**國科會自由軟體專案『系統安裝使用手冊』**

**System Installation Using Manual**

**NSC Open Source Project**

**開放原始碼手機上感測網路代理人之自由軟體系統開發-以個人遠端健康管理應用為例**

**Open source development of WSN agent on Android phone-A Remote Healthcare System Development**

**NSC 99-2220-E-002-029**

**主持人: 潘育群 助研究員**

**台灣大學計算機及資訊網路中心**

**共同主持人:**

**張瑞益 副教授**

**台灣大學工程科學及海洋工程學系**

**王家輝 助理教授**

**銘傳大學資訊工程學系**

**Department of Engineering and Applied Science**

**National Science Council, Taiwan**

**2011/7/31**

目錄

[一. 系統簡介 3](#_Toc298114008)

[1.1 手冊內容 3](#_Toc298114009)

[1.2 手冊範圍 3](#_Toc298114010)

[二. 事前準備 3](#_Toc298114011)

[2.1 可安裝及建議安裝的作業系統或平台 3](#_Toc298114014)

[2.1.1 Linux Ubuntu 作業系統 3](#_Toc298114015)

[2.1.2 Android 作業系統 3](#_Toc298114016)

[2.2 必要的系統環境設定需求 4](#_Toc298114017)

[2.2.1 Apache環境設定 4](#_Toc298114018)

[2.2.2 在 Ubuntu 建立 Android 的編譯環境 13](#_Toc298114019)

[2.3 必要安裝的軟體套件及說明 15](#_Toc298114020)

[2.3.1 Jennic CodeBlocks 開發軟體 15](#_Toc298114021)

[2.4 需額外安裝的軟體套件及說明 16](#_Toc298114022)

[2.5 已知的問題 16](#_Toc298114023)

[三. 安裝步驟與使用方式 16](#_Toc298114024)

[3.1 安裝步驟及注意事項 16](#_Toc298114028)

[3.2.1 自由軟體手機與無線感測網路接合安裝步驟 16](#_Toc298114029)

[3.2.2 感測儀器與人體健康感測程式的結合 19](#_Toc298114030)

[3.2 使用方式及注意事項 23](#_Toc298114031)

[3.2.1 Openmoko及感測儀器的使用方式 23](#_Toc298114032)

[3.2.2 網頁的使用方式 26](#_Toc298114033)

[3.2.3 接收手機傳送感測資訊所使用的系統 28](#_Toc298114034)

[四. 套件設定 29](#_Toc298114035)

[五. 常見問題 29](#_Toc298114036)

# 系統簡介

## 手冊內容

本手冊將針對「開放原始碼手機上感測網路代理人之自由軟體系統開發-以個人遠端健康管理應用為例」功能逐一說明系統架構、各項作業流程及使用者操作方式。並配合相關的畫面圖示以協助本系統操作人員在最簡單、容易的狀態下，熟悉本系統之功能操作介面及使用方式。

## 手冊範圍

本使用手冊分為三大部分:

* 系統簡介:說明使用手冊之內容、範圍。
* 事前準備: 說明軟體系統之安裝環境以及使用者使用此軟體系統需注意之事項。
* 安裝步驟與使用方式:說明各項功能操作及注意事項。

# 事前準備



## 可安裝及建議安裝的作業系統或平台

### Linux Ubuntu 作業系統

為了實現自由軟體與無線感測網路結合，我們必須經由Ubuntu作業系統在自由軟體手機上建立一個支援 USB-to-RS232 的平台，使得該平台可以透過串列埠與嵌入式硬體設備溝通。

### Android 作業系統

我們使用的自由軟體手機平台為 Openmoko 計畫的Neo Free Runner(Gta02)，手機則採用 Android 1.5 作業系統。

## 必要的系統環境設定需求

### Apache環境設定

Apache HTTP Server（簡稱Apache）是Apache軟體基金會的一個開放原始碼的網頁伺服器，可以在大多數電腦作業系統中執行，由於其跨平台和安全性被廣泛使用，是最流行的Web伺服器端軟體之一。首先下載AppServ

**AppServ 2.5.10**

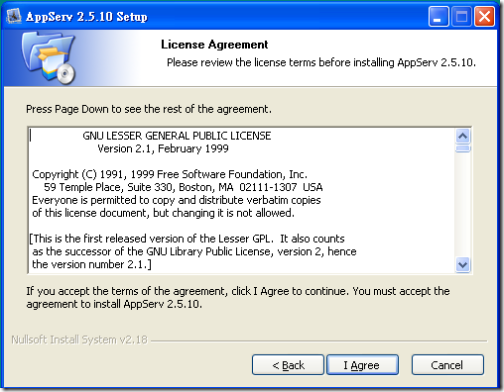
* Apache 2.2.8
* PHP 5.2.6
* MySQL 5.0.51b
* phpMyAdmin-2.10.3

**Download  
Sourceforge.net :** <http://prdownloads.sourceforge.net/appserv/appserv-win32-2.5.10.exe?download>

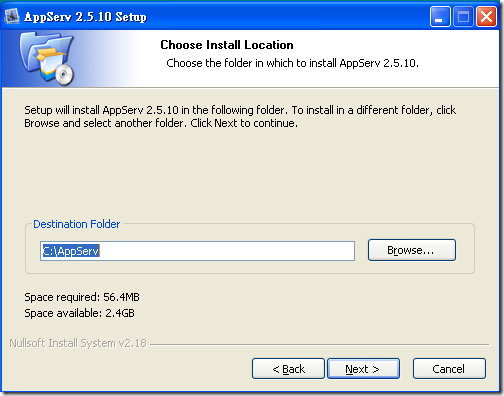
然後double Click程式



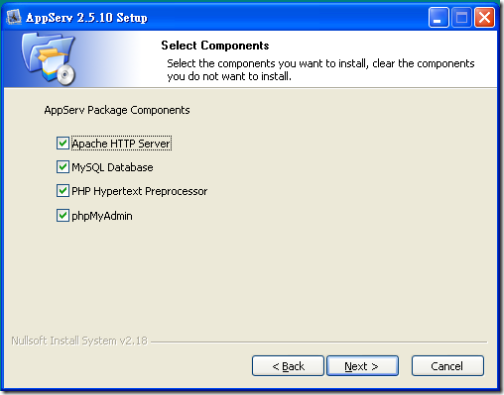
選Next



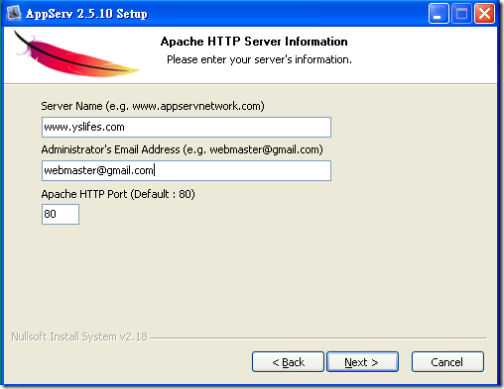
服務條款，選I Agree



選擇安裝路徑按Next



然後選擇要安裝的套件，基本上全選會比較方便使用，否則安裝好後還需要做點設定才可以，建議全選，然後Next

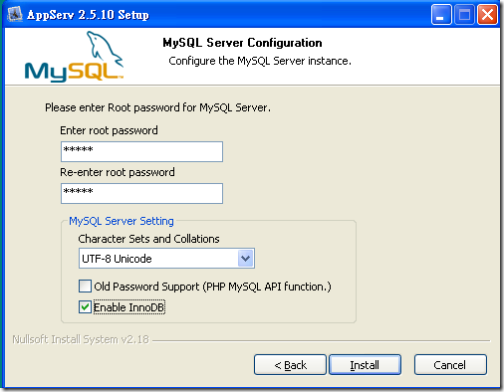


然後設定你的Domain Server Name，如何安裝DNS Server可以參考

[DNS Server –BIND 9 in Windows XP PART1](http://blog.yslifes.com/archives/77)

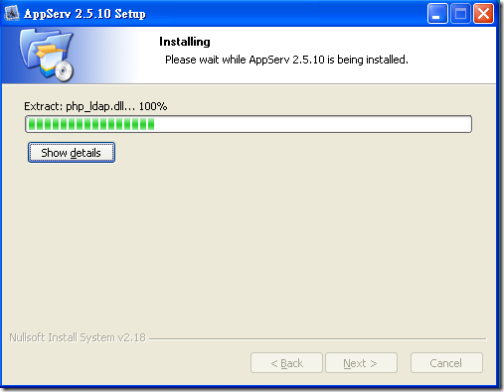
[DNS Server –BIND 9 in Windows XP PART2](http://blog.yslifes.com/archives/114)

還有你的Email及要使用的port，一般都是使用80 port



再來設定你的Mysql 管理者的密碼，這同時也會是你登入phpMyAdmin的密碼（Web化管理Mysql的工具）

這是我們把InnDB enable起來，當然事後再enable也可以，編碼的部份Character Sets and Collations非常建議使用UTF-8 Unicode，省的以後開發程式時發生中文語系問題



然後就開始安裝啦

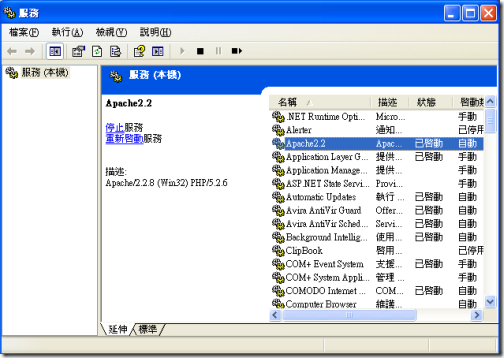


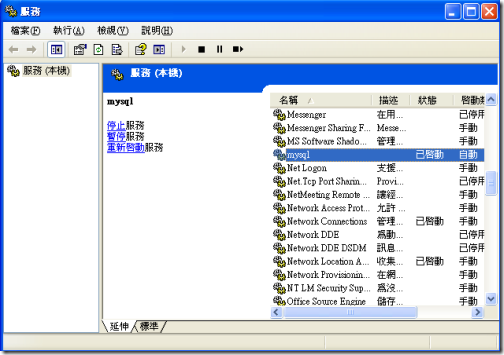
安裝好後，如果你有防火牆FireWall，記得要把對外的port 80打開，或解除封鎖



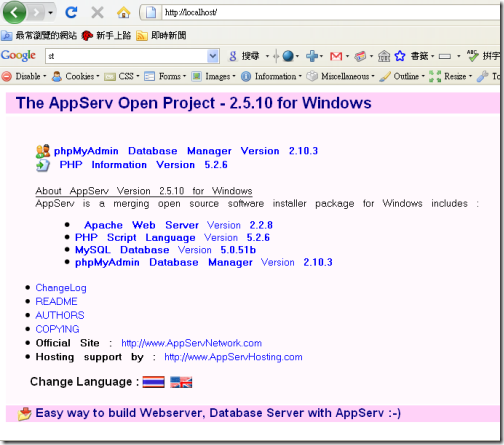
然後完成後，選Finish

Start Apache及Start MySQL我們都打勾



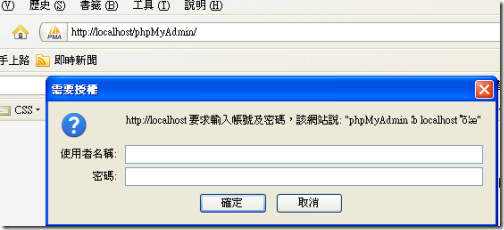


然後可以在，控制台－＞系統管理工具－＞服務 裡找到剛才選的二個服務已經啟動了



然後打開你的Browser，輸入http://localhost 或是http://yourdomain

就可以看上面畫面了。



你也可以輸入http://localhost/phpMyAdmin 或是http://yourdomain/phpMyAdmin就會出現帳號及密碼的對畫框，帳號root，密碼則是剛才在安裝過程設定的MySQL密碼



這樣子就可以對MySQL做控制了

以上就是AppServ的安裝過程

### 在 Ubuntu 建立 Android 的編譯環境

Ubuntu 的系統安裝好之後，開始進行下載 Android 的原始碼。首先安裝 Repo，Repo是一個Git(GNU Interactive Tools)的工具，而 Git 則是分散式的版本控制工具。使用 Repo 可以更容易的取得 Android 的 原 始 碼 ， 有 關 於 Repo 的 更 多 資 訊 ， 請 看 Using Repo and Git(http://source.android.com/download/using-repo)。接下來依照下列步驟進行安裝、初始化與配置Repo：

在家目錄建立一個 bin 資料夾，並且將 bin 資料夾加入到 PATH 中：

$ cd ~

$ mkdir bin

$ vi ~/.bashrc

在最後一行加入 export PATH=$PATH:/home/username/bin/

Username為你的使用者名稱

在進行以下動作及編譯必頇先設定你的編譯環境、安裝套件。首先安裝所需的套件，在Ubuntu 中執行下列安裝指令。

$ sudo apt-get install git-core gnupg flex bison gperf \

libsdl-dev libesd0-dev libwxgtk2.6-dev build-essential zip curl \

libncurses5-dev zlib1g-dev liblzo2-dev libx11-dev uboot-mkimage \

fakeroot curl

下載 repo script 並確保有執行權限：

$ curl <http://android.git.kernel.org/repo> > ~/bin/repo

$ chmod a+x ~/bin/repo

Ununtu 9.10 使用者可以加入下列兩行到/etc/apt/sources.list，並執行第三行來安裝需要的套件：

deb http://us.archive.ubuntu.com/ubuntu/ dapper main restricted

deb http://us.archive.ubuntu.com/ubuntu/ dapper universe multiverse

$sudo aptitude install gcc-4.3 g++-4.3 liblzo2-dev uboot-mkimage sun-java5-jdk

初始化你的 repository，使它連接到目前的開發重點版本 cupcake。由於我們是在 Openmoko 的 freerunner 進行開發，所以下載分支版本 android on freerunner(AOF)。

$ mkdir aof\_source

$ cd aof\_source

$ repo init -u

git://gitorious.org/android-on-freerunner/freerunner\_platform\_manifest.git -b

cupcake

在初始化的過程中，你可能會被詢問名字和 E-mail，這些將會用在之後註解的戳記。現在當你每次執行 Git 或 Repo的指令都會自動連結到 Cupcake版本。

進行原始碼同步，如果你是第一次同步，這個步驟將會花費相當長的時間。

$ repo sync

如果你的網路在防火牆後，可以透過 http proxy 取得 android 原始碼，以下是步驟：下載corkscrew tool，<http://www.agroman.net/corkscrew/corkscrew-2.0.tar.gz>。

$ tar zxvf corkscrew-2.0.tar.gz

$ cd corkscrew-2.0.tar.gz

$ ./configure

$ make

$ cp corkscrew ~/bin

建立一個 shell script „git-proxy‟內容如下：

#!/bin/sh

exec ~/bin/corkcrew proxy port $\*

以你的個人資訊取代 ~/bin/、proxy 和 port，執行下列指令：

$ git config –-global core.gitproxy ~/bin/git-proxy

在編譯之前，先設定編譯環境。你必頇先安裝 sun-java5-jdk，並且需要指定 Android Java變數到正確的位置與環境變數，如下：

$ export JAVA\_HOME= /usr/lib/jvm/java-1.5.0-sun

$ export ANDROID\_JAVA\_HOME=$JAVA\_HOME

$ export CC=gcc-4.3

$ export CXX=g++=4.3

開始編譯，-j 後面的參數根據編譯機器核心數量修改：

$ make –j 4 TARGET\_PRODUCT=freerunner

或著也可以寫一個 shell script：

$vi start.sh

export JAVA\_HOME=/usr/lib/jvm/java-1.5.0-sun

export ANDROID\_JAVA\_HOME=$JAVA\_HOME

export CC=gcc-4.3

export CXX=g++-4.3

make -j 4 TARGET\_PRODUCT=freerunner

## 必要安裝的軟體套件及說明

### Jennic CodeBlocks 開發軟體

利用這套Jennic CodeBlocks開發環境,我們可以自行編譯上所執行的人體健康感測資訊程式。



1. Jennic CodeBlocks開發環境

# 安裝步驟與使用方式



## 安裝步驟及注意事項

### 自由軟體手機與無線感測網路接合安裝步驟

關於自由軟體手機與無線感測網路接合安裝步驟，以下進行 2個步驟，一、修改 Android framework。二、編寫測試程式。

**步驟一：修改 Android framework。**

經過步驟一，我們已經將 Android 原始碼下載到 aof\_source 資料夾，接下來進行修改原始碼來達到我們的目的。首先，這份原始碼並沒有內建 RS232 的驅動程式，我們必頇修改 kernel設定來支援。

$ cd aof\_source/kernel/arch/arm/configs/

$ vi gta02\_android\_defconfig

#CONFIG\_USB\_SERIAL\_PL2303 ...

CONFIG\_USB\_SERIAL\_PL2303=y

大約在 1519 行，可以使用搜尋來尋找這行字，修改成下方那行即完成內建 RS232 驅動程式。不能設定成模組，因為此版本不會安裝驅動模組到系統當中。PL2303 是 USB-to-RS232 的晶片。

USB插到手機時會產生一個 device node，/dev/ttyUSB0，這個 node 的權限預設無法被一般程式讀寫，需要將它設定，讓一般程式可以讀寫。設定方法如下：

$ vi system/core/init/devices.c

在 mode\_t\_get\_device\_perm 的 function 中，大約是 268 行，將第一行改成第二行。

return 0600;

return 0777;

為了將來能夠方便切換 USB 模式，在 vendor/neo/freerunner/init.gta02.rc 中的on boot 底下加入以下兩行，讓一般程式也能讀寫的權限。

chmod 0777 /sys/class/i2c-adapter/i2c-0/0-0073/neo1973-pm-host.0/hostmode

chmod 0777 /sys/devices/platform/s3c-ohci/usb\_mode

**步驟二：撰寫測試程式**

在開始撰寫測詴程式之前，本文根據以下來做為開發的架構。此架構分為兩個部分，Java 與 C語言。Java 是 Android 應用程式與 Framework 的開發語言，Framework 上層的皆由 Java 構成。C語言則是在 Framework 下層與硬體溝通有密切的關係，專門負責硬體的控制並編譯成 library(\*.so)供 Java 語言透過 JNI 進行呼叫。JNI是兩種語言溝通的橋樑。以下要開發的測詴程式包含這兩個部分，Framework 上層與下層。開發方式是在 Android 原始碼裡建立自己的分支，並開發 Uart 應用程式、UartManger、UartService 以及由 C 語言撰寫成的 library 等部分。此一連串的開發將使自由軟體手機擁有與 Uart 溝通的功能，本文將以 Android 1.5 作為開發的版本。

1. Android程式運作架構

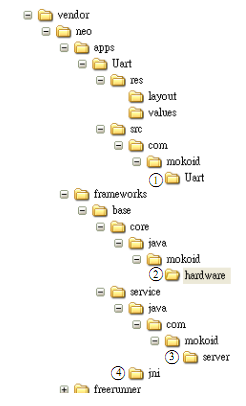
為 aof\_source/vendor/底下的資料夾樹狀圖，其中 neo 為 freerunner 的分支，我們就在這個分支底下進行開發，將分成四個部分由上而下進行開發。

第一部分，首先撰寫的是Android 應用程式 (App) 部分，應用程式名稱為 Uart 。 在 處 建 立 兩 個 Java 檔UartSystemServer.java、Uart.java。UartSystemServer.java 是有關 SystemServer 的部分，在其中它呼叫了 Android 四大元件之一的 Service，進行申請 Uart 的 Service。此後，便可使用 UartService所提供的 API 進行操作。Uart.java 則是此應用程式的核心部分，包含 UI 的連結控制、程式的運作邏輯與流程。主要是提供使用者一個操作介面，有個 edit view 顯示收到的訊息，並有按鍵控制送收與關閉。

第二部分進行的是 Framework 層的撰寫，目的在於建立UartManager 讓 App 能夠透過它呼叫 Service，先在○ 2處建立UartManager.java 與 IUartService.aidl。UartManager.java 是建立一個 class Interface，讓 UART 應用程式能夠透過UartManager class 去存取 UartService，其中 IUartService.aidl只是一個 Interface供 UartManger.java 使用。

第三部分，同屬於 Framework 層，由於 UartService並非內建於 Android Framework 所提供的 Service。 因此需要建立一個 UartService 供 UartManager 存 取 ， 在 ○ 3 建立UartService.java。此 UartService 撰寫了一些 API 並結合 C library 提供 Service 供上層呼叫。其中 C library 是透過 JNI與 UartService接合。

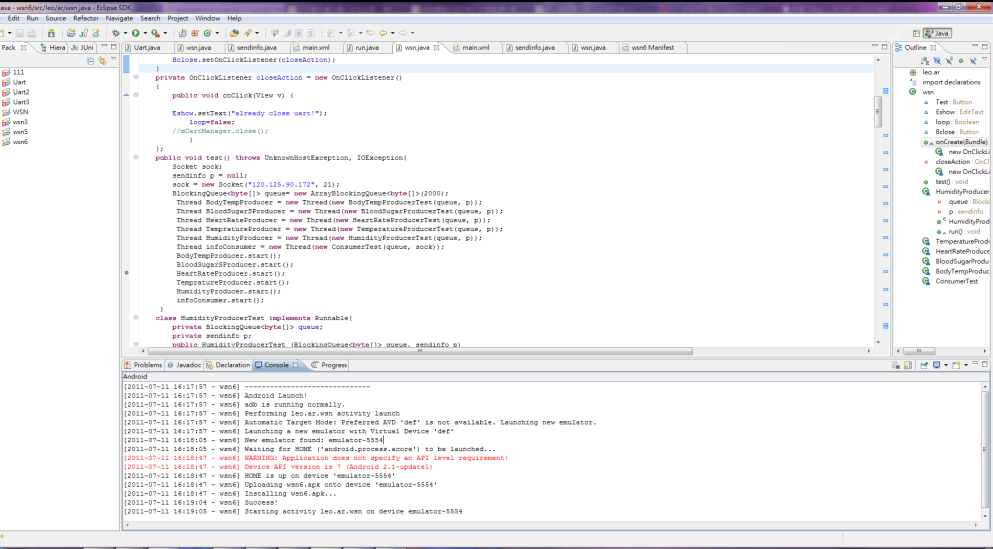
第四部分，C library由 C 語言撰寫並編譯成\*.so檔，它能夠往下與 kernel溝通達到控制硬體的功能。此部分建立一個 com\_mokoid\_server\_UartService.cpp，以 JNI的方式撰寫控制硬體的API，讓自建的 UartService 能夠去呼叫，內含 Uart 的開啟、讀取、寫入與關閉等 API。完成此四部分的撰寫便可達成 Uart 功能的實現，以上是簡略的開發測詴程式的一個流程，編譯以上程式碼之後便可將程式加到手機的 image 裡，將 image 刷入手機之後即可完成。



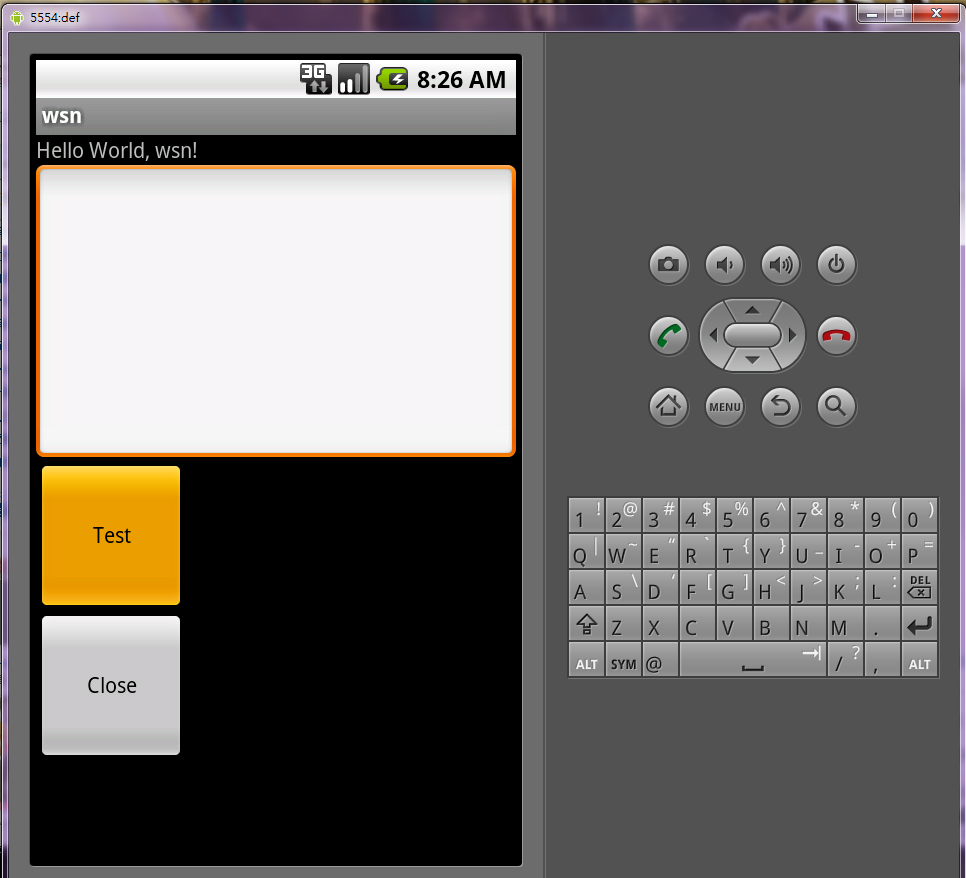
1. aof\_source/vendor/資料夾樹狀圖

### Android HTC 上虛擬資料的程式安裝設定

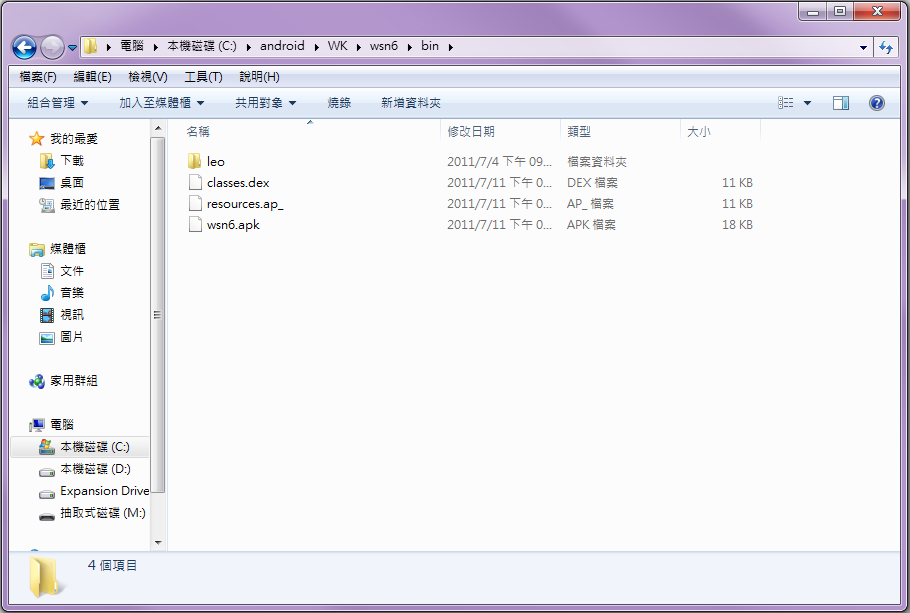
首先我們先將程式碼進行編譯



編譯結束後，就可在Android模擬器上看到執行的結果



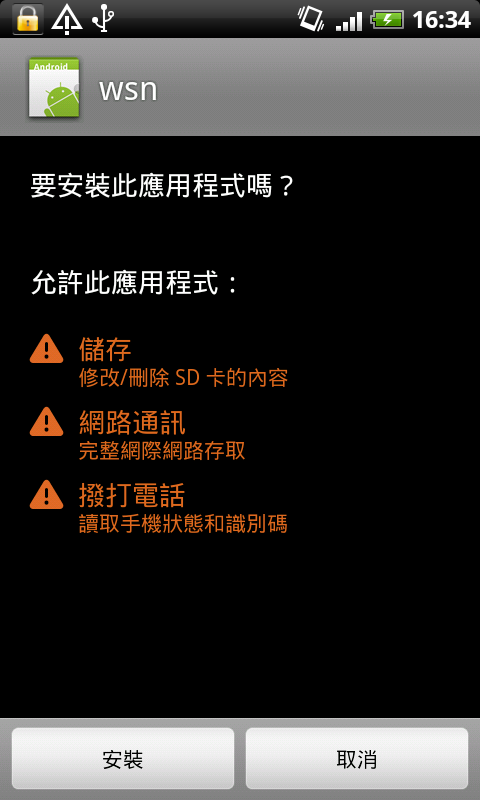
接下來我們將成是放到手機上進行實測，首先先找到編譯完成程式碼的資料夾，接著將資料夾中.apk的檔案放入置手機當中。



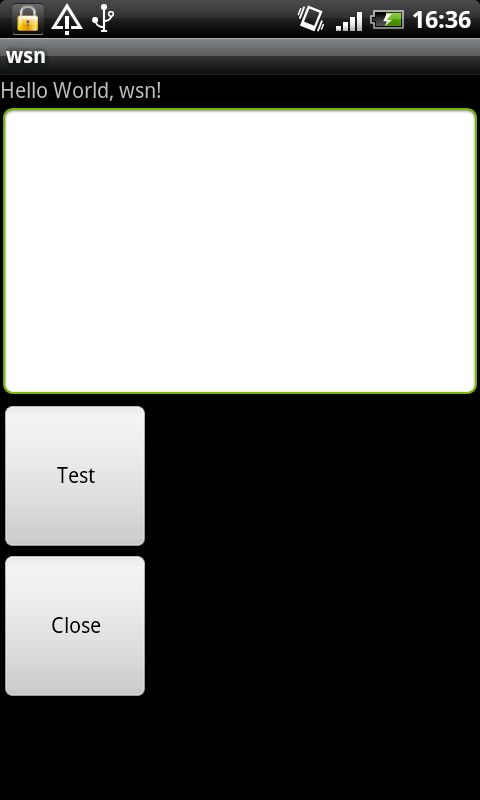
接著選取剛剛放進來的檔案



進行安裝的動作



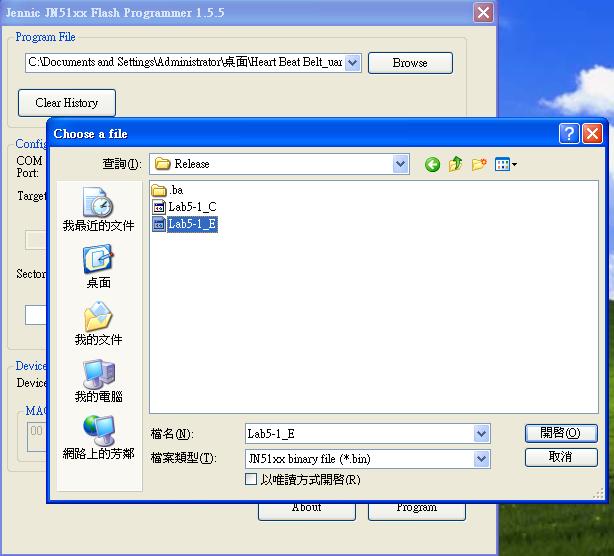
當安裝結束以後，系統就能開始執行虛擬資料的傳輸



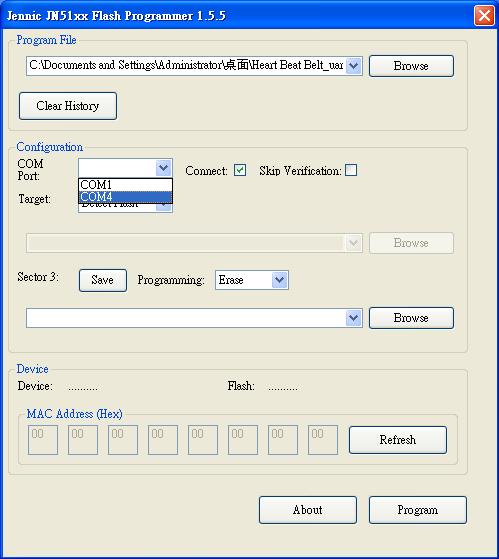
### 感測儀器與人體健康感測程式的結合

關於如何將感測儀器與人體健康感程式做結合，以下我們以圖表逐一介紹:

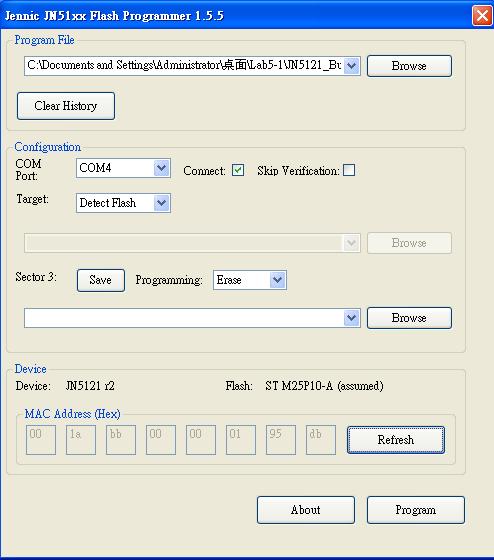
首先，我們先將要燒入的感測器與電腦做連接，並在燒入前進行感測儀器的斷電動作。接著，我們選取編譯完成的程式碼，



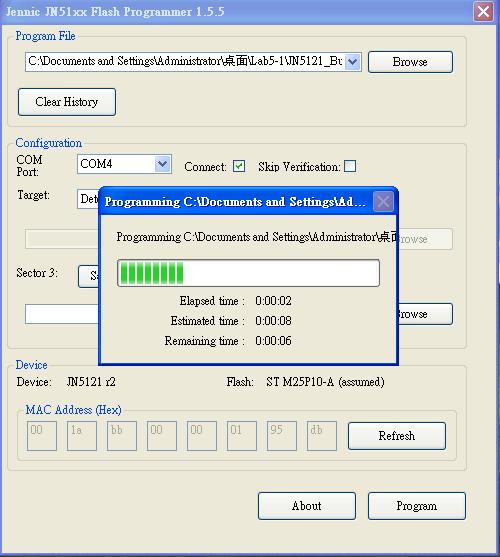
程式碼選擇完畢後，選擇適當的連接埠。



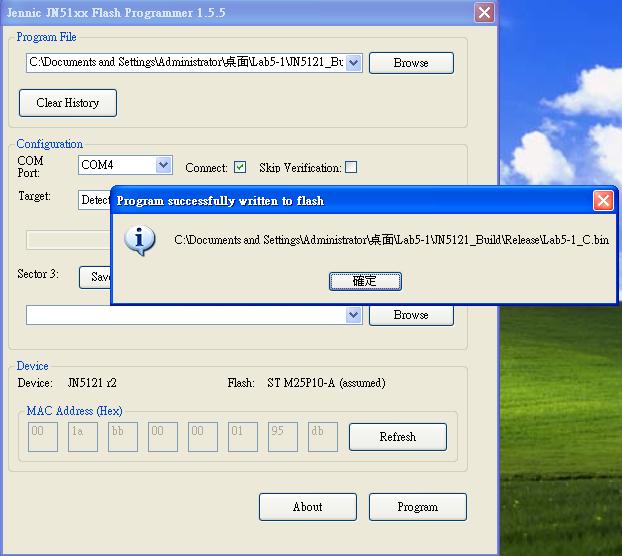
接下來，按一下Refresh這個按鈕(才能確保能正確燒入進去)



接著，按下Program這個按鈕就能開始燒入了



當燒入結束後，會出現燒入完成的視窗，點選確定後，燒入的動作即完成。

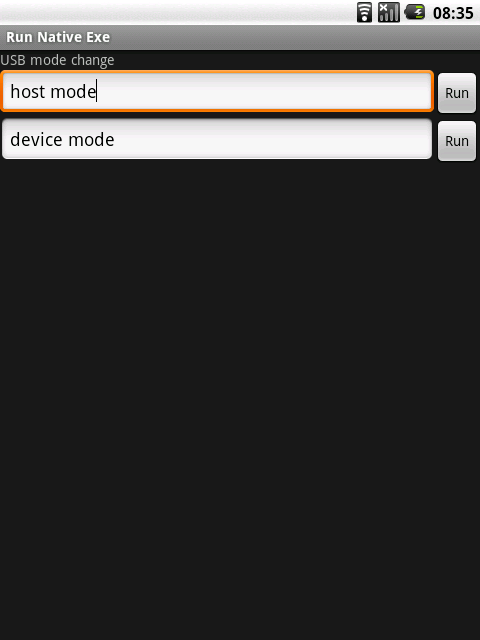


## 使用方式及注意事項

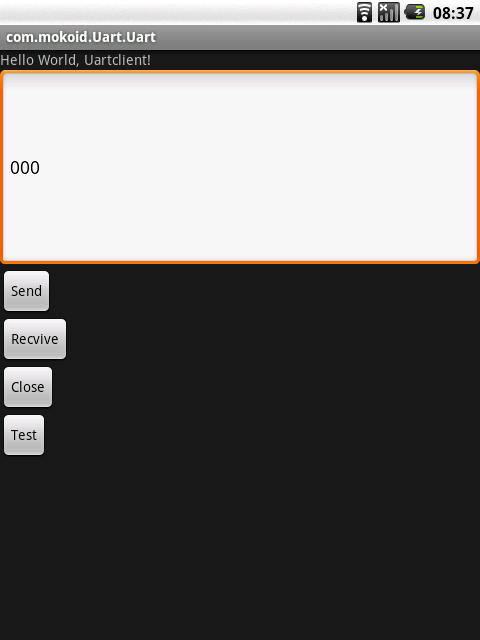
### Openmoko及感測儀器的使用方式

Openmoko的使用方式:

先將模式切換成hostmode



接著轉換到系統執行畫面



點選receive即可開始接收感測資訊

溫濕度感測儀器使用方法:

將感測儀器接起來(編號1為感測端，編號2為接收端)再將接收端經由RS-232連接到openmoko上即可。

下圖為溫濕度感測儀器



心跳感測儀器使用方法:

1

編號1為心跳感測胸帶，編號2為感測心跳資訊所需的擴充套件。將胸帶放置於胸口中間再依序將接收端經由RS-232與手機連結起來即可。

下圖為心跳感測儀器

注意事項:在使用心跳感測器時，需將背面感測區域用水沾濕。



2

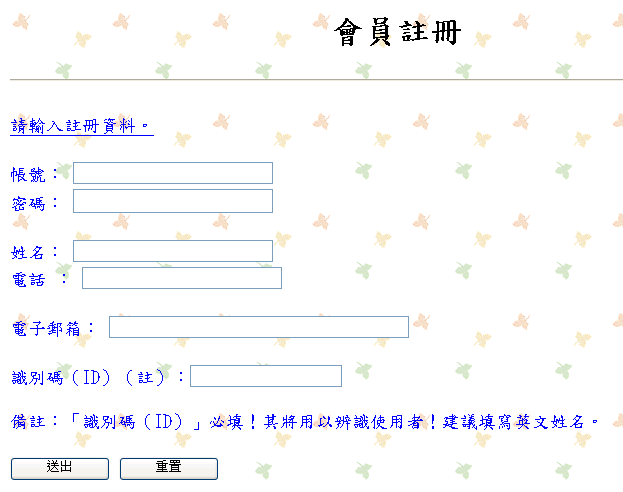
1

### 網頁的使用方式

使用者可透過這個網頁進行登入及註冊的動作



註冊畫面



登入畫面

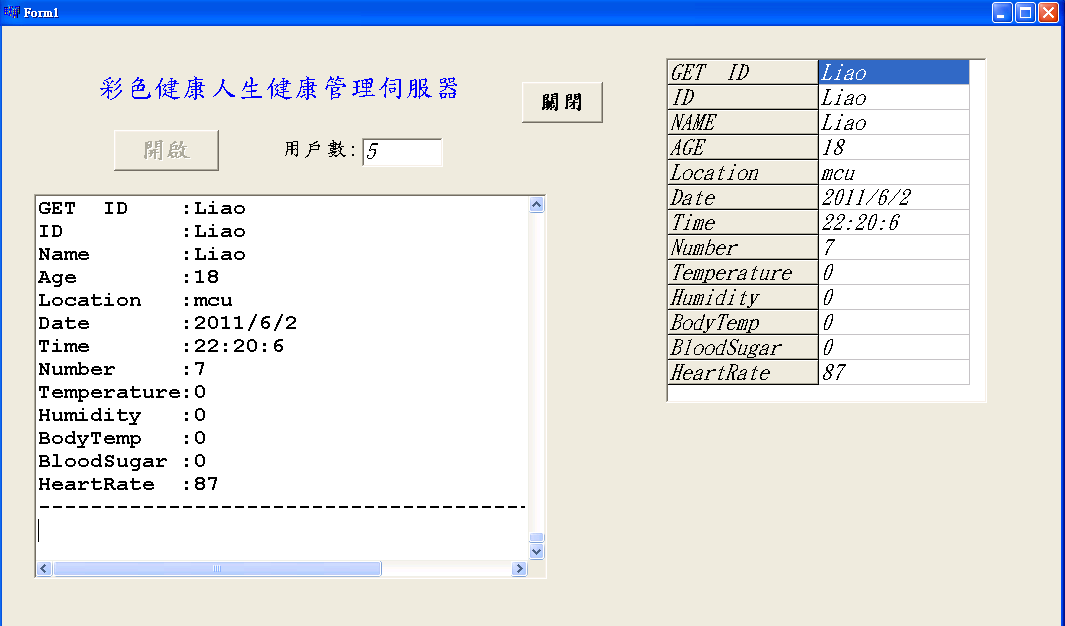


登入成功即可進行人體健康感測資訊的查詢



### 接收手機傳送感測資訊所使用的系統

下圖為接收人體健康感測資訊，並上傳至資料庫之系統



3

2

1

首先輸入最大連接用戶數(編號1)，接著就可以開始執行，執行畫面(編號2)會顯示當下收到的人體健康感測資訊，當要求結束執行時，可以使用關閉(編號3)來關閉程式。

# 常見問題

* 當手機接收不到感測資料時，要怎麼辦?

1. 重新將手機轉換成host mode。
2. 將感測器進行斷電在重新接上的動作
3. 重新啟動整個系統。

* 當傳送感測資料時沒有反應，要怎麼辦?

1. 確定是否有確實連接上網路。
2. 確定Server端是否有開啟，且是否達到最大用戶端之數量。

* 為什麼我無法將系統燒入置感測裝置中?

1. 連接埠選擇不正確。
2. 沒有進行感測儀器的斷電動作。