

藍牙多工傳輸程式設計方法

范揚文 陳建翰 黃渝瑄
朝陽科技大學資管系 朝陽科技大學資管系 朝陽科技大學資管系
ywfan@cyut.edu.tw s10114185@cyut.edu.tw s10114191@cyut.edu.tw

摘要

行動裝置已廣泛使用藍牙做為傳輸資料的媒介，過去這些傳輸的應用大都是一對一的連線架構，然而隨著藍牙技術的演進，一對多連線的系統開發專案也越來越多，這篇論文提出一個完整的程式設計方法，以協助程式設計師開發短距離多點連線的應用。

Apps 的開發已變得越來越快，主要是因為程式開發環境、設計說明文件與樣本範例已相當完善。本論文的範例系統將使用 MIT App Inventor 2 做為程式開發工具，此系統將利用藍牙完成一個一對多連線的對話功能，我們也將詳細描述如何設計藍牙的伺服器端與客戶端，以及連線和資料傳輸控制的方法。

關鍵詞：藍牙，點對多點通信，安卓系統

Abstract

Mobile devices have widely used Bluetooth to do for transmitting media. In the past, the these transmission application was mostly point-to-point connection structure. However, along with Bluetooth technology evolution, point-to-multipoint system projects are developing more and more. This paper proposes a complete formula design method to assist the programmer to develop the short distance point-to-multipoint applications.

The Apps development has become more and more quick. This is mainly because the formula development environment, the design document and the sample model are quite perfect. We will use MIT App Inventor 2 to develop the paper sample system. This system will employ Bluetooth to complete a point-to-multipoint dialogue function. We will also describe the detailed design way of the Bluetooth server and client side, as well as connecting and transmission control method.

Keywords: Bluetooth, point-to-multipoint communication, Android

1.簡介

藍牙起源於 1994 年，最初由易利信 (Ericsson) 公司開發。此技術研究行動電話和其他配件間進行低功耗、低成本無線通信連線的方法。1998 年，由索尼易利信、IBM、英特爾、諾基亞及東芝組成藍牙技術聯盟 (Bluetooth Special Interest Group, 縮寫為 SIG) [4]，這是一個以制定藍牙規範，與推動藍牙技術為宗旨的跨國組織，它擁有藍牙的商標，負責認證製造廠商，授權他們使用藍牙技術與藍牙標誌。同一年正式公布藍牙 1.0 版，確定使用 2.4GHz 頻譜，最高資料傳輸速度 1Mbps。2001 年的 1.1 版正式列入 IEEE 標準，此版本即為 IEEE 802.15.1，在這一年底，SIG 成員公司已超過 2000 家。

藍牙 2.0+EDR(Enhanced Data Rate)版將速度提高到 3Mbps，並減少耗電量。接著，藍牙 3.0+HS(High-Speed)版進一步將速度提高到 24Mbps，正式超過無線區域網路 IEEE802.11 的速度，同樣在電源控制有了進步，耗電量明顯降低。藍牙 4.0 版於 2010 年 7 月推出，此協定包含低功耗藍牙(Bluetooth low energy)、傳統藍牙(Classic Bluetooth)和高速藍牙(Bluetooth high speed)三種運作模式，此協定將藍牙的傳輸距離提升到 100 米以上，並將功耗進一步降低，使藍牙設備的待機時間可長達半年以上。藍牙技術的持續演進，使得相關的應用越來越多，目前已成為物聯網(Internet of Thing)的關鍵技術。最新的版本是 2014 年推出的藍牙 4.2 版，主要增加的功能是與 IP 協定的結合[1]。

藍牙系統提供點對點(point-to-point)與點對多點(point-to-multipoint)的連線方式，分別如下圖 1(a)與(b)，這些共享無線實體訊號的設備所形成網路的稱為微網 (Piconet)。，一個微網內的裝置數目可以有很多，其中包含一個主裝置 (Master device)，以及多個的隨從裝置 (Slave device)，但同時最多只有七個隨從裝置可以處於活動狀態(active state)，其餘的隨從裝置處於待機狀態(parked state)，這些待機狀態

的設備隨時可以轉成活動狀態，不需重新連線。每個微網中的主裝置可以成為另一個微網中的隨從裝置，若幾個微網間有共用的裝置，則由這些微網形成的網路稱為擴散網 (Scatternet)，連線拓譜如下圖 1(c)[1]。

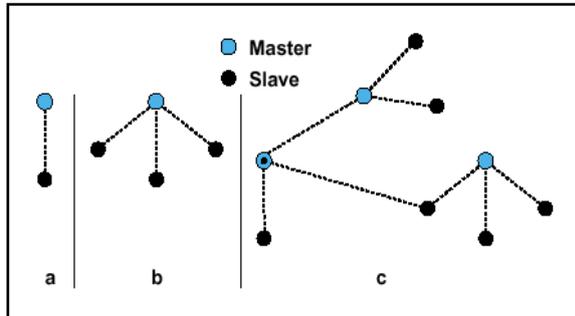


圖 1: (a)一對一連線的 Piconet。(b)一對多連線的 Piconet。(c)多個 Piconets 連線所形成的 Scatternet。

藍牙裝置的規模從 2012 年的 20 億個[1]，根據 ABI Research 研究[5]，2018 年將成長到 49 億以上。主要的應用包含遊戲機、汽車、耳機、手錶及健身設備等之訊號的傳輸或控制；近來逐漸熱門起來的寵物活動力監測、走失協尋、社會安全與老人幼兒照護等；此外與物聯網結合的智慧型回覆與偵測系統也持續在開發中。

其中一對多連線的功能可以讓藍牙的應用更豐富，例如遊戲機中的互動可以更為容易，在監控的應用中也可以對多個目標同時進行，在手機中也可以與多個週邊設備進行互動。在過去開發的系統中，因為一對多連線的程式控制較為複雜，所以相關的應用較少，相信未來隨者藍牙技術的普及，這類型的應用會有可觀的成長。

關於藍牙一對多連線的程式開發，在 Android 系統中的藍牙應用程式開發界面 (Bluetooth APIs) 被定義在 android.bluetooth 這個套件 package 中[6]，此套件包含藍牙類別庫 (classes) 及函式界面的格式，我們可以撰寫連線建立及控制程式，再配合呼叫此函式庫即可以完成藍牙的連線系統。現在已可以在很多文件中找到建立一對一連線的程式方法，但因為一對多連線的建立及控制方法較為複雜，雖然已有些論文提到已完成了連線測試系統[2]，但都沒有完整的程式方法說明，本論文詳細描述建立及控制多點連線的程式技巧，應該對於有心開發藍牙應用系統的技術人員有所幫助。

2. 系統開發的軟體環境

目前主要開發 Apps 的環境有三種，分別是 Eclipse、Android studio 及 Inventor。Eclipse 是跨平台開源整合式開發環境 (IDE)，主要用 Java 語言開發，亦可以外掛模組使其作為 C++、Python、PHP 等其他語言的開發工具。因為 Eclipse 支援多種作業系統的開發，在進行 Apps 程式設計時常會出現版本或平台不相容的情形，因此當 google 公司推出 Android studio 之後，使用 Eclipse 開發系統者已越來越少。Android Studio 是一個專為 Android 平台開發程式的整合式開發環境，第一個穩定版本 1.0 於 2014 年 12 月 8 日發布[7]，目前的穩定的版本是 1.5.1。Android App Inventor 是起先由 Google 提供的應用軟體，現在由麻省理工學院 (MIT) 維護及營運[3]。它可以讓任何熟悉或不熟悉程式設計的人來創造基於 Android 作業系統的應用軟體。2013 年 12 月 MIT 發表了 App Inventor 2 版，到了 2015 年 12 月已經有超過四百萬的註冊使用者，並且開發了超過一億兩千萬個 Apps。

本系統使用 Google Inventor 2 開發此藍牙系統，Inventor 使用網頁介面進行程式開發，所以只要能連上 MIT 網頁[8]的電腦即滿足程式開發需求。測試 Apps 的軟體硬體規格如下表：

表 1：測試 Apps 的軟體硬體規格

	HTC Butterfly 2	Google Nexus 7 二代
作業系統	Android 5.0.2	Android 4.4.4
記憶體容量	2GB	2GB
藍牙版本	藍牙 4.0	藍牙 4.0
可連線距離	105~110M	105~110M

3. 藍牙多點連線系統開發的主要步驟

因為 Inventor 2 提供很便利的操作介面，所以只要依循正確的設計步驟，就能將功能建立起來。以下先說明程式設計的主要步驟，以幫助程式設計師了解整個專案進行的過程，下

一段接著闡述詳細的設計方法。

步驟一：建立 Android Application 專案
工作內容為設定專案名稱。

步驟二：設計 Android 的使用者介面
利用調色盤(Palette)工具箱，在 Design Window 環境下設計使用者介面。

步驟三：新增 Bluetooth component
在 Design Window 環境下新增 Bluetooth server component 及 Bluetooth client component。

步驟四：編寫 Bluetooth 連線控制碼
在 Blocks Editor 環境下撰寫程式碼，對 Bluetooth server 與 Bluetooth client 連線的建立及中止進行控制。

步驟五：製作 apk 檔
利用 App Inventor 2 的 build 工具建立 apk 檔。

步驟六：進行 Bluetooth 連線測試
將 apk 程式檔下載到行動裝置中，進行 bluetooth 配對後，即可做 bluetooth 連線測試。

4.設計方法與成果

在這個小節首先將完整說明程式設計的過程，隨後展示執行的結果。

步驟一：建立 Android Application 專案[9]

工作內容為設定專案名稱，依提示畫面輸入適當名稱(如圖 2)，注意不可輸入空白鍵。當按 OK 鍵確認後，此專案資料即儲存在雲端的資料庫中。

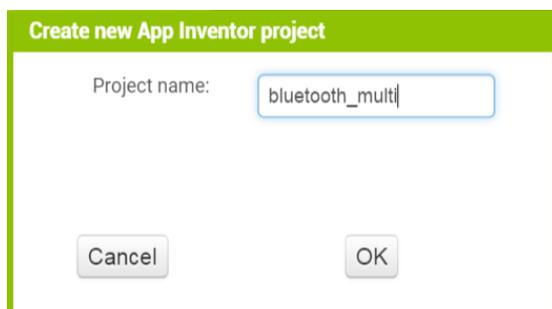


圖 2：啟動專案後即可依指示輸入名稱

步驟二：設計 Android 的使用者介面
利用調色盤(Palette)工具箱，在 Design

Window 環境下設計使用者介面。首先布置底稿的框架物件 align，其次選用標記元件 label 即按鈕元件 button，最後選擇 text 元件以接收傳送與接受的訊息，並給每一個元件適當的名稱與屬性設定，完成的畫面如下(如圖 3)。



圖 3：先布置框架再拉入元件即可排列整齊

步驟三：新增 Bluetooth component

在 Design Window 環境下新增 Bluetooth server component 及 Bluetooth client component。在此專案中我們將進行一對三的藍牙連線，所以新增三個伺服器元件以等待客戶端發出請求，及新增一個客戶端的元件，此元件用來發出藍牙連線請求。這些元件會顯示在設計的版面上(如圖 4)，當執行程式時，這些原件不會顯示出來。當新增完成後，相關設計介面如下。

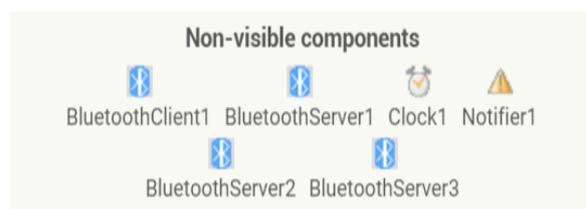


圖 4：不可視的藍牙連線元件

步驟四：編寫 Bluetooth 連線控制碼

在 Blocks Editor 環境下撰寫程式碼，對 Bluetooth server 與 Bluetooth client 連線的建立及中止進行控制。因為藍牙是一種對等式網路連線，每一端的連線設備都有可能扮演伺服器端或客戶端，但藍牙技術只支援伺服器端連線多台客戶端，不支援一個客戶端連線多個伺服器端值，因此當設備發出連線請求之後，就只能成為客戶端的角色，不能同時成為伺服器端。以下分別說明客戶端及伺服器端的程式設計方法。

一、客戶端的程式設計

當發出連線請求之後，列出經過藍牙配對成功的設備，提供客戶端選擇要連線的設備。

接著依 MAC Address 選擇指定的連線機器後，判斷是否連線成功，若連線成功，則將自己的設備指定為藍牙的客戶端，並中止藍牙的伺服器等待(如圖 5)。

當連線完成後，即可以利用此連線進行資料傳輸，若要主動中止連線，則發出連線中止請求。

```
when lipConnect .AfterPicking
do
  call print
  textToPrint join " trying to connect to " lipConnect . Selection
  if BluetoothServer1 . IsAccepting
  then call BluetoothServer1 . StopAccepting
  if BluetoothServer2 . IsAccepting
  then call BluetoothServer2 . StopAccepting
  if BluetoothServer3 . IsAccepting
  then call BluetoothServer3 . StopAccepting
  if call BluetoothClient1 . Connect
  address lipConnect . Selection
  then
    set global boolServer1 to false
    set global boolServer2 to false
    set global boolServer3 to false
    set global boolClient to true
    set Screen1 . Title to " 角色=Client "
```

圖 5：發出連線請求後，先中止 Server 的連線等待，再判斷請求是否成功。

二、伺服器端的程式設計

當系統啟動之後，設定完變數的初始值之後，即可啟動藍牙的伺服器端等待連線程式，這裡我們提供三個等待連線服務的例子(如圖 6)。

```
to init
do
  call BluetoothServer1 . AcceptConnection
  serviceName " "
  call BluetoothServer2 . AcceptConnection
  serviceName " "
  call BluetoothServer3 . AcceptConnection
  serviceName " "
  set Screen1 . Title to " 角色=undefined "
```

圖 6：啟動三個等待連線的 Server。

我們需要定時掃描通訊訊號，查看是否有藍牙連線請求，若有連線請求進來，給定連線旗標編號，及將自己的設備指定為藍牙的伺服器端(如圖 7)，並繼續掃描通訊訊號，準備其他的連線建立。此外，也要將自己的客戶端的服務關閉。

```

when BluetoothServer3 .ConnectionAccepted
do
  call print
  textToPrint "connection accepted"
  set global boolServer3 to true
  set Screen1 . Title to "角色=Server"
  call connected
  trueOrFalse true

```

圖 7：連線建立，設定旗標與角色。

當連線完成後，即可以依連線旗標編號，判斷可以使用的藍牙連線(如圖 8)，並能將資料同時送到指定的連線(如圖 9)。

依 MAC Address 選擇指定的連線機器後，判斷是否連線成功，若連線成功，則將自己的設備指定為藍牙的客戶端，並中止藍牙的伺服器等待。

當資料傳送後，可以發出中止連線請求(如圖 10)，若連線中止後，將系統帶回起始狀態，再次準備成為藍牙的客戶端或伺服器端。

```

to disconnect
do
  call print
  textToPrint join txbNickname . Text
  " disconnected "
  if get global boolServer1
  then call BluetoothServer1 .Disconnect
  if get global boolServer2
  then call BluetoothServer2 .Disconnect
  if get global boolServer3
  then call BluetoothServer3 .Disconnect
  if get global boolClient
  then call BluetoothClient1 .Disconnect

```

圖 10：判斷連線的狀態，並關閉活動中的連線。

```

when Clock1 .Timer
do
  if get global boolServer1
  then
    if call BluetoothServer1 .BytesAvailableToReceive > 0
    then
      call print
      textToPrint call BluetoothServer1 .ReceiveText
      numberOfBytes call BluetoothServer1 .BytesAvailableToReceive

```

圖 8：定時掃描各個連線，判斷是否有資料傳入。

```

to send message
do
  if get global boolServer1
  then call BluetoothServer1 .SendText
  text get message
  if get global boolServer2
  then call BluetoothServer2 .SendText
  text get message
  if get global boolServer3
  then call BluetoothServer3 .SendText
  text get message
  if get global boolClient
  then call BluetoothClient1 .SendText
  text get message

```

圖 9：依旗標將資料送到指定的連線。

步驟五：製作 apk 檔

利用 App Inventor 2 的 build 工具建立 apk 檔。在設計畫面的上方按下 Build 鍵，會有如下圖的兩個選項可以選擇。

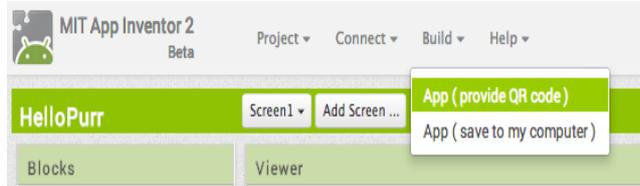


圖 11: 建立完成的 apk 檔可以放在雲端或本機。

選擇一可以產生 Barcode(或稱 QR Code)，再利用自己的設備去掃描，即可以將程式下載下來執行。此 Barcode 只針對自己的 google 帳號，所以其他帳號的設備無法下載。

選擇二會產生一個 apk 檔，並下載到自己的電腦，若要執行此程式，將 apk 檔安裝到行動裝置上即可執行。

步驟六：進行 Bluetooth 連線測試

將 apk 程式檔下載到行動裝置中，進行 bluetooth 配對後，即可做 bluetooth 連線測試。以下為連線測試的成果展示與相關說明(如圖 12 與 13)。



圖 12: 已連線完成，Server 發出訊息給所有的 Client。



圖 13: 將自己的腳色設成 Client，及回應訊息給 Server。

5. 結論

行動 apps 的應用如雨後春筍大量的冒出，其中關於中短距離無線連線的創意應用特別的吸引人，本論文提出完整的一對多的藍牙連線程式設計方法，除了驗證藍牙多點連線的可能性，也提供給行動裝置系統開發者，關於無線連線控制的詳細參考技術文件。

在未來我們將利用藍牙省電與對等連線的特性，開發更多相關的應用系統；也會追蹤物聯網的應用趨勢，結合 RFID 或 Barcode 技術，研究藍牙應用的各種可能。

6. 參考文獻

[1] Bluetooth Core Specification Version 4.1, 03 December 2013, Bluetooth SIG, Inc.

[2] Brian Wirsing, "Sending and Receiving Data via Bluetooth with an Android Device," March 26, 2014.

[3] Jeffrey Schiller, Franklyn Turbak, Mark Friedman, Hal Abelson, Johann Okerlund, José Dominguez, Andrew McKinney, "Live Programming of Mobile Apps in App Inventor," PROMOTO 2014, October 2014.

[4] <https://www.bluetooth.com/media/our-history>

[5] <https://www.abiresearch.com/market-research/product/1021794-bluetooth/>

[6] <http://developer.android.com/guide/topics/connectivity/bluetooth.html>

[7] <http://android-developers.blogspot.tw/2014/12/android-studio-10.html>

[8] <http://ai2.appinventor.mit.edu/>

[9] http://appinventor.mit.edu/explore/sites/all/files/hourofcode/TalkToMePart1_2perpage.pdf