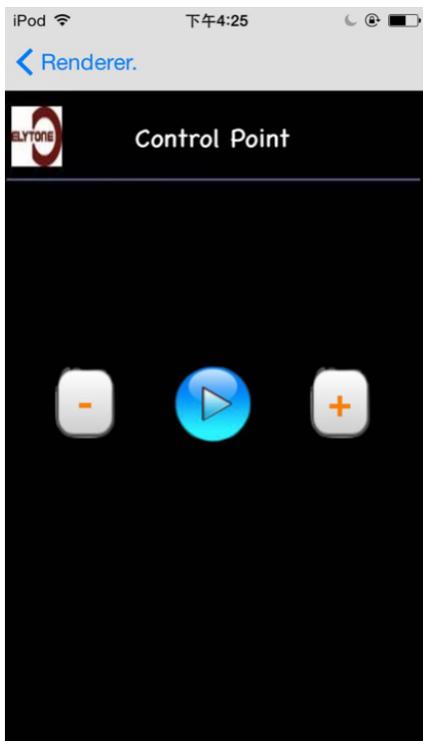


Figure 8. UPnP播放頁面,包含音量控制以及播放/暫停功能

B. Internet Radio

在首頁點選Internet Radio之後,進入Radio控制頁面(如Figure 6.所示),在此頁面中,我們可以看到有一個序列式的選單,一個方型的暫停鍵,一個Label,還有一張圖片,點選下面的序列式選單即開始播放電台,而點選暫停鍵會暫停播放,Label會顯示電台名稱,上面的圖片會隨電台種類做相對應的變化。



Figure 6. Radio控制頁面,含電台選單、暫停鍵、顯示電台的Label、隨電台種類做更改的圖片

七、結論及未來展望

在這個主題之下,我們實作了iOS系統上UPnP Controller,以及Internet Radio Controller,在音訊處理的部分,我們也利用ALSA處理左右聲道之分離。

我們希望未來的架構可以做到UPnP的一對多,還有讓UPnP不只利用於音訊相關的檔案。

八、銘謝

這次專題我們得到許多人的幫助,除了指導教授耐心且細心的觀念指點還有引導,也要感謝碩士班學長的教學,如果沒有他們的協助,這次專題將會有許多困難無法被解決。

也很謝謝我們彼此之間付出的努力,才可以讓我們走到這一步。

九、參考文獻

- [1]upnpx函式庫參考與教學
<https://code.google.com/p/upnpx/>
- [2]Platinum UPnP參考與教學
<https://github.com/gravitymobile/Platinum>
- [3]Platinum UPnP官方網站
<http://www.plutinosoft.com/platinum/>
- [4]iOS AsyncSocket參考
<http://blog.csdn.net/bjx327660180/article/details/8105289>
- [5]UDP Broadcast參考
<http://www.tack.ch/multicast/broadcast.shtml>
- [6]Socket in python參考
<http://www.binarytides.com/programming-udp-sockets-in-python/>
- [7]Internet Radio參考
<https://www.youtube.com/watch?v=KM4n2OtwGI0>
- [8]ALSA音訊處理參考
http://www.alsa-project.org/main/index.php/Main_Page