『物聯網概論』實驗手冊

實驗模組名稱: 智慧手機與閘道器整合之燈控系統實作

開發教師: 張志勇教授 (cychang@mail.tku.edu.tw)

開發學生: 謝嘉濬 (699410261@s99.tku.edu.tw)

高正豪 (600420409@s00.tku.edu.tw)

學校系所: 淡江大學資訊工程學系

實驗平台: Octopus II

實驗主軸: 智慧手機與異質網路閘道器整合

實驗內容關鍵字: Octopus II ,Tiny OS , Microwave motion sensor module

本實驗手冊乃『教育部網路通訊人才培育先導型計畫-課程發展計畫』支持所發展之實驗手冊

實驗目的

本實驗整合 Octopus II 嵌入式感測器、Arduino 實驗平台以及藍芽無線傳輸模組。利用 Android 手機透過藍芽無線傳輸模組，下達命令給 Octopus II 嵌入式感測器、Arduino 實驗平台以及藍芽無線傳輸模組所組成的 Gateway(閘道器)，送命令給所控制的電燈，而 Android 手機的軟體介面將對電燈下達開關電燈之命令，藉此與使用者達到互動的效果。本實驗將可讓學生練習 Android 軟體使用，以及 Arduino 和 Octopus II 程式寫作並利用 Arduino 和 Octopus II 輸入與輸出腳位之電路連結與控制，我們利用藍芽無線模組當作 Android 手機的輸入，並利用電燈當作 Gateway 的輸出，透過本實驗，同學將可以更加熟悉 Octopus II 嵌入式感測器、Arduino 實驗平台的操作、控制以及如何透過輸出入腳位整合其他模組。

實驗器材

- Octopus II 無線感測節點
  - 中央處理器使用 TI 公司的 MSP430 處理器
  - 超低功耗架構，可延長電池使用時間
  - 儲存單元使用 ST 公司的 MP25P80 快閃記憶體，容量為 1MB
  - 具有 50 個擴充腳位，可整合其他感測器與模組

- Arduino 實驗平台
  - 基於 創用 CC 開放原始碼的電路圖設計
  - Arduino 可使用 ICSP 線上燒入器，將 Bootloader 燒入新的 IC 晶片。
  - 可依據官方電路圖，簡化 Arduino 模組，完成獨立運作的微處理控制。
  - 可簡單地與感測器，各式各樣的電子元件連接，如 紅外線、超音波、熱敏電阻、光敏電阻、伺服馬達…等。
  - 支援多樣的互動程式，如 Adobe Flash, Max/MSP, VVVV, Pure Data, C, Processing… 等。
  - USB 介面，不需外接電源。另外有提供 9V 直流電源輸入
  - 使用低價格的微處理控制器 (ATMEGA 8–328)。

- 藍芽無線傳輸模組
  - IC 為 class 1 bluegiga WT11
  - 啟動電壓為 1.2V
  - 傳遞距離為 100 公尺
  - 增加兩倍的資料寫入量，而且有 14KB 的儲存空間
  - 電源可由 USB 供電或將 5V 直流電源連結於 Vcc 供電
  - 提供另外兩組 analog 輸入端，總共八個類比輸入

實驗所需軟體

- PC
  - Windows XP
  - Java 1.5 (Java 5) JDK 以上
  - TinyOS
  - Cygwin
  - MSP430 tool chain
  - nesC 編譯器
  - OctopusII 驅動程式
  - OctopusII 燒錄程式
  - OctopusII 編譯文件
  - VS1000 燒錄程式
  - Eclipse 程式語言
  - Arduino 編譯器

Table of Contents

Part I 簡介 Octopus II、Arduino 和周邊整合裝置 5
Step1 瞭解 Octopus II 5
Step2 瞭解 Arduino 6
Step3 瞭解藍芽無線傳輸模組 7
Step4 瞭解 Android 實驗平台開發環境 9

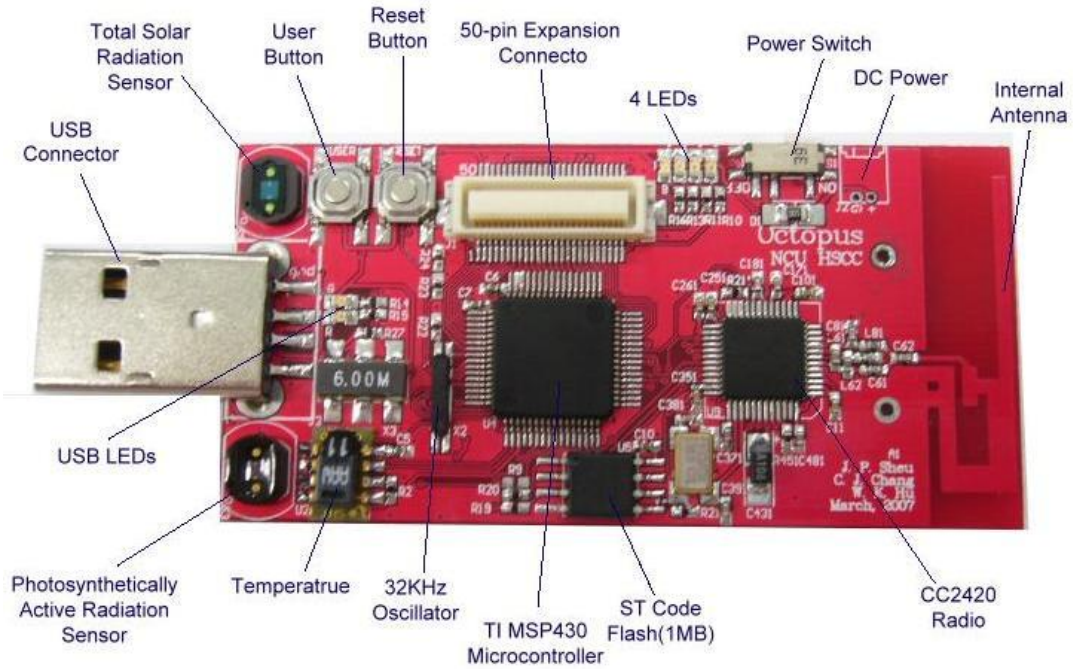
Part II 環境建置 10
Step1 介紹實驗所需之軟體 10
Step2 安裝 JAVA 10
Step3 安裝 Cygwin1.7 (線上安裝) 16
Step4 安裝 TinyOS 2.1 27
Step5 安裝 Eclipse 34
Step6 安裝 Arduino 41

Part III 硬體電路整合 43
Step1 實驗架構流程圖 43
Step2 選擇支援 GPIO 與 UART 的腳位 44
Part IV 設計與製作智慧手機與閘道器整合之燈控系統 45
Step1 編譯 Octopus II 程式碼 45
Step2 燒錄 Octopus II 程式 45
Step3 將程式安裝至 Android 手機上 ..... **錯誤! 尚未定義書籤。**
Part V 附錄程式碼 50

Part I 簡介 Octopus II 和周邊整合裝置

Step1 瞭解 Octopus II

Octopus 是由清華大學和中央大學團隊所共同研發的無線感測器，目前已開發出三個版本：Octopus I、Octopus II 以及 Octopus X，三個版本所使用的無線通訊模組皆為 CC2420 RF 晶片模組，第一版的 Octopus I 參照 Crossbow 的 MICAz 感測節點為其主要開發架構，並搭載 8-bit AVR MCU 為中央處理單元，具有 4 KB 的記憶體空間以及 128 KB 的快閃記憶體容量；第二版的 Octopus II 則參照 MoteIV 的 Tmote Sky 感測節點為其主要開發架構，搭載 16-bit MSP430 MCU 為中央處理單元，具有 10 KB 的記憶體空間以及 1MB 的快閃記憶體容量，相較於前一代的 Octopus I 具有更強的運算能力與更多的資料儲存空間；第三版的 Octopus X 全部架構皆為自行設計，其主要訴求為使感測節點體積變得更為輕巧，搭載了 CC2430 整合型晶片(整合了 8051 MCU 與 CC2420 RF)，使 Octopus X 感測節點體積縮小為 28mm×28mm×7mm，約為 Octopus II 體積(65mm×31mm×15mm, Octopus II－A)之 1/4 大小。



本範例實驗採用 Octopus II 感測節點，如上圖所示，其中央處理器晶片與儲存單元分別採用了 16 位元的 MSP430 RISC 處理器以及具有 1MByte 容量的 MP25P80 快閃記憶體。Octopus II 系列可分為 Octopus II－A 和 Octopus II－B，其尺寸分別為 65mm×31mm (Octopus II－A)及 80mm×31mm (Octopus II－B)，兩者皆採用雙天線設計，Octopus II－A 的傳輸距離可以達到 200 公尺，Octopus II

－B 比 Octopus II－A 多了 RF 訊號放大處理，因此 Octopus II－B 的傳輸距離可藉此增加到 450 公尺。這兩款 Octopus II 系列在訊號強度和傳輸距離上，均比市面上同類型無線感測器平台有更佳的線性化。

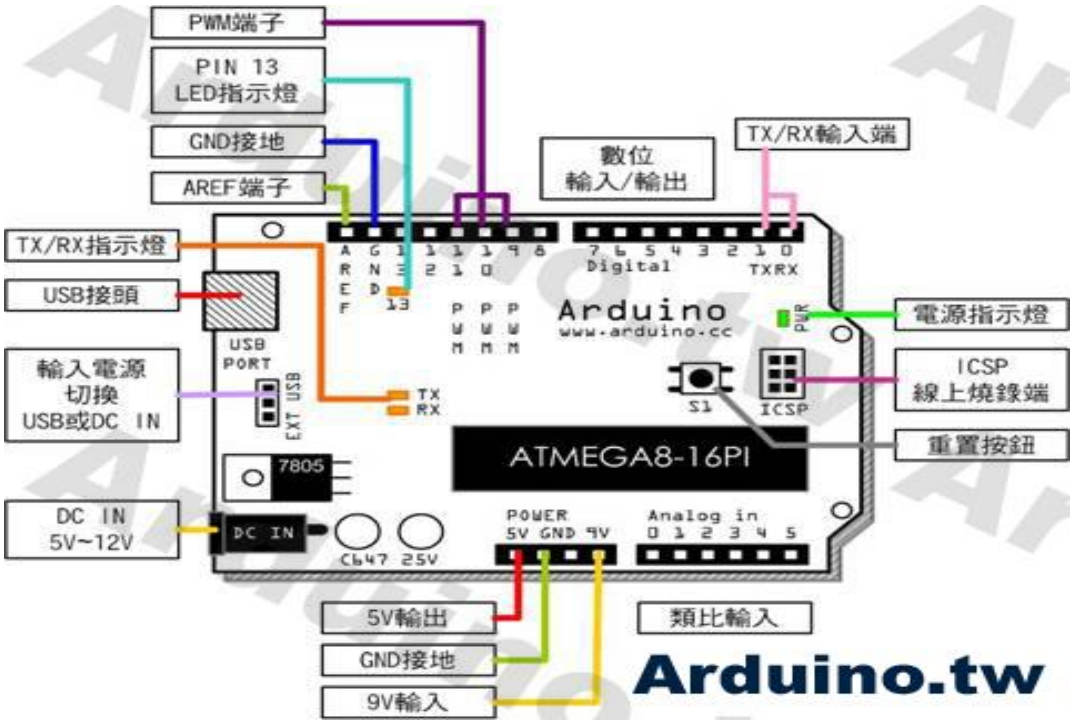
在感測器的部分，Octopus II 也採取跟 Octopus X 系列相同的分離式模組設計，採用 MICAz 的 PCB 線路設計，設置了 50 個擴充腳位，讓使用者可以靈活的運用以符合各種需求，藉此增加無線感測器平台的擴充性，而其感測器擴充板包含了四種基本的感測器，分別為溫溼度感測器、光感測器、陀螺儀和三軸加速度計。

在電力管理的部分，Octopus II 的電力來源主要是兩顆 3 號電池，而 MSP430 處理器提供了 5 種省電模式，因此，使用者可藉由進入省電模式延長無線感測節點的使用時間，而其中一種省電模式更可以藉由關掉所有震盪器功能，讓 MSP430 達到極低的 0.2 μA 耗電量。

與 PC 端的橋接的部分，Octopus II 使用了 FTDI 的 FT232BL 為其 USB 與 UART 橋接晶片，並與處理單元 MSP430 F1611 MCU 銜接，只須將 Octopus II 的 USB 端直接插入 PC，再配合相關應用程式，便可進行程式燒錄及除錯。

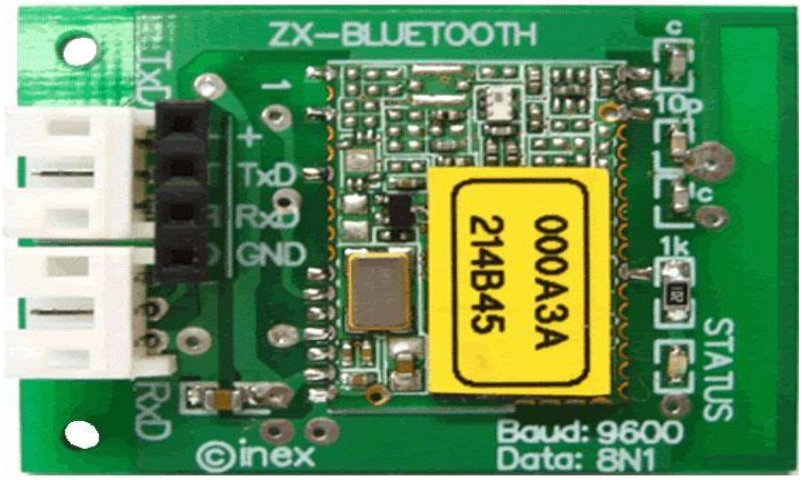
Step2 瞭解 Arduino 實驗平台

Arduino是由義大利高科技設計學校的老師和西班牙籍晶片工程師所共同研發的實驗平台，Arduino可以獨立運作成為一個可以跟軟體溝通的介面，Arduino採用Atmel Atmega8/168/328 單晶片，參照AVR-GCC編譯器與avr-libc, AVRlib函式庫與Processing程式語言為基礎 ，並搭載8-bit AVR MCU為中央處理單元，具有1 KB的記憶體空間以及16 KB的快閃記憶體容量；其主要訴求讓我們可以快速使用 Arduino 語言與 Macromedia Flash, Processing, Max/MSP, Pure Data, SuperCollider…等軟體，作出互動作品。Arduino可以使用開發完成的電子元件例如Switch或感測器或其他控制器件、LED、步進馬達或其他輸出裝置。
Arduino大小尺寸：寬70mm X 高54mm。



Step3 瞭解藍芽無線傳輸模組

藍芽無線傳輸模組主要功能在於利用9600 bps speeds and 8N1 data transfer與其他藍芽裝置進行通訊，並且只需要連接RX, TX 就可以直接傳資料, 不需額外設定程式；使用者, 程式與原來的Serial傳輸程式一樣，可以不用修改，且與其他藍芽協定的裝置相容，所以常常應用於其他藍芽裝置進行配對，使用UART通訊。



低單價，適用任何微處理器(包括 BASIC Stamp, Arduino, Atmel, 8051，等等) 與其他藍牙裝置連線，體積小 : 3cm x 4cm，標準 TTL 電壓輸入 5V，最大有效通訊距離 30 公尺。

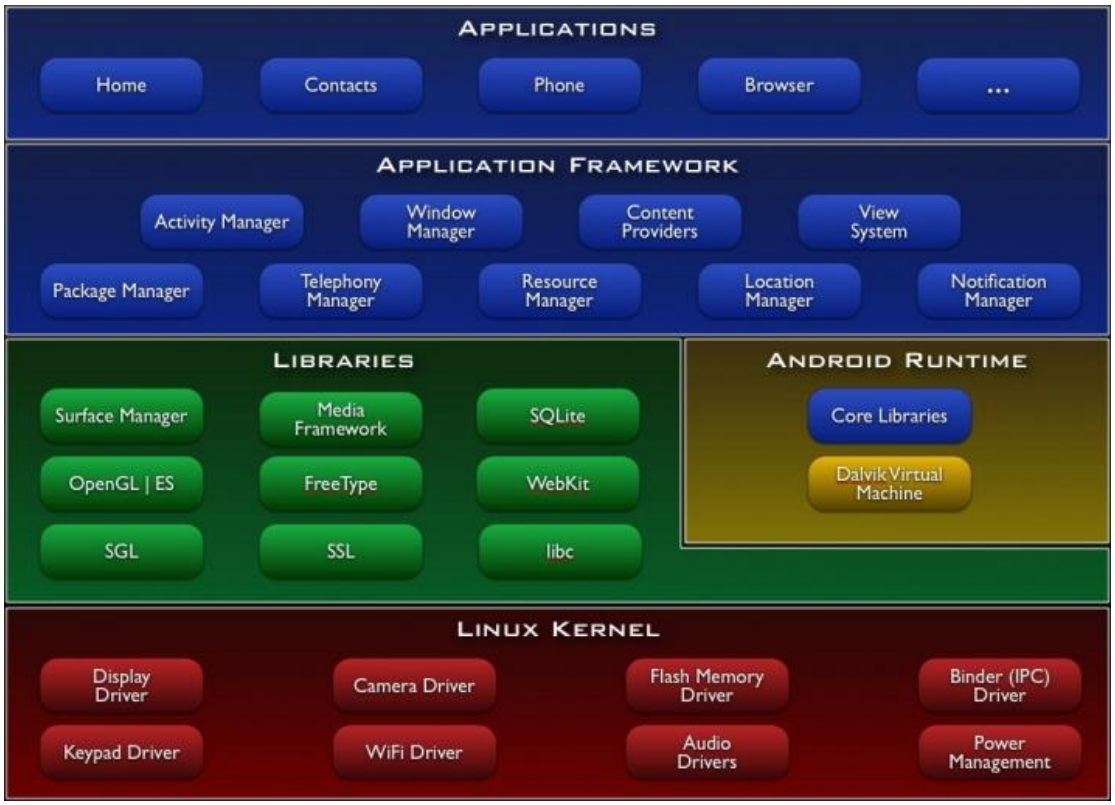
以下為 X-Band Motion Detector 之特色與規格：

- 內建天線
- 最大傳輸距離 30M
- 支援串列傳輸 Serial Port Profile (SPP)
- 9600 bps speeds and 8N1 data transfer
- 只需要連接 RX, TX 就可以直接傳資料，不需額外設定程式；使用者，程式與原來的 Serial 傳輸程式一樣，可以不用修改
- 使用 UART 通訊
- 本身只能當成 Slave bluetooth 通訊
- 可以於其他藍牙協定的裝置相容
- 標準輸入電壓 +5v
- Able to work with Computers with SSP Profile based Bluetooth Devices.
- Compatible with ESD-02 and ESD100 Devices
- 尺寸: 3cm by 4cm

Step4 瞭解 Android 實驗平台

Android 實驗平台是以 Linux 作業系統為基礎，擴充 Android 執行作業環境(Android Runtime)和應用程式開發平台(Application Platform)，Android 作業環境包括有：

- 應用程式(Application)
- 應用程式開發平(Application Platform)
- Android 執行作業環境(Runtime)和各類程式庫(Libraries)
- Linux 作業系統核心(OS Kernel)和各類驅動程式(Drivers)



參考於 http://sites.google.com/site/io/anatomy--physiology-of-an-android

Android 應用程式是採用 Java 語言編寫的，是用戶和設備的人機介面，提供用戶可以操作各式的應用功能。目前 Android 作業系統本身已提供基本桌面(Home) 、電話簿(Contact) 、數位電話(Phone)和網頁瀏覽器(Browser)等核心應用程式。開發人員採用 Android 開發平台所開發出來的應用程式也安置在這一層。

Part II 環境建置

Step1 介紹實驗所需之軟體

- Java SE Development Kit
  提供 TinyOS 的執行引擎，並且提供了 JAVA 用戶端程式用以監控感測器目前所感測的資料。

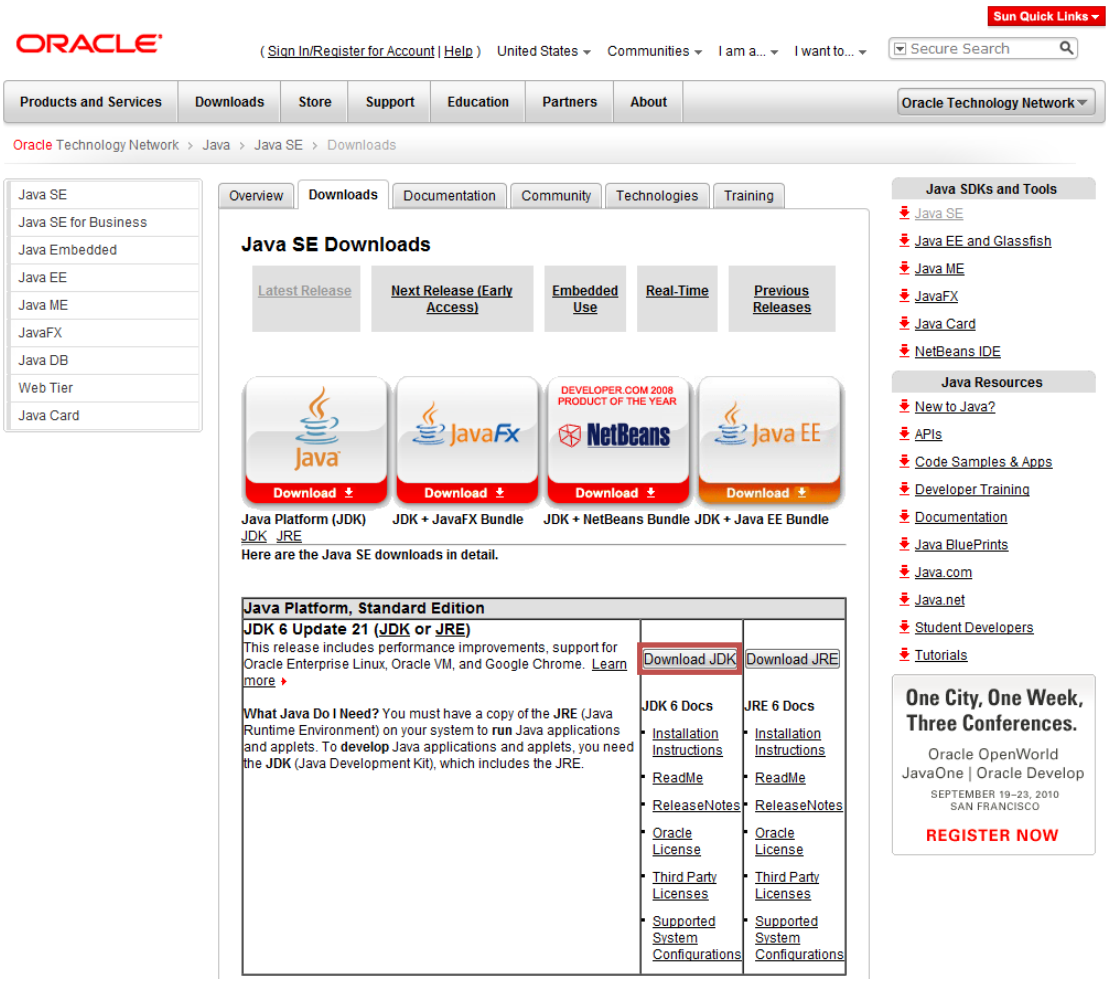
- Cygwin (with TinyOS)
  提供在 Windows 下模擬 UNIX 的 shell 環境以及開發 TinyOS 時需使用的 UNIX 工具(如 perl 和 shell 腳本)。在此模擬環境下，程式開發者將可直接使用 make 指令編譯 TinyOS 程式碼，並可在模擬環境下監控感測器所傳回之資料。

- Octopus II Programmer
  這是一套由清華大學和中央大學團隊共同開發之 Octopus II 燒錄程式，能將編譯完成後所產生的 TinyOS ihex 檔案直接利用 USB 燒錄到 Octopus II。

Step2 安裝 JAVA

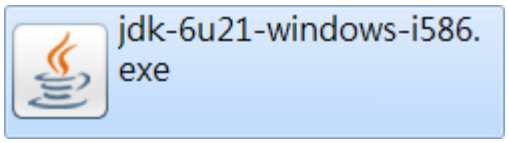
由於 TinyOS2.x 需要有 JAVA VM 環境的支援，因此我們將先介紹 JAVA JDK 之安裝。

- JAVA JDK 需使用 1.5 以上的版本
- 下載連結：
  http://www.oracle.com/technetwork/java/javase/downloads/index.html

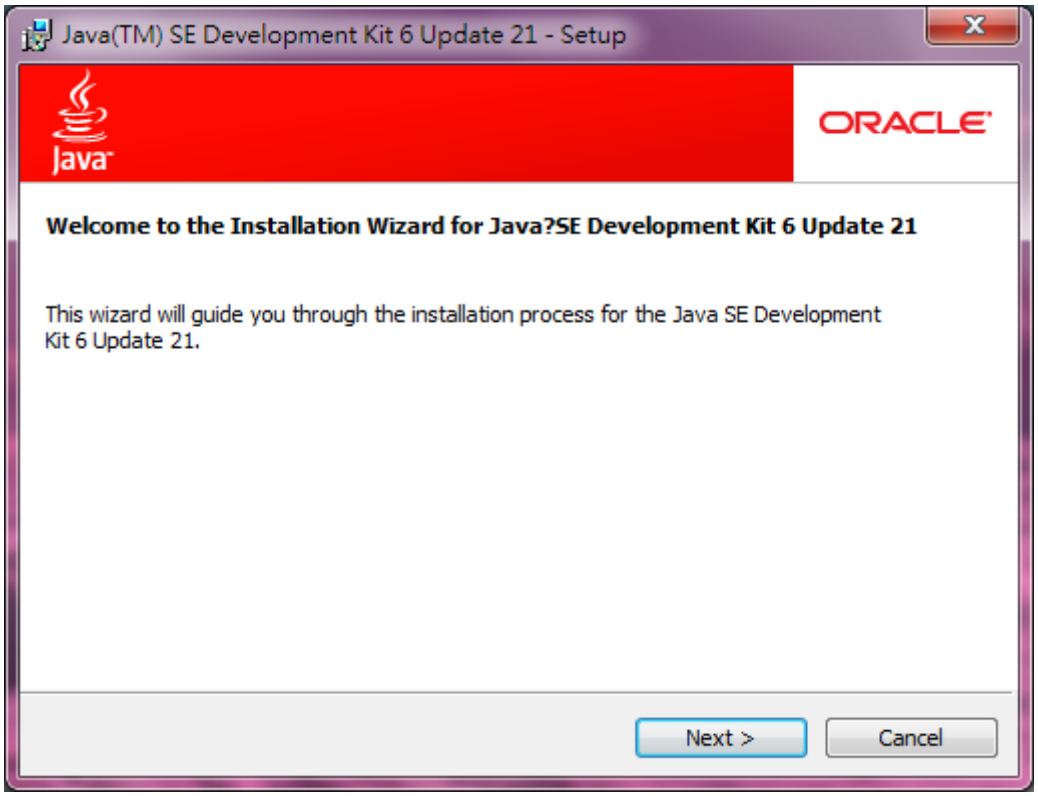


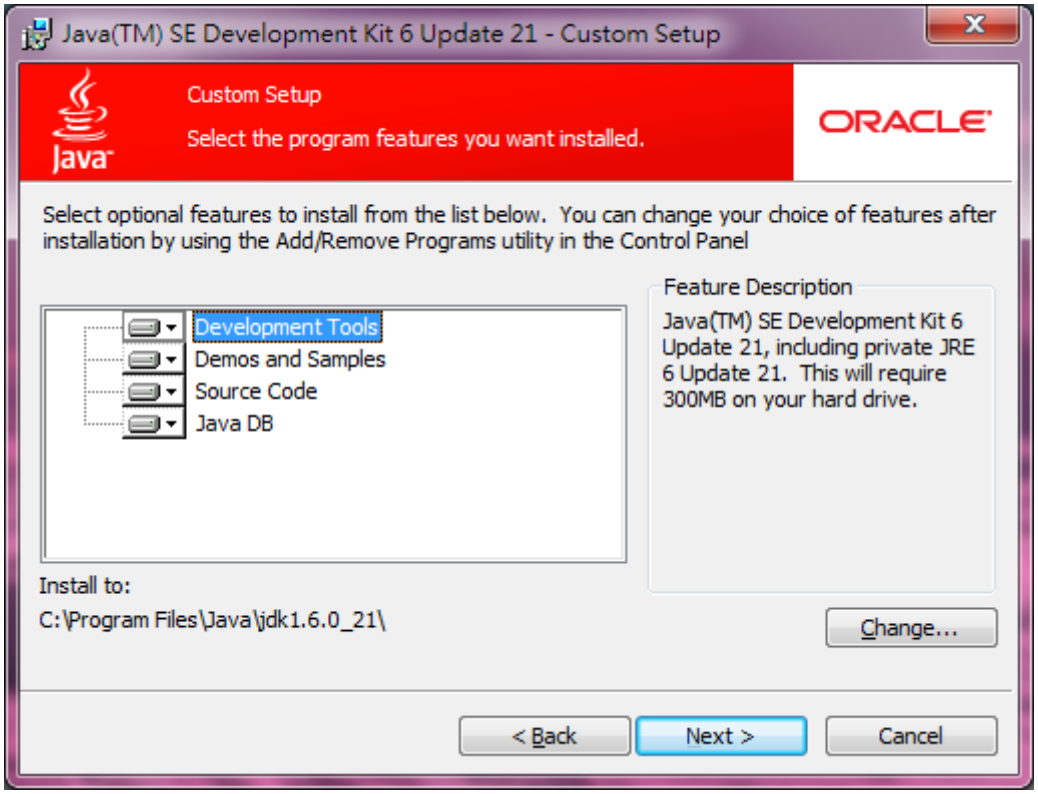
安裝 Java SDK

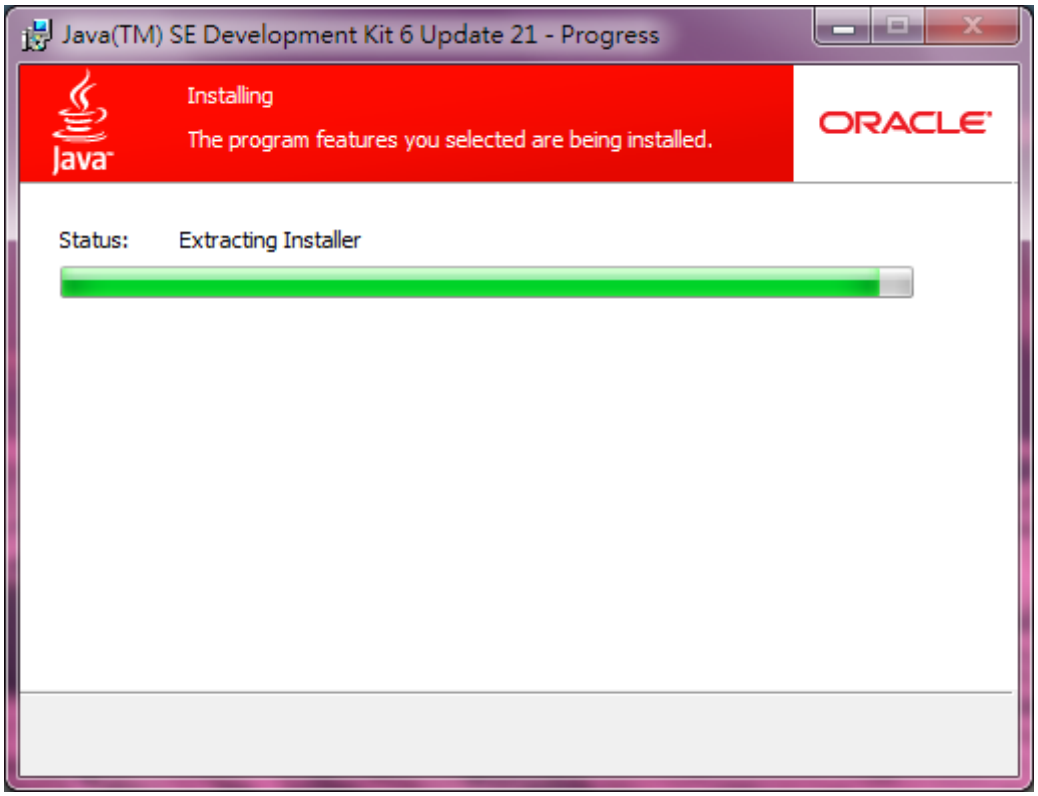
先下載安裝檔並執行：

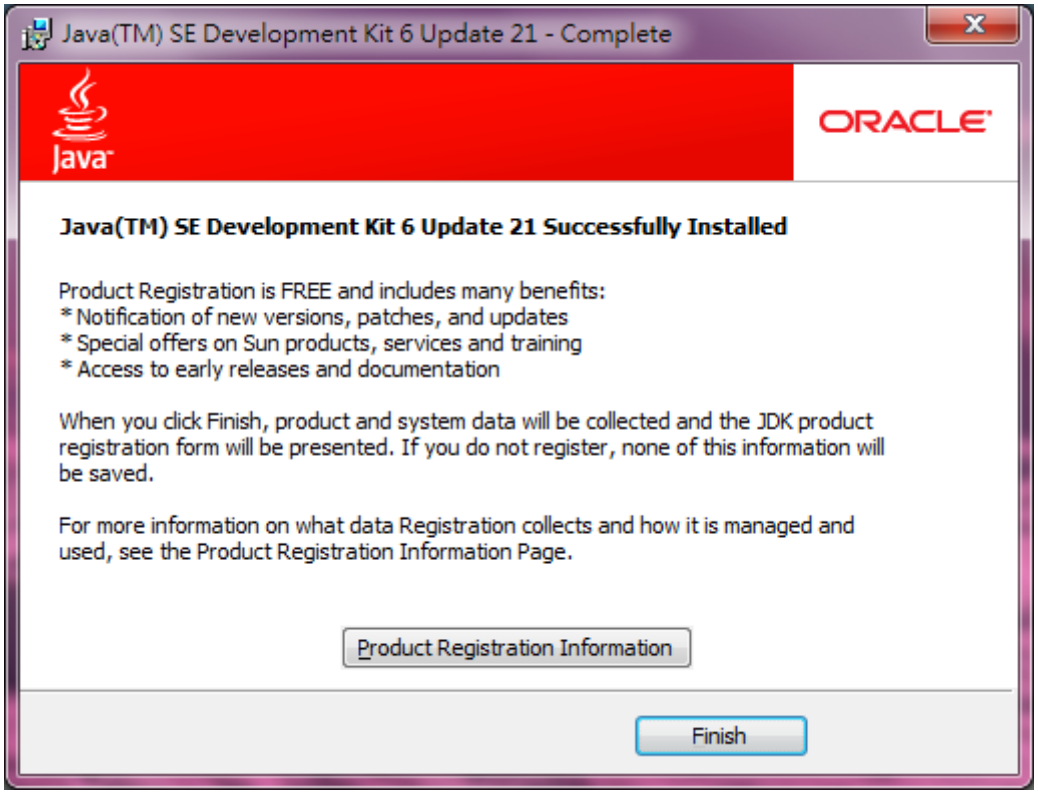


開啟後選擇 Next，順著安裝精靈的指示下一步直到完成安裝。


Java(TM) SE Development Kit 6 Update 21 - Setup
Java
ORACLE
Welcome to the Installation Wizard for Java?SE Development Kit 6 Update 21
This wizard will guide you through the installation process for the Java SE Development Kit 6 Update 21.
Next >
Cancel

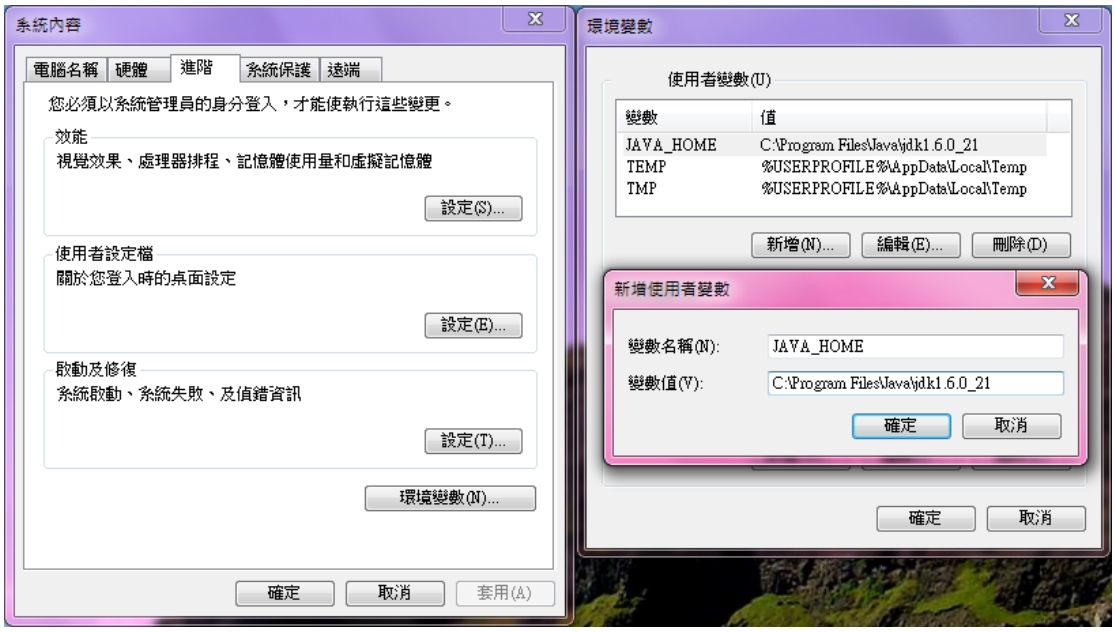

Java(TM) SE Development Kit 6 Update 21 - Custom Setup
Java
Custom Setup
Select the program features you want installed.
ORACLE
Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel
Development Tools
Demos and Samples
Source Code
Java DB
Feature Description
Java(TM) SE Development Kit 6 Update 21, including private JRE 6 Update 21. This will require 300MB on your hard drive.
Install to:
C:\Program Files\Java\jdk1.6.0_21\
Change...
< Back
Next >
Cancel


Java(TM) SE Development Kit 6 Update 21 - Progress
Installing
The program features you selected are being installed.
ORACLE
Status:
Extracting Installer


Java(TM) SE Development Kit 6 Update 21 - Complete
ORACLE
Java(TM) SE Development Kit 6 Update 21 Successfully Installed
Product Registration is FREE and includes many benefits:
* Notification of new versions, patches, and updates
* Special offers on Sun products, services and training
* Access to early releases and documentation
When you click Finish, product and system data will be collected and the JDK product registration form will be presented. If you do not register, none of this information will be saved.
For more information on what data Registration collects and how it is managed and used, see the Product Registration Information Page.
Product Registration Information
Finish

設定環境變數

先在「**我的電腦**」上面按右鍵選「**內容**」→「**進階**」→「**環境變數**」，在下面的變數值貼上安裝 Java SDK 的所在位置。



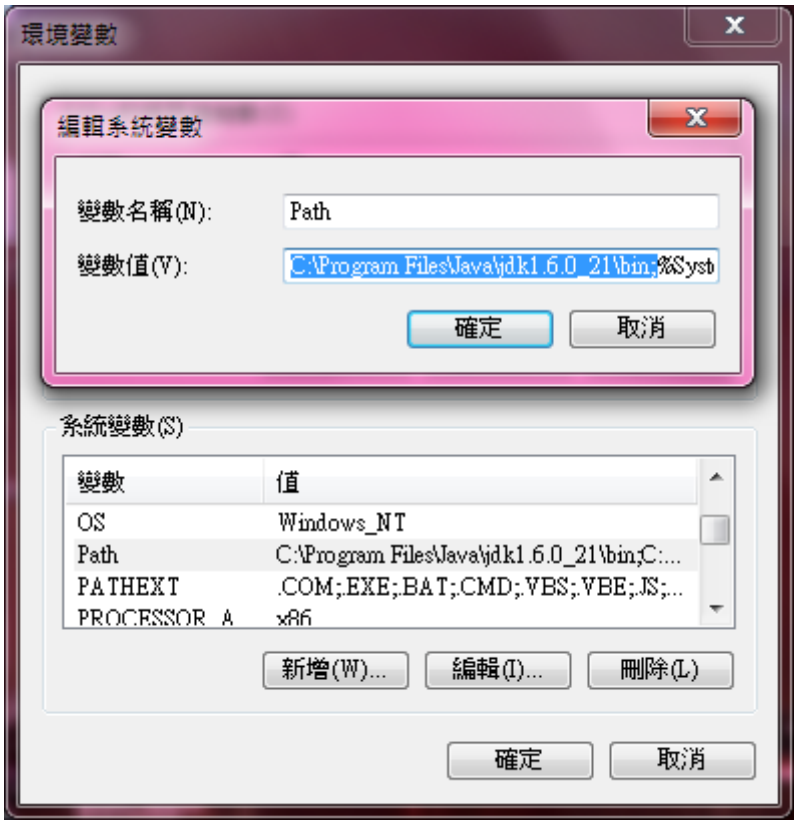
變數名稱：**JAVA_HOME**

路徑位置：**C:\Program Files\Java\jdk1.6.0_21**

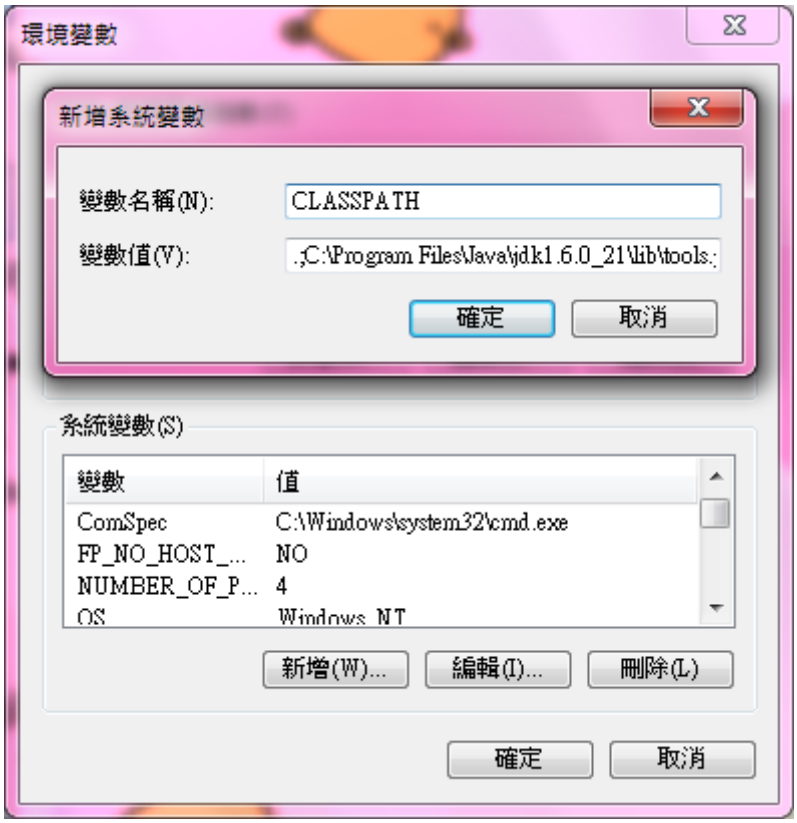
(此為安裝 Java 時選擇之安裝路徑)

接著在下方的系統變數選至「**Path**」，並選擇編輯。

在 Path 系統變數的最前面加入「**C:\Program Files\Java\jdk1.6.0_21\bin;**」，按下確定。



在下方的系統變數新增一個變數，名稱：「**CLASSPATH**」，
變數值：「**.;C:\Program Files\Java\jdk1.6.0_21\lib\tools.jar;**」，按下確定。



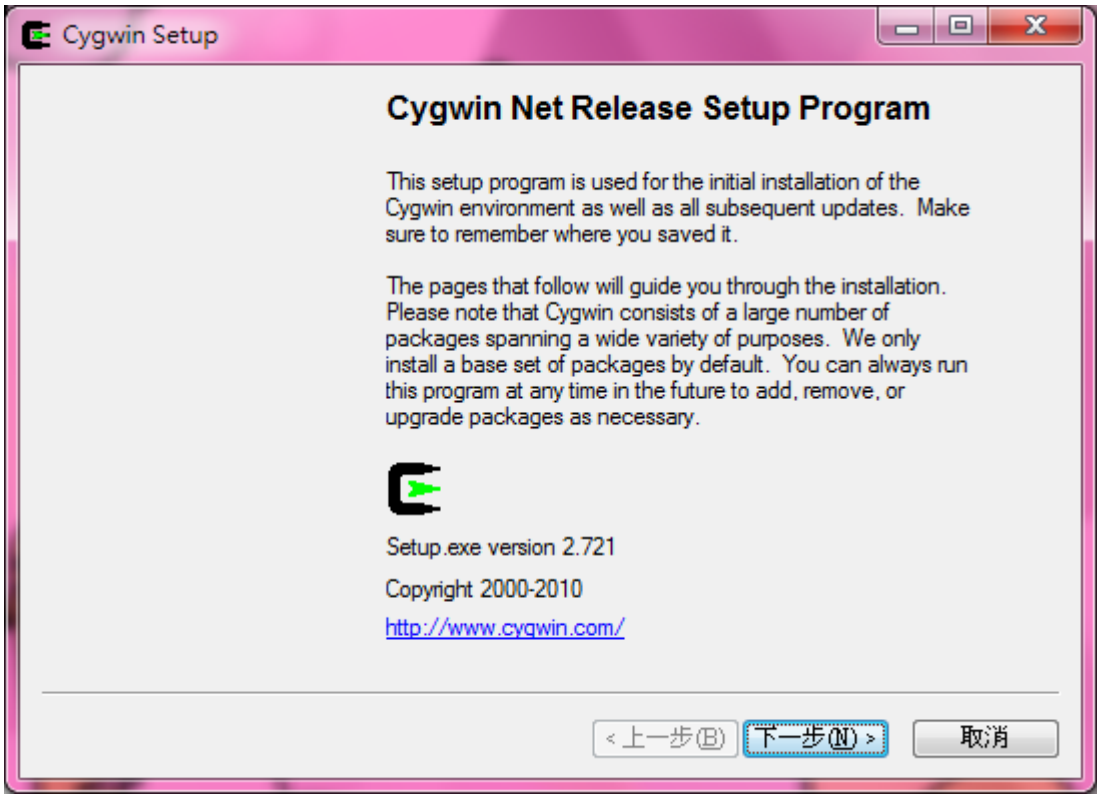
至此 Java SDK 的安裝與設定就告一段落。

Step3 安裝 Cygwin1.7 (線上安裝)

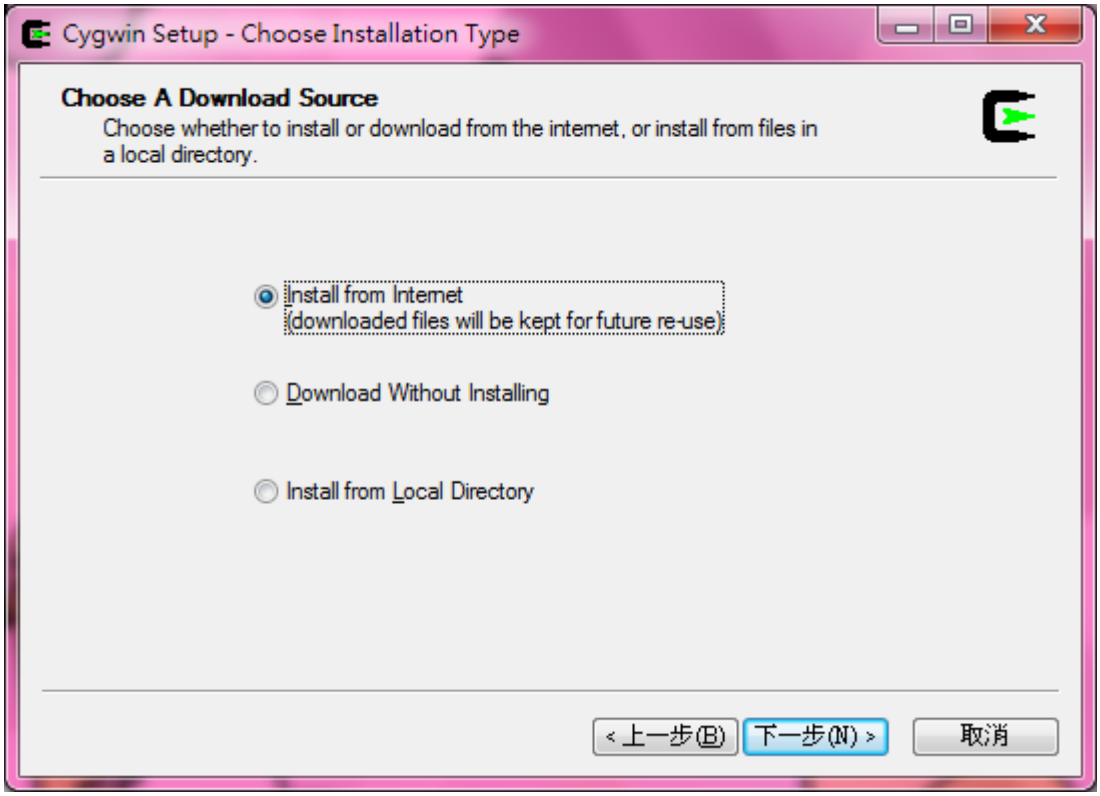
前往 http://www.cygwin.com

執行 **setup.exe**

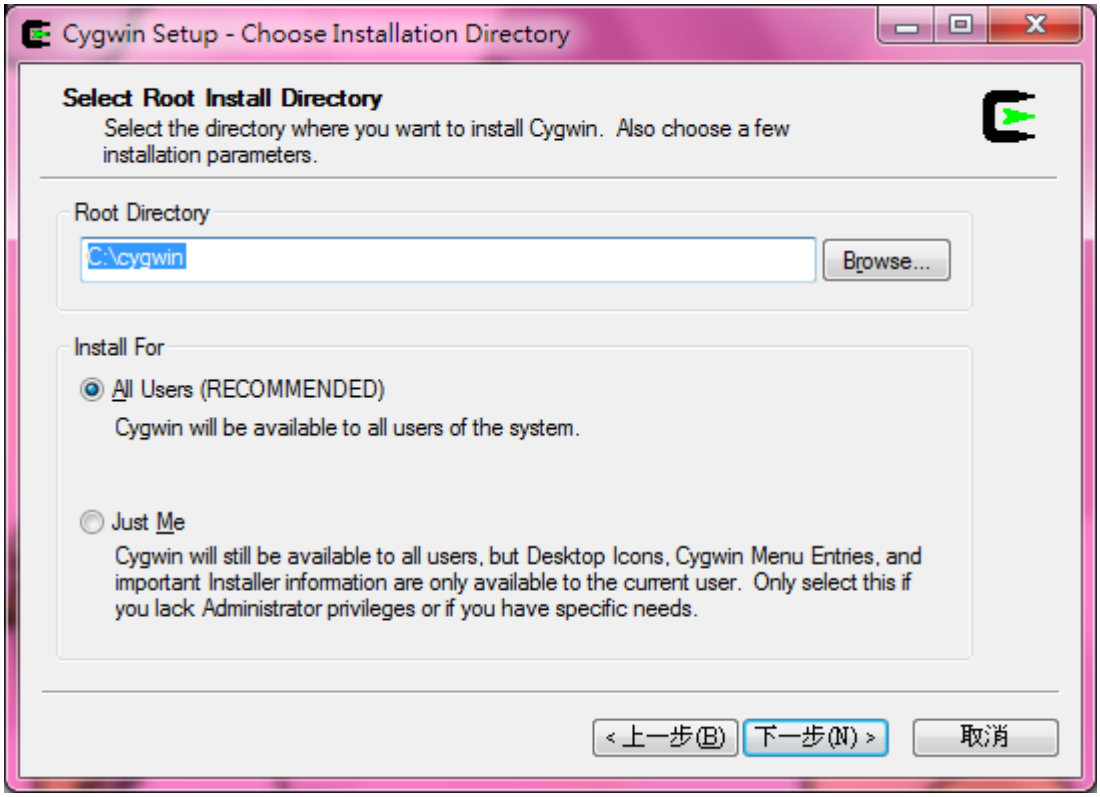
第一個 Setup 畫面直接按下下一步



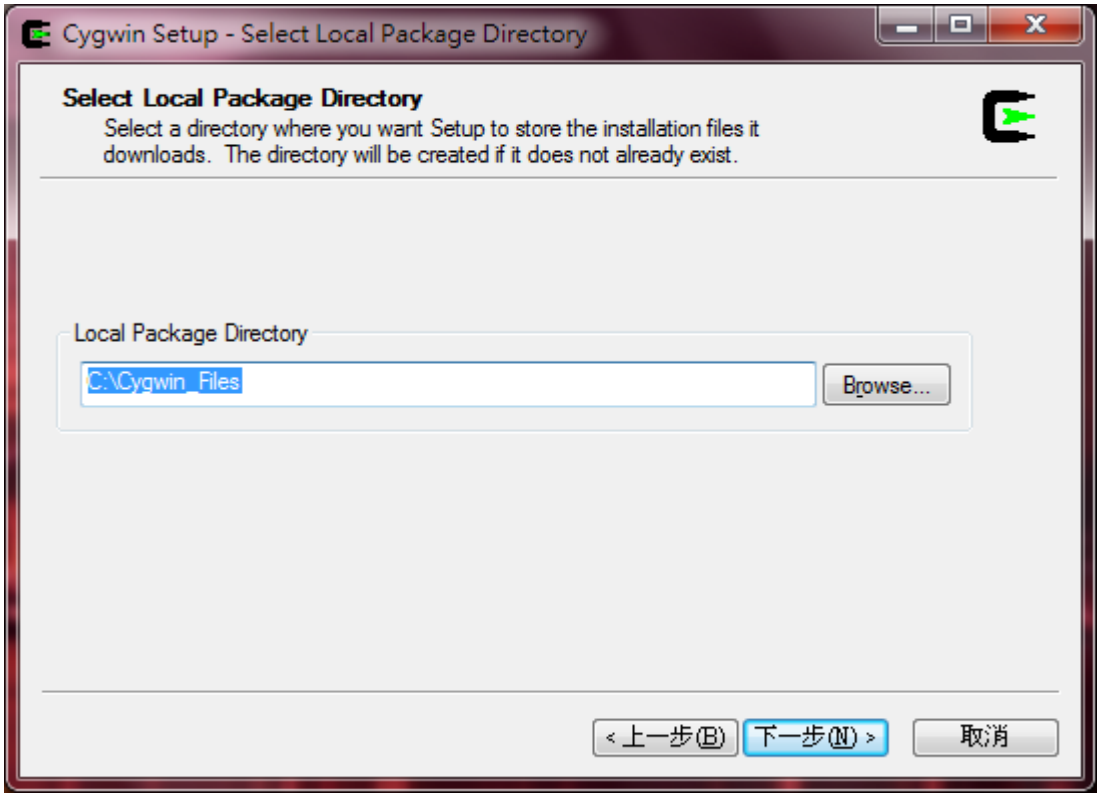
在 **Choose Installation Type** 時，選擇「**Install from Internet**」。



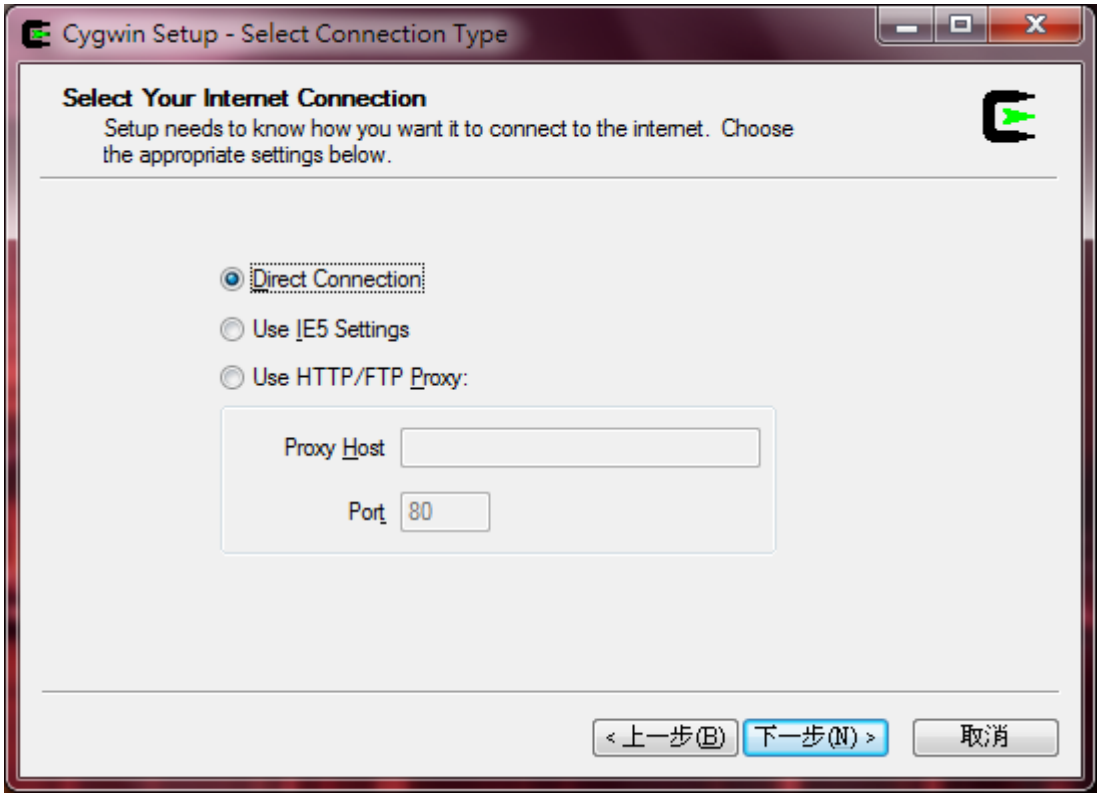
在 **Choose Installation Direction** 時，照預設值直接按下一步。



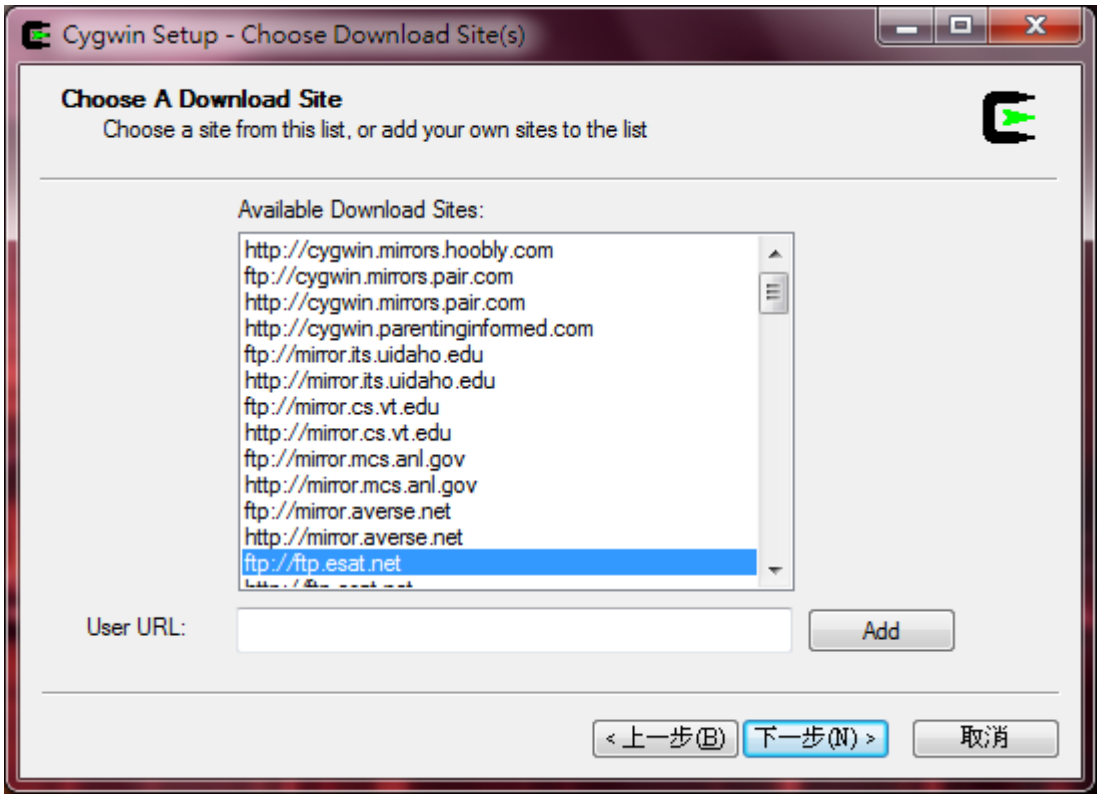
接著 **Select Local Package Direction** 為選擇「**存放由網路下載回來檔案的資料夾**」，設定完之後按下下一步。



在 **Select Connection Type** 時，選擇「**Direct Connection**」。

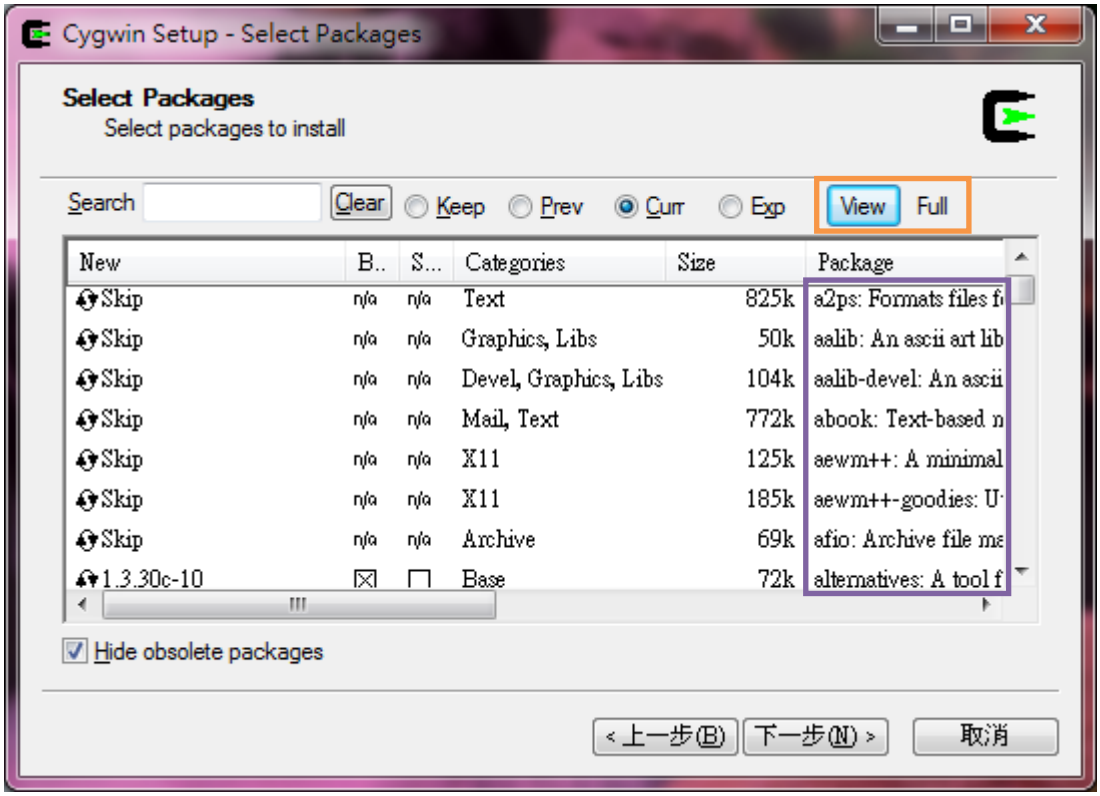


接著選擇下載來源 **Choose Download Sites**。建議選擇離自己較近的網站或台灣的網站（如 NTU 等）。



接著來到 **Select Packages** 部分。

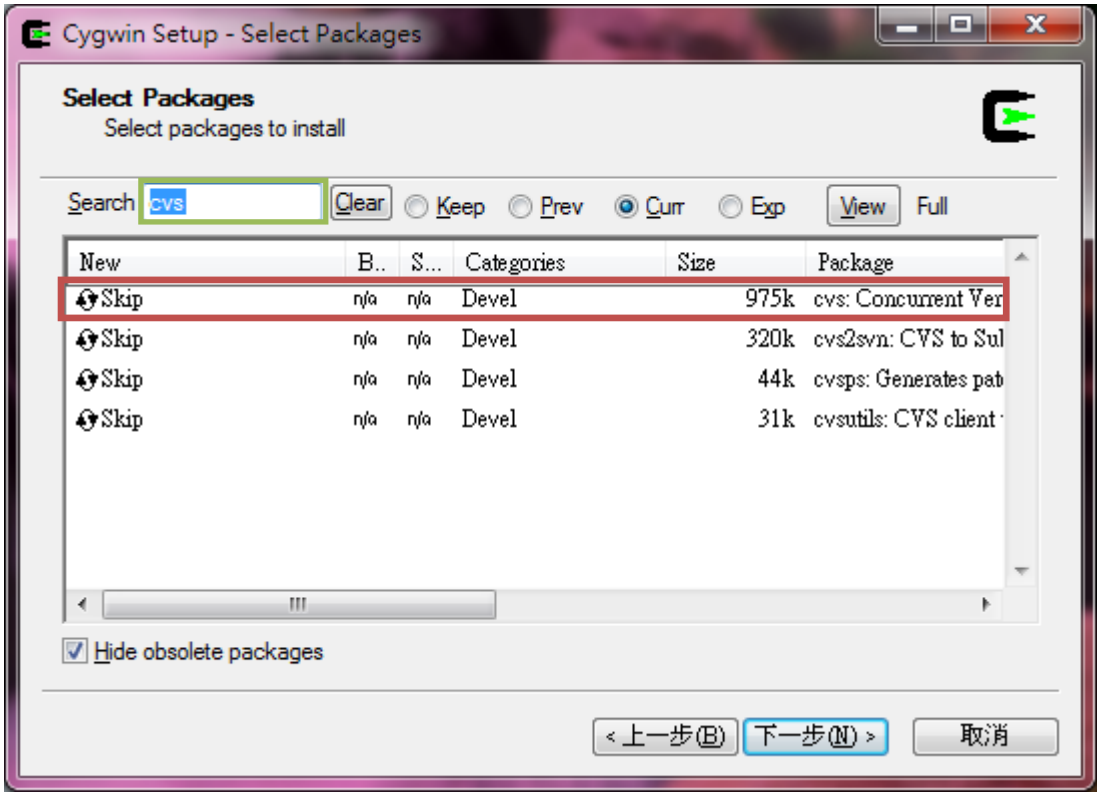
先按下右上角的“**View**”按鈕，使旁邊變成“**full**”，之後 packages 會變成由 a-z 順序列出。



安裝“**cvs**”，“**gcc**”，“**make**”，“**perl**”，“**rpm**”，和“**wget**”這六個 package。（點選 ***Skip*** 的圖案安裝）

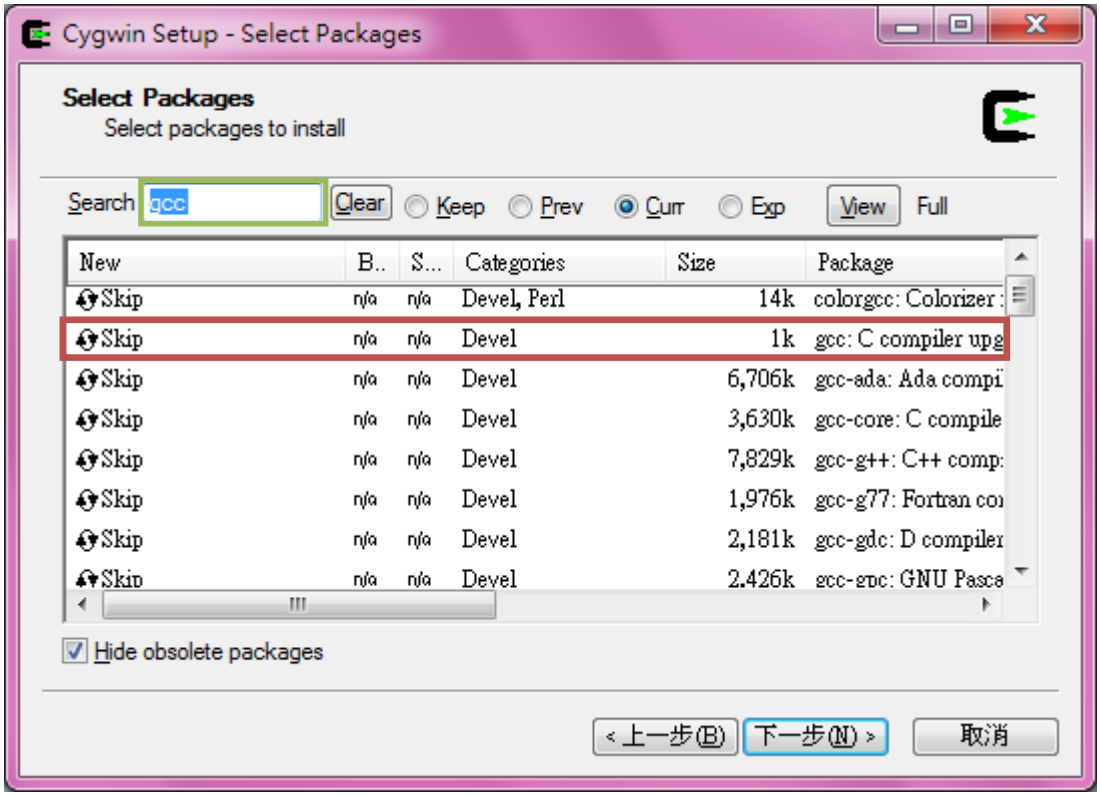
***cvs*：於“Search”欄位中輸入“cvs”，並點選“Skip”圖示進行安裝**

CVS（Concurrent Versions System）是一種版本控制系統，方便軟體的開發和使用者協同工作，很多開放原始碼或自由軟體皆使用 CVS 以便結合各程式設計師的改進和更改，這些項目包括 Gnome、KDE、GIMP、Wine 等，CVS 以獲得 GNU 的通用公共許可證授權，CVS 系統主要是利用將一組文件放在層次目錄樹中以保持同步，程式設計師可從 CVS 伺服器上更新他們本地層次樹的副本，並將修改結果或新文件發回，亦可刪除舊文件。



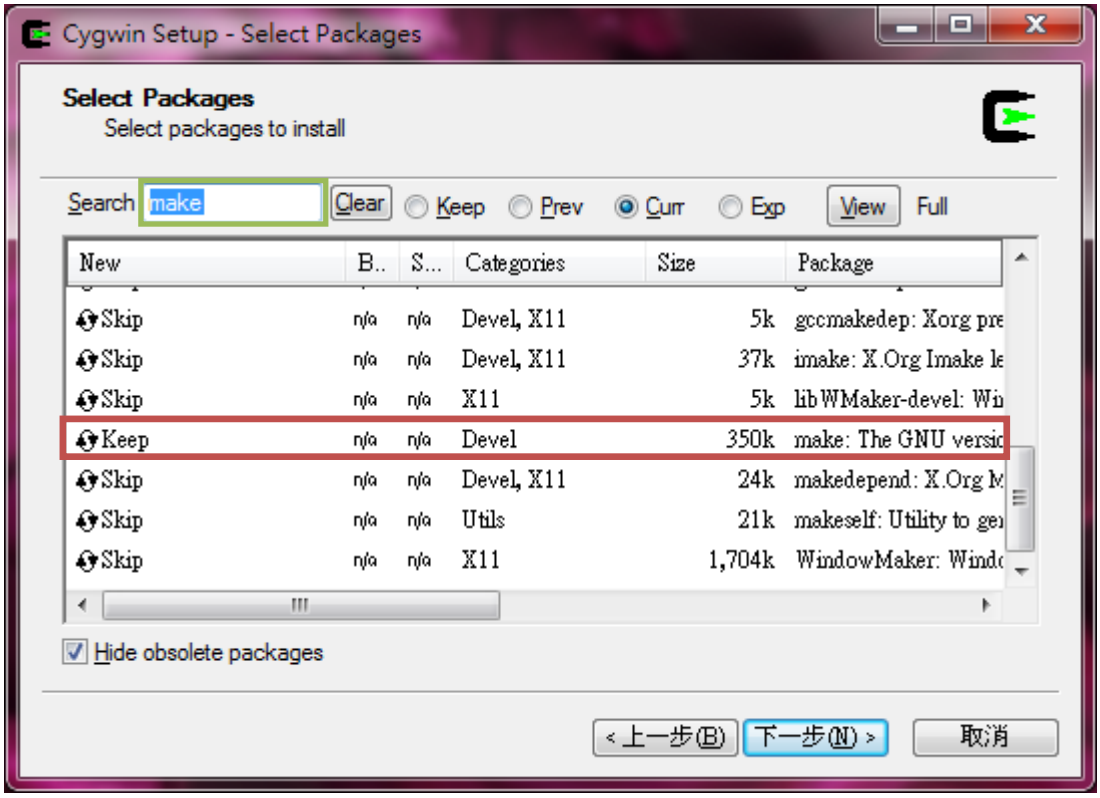
***gcc*：於“Search”欄位中輸入“gcc”，並點選“Skip”圖示進行安裝**

GCC（GNU Compiler Collection，GNU 編譯器套裝），是一套由 GNU 開發的編程語言編譯器，除了是一套擁有 GPL 及 LGPL 許可證的自由軟體外，亦是 Unix 及蘋果電腦 Mac OS X 作業系統的標準編譯器，GCC 原名為 GNU C 語言編譯器（GNU C Compiler），因為原本只能處理 C 語言，但後來快速地擴展，使其可處理 C++等語言，我們使用 GCC 主要是用來編譯 Octopus II 的 nes C 程式。



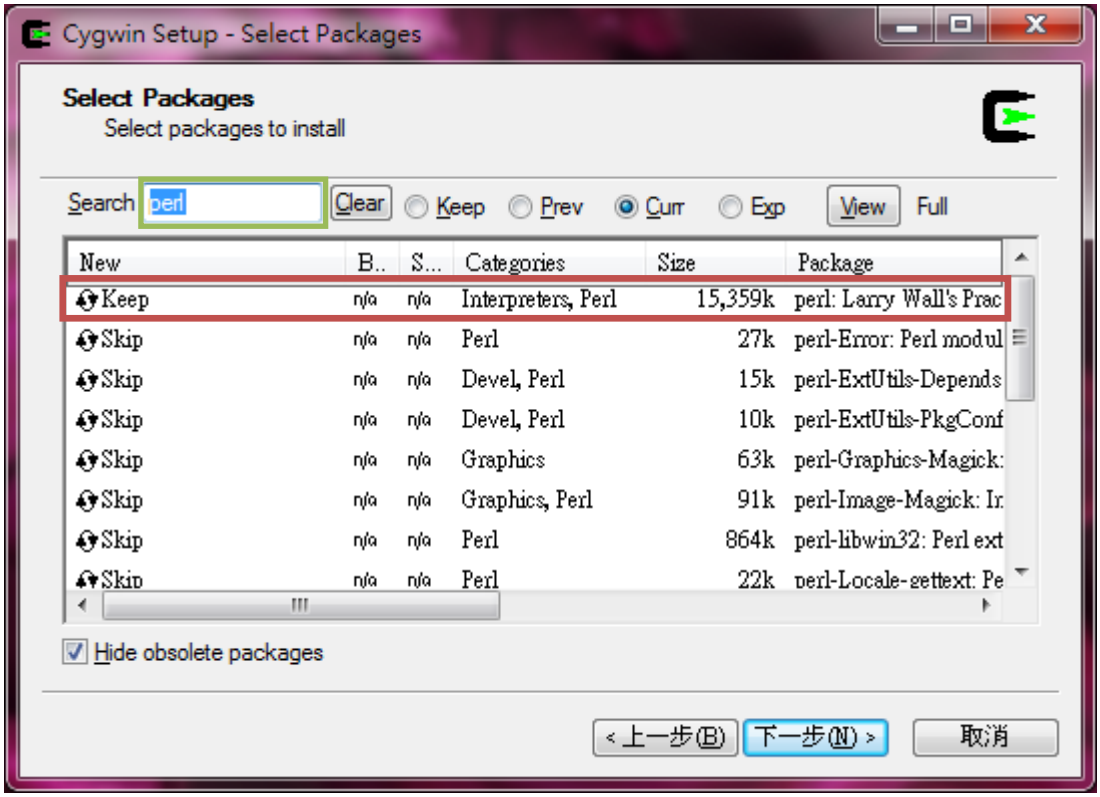
make：**於“Search”欄位中輸入“make”，並點選“Skip”圖示進行安裝**

MAKE 是一個會自動讀取原始碼、自動連結需要的編譯工具、並將原始碼轉譯為指定平台之專屬可執行檔。



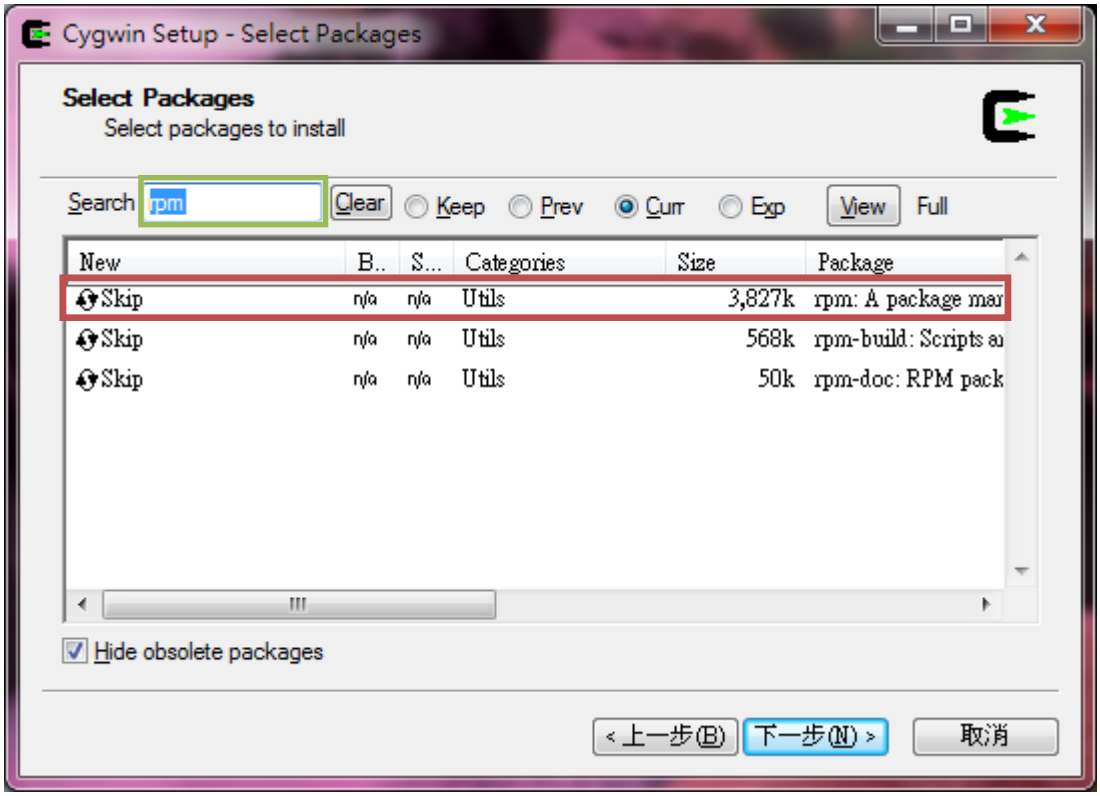
***perl*：於“Search”欄位中輸入“perl”，並點選“Skip”圖示進行安裝**

安裝 perl 的 Interpreter 主要是為了能夠順利使用 cygwin 在 windows 下模擬及執行 perl 程式。



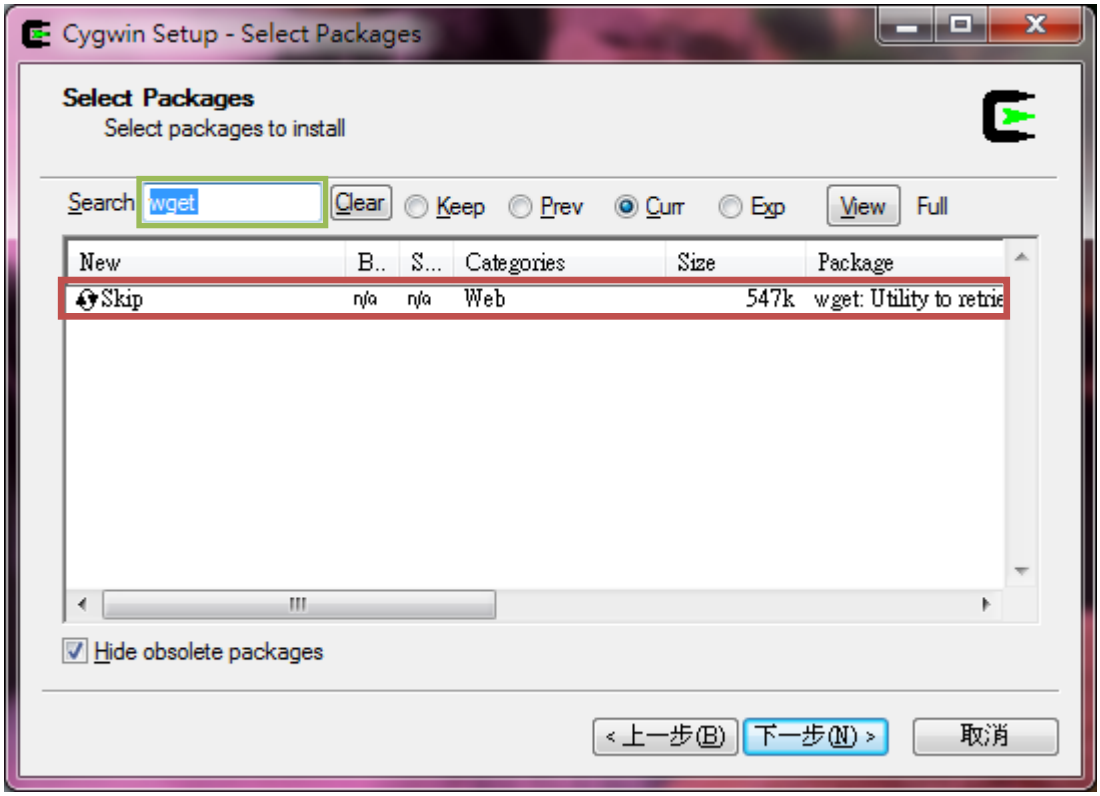
***rpm*：於“Search”欄位中輸入“rpm”，並點選“Skip”圖示進行安裝**

RPM 套件管理員（簡稱 RPM，全稱為 The RPM Package Manager），是在 Linux 下被廣泛使用的套件管理軟體，最早是由 Red Hat 公司研發，現在則由開源社群接續開發，RPM 套件分為二進制（Binary）、原始碼（Source）和 Delta 三種，二進制套件可直接安裝於電腦中，而原始碼套件將會由 RPM 自動編譯與安裝，原始碼套件經常以 src.rpm 作為副檔名。

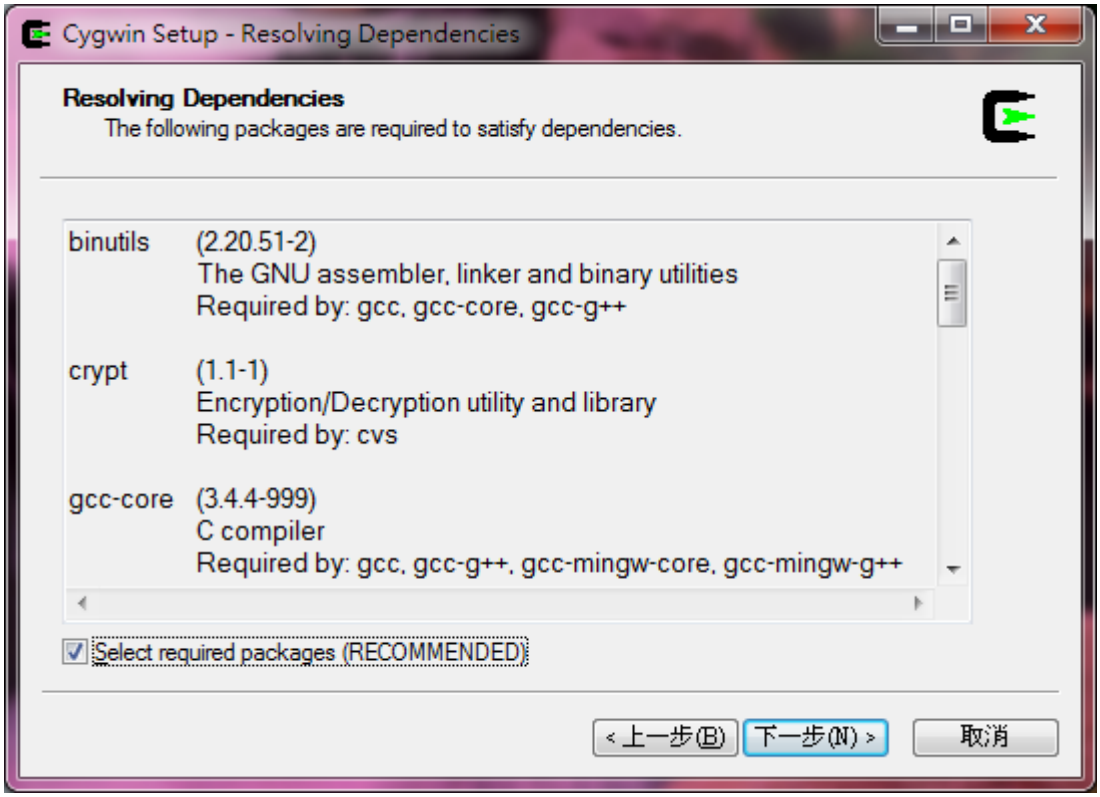


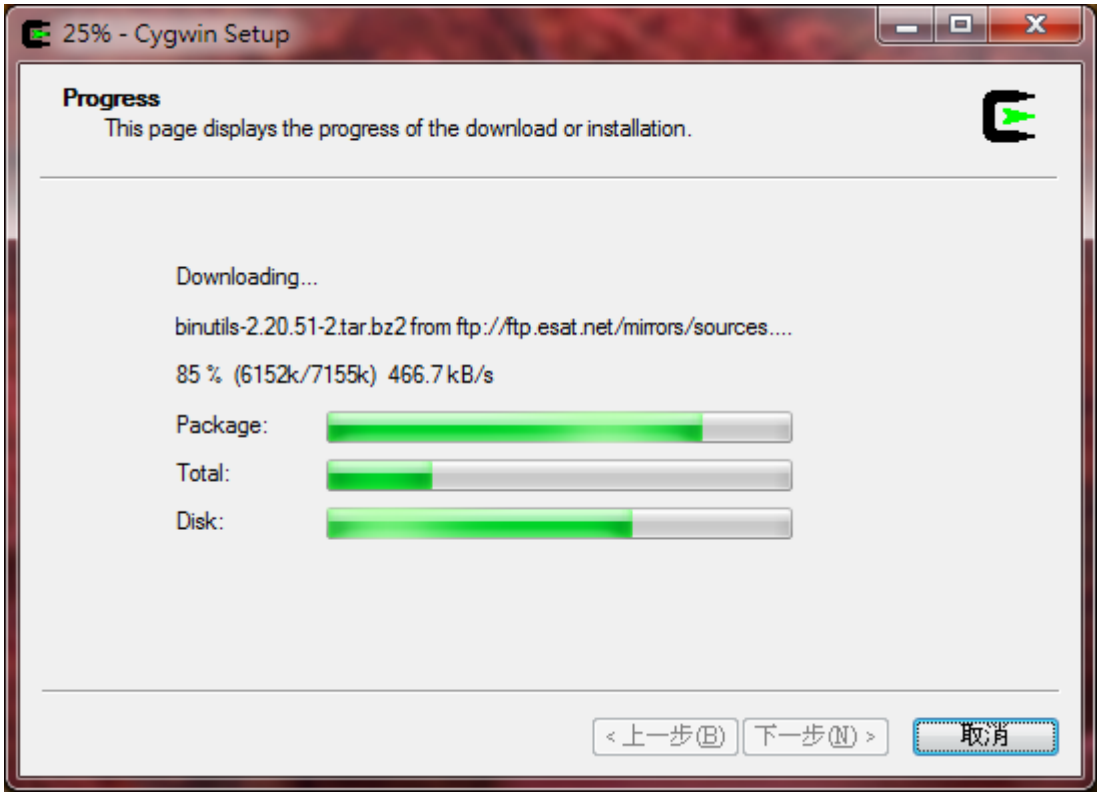
***wget*：於“Search”欄位中輸入“wget”，並點選“Skip”圖示進行安裝**

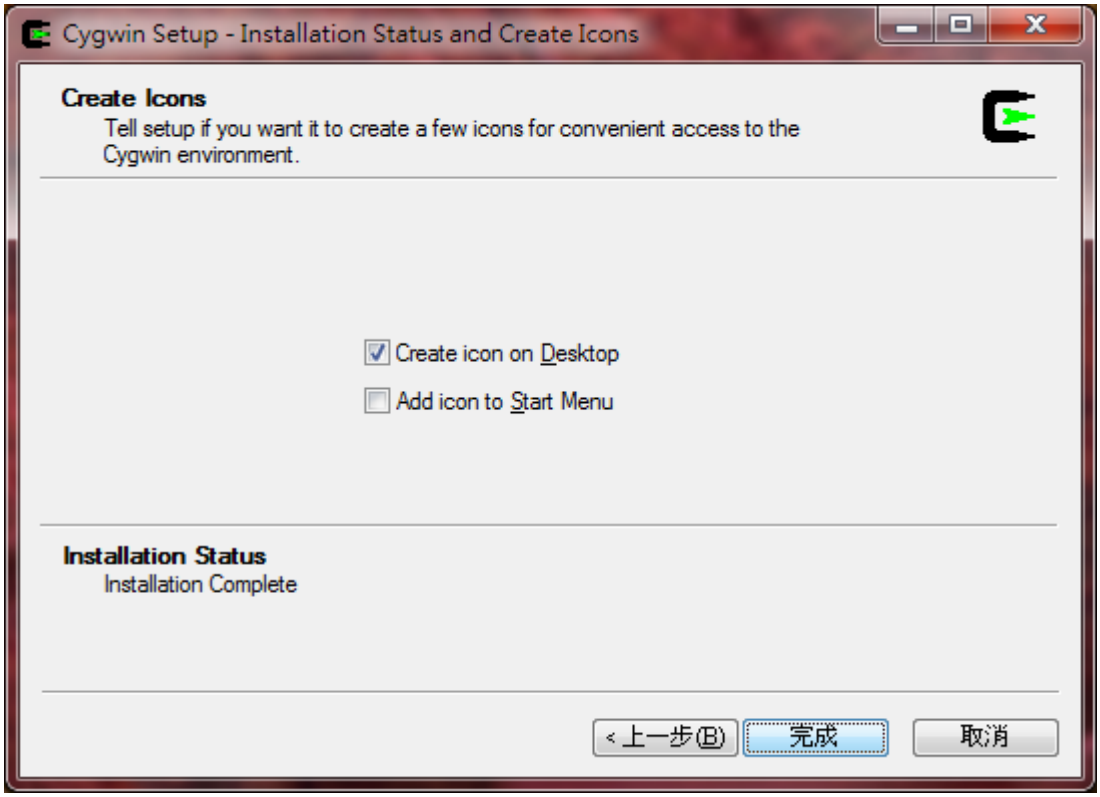
WGET 是一套在網路上進行程式下載的自由軟體，WGET 簡稱是由「World Wide Web」和「GET」結合，目前支援 HTTP、HTTPS 以及 FTP 三種最常見的網路下載。



選下一步以進行下載與安裝。







到此步驟便完成 Cygwin 環境的安裝。

Step4 安裝 TinyOS 2.1

安裝目標節點編譯器

因為我們要編譯 MCU 上的程序，所以需要能夠生成相應匯編語言的編譯器。如果你使用 mica 系列的節點，就需要 AVR 的 tool chain；如果是 telos 系列的節點，那就需要 MSP430 tool chain。

OctopusII 是使用 TI 公司的 MSP430 處理器，故要安裝 MSP430 的編譯器，請至 http://www.tinyos.net/dist-2.1.0/tools/windows/ 下載，共 5 個套件。

Base

http://www.tinyos.net/dist-2.1.0/tools/windows/msp430tools-base-0.1-20050607.cygwin.i386.rpm

python tools

http://www.tinyos.net/dist-2.1.0/tools/windows/msp430tools-python-tools-1.0-1.cygwin.noarch.rpm

binutils

http://www.tinyos.net/dist-2.1.0/tools/windows/msp430tools-binutils-2.16-20050607.cygwin.i386.rpm

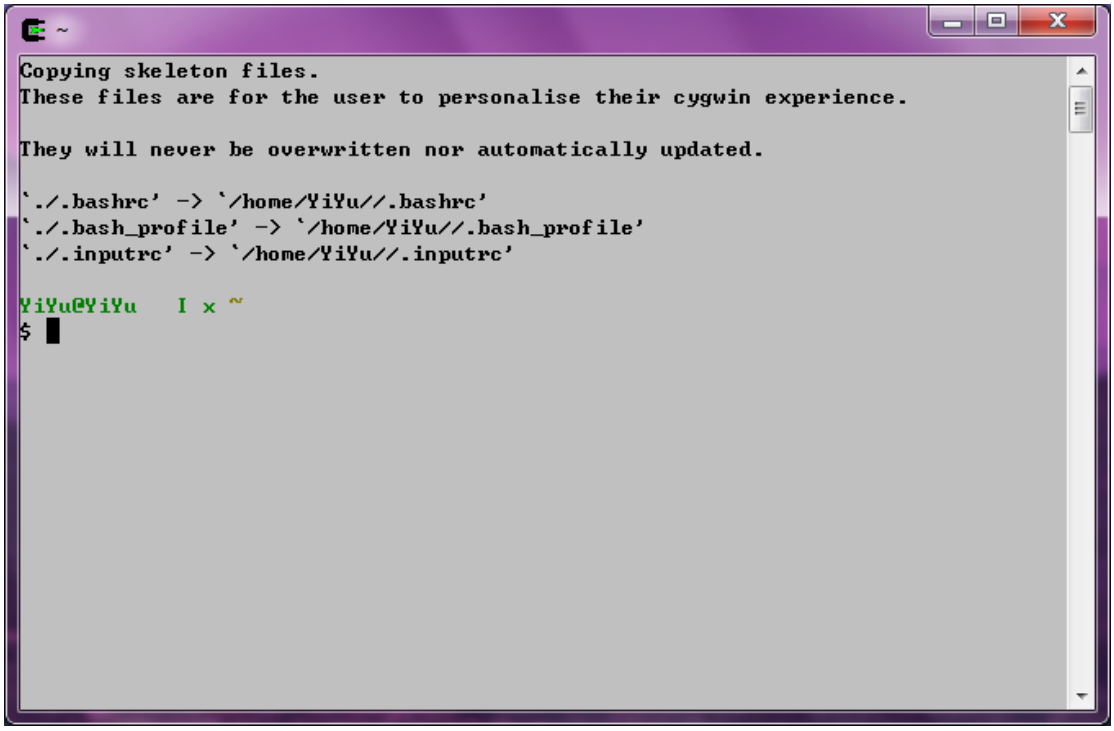
gcc

http://www.tinyos.net/dist-2.1.0/tools/windows/msp430tools-gcc-3.2.3-20050607.cygwin.i386.rpm

libc

http://www.tinyos.net/dist-2.1.0/tools/windows/msp430tools-libc-20080808-1.cygwin.i386.rpm

先將下載的安裝檔存至「**c:\cygwin\rpm\msp430rpm**」下（自行新增 rpm 資料夾），並開啟 Cygwin。



輸入 **cd /rpm/msp430rpm/**，意思為切換至安裝檔 msp430rpm 目錄下。

輸入 **rpm -ivh --nodeps --ignoreos msp***以安裝所需的工具。

```
/rpm/msp430rpm
Copying skeleton files.
These files are for the user to personalise their cygwin experience.

They will never be overwritten nor automatically updated.

`./.bashrc' -> `/home/YiYu//.bashrc'
`./.bash_profile' -> `/home/YiYu//.bash_profile'
`./.inputrc' -> `/home/YiYu//.inputrc'

YiYu@YiYu   I x ~
$ cd /rpm/msp430rpm/

YiYu@YiYu   I x /rpm/msp430rpm
$ rpm -ivh --nodeps --ignoreos msp*
Preparing...                ########################################### [100%]
   1:msp430tools-python-tool########################################### [ 20%]
   2:msp430tools-base       ########################################### [ 40%]
   3:msp430tools-binutils   ########################################### [ 60%]
   4:msp430tools-gcc        ########################################### [ 80%]
   5:msp430tools-libc       ########################################### [100%]

YiYu@YiYu   I x /rpm/msp430rpm
$
```

安裝 nesC 編譯器

TinyOS 用 nesC 寫成。nesC 是 C 的一個分支，它支援 TinyOS 中的併發模型並且使用基於組件（component）的編程。nesC 編譯器是相對於節點而獨立的，它的編譯輸出會傳遞給目標節點的編譯器，所以你可以充分發揮你的主觀能動性對代碼進行優化。

請至 http://www.tinyos.net/dist-2.1.0/tinyos/windows/ 下載 nesC 與 TinyOS-tools 存至「c:\cygwin\rpm\tinyosrpm」下。

NesC

http://www.tinyos.net/dist-2.1.0/tinyos/windows/nesc-1.3.0-1.cygwin.i386.rpm

Deputy

http://www.tinyos.net/dist-2.1.0/tinyos/windows/tinyos-deputy-1.1-1.cygwin.i386.rpm

TinyOS-tools

http://www.tinyos.net/dist-2.1.0/tinyos/windows/tinyos-tools-1.3.0-1.cygwin.i386.rpm

輸入 **cd /rpm/tinyosrpm/**切換至安裝檔 tinyosrpm 目錄下

輸入 **rpm -Uvh --ignoreos nesc-1.3.0-1.cygwin.i386.rpm**

輸入 **rpm -Uvh --ignoreos tinyos-deputy-1.1-1.cygwin.i386.rpm**

輸入 **rpm -Uvh --ignoreos tinyos-tools-1.3.0-1.cygwin.i386.rpm**

```
/rpm/tinyosrpm

YiYu@YiYu   I x ~
$ cd /rpm/tinyosrpm/

YiYu@YiYu   I x /rpm/tinyosrpm
$ rpm -Uvh --ignoreos nesc-1.3.0-1.cygwin.i386.rpm
Preparing...                ########################################### [100%]
   1:nesc                   ########################################### [100%]

YiYu@YiYu   I x /rpm/tinyosrpm
$ rpm -Uvh --ignoreos tinyos-deputy-1.1-1.cygwin.i386.rpm
Preparing...                ########################################### [100%]
   1:deputy                 ########################################### [100%]

YiYu@YiYu   I x /rpm/tinyosrpm
$ rpm -Uvh --ignoreos tinyos-tools-1.3.0-1.cygwin.i386.rpm
Preparing...                ########################################### [100%]
   1:tinyos-tools           ########################################### [100%]
giveio-install: Driver already installed
Installing Java JNI code in /cygdrive/c/Program Files/Java/jdk1.6.0_21/jre/bin .
..
done.

YiYu@YiYu   I x /rpm/tinyosrpm
$
```

安裝 TinyOS Source Tree

使用 TinyOS Source Tree 做為編譯和下載 TinyOS 的程式。

TinyOS 2.x

http://www.tinyos.net/dist-2.1.0/tinyos/windows/tinyos-2.1.0-2.cygwin.noarch.rpm

同樣在 tinyosrpm 目錄下：

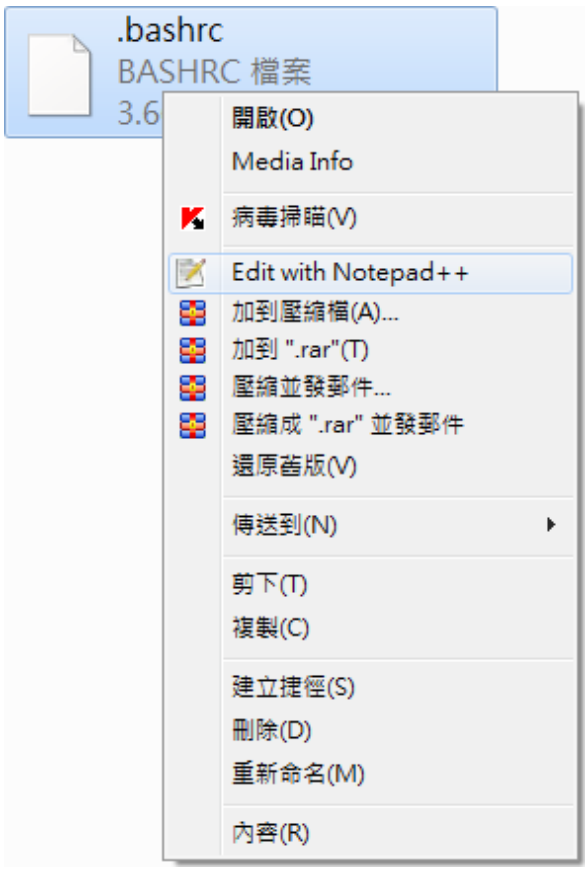
輸入 **rpm -ivh --ignoreos tinyos-2.1.0-2.cygwin.noarch.rpm**

```
/rpm/tinyosrpm
$ rpm -Uvh --ignoreos nesc-1.3.0-1.cygwin.i386.rpm
Preparing...                ########################################### [100%]
   1:nesc                   ########################################### [100%]

YiYu@YiYu   I x /rpm/tinyosrpm
$ rpm -Uvh --ignoreos tinyos-deputy-1.1-1.cygwin.i386.rpm
Preparing...                ########################################### [100%]
   1:deputy                 ########################################### [100%]

YiYu@YiYu   I x /rpm/tinyosrpm
$ rpm -Uvh --ignoreos tinyos-tools-1.3.0-1.cygwin.i386.rpm
Preparing...                ########################################### [100%]
   1:tinyos-tools           ########################################### [100%]
giveio-install: Driver already installed
Installing Java JNI code in /cygdrive/c/Program Files/Java/jdk1.6.0_21/jre/bin .
..
done.

YiYu@YiYu   I x /rpm/tinyosrpm
$ rpm -ivh --ignoreos tinyos-2.1.0-2.cygwin.noarch.rpm
Preparing...                ########################################### [100%]
   1:tinyos                 ########################################### [100%]

YiYu@YiYu   I x /rpm/tinyosrpm
$
```

完成後關閉 cygwin。

<u>修改環境變數</u>

在使用者目錄下（C:\cygwin\home\使用者\），用 Notepad++（或記事本）開啟**.bashrc**。



將下列文字貼在最後一行下面

```
# TinyOS 2.x
export TOSROOT="/opt/tinyos-2.x"
export TOSDIR="$TOSROOT/tos"
export CLASSPATH="C:\cygwin\opt\tinyos-2.x\support\sdk\java\tinyos.jar;."
export MAKERULES="$TOSROOT/support/make/Makerules"
```

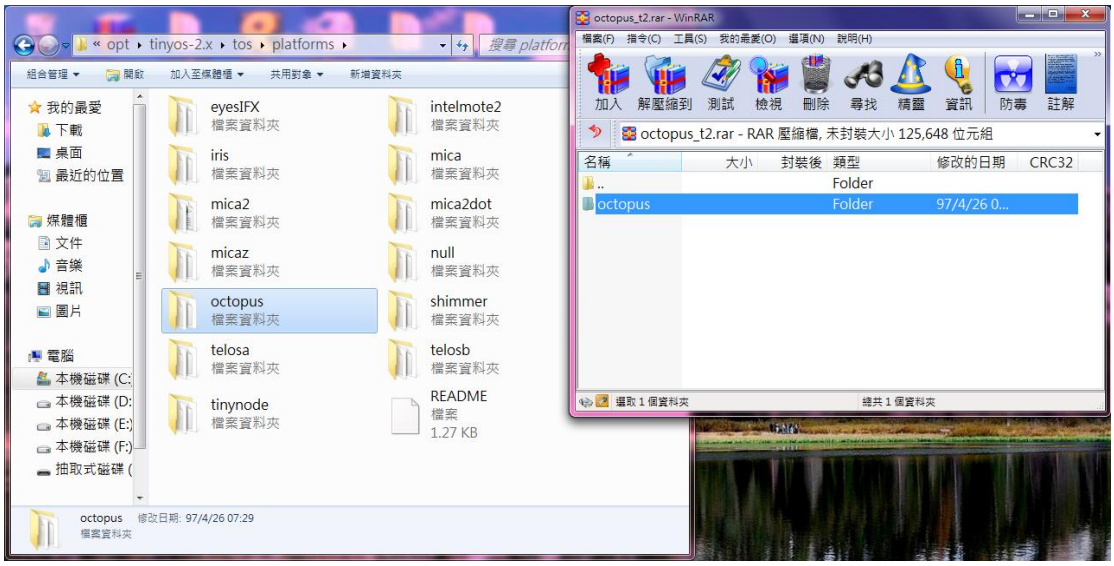
（紅色字體部分請根據安裝 Cygwin 的位置作變更，要注意換行符號的問題。)

```
124
125  # Functions
126  # #########
127
128  # Some example functions
129  # function settitle() { echo -ne "\e]2;$@\a\e]1;$@\a"; }
130
131  # TinyOS 2.x
132  export TOSROOT="/opt/tinyos-2.x"
133  export TOSDIR="$TOSROOT/tos"
134  export CLASSPATH="C:\cygwin\opt\tinyos-2.x\support\sdk\java\tinyos.jar;."
135  export MAKERULES="$TOSROOT/support/make/Makerules"
136
```

安裝 OctopusII 所需的編譯文件

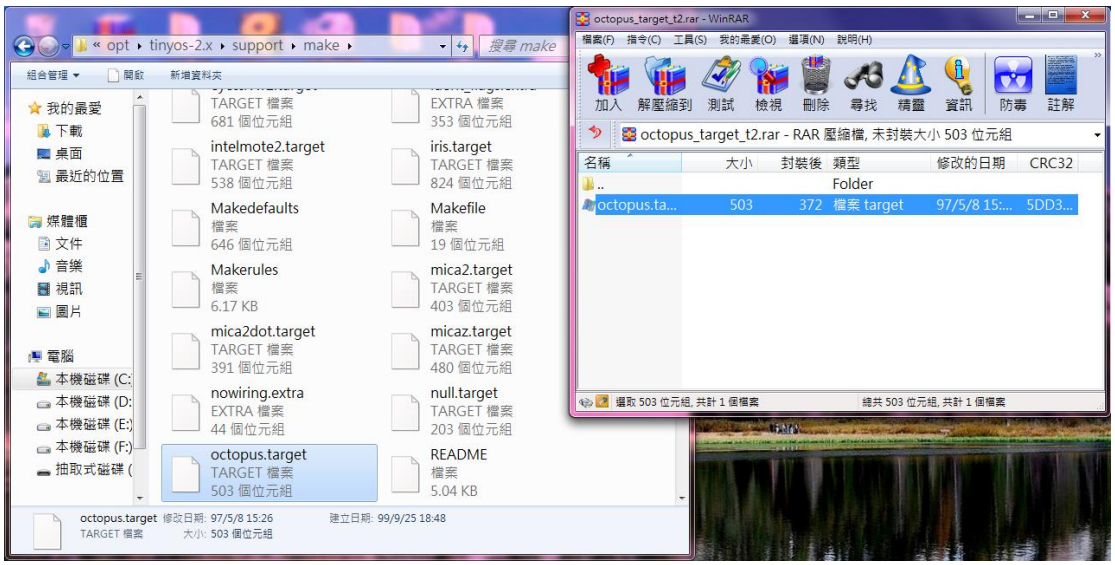
解壓縮 **octopus_t2.rar**，放至

/opt/tinyos-2.x/tos/platforms/中。



解壓縮 **octopus_target_t2.rar**，放至

/opt/tinyos-2.x/support/make/中。

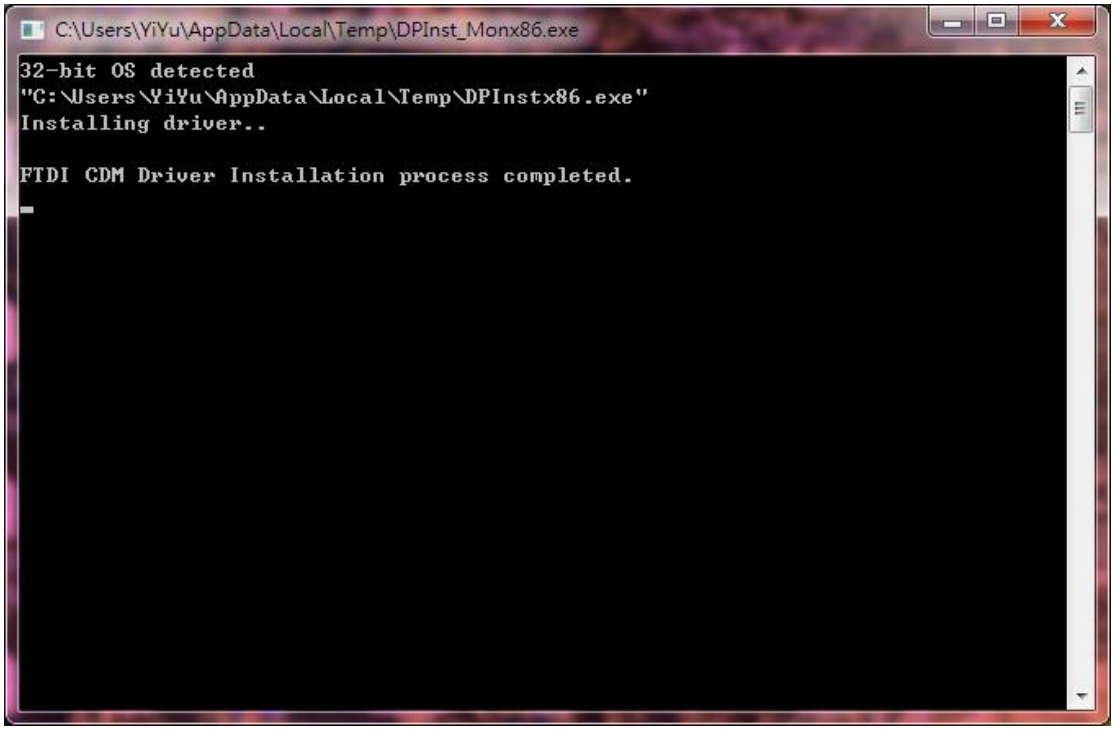


安裝 OctopusII Driver

至 FTDI VirtualComm USB Drivers（http://www.ftdichip.com/Drivers/VCP.htm）下載 USB 轉 RS232 驅動程式。

Operating System	Devices Supported	Driver Version	Release Date	Comments
Windows Server 2008 R2 Windows 7 Windows 7 x64 Windows Server 2008 Windows Server 2008 x64 Windows Vista Windows Vista x64 Windows XP Windows XP x64 Windows 2000 Windows Server 2003 Windows Server 2003 x64	FT2232H, FT4232H, FT232R, FT245R, FT2232, FT232B, FT245B, FT8U232AM, FT8U245AM	2.08.02 (zip file) 2.08.02 (setup exe)	11th August 2010	Microsoft WHQL certified. For custom VID and PID combinations see AN232R-03. Combined driver model (D2XX and VCP). Devices programmed as VCP will expose a COM port, as will AM and BM devices. Release Notes

下載完後安裝即可。


C:\Users\YiYu\AppData\Local\Temp\DPInst_Monx86.exe
32-bit OS detected
"C:\Users\YiYu\AppData\Local\Temp\DPInstx86.exe"
Installing driver..
FTDI CDM Driver Installation process completed.

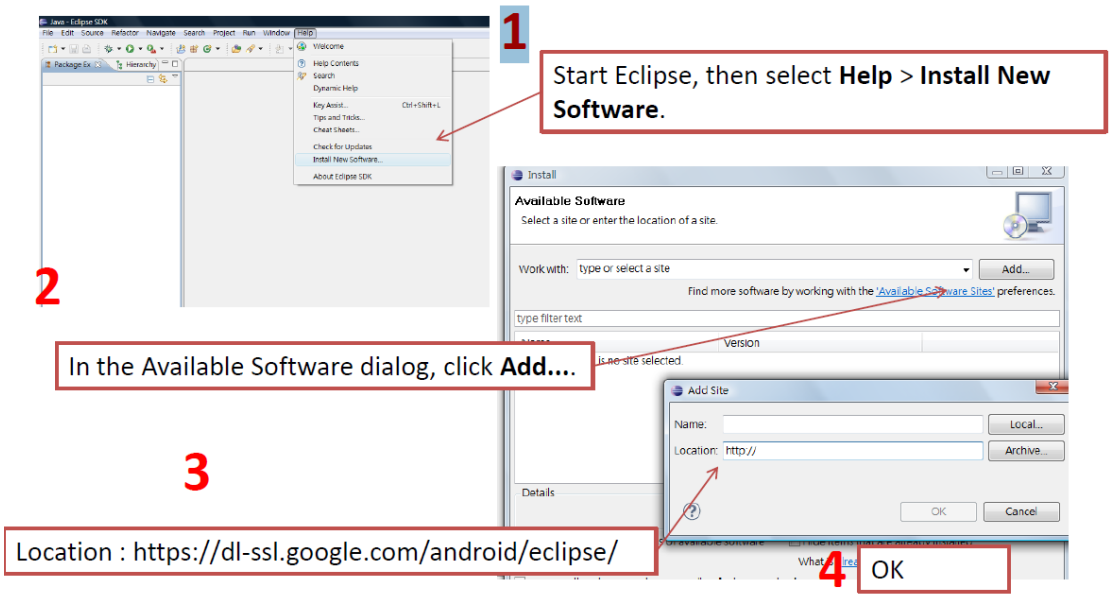
Step5 安裝 Eclipse

由於 Android 需要有 Eclipse 環境的支援，因此我們將先介紹 Eclipse 之安裝。

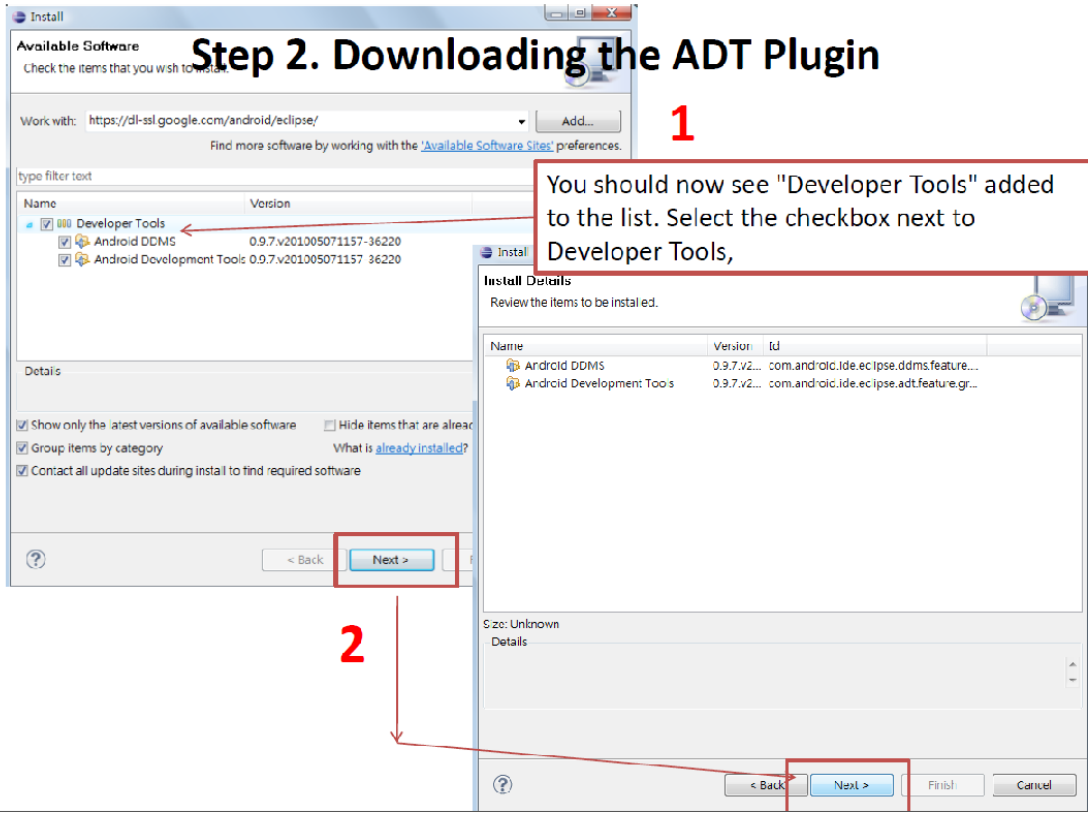
- 下載連結： http://www.eclipse.org/downloads/
- 版本建議: （建議使用 3.5.2 或更新的）

下載ADT Plugin

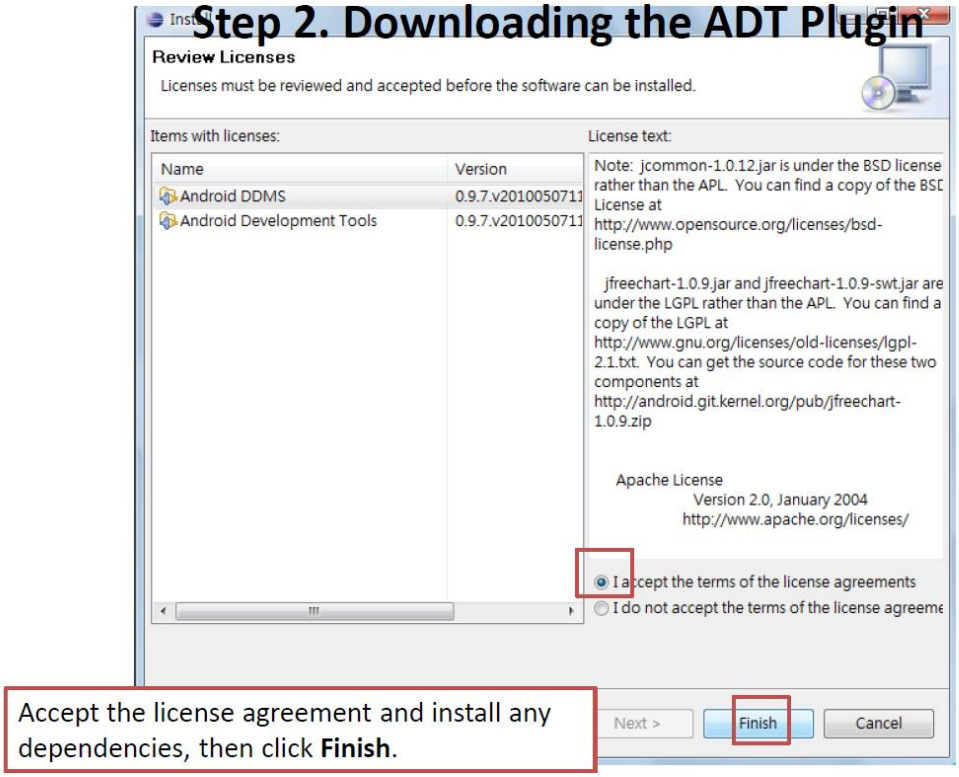
使用Eclipse安裝的更新管理器功能，在電腦上安裝 ADT的最新版本。



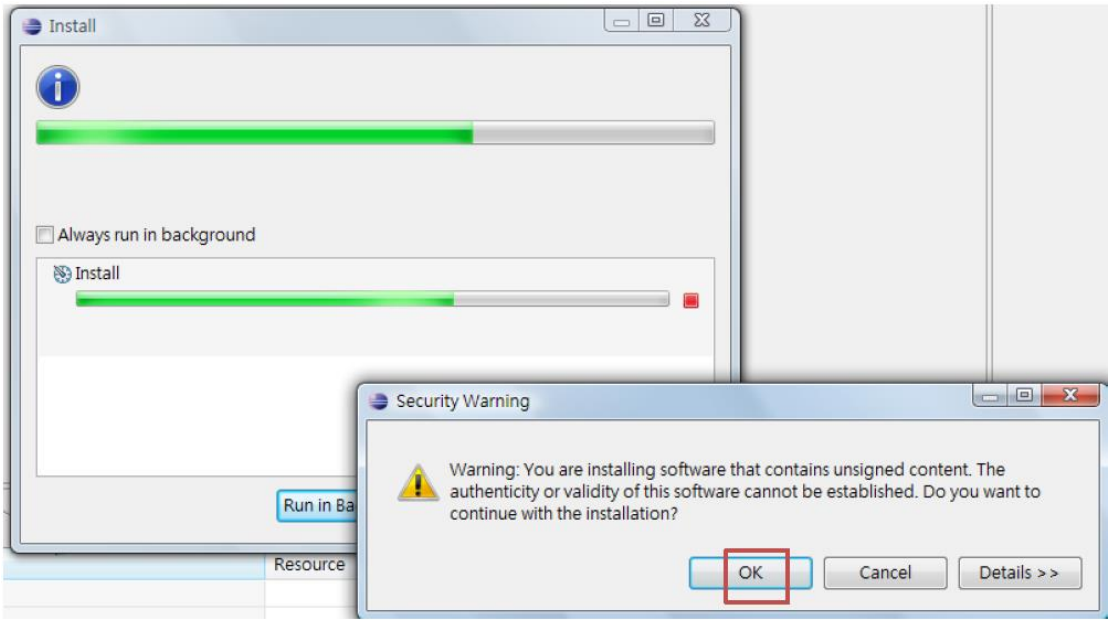
依序按照圖片步驟進行



依序按照圖片步驟進行

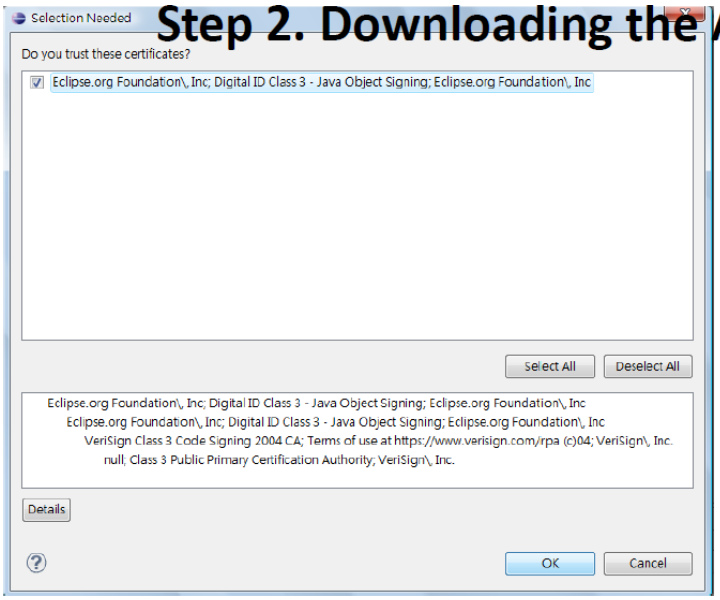


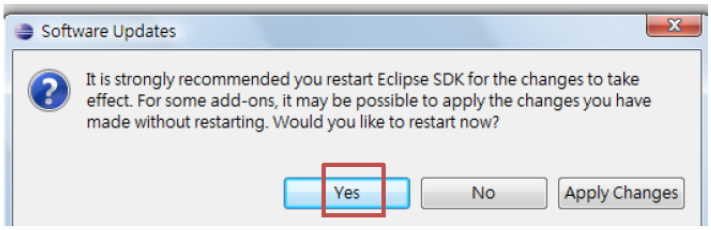
依序按照圖片步驟進行



依序按照圖片步驟進行

Step 2. Downloading the ADT Plugin



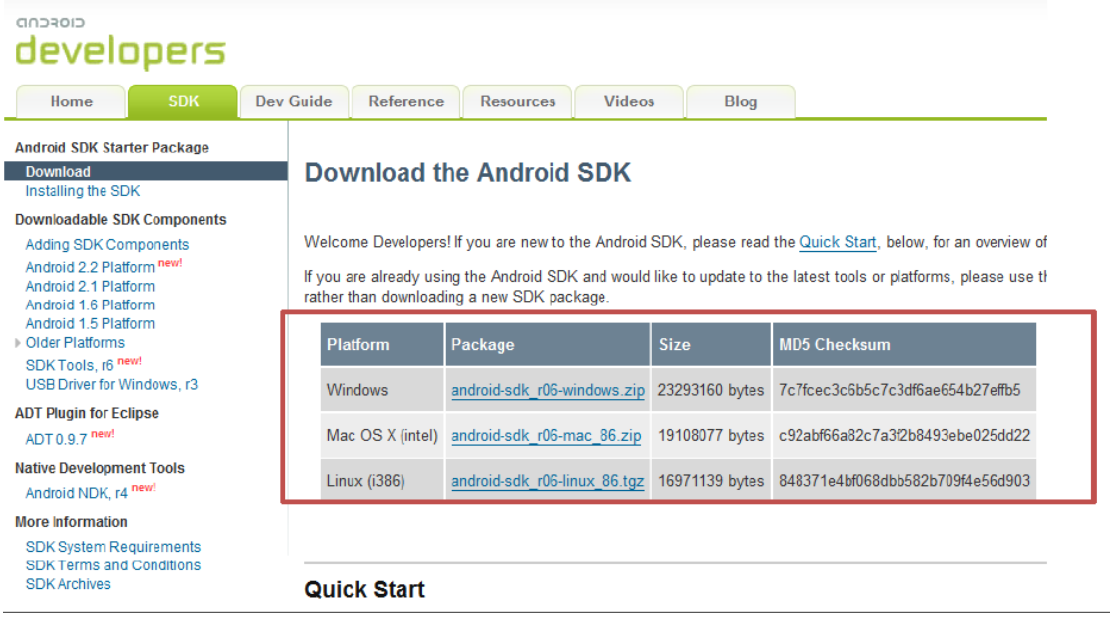


下載SDK封包

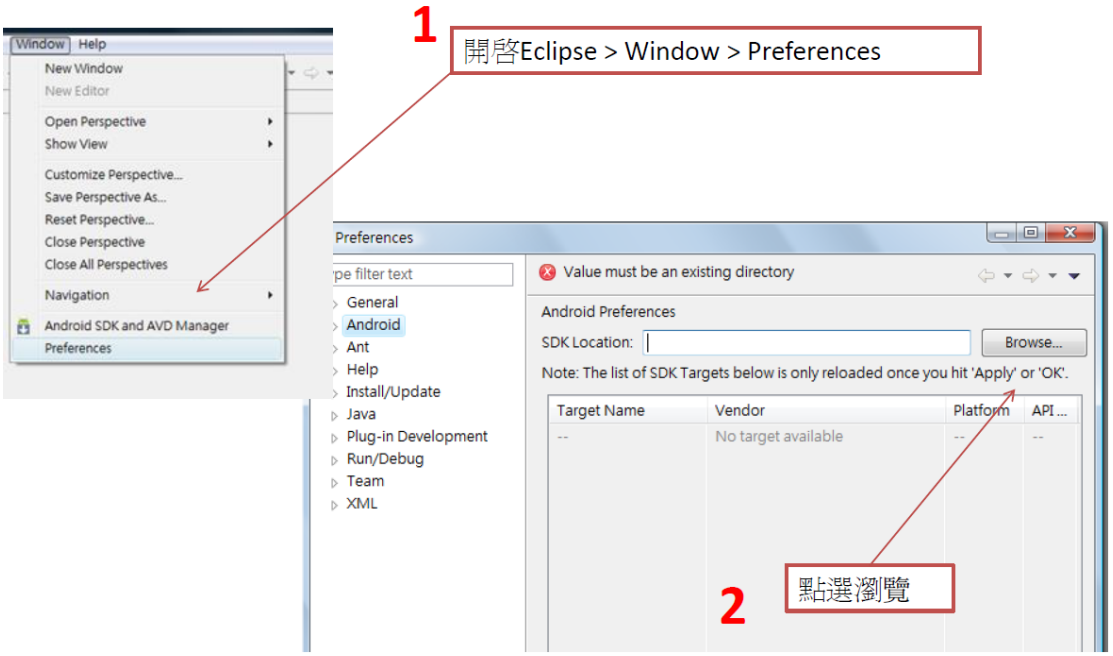
可從此網站下載最新的SDK封包

下載網址:http://developer.android.com/sdk/index.html

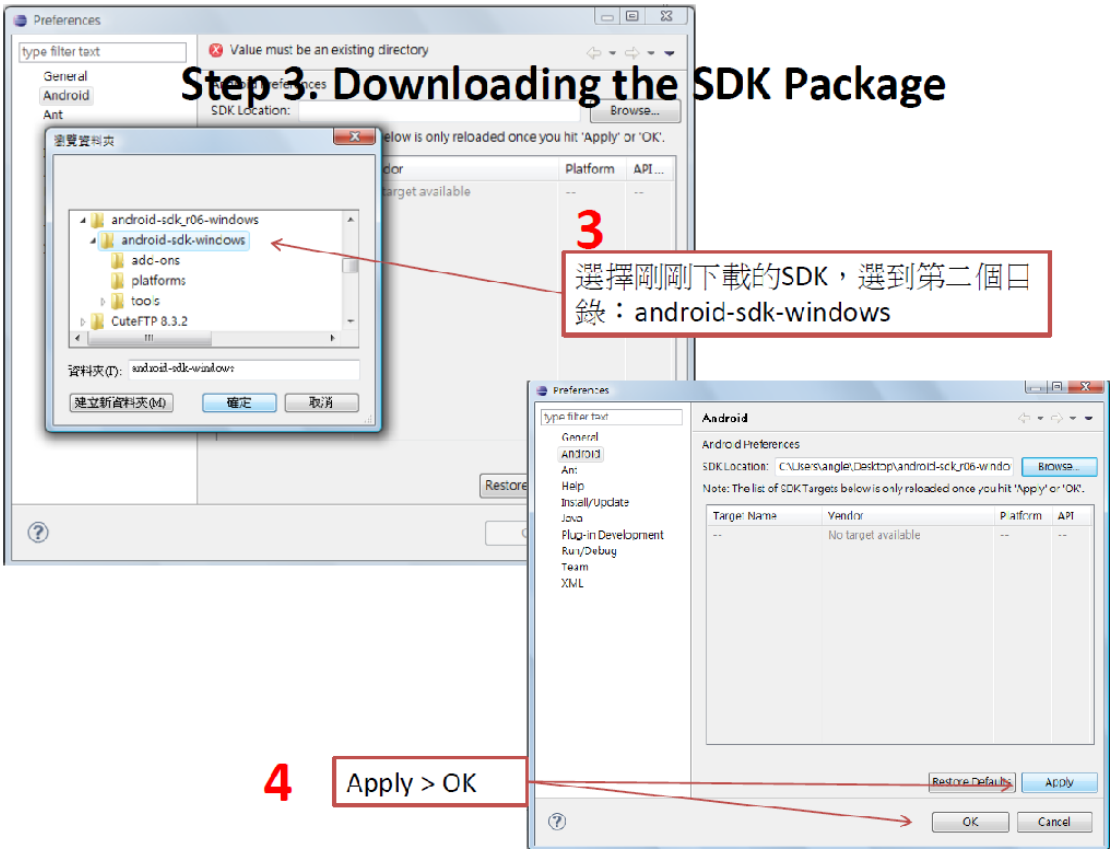
下載時請注意是否符合所需的版本



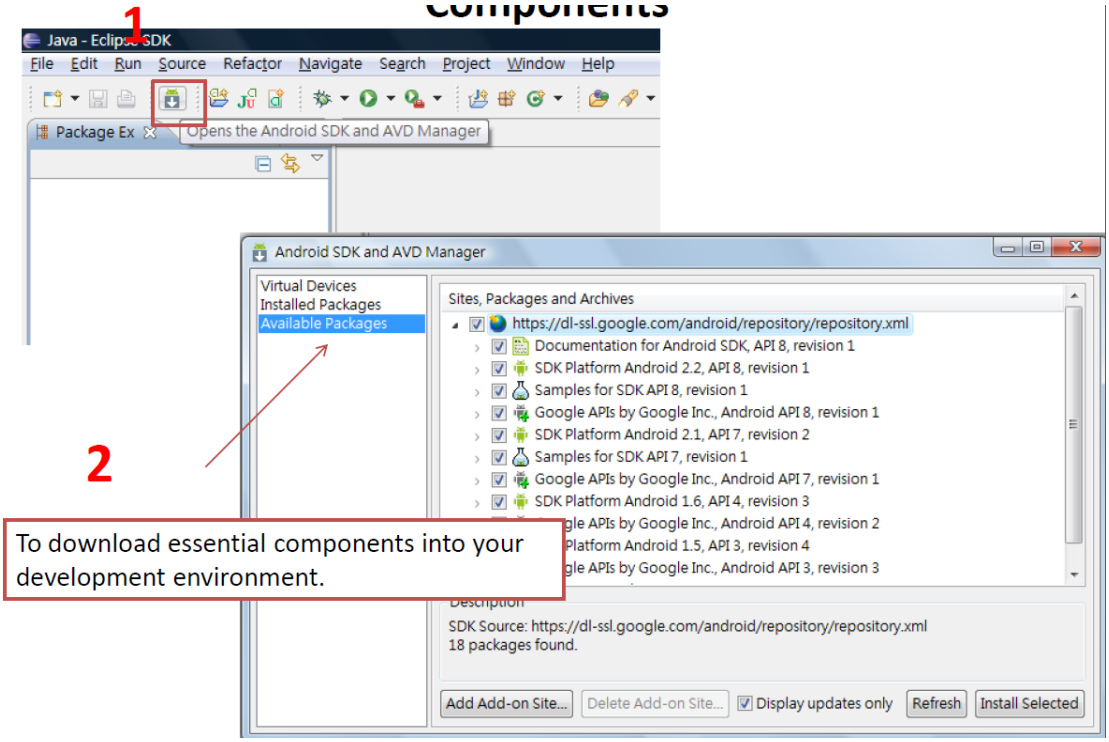
依序按照圖片步驟進行



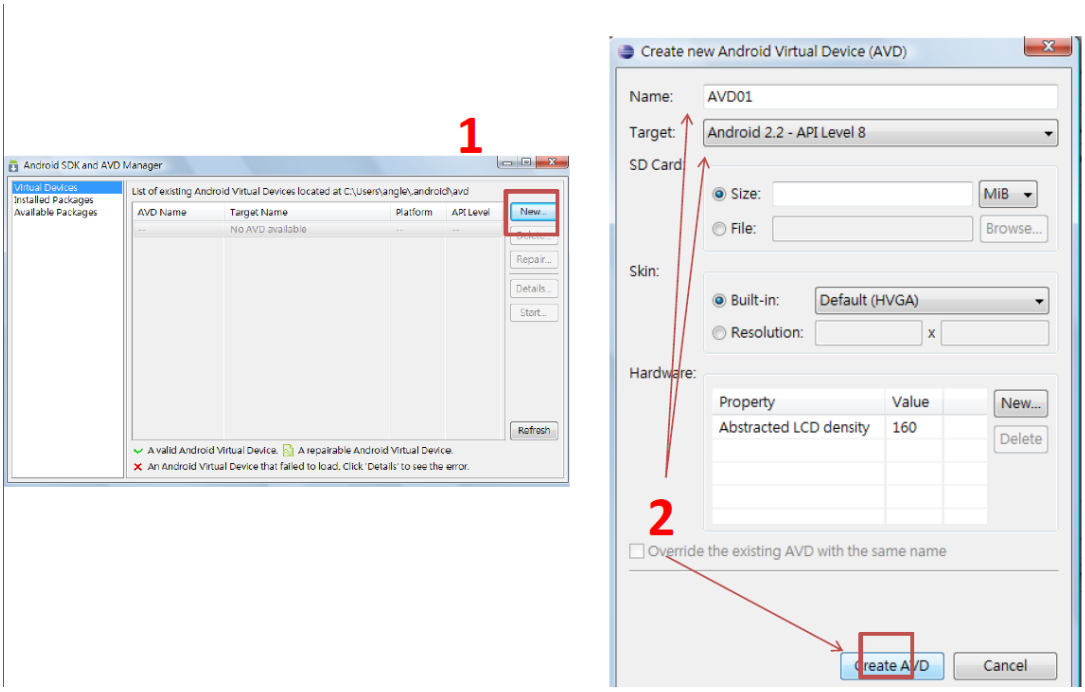
依序按照圖片步驟進行


Step 3. Downloading the SDK Package
3
選擇剛剛下載的SDK，選到第二個目錄：android-sdk-windows
4
Apply > OK

增加 Android 實驗平台和其他套件所需的實驗內容

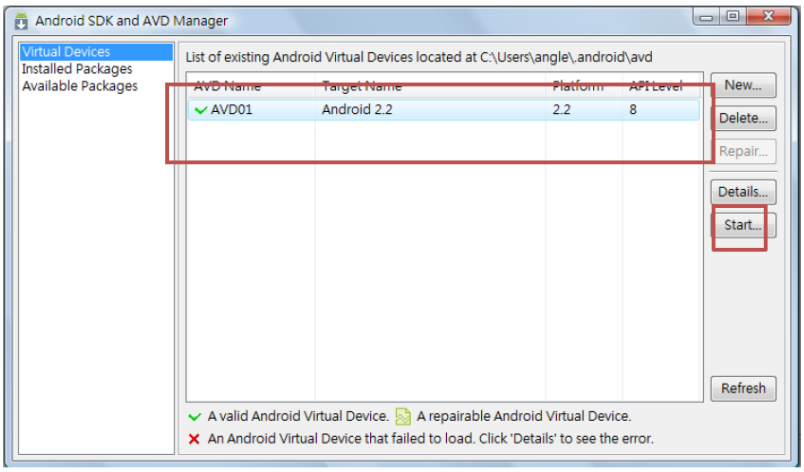


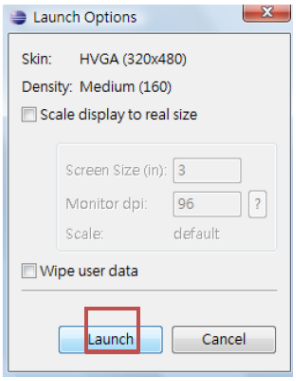
創新 Android 的虛擬創作環境



依序按照圖片步驟進行

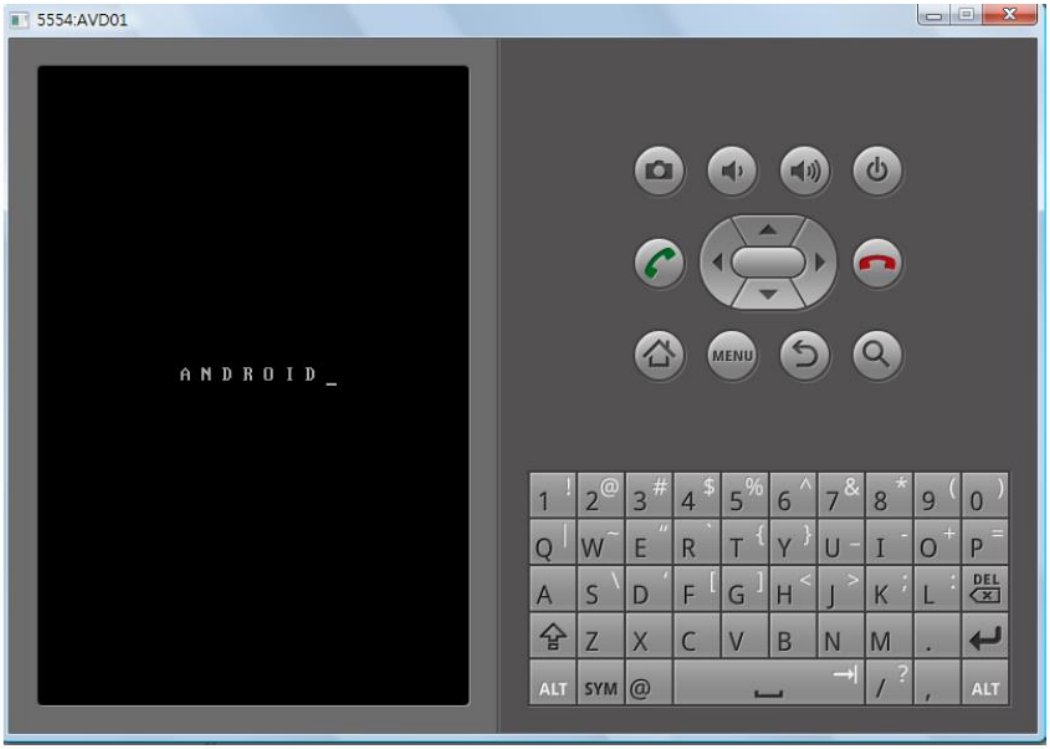
1





2

依序按照圖片步驟進行(到此結束)

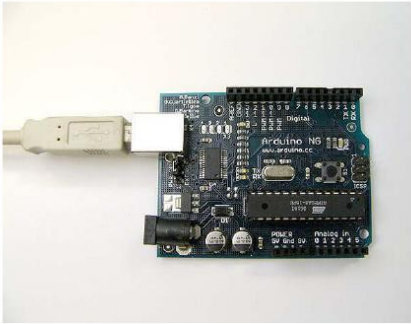


DONE

Step6 安裝 Arduino

安裝Arduino

- 將 Arduino 接上方頭的USB線，另外一端連接電腦
- 接上之後會出現FTDI的驅動程式安裝畫面



安裝Arduino

- 指定驅動程式安裝目錄，一般下載的Arduino資料夾裡面即會包含
- （EX：C:\Arduino-0022\drivers\FTDI USB Drivers）

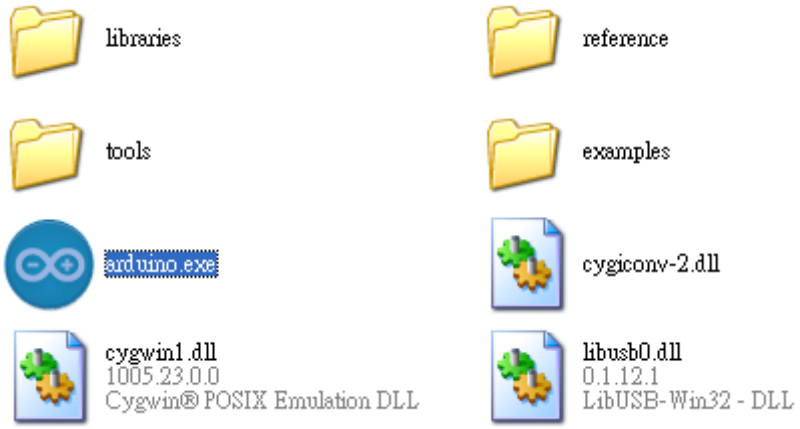
連接好硬體後，可以檢查所連接的 Arduino 目前是位於哪個 COM Port

你可以從控制台 → 系統 → 切換至「硬體」標籤 → 裝置管理員

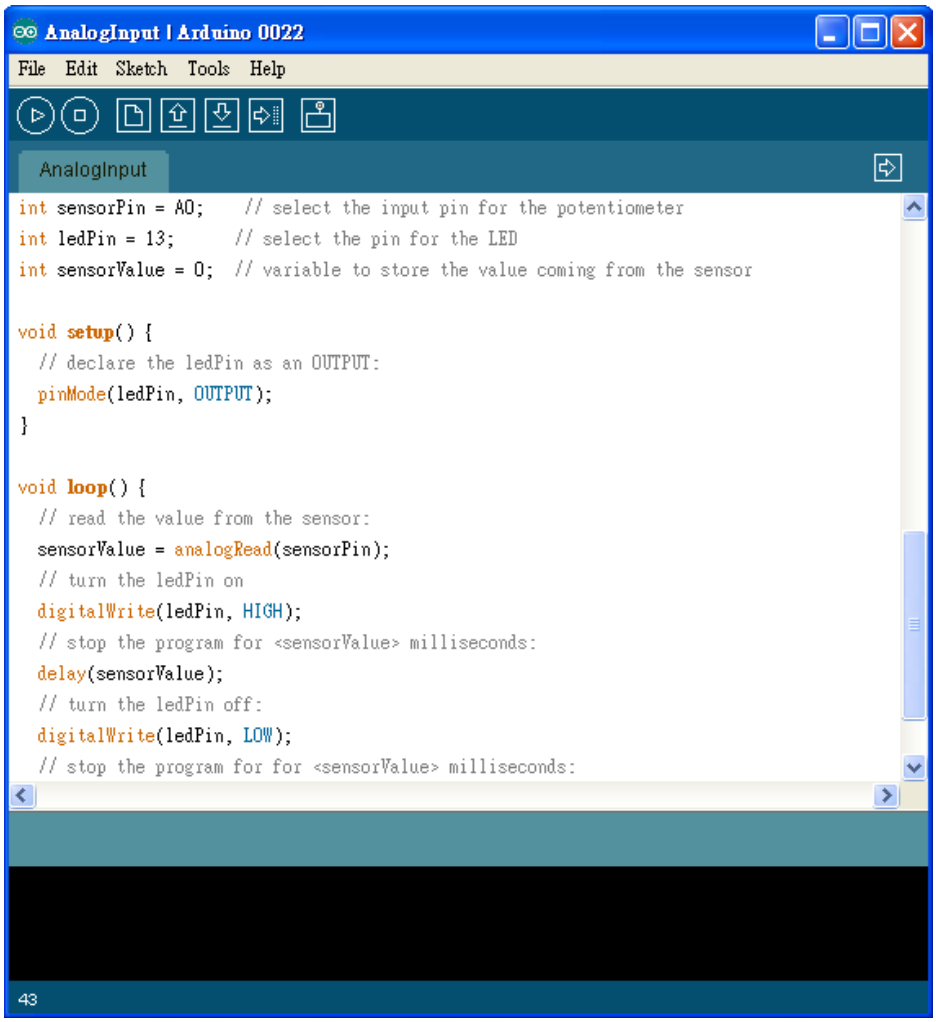
→ 連接埠 (COM 和 LPT)→ 檢查 USB Serial Port (COM4)，括號中的數字即為

你目前所連接的 Arduino Port

Arduino 程式開發工具



即可進入到 Arduino

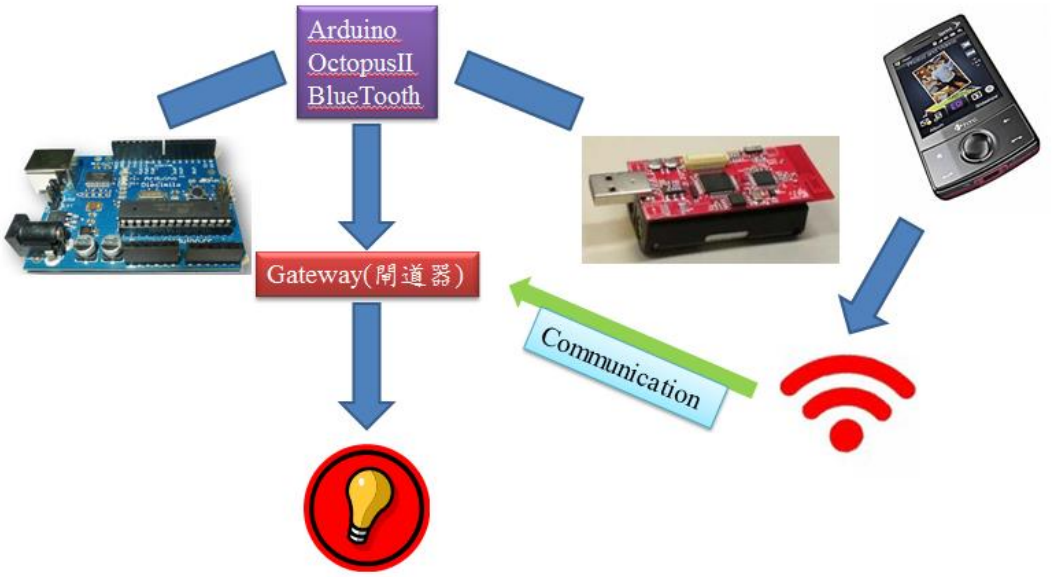


Part III 硬體電路整合

Step1 實驗架構流程圖

首先我們將介紹本實驗的整體架構，Octopus II 以及 Arduino 在本實驗中扮演控制者的角色，負責藍芽無線傳輸以及電燈的溝通與協調，使其結合形成可與使用者互動的智慧燈控系統。

在本實驗中，如下圖所示，手機將透過藍芽無線傳輸下達命令給閘道器，閘道器是由 Arduino，Octopus II，藍芽無線傳輸模組所組成的，手機下達命令時，Octopus II,Arduino 將會透過藍芽無線傳輸介面命令電燈控制開關。



Step2 選擇支援 GPIO 與 UART 的腳位

由 Step1 的實驗架構流程圖可以得知，為了接收 PIR 紅外線感測器的感測結果，我們需於 OctopusII 上選擇一個支援 GPIO 的腳位用以接收 PIR 紅外線感測器的 OUT 腳位輸出，此外，為了控制 VS1000D 音樂模組撥放音樂，我們需另外於 OctopusII 上選擇一個支援 UART-TX 的腳位用以傳送控制訊號給 VS1000D 音樂模組，由於 Octopus 之 MCU 為 MSP430，我們可由 MSP430 的 datasheet 查出支援 GPIO 的腳位，經由下表可得知各腳位可支援的通訊介面，若 datasheet 中的 Description 欄位內有標示 General-Purpose digital I/O 即代表該腳位支援 GPIO，因此腳位 P1.0(NO.12)至 P5.3(NO.47)皆可支援 GPIO 通訊界面，而 P3.3(NO.31)至 P3.7 (NO.35)皆可支援 UART 通訊界面，其中 P3.4 與 P3.6 為 UART 之 TX 傳送端腳位，P3.5 與 P3.7 則為 UART 之 RX 接收端腳位。

Terminal Functions

TERMINAL NAME	NO.	I/O	DESCRIPTION
AV_{CC}	64		Analog supply voltage, positive terminal. Supplies only the analog portion of ADC12 and DAC12.
AV_{SS}	62		Analog supply voltage, negative terminal. Supplies only the analog portion of ADC12 and DAC12.
DV_{CC}	1		Digital supply voltage, positive terminal. Supplies all digital parts.
DV_{SS}	63		Digital supply voltage, negative terminal. Supplies all digital parts.
P1.0/TACLK	12	I/O	General-purpose digital I/O pin/Timer_A, clock signal TACLK input
P1.1/TA0	13	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI0A input, compare: Out0 output/BSL transmit
P1.2/TA1	14	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI1A input, compare: Out1 output
P1.3/TA2	15	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI2A input, compare: Out2 output
P1.4/SMCLK	16	I/O	General-purpose digital I/O pin/SMCLK signal output
P1.5/TA0	17	I/O	General-purpose digital I/O pin/Timer_A, compare: Out0 output
P1.6/TA1	18	I/O	General-purpose digital I/O pin/Timer_A, compare: Out1 output
P1.7/TA2	19	I/O	General-purpose digital I/O pin/Timer_A, compare: Out2 output
P2.0/ACLK	20	I/O	General-purpose digital I/O pin/ACLK output
P2.1/TAINCLK	21	I/O	General-purpose digital I/O pin/Timer_A, clock signal at INCLK
P2.2/CAOUT/TA0	22	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI0B input/Comparator_A output/BSL receive
P2.3/CA0/TA1	23	I/O	General-purpose digital I/O pin/Timer_A, compare: Out1 output/Comparator_A input
P2.4/CA1/TA2	24	I/O	General-purpose digital I/O pin/Timer_A, compare: Out2 output/Comparator_A input
P2.5/Rosc	25	I/O	General-purpose digital I/O pin/input for external resistor defining the DCO nominal frequency
P2.6/ADC12CLK/ DMAE0	26	I/O	General-purpose digital I/O pin/conversion clock – 12-bit ADC/DMA channel 0 external trigger
P2.7/TA0	27	I/O	General-purpose digital I/O pin/Timer_A, compare: Out0 output
P3.0/STE0	28	I/O	General-purpose digital I/O pin/slave transmit enable – USART0/SPI mode
P3.1/SIMO0/SDA	29	I/O	General-purpose digital I/O pin/slave in/master out of USART0/SPI mode, I^2C data – USART0/I^2C mode
P3.2/SOMI0	30	I/O	General-purpose digital I/O pin/slave out/master in of USART0/SPI mode
P3.3/UCLK0/SCL	31	I/O	General-purpose digital I/O pin/external clock input – USART0/UART or SPI mode, clock output – USART0/SPI mode, I^2C clock – USART0/I^2C mode
P3.4/UTXD0	32	I/O	General-purpose digital I/O pin/transmit data out – USART0/UART mode
P3.5/URXD0	33	I/O	General-purpose digital I/O pin/receive data in – USART0/UART mode
P3.6/UTXD1†	34	I/O	General-purpose digital I/O pin/transmit data out – USART1/UART mode
P3.7/URXD1†	35	I/O	General-purpose digital I/O pin/receive data in – USART1/UART mode
P4.0/TB0	36	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI0A/B input, compare: Out0 output
P4.1/TB1	37	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI1A/B input, compare: Out1 output
P4.2/TB2	38	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI2A/B input, compare: Out2 output
P4.3/TB3†	39	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI3A/B input, compare: Out3 output
P4.4/TB4†	40	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI4A/B input, compare: Out4 output
P4.5/TB5†	41	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI5A/B input, compare: Out5 output
P4.6/TB6†	42	I/O	General-purpose digital I/O pin/Timer_B, capture: CCI6A input, compare: Out6 output
P4.7/TBCLK	43	I/O	General-purpose digital I/O pin/Timer_B, clock signal TBCLK input
P5.0/STE1†	44	I/O	General-purpose digital I/O pin/slave transmit enable – USART1/SPI mode
P5.1/SIMO1†	45	I/O	General-purpose digital I/O pin/slave in/master out of USART1/SPI mode
P5.2/SOMI1†	46	I/O	General-purpose digital I/O pin/slave out/master in of USART1/SPI mode
P5.3/UCLK1†	47	I/O	General-purpose digital I/O pin/external clock input – USART1/UART or SPI mode, clock output – USART1/SPI mode

† 16x, 161x devices only

Part IV 設計與製作智慧型手機與閘道器整合之燈控系統

Step1 編譯 Octopus II 程式

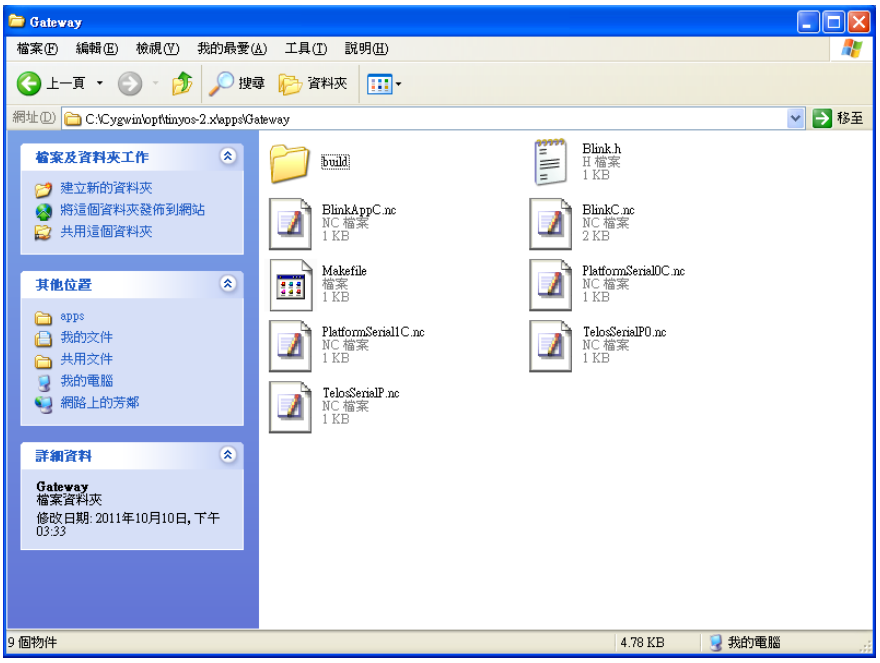
1.在 cygwin 命令列之下開啟 Octopus II 專案，此專案原始碼於 PartV 附錄程式碼中。

```
root@Shrimp ~
$ cd /opt/tinyos-2.x/apps

root@Shrimp /opt/tinyos-2.x/apps
$ cd Gateway

root@Shrimp /opt/tinyos-2.x/apps/Gateway
```

2.輸入 make octopus，編譯 Octopus II 程式

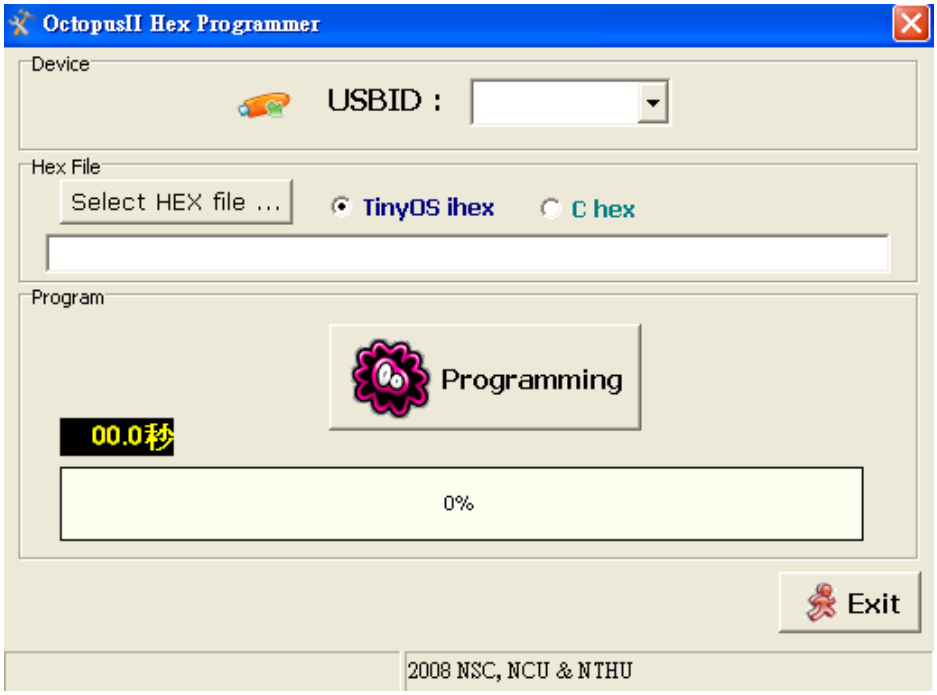
```
root@Shrimp /opt/tinyos-2.x/apps/Gateway
$ make octopus
mkdir -p build/octopus
    compiling BlinkAppC to a octopus binary
ncc -o build/octopus/main.exe  -Os -O -mdisable-hwmul -fnesc-separator=__ -Wall
-Wshadow -Wnesc-all -target=octopus -fnesc-cfile=build/octopus/app.c -board= -DD
EFINED_TOS_AM_GROUP=0x22 -DIDENT_PROGRAM_NAME=\"BlinkAppC\" -DIDENT_USER_ID=\"ro
ot\" -DIDENT_HOSTNAME=\"Shrimp\" -DIDENT_USER_HASH=0xa3bcf819L -DIDENT_UNIX_TIME
=0x4e929fd7L -DIDENT_UID_HASH=0x5a8aa5d4L  BlinkAppC.nc -lm
/opt/tinyos-2.x/tos/chips/cc2420/lpl/DummyLplC.nc:39:2: warning: #warning "*** L
OW POWER COMMUNICATIONS DISABLED ***"
    compiled BlinkAppC to build/octopus/main.exe
            11096 bytes in ROM
              267 bytes in RAM
msp430-objcopy --output-target=ihex build/octopus/main.exe build/octopus/main.ih
ex
    writing TOS image

root@Shrimp /opt/tinyos-2.x/apps/Gateway
$
```

3.編譯完成後，專案底下會生成 build 資料夾

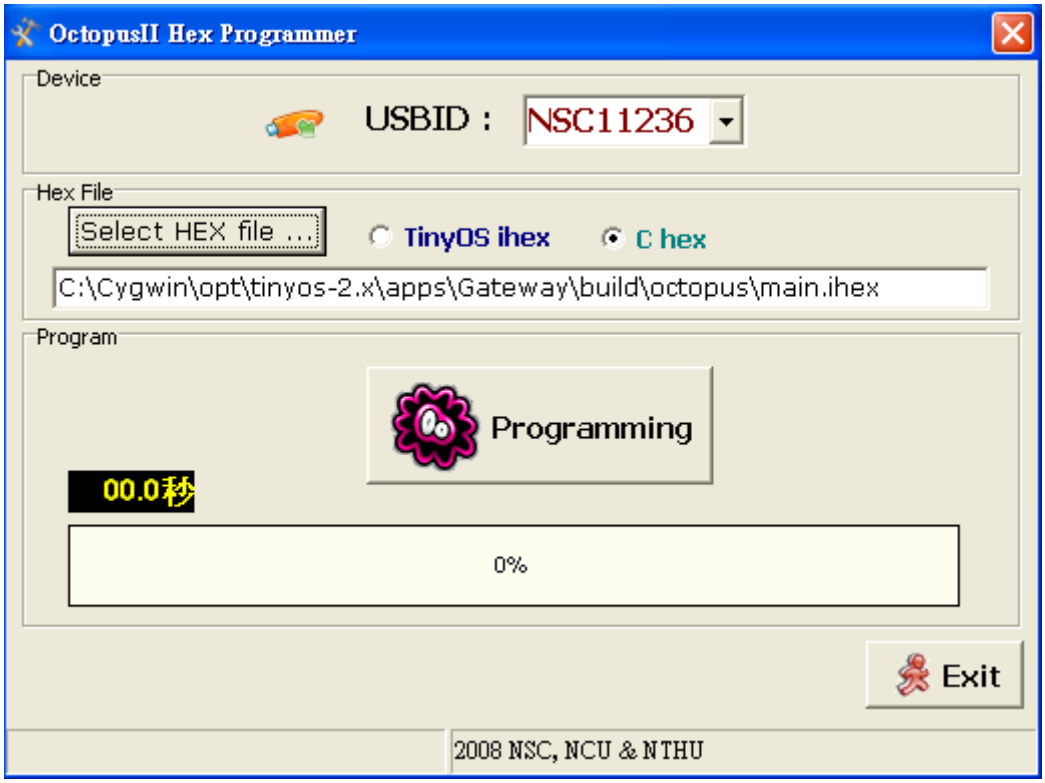

Gateway
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
資料夾
網址(D) C:\Cygwin\opt\tinyos-2.x\apps\Gateway
移至
檔案及資料夾工作
建立新的資料夾
將這個資料夾發佈到網站
共用這個資料夾
其他位置
apps
我的文件
共用文件
我的電腦
網路上的芳鄰
詳細資料
Gateway
檔案資料夾
修改日期: 2011年10月10日, 下午 03:33
build
Blink.h
H 檔案
1 KB
BlinkAppC.nc
NC 檔案
1 KB
BlinkC.nc
NC 檔案
2 KB
Makefile
檔案
1 KB
PlatformSerial0C.nc
NC 檔案
1 KB
PlatformSerial1C.nc
NC 檔案
1 KB
TelosSerialP0.nc
NC 檔案
1 KB
TelosSerialP.nc
NC 檔案
1 KB
9 個物件
4.78 KB
我的電腦

Step2 燒錄 Octopus II 程式

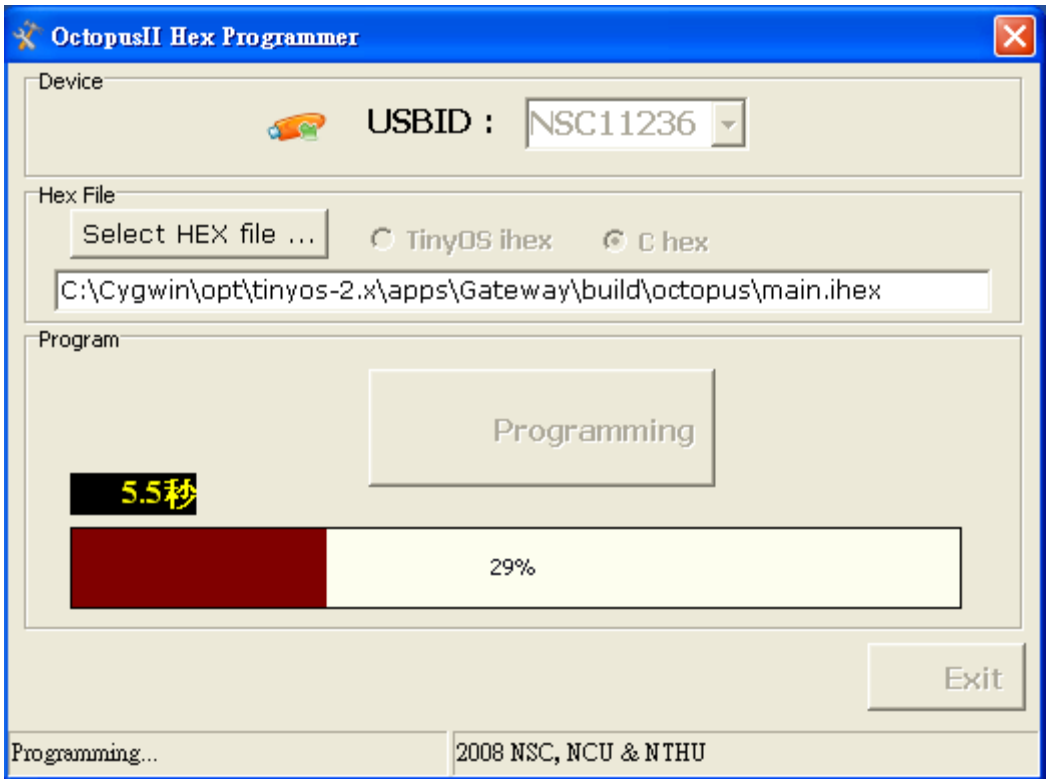
1. 開啟 Octopus II Hex Programmer 燒錄程式



2. 選擇 Gateway\build\octopus 底下的 main.ihex 檔案，並設定燒錄檔案為 C hex

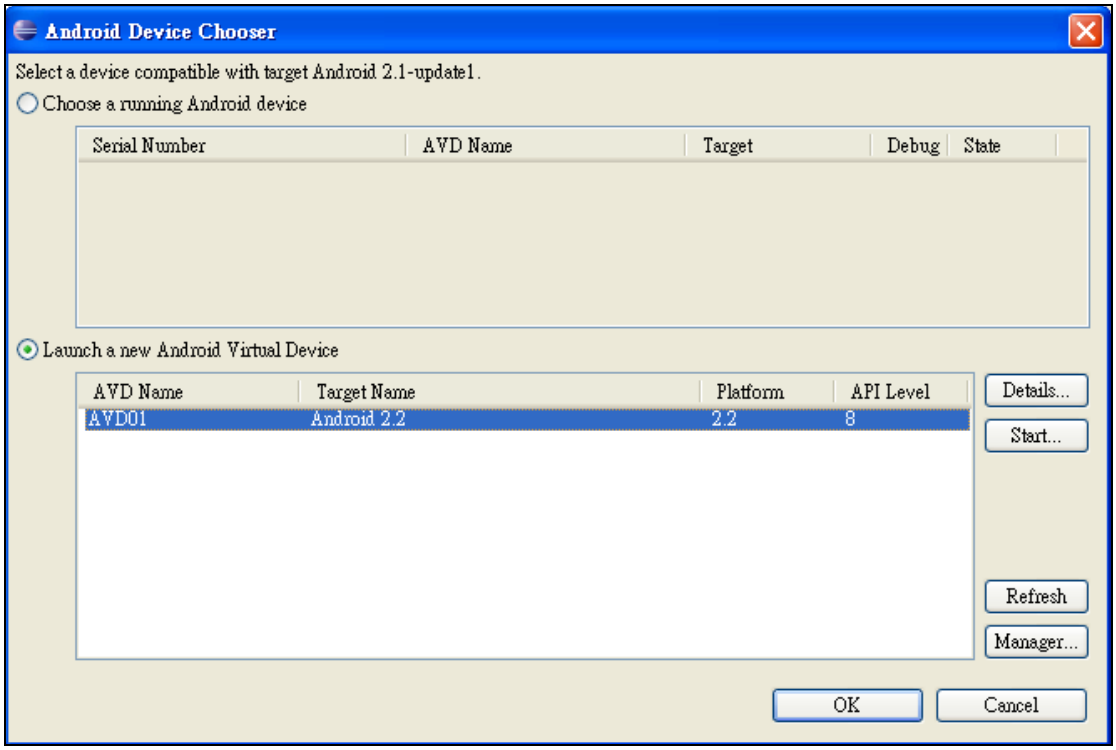


3. 按下 Programming 即可燒錄程式至 Octopus II



Step3 將程式安裝至 Android 手機上

打開 Andoird 專案後，若要編譯時，請先確定 Andoird 手機是否以連接至電腦上，若有，在 Android Device 裝置上會顯示其裝置 ID，點選 OK 即開始將程式安裝至手機上。



Part V –附錄程式碼

在 Android 程式上，我們必須賦予藍芽模組連線之能力，以及使用者介面，以下為我們所提供的介面定義：

使用者介面

CLASS	**METHOD**	**DERSCRIBE**
Activity	onCreate(Bundle)	程式啟動時將執行 onCreate(Bundle),在此我們利用 Properties 讀出程式之參數設定,並啟動藍芽背景連線程式
	startService(Intent)	啟動藍芽背景連線程式
	onResume()	程式起動後之將執行 onResume(),在此我們將藉由 ServiceConnection 取得藍芽背景連線程式之參考
	onPause()	程式若暫離前景，將執行 onPause()在此我們會要求藍芽連線暫停
	onDestroy()	程式若結束時會執行 onDestroy()，在此我們會關閉藍芽背景連線程式
OnClickListener	onClick(View)	藉由 OnClickListener 我們可以取的使用者操作介面之點擊事件，並將對應之程序寫於 onClick(View)
ServiceConnection	onServiceConnected(ComponentName, IBinder)	當程式和藍芽背景連線程式連結時將執行 onServiceConnected(ComponentName, IBinder)，我們在此取得藍芽背景連線程式之參考
	onServiceDisconnected(ComponentName)	當程式和藍芽背景連線程式斷連時將執行 onServiceDisconnected(ComponentName)，我們在此釋放藍芽背景連線程式之參考
Properties	loadFromXML(InputStream)	我們可以利用 Properties 之 loadFromXML(InputStream) 取得程式之參數設定

藍芽背景連線程式

CLASS	METHOD	DERSCRIBE
Service	onCreate(Bundle)	當藍芽背景連線程式建立時，將執行 onCreate(Bundle)
	onStartCommand(Intent,int,int startId)	當藍芽背景連線程式接收到啟動命令時將執行 onStartCommand(Intent,int,int startId)，我們在此撰寫藍芽連線程式
	onBind(Intent)	傳回和此藍芽背景連線程式溝通之管道
	onDestroy()	當藍芽背景連線程式中止時，將執行 onDestroy()，我們在此斷開藍芽連線
BluetoothSocket	getRemoteDevice()	Get the remote device this socket is connecting, or connected, to.
	connect()	Attempt to connect to a remote device.
	getInputStream()	Get the input stream associated with this socket.
	getOutputStream()	Get the output stream associated with this socket.
	close()	Immediately close this socket, and release all associated resources.
BluetoothAdapter	startDiscovery()	Start the remote device discovery process.
	getAddress()	Returns the hardware address of the local Bluetooth adapter.
	getRemoteDevice(String address)	Get a BluetoothDevice object for the given Bluetooth hardware address.
BluetoothDevice	Represents a remote Bluetooth device. A BluetoothDevice lets you create a connection with the respective device or query information about it, such as the name, address, class, and bonding state.	

ConnectService.java.

```
1.  package efa.gateway;
2.  import java.io.IOException;
3.  import java.util.UUID;
4.  import android.app.Activity;
5.  import android.app.ProgressDialog;
6.  import android.app.Service;
7.  import android.bluetooth.BluetoothAdapter;
8.  import android.bluetooth.BluetoothDevice;
9.  import android.bluetooth.BluetoothSocket;
10. import android.content.Intent;
11. import android.os.Binder;
12. import android.os.Handler;
13. import android.os.IBinder;
14. import android.util.Log;
15. public class ConnectService extends Service {
16.         // Bluetooth
17.         private static final int REQUEST_ENABLE_BT = 2;
18.         private static final UUID MY_UUID1 = UUID
19.                         .fromString("00001101-0000-1000-8000-00805F9B34FB");
20.         private static final String LOG_TAG = "ConnectService";
21.         private static String address = "00:0A:3A:2E:F4:29";
22.         private BluetoothSocket bluetoothSocket = null;
23.         private BluetoothAdapter bluetoothAdapter = null;
24.         private BluetoothDevice bluetoothDevice = null;
25.         private boolean D = true;
26.         private Activity activity;
27.         private Handler mHandler;
28.         private ProgressDialog pd;
29.         private final IBinder mBinder = new LocalBinder();
30.         public ProgressDialog getProgressDialog (){
31.                 return pd;
32.         }
33.         public void pdDismiss(){
34.                 if(pd!=null){
35.                         pd.dismiss();
```

```
36.             }
37.         }
38.         public void setActivity(Activity activity) {
39.             if (D)
40.                 Log.d(LOG_TAG, "setActivity = "+activity);
41.             this.activity = activity;
42.         }
43.         public void setHandler(Handler mHandler) {
44.             if (D)
45.                 Log.d(LOG_TAG, "setHandler = "+mHandler);
46.             this.mHandler = mHandler;
47.         }
48.         public void closeBluetoothSocket() throws IOException{
49.             if (D)
50.                 Log.d(LOG_TAG, "closeBluetoothSocket");
51.             if(bluetoothSocket!=null){
52.                 bluetoothSocket.close();
53.             }
54.         }
55.         public void writeOut(int hex) throws IOException{
56.             if (D)
57.                 Log.d(LOG_TAG, "write = "+hex);
58.             bluetoothSocket.getOutputStream().write(hex);
59.         }
60.         public void initBluetooth() throws IOException {
61.             if (D)
62.                 Log.d(LOG_TAG, "initBluetooth");
63.             bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
64.             bluetoothDevice = bluetoothAdapter.getRemoteDevice(address);
65.             bluetoothSocket = bluetoothDevice
66.                     .createRfcommSocketToServiceRecord(MY_UUID1);
67.             Thread connecteThread = new Thread(new Runnable() {
68.                 public void run() {
69.                     try {
70.                         if (D)
71.                             Log.d(LOG_TAG, "try to connect");
72.                         bluetoothSocket.connect(); // this will block thread
73.                         if (D)
```

```
74.                                     Log.d(LOG_TAG, "connect success");
75.                                 // connect success
76.                                 mHandler.obtainMessage(EntryActivity.CONNECT_SUCCESS,
77.                                         bluetoothSocket).sendToTarget();
78.                             } catch (IOException e) {
79.                                 // connect fail
80.                                 mHandler.obtainMessage(EntryActivity.CONNECT_FAIL, e)
81.                                         .sendToTarget();
82.                                 if (D)
83.                                     Log.getStackTraceString(e);
84.                             }
85.                         }
86.                 });
87.                 pd = ProgressDialog.show(activity, "", "連結至閘道器中...");
88.                 connecteThread.start();
89.                 // connect success
90.             }
91.             public class LocalBinder extends Binder {
92.                 ConnectService getService() {
93.                     return ConnectService.this;
94.                 }
95.             }
96.             @Override
97.             public IBinder onBind(Intent intent) {
98.                 return mBinder;
99.             }
100.
101.
102.            @Override
103.            public void onCreate() {
104.                // TODO Auto-generated method stub
105.                super.onCreate();
106.            }
107.            @Override
108.            public void onDestroy() {
109.                // TODO Auto-generated method stub
110.                super.onDestroy();
111.            }
```

```
112.
113.        @Override
114.        public int onStartCommand(Intent intent, int flags, int startId) {
115.        return super.onStartCommand(intent, flags, startId);
116.        }
117.        @Override
118.        public boolean onUnbind(Intent intent) {
119.                // TODO Auto-generated method stub
120.                return super.onUnbind(intent);
121.        }
122.  }
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
```

150..	

DeviceDemo.java

```
1.  package efa.gateway;
2.  import java.io.IOException;
3.  import java.util.Set;
4.  import java.util.UUID;
5.  import android.app.Activity;
6.  import android.app.AlertDialog;
7.  import android.app.Service;
8.  import android.bluetooth.BluetoothAdapter;
9.  import android.bluetooth.BluetoothDevice;
10. import android.bluetooth.BluetoothSocket;
11. import android.content.ComponentName;
12. import android.content.Context;
13. import android.content.DialogInterface;
14. import android.content.Intent;
15. import android.content.ServiceConnection;
16. import android.graphics.Bitmap;
17. import android.graphics.BitmapFactory;
18. import android.graphics.Canvas;
19. import android.graphics.Paint;
20. import android.graphics.Path;
21. import android.hardware.Sensor;
22. import android.hardware.SensorListener;
23. import android.hardware.SensorManager;
24. import android.media.MediaPlayer;
25. import android.os.Bundle;
26. import android.os.IBinder;
27. import android.os.Vibrator;
28. import android.util.DisplayMetrics;
29. import android.util.Log;
30. import android.view.Menu;
31. import android.view.MenuItem;
32. import android.view.MotionEvent;
33. import android.view.View;
34. import android.widget.TextView;
35. import android.widget.Toast;
36. public class DriverDemo extends Activity {
```

```
37.         private static final String LOG_TAG = "DriverDemo";
38.          /** Called when the activity is first created. */
39.          long ptime;
40.          private TextView showT;
41.         private SensorManager sensorM;
42.         private myView mv;
43.         boolean isOpen=false;
44.          private ConnectService mBoundService;
45.         public void onCreate(Bundle savedInstanceState) {
46.                 super.onCreate(savedInstanceState);
47.                 setContentView(R.layout.drive);
48.                 showT = (TextView)this.findViewById(R.id.showT);
49.                 sensorM = (SensorManager) this.getSystemService(Context.SENSOR_SERVICE);
50.                 mv =    new myView(DriverDemo.this);
51.                 setContentView(mv);
52.         }
53.         protected void onResume(){
54.          doBindService();
55.          sensorM.registerListener(slistener, SensorManager.SENSOR_ACCELEROMETER,
56.  SensorManager.SENSOR_DELAY_NORMAL);
57.                  super.onResume();
58.         }
59.         protected void onPause(){
60.                  try {
61.                          mBoundService.writeOut(0x40); //@
62.                  } catch (IOException e) {
63.                          // TODO Auto-generated catch block
64.                          Log.getStackTraceString(e);
65.                  }
66.            sensorM.unregisterListener(slistener);
67.            doUnbindService();
68.                 super.onPause();
69.         }
70.         public boolean onCreateOptionsMenu(Menu menu){
71.          menu.add(0,1,0,"關於");
72.          menu.add(0,2,1,"離開");
73.          return super.onCreateOptionsMenu(menu);
74.         }
```

```
75.
76.         private ServiceConnection mConnection = new ServiceConnection() {
77.                 public void onServiceConnected(ComponentName className, IBinder service) {
78.                         mBoundService = ((ConnectService.LocalBinder) service).getService();
79.                 }
80.                 public void onServiceDisconnected(ComponentName className) {
81.                         try {
82.                                 mBoundService.closeBluetoothSocket();
83.                         } catch (IOException e) {
84.                                 Log.getStackTraceString(e);
85.                         }
86.                         mBoundService = null;
87.                 }
88.         };
89.         private boolean mIsBound;
90.         void doBindService() {
91.                 // Establish a connection with the service. We use an explicit
92.                 // class name because we want a specific service implementation that
93.                 // we know will be running in our own process (and thus won't be
94.                 // supporting component replacement by other applications).
95.                 bindService(new Intent(DriverDemo.this,
96.                                 ConnectService.class), mConnection,
97.     Context.BIND_AUTO_CREATE);
98.                 mIsBound = true;
99.         }
100.        void doUnbindService() {
101.                if (mIsBound) {
102.                        // Detach our existing connection.
103.                        unbindService(mConnection);
104.                        mIsBound = false;
105.                }
106.        }
107.        public void doSesnd(int msg){
108.          try {
109.                  mBoundService.writeOut(msg);
110.                        Log.d(LOG_TAG, "Send Message = "+msg);
111.                } catch (IOException e) {
112.                        // TODO Auto-generated catch block
```

```
113.                    e.printStackTrace();
114.            }
115.    }
116.    private final SensorListener slistener =new SensorListener(){
117.            @Override
118.            public void onAccuracyChanged(int arg0, int arg1) {
119.                    // TODO Auto-generated method stub
120.            }
121.            @Override
122.            public void onSensorChanged(int sensor, float[] values) {
123.                    // TODO Auto-generated method stub
124.                    if(sensor==SensorManager.SENSOR_ACCELEROMETER){
125.                            long ctime = System.currentTimeMillis();
126.                            if((ctime-ptime)>400){
127.                            float x = values[SensorManager.DATA_X];
128.                                float y = values[SensorManager.DATA_Y];
129.                                float z = values[SensorManager.DATA_Z];
130.                                Log.d("Sensor", "x = "+x+",y = "+y+",z = "+z);
131.                                if(y>3){
132.                                  if(x>2){
133.                                          mv.setDraw(3);
134.                                        }
135.                                        else if(x<-2){
136.                                          mv.setDraw(2);
137.                                        }else{
138.                                          mv.setDraw(1);
139.                                        }
140.                                }
141.                                else if(y<-3){
142.                                  if(x>2){
143.                                          mv.setDraw(6);
144.                                        }
145.                                        else if(x<-2){
146.                                          mv.setDraw(5);
147.                                        }else{
148.                                          mv.setDraw(4);
149.                                        }
150.                                }
```

```
151.                                    else{
152.                                      mv.setDraw(0);
153.                                    }
154.                                    mv.invalidate();
155.                                    ptime=ctime;
156.                                }
157.                            }
158.                    }
159.            };
160.            class myView extends View{
161.                    private static final String LOG_TAG = "myView";
162.                      private int state,width,height;
163.                    Bitmap d1,d2,u1,u2,l1,l2,r1,r2,on,off,car,title;
164.                    private Vibrator v;
165.                    private MediaPlayer player;
166.                      public myView(Context context) {
167.                            super(context);
168.                            DisplayMetrics dm = new DisplayMetrics();
169.                            getWindowManager().getDefaultDisplay().getMetrics(dm);
170.                            width = dm.widthPixels;
171.                            height = dm.heightPixels;
172.                            d1 = BitmapFactory.decodeResource(getResources(), R.drawable.d1);
173.                            d2 = BitmapFactory.decodeResource(getResources(), R.drawable.d2);
174.                            u1 = BitmapFactory.decodeResource(getResources(), R.drawable.u1);
175.                            u2 = BitmapFactory.decodeResource(getResources(), R.drawable.u2);
176.                            l1 = BitmapFactory.decodeResource(getResources(), R.drawable.l1);
177.                            l2 = BitmapFactory.decodeResource(getResources(), R.drawable.l2);
178.                            r1 = BitmapFactory.decodeResource(getResources(), R.drawable.r1);
179.                            r2 = BitmapFactory.decodeResource(getResources(), R.drawable.r2);
180.                            off = BitmapFactory.decodeResource(getResources(), R.drawable.off);
181.                            on = BitmapFactory.decodeResource(getResources(), R.drawable.on);
182.                            car = BitmapFactory.decodeResource(getResources(), R.drawable.car);
183.                            title = BitmapFactory.decodeResource(getResources(), R.drawable.title);
184.                        v =(Vibrator)
185.  getApplication().getSystemService(Service.VIBRATOR_SERVICE);
186.                            // TODO Auto-generated constructor stub
187 .                     }
188.                      public void setDraw(int i){
```

```
189.                    this.state=i;
190.            }
191.            public void onDraw(Canvas    canvas){
192.                    super.onDraw(canvas);
193.                    Paint paint = new Paint();
194.                    paint.setAntiAlias(true);
195.                    canvas.drawBitmap(title, width/2-75,height/2-150,paint);
196.                    canvas.drawBitmap(car, width/2-28,height/2-30,paint);
197.                    canvas.drawBitmap(d1, width/2-20,height/2+30,paint);
198.                    canvas.drawBitmap(u1, width/2-20,height/2-115,paint);
199.                    canvas.drawBitmap(l1, width/2-115,height/2-20,paint);
200.                    canvas.drawBitmap(r1, width/2+40,height/2-20,paint);
201.                  switch(state){
202.                          case 0:
203.                              doSesnd(state);
204.                              break;
205.                          case 1:
206.                              canvas.drawBitmap(u2, width/2-20,height/2-115,paint);
207.                              doSesnd(state);
208.                              break;
209.                          case 2:
210.                              canvas.drawBitmap(u2, width/2-20,height/2-115,paint);
211.                              canvas.drawBitmap(l2, width/2-115,height/2-20,paint);
212.                              doSesnd(state);
213.                              break;
214.                          case 3:
215.                              canvas.drawBitmap(u2, width/2-20,height/2-115,paint);
216.                              canvas.drawBitmap(r2, width/2+40,height/2-20,paint);
217.                              doSesnd(state);
218.                                      break;
219.                          case 4:
220.                              canvas.drawBitmap(d2, width/2-20,height/2+30,paint);
221.                              doSesnd(state);
222.                                      break;
223.                          case 5:
224.                              canvas.drawBitmap(d2, width/2-20,height/2+30,paint);
225.                              canvas.drawBitmap(l2, width/2-115,height/2-20,paint);
226.                              doSesnd(state);
```

```
                                break;
                        case 6:
                            canvas.drawBitmap(d2, width/2-20,height/2+30,paint);
                            canvas.drawBitmap(r2, width/2+40,height/2-20,paint);
                            doSesnd(state);
                                break;
                       default:
                            state=0;
                    }
                    canvas.drawBitmap(on, width/2-230,height/2-30,paint);
                }

        }
}
```

265..	

EntryActivity.java

```
package efa.gateway;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.InvalidPropertiesFormatException;
import java.util.Properties;
import java.util.UUID;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.ComponentName;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.ServiceConnection;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Message;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnLongClickListener;
import android.widget.ImageButton;
import android.widget.Toast;
public class EntryActivity extends Activity implements OnClickListener,
                OnLongClickListener {
        private ImageButton tvButton, device1Button, device2Button, carButton,
```

```
                device3Button, device4Button;

    // Handle msg
    protected static final int MESSAGE_READ = 0;
    protected static final int CONNECT_FAIL = 1;
    protected static final int CONNECT_SUCCESS = 3;
    protected static final int CONNECT_DONE = 4;
    private static final String LOG_TAG = "EntryActivity";
    // Other
    private boolean D = true;
    private ConnectService mBoundService;
    private DisplayMetrics dm;
    // button image
    private Bitmap device1Image, device2Image, device3Image, device4Image;
    // button path
    private String device1path, device2path, device3path, device4path;
    // button flag
    boolean isDevice1On = false;
    boolean isDevice2On = false;
    boolean isDevice3On = false;
    boolean isDevice4On = false;
    //
    private Properties properties;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        findViewById();
        // start and bind srvice
        this.startService(new Intent(this, ConnectService.class));
        if (mBoundService == null)
            doBindService();
        // get window's height and width
        dm = new DisplayMetrics();
        this.getWindowManager().getDefaultDisplay().getMetrics(dm);
        //
        properties = new Properties();
        //
```

```
// load image
try {
        properties.loadFromXML(new FileInputStream(new File(
                        "/sdcard/gateway/properties.dtd")));
        Log.d(LOG_TAG, " load :properties =" + properties);
} catch (IOException e) {
        // TODO Auto-generated catch block
        Log.getStackTraceString(e);
}
device1path = properties.getProperty("device1path");
device2path = properties.getProperty("device2path");
device3path = properties.getProperty("device3path");
device4path = properties.getProperty("device4path");
if (device1path != null) {
        device1Image = getScaleImageByPath(device1path);
        device1Button.setImageBitmap(device1Image);
}
if (device2path != null) {
        device2Image = getScaleImageByPath(device2path);
        device2Button.setImageBitmap(device2Image);
}
if (device3path != null) {
        device3Image = getScaleImageByPath(device3path);
        device3Button.setImageBitmap(device3Image);
}
if (device4path != null) {
        device4Image = getScaleImageByPath(device4path);
        device4Button.setImageBitmap(device4Image);
}
String r1 = properties.getProperty("isDevice1On");
if (r1 != null && r1.equals("yes")) {
        device1Button.setBackgroundResource(R.drawable.device1on);
}
String r2 = properties.getProperty("isDevice2On");
if (r2 != null && r2.equals("yes")) {
        device2Button.setBackgroundResource(R.drawable.device2on);
}
String r3 = properties.getProperty("isDevice3On");
```

```
        if (r3 != null && r3.equals("yes")) {
            device3Button.setBackgroundResource(R.drawable.device3on);
        }
        String r4 = properties.getProperty("isDevice4On");
        if (r4 != null && r4.equals("yes")) {
            device4Button.setBackgroundResource(R.drawable.device4on);
        }
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(LOG_TAG, "Destroy");
        doUnbindService();
        // save image state
        try {
            properties.storeToXML(new FileOutputStream(new File(
                    "/sdcard/gateway/properties.dtd")), null);
            Log.d(LOG_TAG, " store :properties =" + properties);

        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    @Override
    protected void onResume() {
        Log.d(LOG_TAG, "onResume");
        super.onResume();
    }

    @Override
    protected void onPause() {
        Log.d(LOG_TAG, "onPause");
        super.onPause();
```

```
}
@Override
protected void onStop() {
	Log.d(LOG_TAG, "onStop");
	super.onStop();
}
private ServiceConnection mConnection = new ServiceConnection() {
	public void onServiceConnected(ComponentName className, IBinder service) {
		// This is called when the connection with the service has been
		// established, giving us the service object we can use to
		// interact with the service. Because we have bound to a explicit
		// service that we know is running in our own process, we can
		// cast its IBinder to a concrete class and directly access it.
		mBoundService = ((ConnectService.LocalBinder) service).getService();
		mBoundService.setHandler(mHandler);
		mBoundService.setActivity(EntryActivity.this);
		try {
			mBoundService.initBluetooth();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			Log.getStackTraceString(e);
		}
	}
	public void onServiceDisconnected(ComponentName className) {
		// This is called when the connection with the service has been
		// unexpectedly disconnected -- that is, its process crashed.
		// Because it is running in our same process, we should never
		// see this happen.
		try {
			mBoundService.closeBluetoothSocket();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			Log.getStackTraceString(e);
		}
		mBoundService = null;
	}
};
```

```
private boolean mIsBound;
void doBindService() {
        // Establish a connection with the service. We use an explicit
        // class name because we want a specific service implementation that
        // we know will be running in our own process (and thus won't be
        // supporting component replacement by other applications).
        bindService(new Intent(EntryActivity.this, ConnectService.class),
                        mConnection, Context.BIND_AUTO_CREATE);
        mIsBound = true;
}
void doUnbindService() {
        try {
                mBoundService.closeBluetoothSocket();
        } catch (IOException e) {
                // TODO Auto-generated catch block
                Log.getStackTraceString(e);
        }
        if (mIsBound) {
                // Detach our existing connection.
                unbindService(mConnection);
                mIsBound = false;
        }
}
public void findViewById() {
        tvButton = (ImageButton) this.findViewById(R.id.tvButton);
        tvButton.setOnClickListener(this);
        device1Button = (ImageButton) this.findViewById(R.id.device1Button);
        device1Button.setOnClickListener(this);
        device1Button.setOnLongClickListener(this);
        device2Button = (ImageButton) this.findViewById(R.id.device2Button);
        device2Button.setOnClickListener(this);
        device2Button.setOnLongClickListener(this);
        carButton = (ImageButton) this.findViewById(R.id.carButton);
        carButton.setOnClickListener(this);
        device3Button = (ImageButton) this.findViewById(R.id.device3Button);
        device3Button.setOnClickListener(this);
        device3Button.setOnLongClickListener(this);
        device4Button = (ImageButton) this.findViewById(R.id.device4Button);
```

```
            device4Button.setOnClickListener(this);
            device4Button.setOnLongClickListener(this);
        }
        @Override
        public void onClick(View v) {
            try {
                switch (v.getId()) {
                case R.id.tvButton:
                    mBoundService.writeOut(0x74); // t
                    Intent intent_i = new Intent(EntryActivity.this,
                                TvControl.class);
                    startActivity(intent_i);
                    Log.d(LOG_TAG, "Click tvButton");
                    break;
                case R.id.device1Button: // device 1
                    Log.d(LOG_TAG, "OnClick device1Button");
                    mBoundService.writeOut(0x64);// d
                    if (!isDevice1On) {
                        Log.d(LOG_TAG, "isDevice1On = true");
                        device1Button.setBackgroundResource(R.drawable.device1on);
                        properties.setProperty("isDevice1On", "yes");
                        isDevice1On = true;
                        mBoundService.writeOut(0x0B);// 11      (1:devie number, 1:on)
                    } else {
                        Log.d(LOG_TAG, "isDevice1On = false");
                        device1Button.setBackgroundResource(R.drawable.device1off);
                        properties.setProperty("isDevice1On", "no");
                        isDevice1On = false;
                        mBoundService.writeOut(0x0A);// 10      (1:devie number,
0:off)
                    }
                    Log.d(LOG_TAG, "Click device1Button");
                    //mBoundService.writeOut(0x40);// @ (end cmd)
                    break;
                case R.id.device2Button: // device 2
                    mBoundService.writeOut(0x64);// d
                    if (!isDevice2On) {
                        device2Button.setBackgroundResource(R.drawable.device2on);
```

```
                    properties.setProperty("isDevice2On", "yes");
                    isDevice2On = true;
                    mBoundService.writeOut(0x15);// 21      (2:devie number, 1:on)
                } else {
                    device2Button.setBackgroundResource(R.drawable.device2off);
                    properties.setProperty("isDevice2On", "no");
                    isDevice2On = false;
                    mBoundService.writeOut(0x14);// 20      (2:devie number, 0:off)
                }
                //mBoundService.writeOut(0x40);// @ (end cmd)
                break;
            case R.id.carButton:
                mBoundService.writeOut(0x63);// c
                Intent intent_c = new Intent(EntryActivity.this,
                        DriverDemo.class);
                startActivity(intent_c);
                Log.d(LOG_TAG, "Click carButton");
                break;
            case R.id.device3Button:// device 3
                mBoundService.writeOut(0x64);// d
                if (!isDevice3On) {
                    device3Button.setBackgroundResource(R.drawable.device3on);
                    isDevice3On = true;
                    properties.setProperty("isDevice3On", "yes");
                    mBoundService.writeOut(0x1F);// 31      (3:devie number, 1:on)
                } else {
                    device3Button.setBackgroundResource(R.drawable.device3off);
                    properties.setProperty("isDevice3On", "no");
                    isDevice3On = false;
                    mBoundService.writeOut(0x1E);// 30      (3:devie number,
0:off)
                }
                mBoundService.writeOut(0x40);// @ (end cmd)
                break;
            case R.id.device4Button:    // device 4
                mBoundService.writeOut(0x64);// d
                if (!isDevice4On) {
                    device4Button.setBackgroundResource(R.drawable.device4on);
```

```
                    isDevice4On = true;
                    properties.setProperty("isDevice4On", "yes");
                    mBoundService.writeOut(0x29);// 41      (4:devie number, 1:on)
                } else {
                    device4Button.setBackgroundResource(R.drawable.device4off);
                    properties.setProperty("isDevice4On", "no");
                    isDevice4On = false;
                    mBoundService.writeOut(0x28);// 40      (3:devie number, 0:off)
                }
            //      mBoundService.writeOut(0x40);// @ (end cmd)
                break;
            }
        } catch (IOException e) {
            Log.getStackTraceString(e);
        }
    }
    @Override
    public boolean onLongClick(View v) {
        Log.d(LOG_TAG, "onLongClick");
        switch (v.getId()) {
        case R.id.device1Button: // device 1
            Log.d(LOG_TAG, "LongClick device1Button");
            Intent intent_1 = new Intent(EntryActivity.this,
                    GalleryActivity.class);
            startActivityForResult(intent_1, R.id.device1Button);
            break;
        case R.id.device2Button: // device 2
            Log.d(LOG_TAG, "LongClick device2Button");
            Intent intent_2 = new Intent(EntryActivity.this,
                    GalleryActivity.class);
            startActivityForResult(intent_2, R.id.device2Button);
            break;
        case R.id.device3Button:// device 3
            Intent intent_3 = new Intent(EntryActivity.this,
                    GalleryActivity.class);
            startActivityForResult(intent_3, R.id.device3Button);
            Log.d(LOG_TAG, "LongClick device3Button");
            break;
```

```
            case R.id.device4Button: // device 4
                Intent intent_4 = new Intent(EntryActivity.this,
                        GalleryActivity.class);
                startActivityForResult(intent_4, R.id.device4Button);
                Log.d(LOG_TAG, "LongClick device4Button");
                break;
            }
            return true;
        }
        private final Handler mHandler = new Handler() {
            public void handleMessage(Message msg) {
                switch (msg.what) {
                case CONNECT_SUCCESS:
                    if (D)
                        Log.d(LOG_TAG, "CONNECT_SUCCESS");
                    mBoundService.pdDismiss();
                    Toast.makeText(EntryActivity.this, "連結成功",
Toast.LENGTH_SHORT)
                            .show();
                    break;
                case CONNECT_FAIL:
                    if (D)
                        Log.d(LOG_TAG, "CONNECT_FAIL");
                    mBoundService.pdDismiss();
                    new AlertDialog.Builder(EntryActivity.this).setTitle("連結失敗")
                            .setMessage("程式將關閉").setPositiveButton("確定",
                                    new DialogInterface.OnClickListener() {
                                        @Override
                                    public void onClick(DialogInterface dialog,
                                            int which) {
                                        // TODO Auto-generated method stub
                                            EntryActivity.this.finish();
                                        }
                                    }).show();
                    break;
                }
            }
        };
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    if (data != null) {
        Bundle bundle = data.getExtras();
        String path = bundle.getString("path");
        switch (requestCode) {
        case R.id.device1Button: // device 1
            device1Image = getScaleImageByPath(path);
            properties.setProperty("device1path", path);
            device1Button.setImageBitmap(device1Image);
            Log.d(LOG_TAG, "onActivityResult device1path = " + path);
            break;
        case R.id.device2Button: // device 2
            device2Image = getScaleImageByPath(path);
            properties.setProperty("device2path", path);
            device2Button.setImageBitmap(device2Image);
            Log.d(LOG_TAG, "onActivityResult device1path = " + path);
            break;
        case R.id.device3Button:// device 3
            device3Image = getScaleImageByPath(path);
            properties.setProperty("device3path", path);
            device3Button.setImageBitmap(device3Image);
            Log.d(LOG_TAG, "onActivityResult device1path = " + path);
            break;
        case R.id.device4Button: // device 4
            device4Image = getScaleImageByPath(path);
            properties.setProperty("device4path", path);
            device4Button.setImageBitmap(device4Image);
            Log.d(LOG_TAG, "onActivityResult device1path = " + path);
            break;
        }
    }
}
private Bitmap getScaleImageByPath(String path) {
    Bitmap bm = BitmapFactory.decodeFile(path);
    bm = Bitmap.createScaledBitmap(bm, dm.widthPixels / 3,
```

```
                                dm.widthPixels / 3, true);
                return bm;
        }
}
```

TvControl.java

```
package efa.gateway;
import java.io.IOException;
import java.util.UUID;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.ComponentName;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Message;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
public class TvControl extends Activity implements OnClickListener {
	private static final String LOG_TAG = "TvControl";
	// UI
	private Button number_1_button, number_2_button, number_3_button,
			number_4_button, number_5_button, number_6_button, number_7_button,
			number_8_button, number_9_button, number_0_button;
	private Button switch_input_button, on_off_button, volume_up_button,
			volume_down_button, ch_pre_button, ch_next_button;
	private ConnectService mBoundService;
	private boolean D = true;
	@Override
	public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.tv);
        findViewById();
    }
    @Override
    protected void onResume() {
        // TODO Auto-generated method stub
        super.onResume();
        doBindService();
    }
    @Override
    protected void onPause() {

        Log.d(LOG_TAG, "onPause");
        try {
            mBoundService.writeOut(0x42); //B
            mBoundService.writeOut(0x41); //A
            mBoundService.writeOut(0x43); //C
            mBoundService.writeOut(0x4B); //K
            mBoundService.writeOut(0x21);//!
        } catch (IOException e) {
            // TODO Auto-generated catch block
            Log.getStackTraceString(e);
        }
        doUnbindService();
        super.onPause();
    }
    private ServiceConnection mConnection = new ServiceConnection() {
        public void onServiceConnected(ComponentName className, IBinder service) {
            mBoundService = ((ConnectService.LocalBinder) service).getService();
        }
        public void onServiceDisconnected(ComponentName className) {
            try {
                mBoundService.closeBluetoothSocket();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                Log.getStackTraceString(e);
            }
            mBoundService = null;
```

```
            }
        };
        private boolean mIsBound;
        void doBindService() {
            bindService(new Intent(TvControl.this,
                        ConnectService.class), mConnection,
Context.BIND_AUTO_CREATE);
            mIsBound = true;
        }
        void doUnbindService() {
            if (mIsBound) {
                // Detach our existing connection.
                unbindService(mConnection);
                mIsBound = false;
            }
        }
        private void findViewById() {
            number_1_button = (Button) this.findViewById(R.id.number_1);
            number_1_button.setOnClickListener(this);
            number_2_button = (Button) this.findViewById(R.id.number_2);
            number_2_button.setOnClickListener(this);
            number_3_button = (Button) this.findViewById(R.id.number_3);
            number_3_button.setOnClickListener(this);
            number_4_button = (Button) this.findViewById(R.id.number_4);
            number_4_button.setOnClickListener(this);
            number_5_button = (Button) this.findViewById(R.id.number_5);
            number_5_button.setOnClickListener(this);
            number_6_button = (Button) this.findViewById(R.id.number_6);
            number_6_button.setOnClickListener(this);
            number_7_button = (Button) this.findViewById(R.id.number_7);
            number_7_button.setOnClickListener(this);
            number_8_button = (Button) this.findViewById(R.id.number_8);
            number_8_button.setOnClickListener(this);
            number_9_button = (Button) this.findViewById(R.id.number_9);
            number_9_button.setOnClickListener(this);
            number_0_button = (Button) this.findViewById(R.id.number_0);
            number_0_button.setOnClickListener(this);
```

```
        switch_input_button = (Button) this.findViewById(R.id.switch_input);
        switch_input_button.setOnClickListener(this);
        on_off_button = (Button) this.findViewById(R.id.on_off);
        on_off_button.setOnClickListener(this);
        volume_up_button = (Button) this.findViewById(R.id.volume_up);
        volume_up_button.setOnClickListener(this);
        volume_down_button = (Button) this.findViewById(R.id.volume_down);
        volume_down_button.setOnClickListener(this);
        ch_pre_button = (Button) this.findViewById(R.id.ch_pre);
        ch_pre_button.setOnClickListener(this);
        ch_next_button = (Button) this.findViewById(R.id.ch_next);
        ch_next_button.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        try {
            mBoundService.writeOut(0x74);//t
            mBoundService.writeOut(0x76);//v
            mBoundService.writeOut(0x2C);//,
            switch (v.getId()) {
            case R.id.number_1:
                mBoundService.writeOut(0x31);//1
                break;
            case R.id.number_2:
                mBoundService.writeOut(0x32);//2
                break;
            case R.id.number_3:
                mBoundService.writeOut(0x33);//3
                break;
            case R.id.number_4:
                mBoundService.writeOut(0x34);//4
                break;
            case R.id.number_5:
                mBoundService.writeOut(0x35);//5
                break;
            case R.id.number_6:
                mBoundService.writeOut(0x36);//6
                break;
```

```
case R.id.number_7:
        mBoundService.writeOut(0x37);//7
        break;
case R.id.number_8:
        mBoundService.writeOut(0x38);//8
        break;
case R.id.number_9:
        mBoundService.writeOut(0x39);//9
        break;
case R.id.number_0:
        mBoundService.writeOut(0x30);//0
        break;
case R.id.switch_input:
        mBoundService.writeOut(0x73);//s
        break;
case R.id.on_off:
        mBoundService.writeOut(0x6F);//o
        mBoundService.writeOut(0x6E);//n
        break;
case R.id.volume_up:
        mBoundService.writeOut(0x75);//u
        mBoundService.writeOut(0x70);//p
        break;
case R.id.volume_down:
        mBoundService.writeOut(0x64);//d
        mBoundService.writeOut(0x6F);//o
        break;
case R.id.ch_pre:
        mBoundService.writeOut(0x70);//p
        mBoundService.writeOut(0x72);//r
        mBoundService.writeOut(0x65);//e
        break;
case R.id.ch_next:
        mBoundService.writeOut(0x6E);//n
        mBoundService.writeOut(0x65);//e
        mBoundService.writeOut(0x78);//x
        mBoundService.writeOut(0x74);//t
        break;
```

```
                }

                mBoundService.writeOut(0x21);//!

            } catch (IOException e) {
                // TODO Auto-generated catch block
                Log.getStackTraceString(e);
            }
        }
}
```

Arduino Code

```
#include <NewSoftSerial.h>
#include <IRremote.h>
// for all
   byte incomingByte;
  int DEMO_TYPE;
  boolean isDemoing = false;
  // soft uart for octopus
  NewSoftSerial mySerial(6, 7);
  //car
  int uPin = 5;
int dPin = 4;
int lPin = 3;
int rPin = 2;
// ir send
char buffer[100];
char read;
int i,j;
void setup()
{
   Serial.begin(9600);
   Serial.println("gateway start");
   // set the data rate for the NewSoftSerial port
   mySerial.begin(115200);
  //mySerial.println("Hello, world?");
}
void loop()                                // run over and over again
{
     if(!isDemoing){ // if is not demoing
          byte incomingByte;
            while (Serial.available() && !isDemoing) {
                    incomingByte = Serial.read();
                    Serial.print("receive byte = ");
                    Serial.println(incomingByte);
                    switch (incomingByte){
                              Serial.println(incomingByte);
```

```
                    case 0x74: // t (ir_send)
                            DEMO_TYPE=3;
                            isDemoing=true;
                            Serial.println("TvControl GO");
                    break;
                    case 0x64: //d (device)
                            DEMO_TYPE=1;
                            isDemoing=true;
                            Serial.println("DeviceDemo GO");
                    break;
                    case 0x63: // c (carDemo)
                        pinMode(uPin, OUTPUT);
                        pinMode(dPin, OUTPUT);
                        pinMode(lPin, OUTPUT);
                        pinMode(rPin, OUTPUT);
                          DEMO_TYPE=2;
                          isDemoing=true;
                          Serial.println("CarDemo GO");
                      break;
                }// end switch
        }//end while
    }else{
      Serial.print("DEMOING    DEMO_TYPE = ");
      Serial.println(DEMO_TYPE);
      switch(DEMO_TYPE){
          case 0:
          isDemoing=false;
          break;
          case 1: // Device Demo
//                if (mySerial.available()) {
//                Serial.print((char)mySerial.read());
//                }
              while (!Serial.available()) {} // block
                    incomingByte = Serial.read();
                    mySerial.print((char)incomingByte);
                    Serial.println(incomingByte);
              Serial.println("BACK");
```

```
        isDemoing=false;
    break;
    case 2: // carDemo
    while(true){
        if (Serial.available()) {
            incomingByte = Serial.read();
            // code
              if(incomingByte==1 ){
                  Serial.print("UP");
                  digitalWrite(uPin, HIGH);
            }
            if(incomingByte==2 ){
                  Serial.print("UP_LEFT");
                  digitalWrite(uPin, HIGH);
                  digitalWrite(lPin, HIGH);
            }
            if(incomingByte==3 ){
                  Serial.print("UP_RIGHT");
                  digitalWrite(uPin, HIGH);
                  digitalWrite(rPin, HIGH);
            }
            if(incomingByte==4 ){
                  Serial.print("DOWN");
                  digitalWrite(dPin, HIGH);
            }
            if(incomingByte==5 ){
                  Serial.print("DOWN_LEFT");
                  digitalWrite(dPin, HIGH);
                  digitalWrite(lPin, HIGH);
            }
            if(incomingByte==6 )      {
                  Serial.print("DOWN_RIGHT");
                  digitalWrite(dPin, HIGH);
                  digitalWrite(rPin, HIGH);
            }
            if(incomingByte==0){
                digitalWrite(uPin, LOW);
                digitalWrite(dPin, LOW);
```

```
                digitalWrite(rPin, LOW);
                digitalWrite(lPin, LOW);
            }
            if(incomingByte==0x40){//!
              isDemoing=false;
              break;
            }
            delay(200);
        }// end if
    }// end while
     Serial.println("BACK");
    break;
     case 3: // ir_send tv control
     while(true){
          IRsend irsend;
           buffer[0]='\0';
           i=0;
           j=0;
            while(    (read=Serial.read()) !=-1 && read !='!' )
            {
              buffer[i]=read;
              i++;
                Serial.print("read = ");
                Serial.println(read);
            }
            if(chkequ(buffer,"tv,on")||chkequ(buffer,"tv,off"))//開關電視
            {
                  irsend.sendNEC(0x6f900ff, 32);
                  Serial.println("tv,on,");
            }
            else if(chkequ(buffer,"tv,s"))//切換訊號來源
            {
                  irsend.sendNEC(0x6f908f7, 32);
                  Serial.println("tv,signal,");
            }
            else if(chkequ(buffer,"tv,up"))//音量大
            {
                  irsend.sendNEC(0x6f99867, 32);
```

```
        Serial.println("tv,up,");
    }
    else if(chkequ(buffer,"tv,do"))    //音量小
    {
        irsend.sendNEC(0x6f9a857, 32);
        Serial.println("tv,down,");
    }
    else if(chkequ(buffer,"tv,pre"))    //頻道上
    {
        irsend.sendNEC(0x6f93ac5, 32);
        Serial.println("tv,pre,");
    }
    else if(chkequ(buffer,"tv,next"))    //頻道下
    {
        irsend.sendNEC(0x6f928d7, 32);
        Serial.println("tv,next,");
    }
    else if(chkequ(buffer,"tv,1"))    //1
    {
        irsend.sendNEC(0x6f920df, 32);
        Serial.println("tv,1");
    }
    else if(chkequ(buffer,"tv,2"))    //2
    {
        irsend.sendNEC(0x6F9A05F, 32);
        Serial.println("tv,2");
    }
    else if(chkequ(buffer,"tv,3"))    //3
    {
        irsend.sendNEC(0x6F9FA05, 32);
        Serial.println("tv,3");
    }
        else if(chkequ(buffer,"tv,4"))    //4
    {
        irsend.sendNEC(0x6F9E01F, 32);
        Serial.println("tv,4");
    }
        else if(chkequ(buffer,"tv,5"))    //5
```

```
{
        irsend.sendNEC(0x6F910EF, 32);
        Serial.println("tv,5");
}
          else if(chkequ(buffer,"tv,6"))    //6
{
        irsend.sendNEC(0x6F9906F, 32);
        Serial.println("tv,6");
}
          else if(chkequ(buffer,"tv,7"))    //7
{
        irsend.sendNEC(0x6F950AF, 32);
        Serial.println("tv,7");
}
          else if(chkequ(buffer,"tv,8"))    //8
{
        irsend.sendNEC(0x6F9D02F, 32);
        Serial.println("tv,8");
}
          else if(chkequ(buffer,"tv,9"))    //9
{
        irsend.sendNEC(0x6F930CF, 32);
        Serial.println("tv,9");
}
          else if(chkequ(buffer,"tv,0"))    //0
{
        irsend.sendNEC(0x6F9C03F, 32);
        Serial.println("tv,0");
}
         else if(chkequ(buffer,"BACK"))    //BACK
{
        isDemoing=false;
        Serial.println("BACK");
        incomingByte=0;
      break;
}
buffer[0]='\0';
delay(500);
```

```
            }// end while
            break;
              default:
            isDemoing=false;
        } // end swirch
    }// end else
//    Serial.flush();
} // end loop
boolean chkequ(char b[],char a[])        //read  字串比對
{
   int k=0;
   while(a[k]!=0)
   {
      if(a[k]!=b[k])
         return false;
      k++;
   }
   return true;
}
```