



# TAROKO 使用手冊

## RevA

1. 介紹 TAROKO 的硬體架構
2. 簡易的 TinyOS 及驅動程式安裝
3. 說明如何使用 TinyOS 寫入程式到 TAROKO 及收集感測資料

**CHNDS**

**8/15/2006**

# Index

1. TAROKO 平台	-----	3
a. 簡介	-----	3
b. 技術規格	-----	4
c. 各部位圖解	-----	4
d. 主要硬體元件	-----	6
e. 電源	-----	7
f. 感測器	-----	7
2. 安裝 TinyOS 1.x	-----	9
3. 安裝 USB 驅動程式	-----	15
4. 使用 TinyOS	-----	22
5. 收集資料	-----	25

# 1. TAROKO 平台

## a、簡介

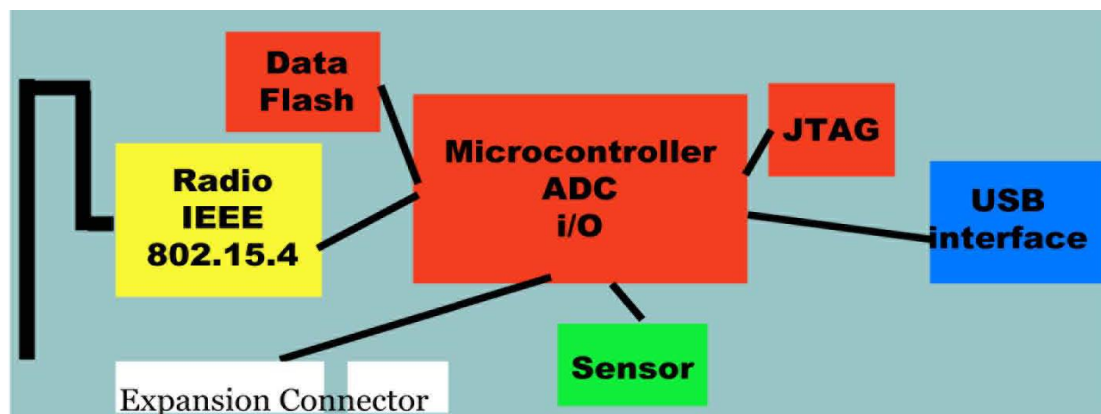
TAROKO 是一個低耗電、可程式、無線傳輸的感測網路硬體平台。TAROKO 平台所使用微處理器為德州儀器(TI)的 MSP430，此微處理器的最大優點為其低耗電量，因此 TAROKO 亦較一般其他的無線感測網路平台更省電。TAROKO 使用 IEEE 802.15.4 做為無線傳輸的通訊協定標準，相容於 Zigbee。室外最長的傳輸距離可達 100 米，而室內直線傳輸距離可達 50 米。TAROKO 具有 USB 介面，可直接利用電腦的 USB 做為供電、燒錄程式及收集資料之用途。除此之外，TAROKO 還具有 ADC、DAC、UART、SPI、I2C 等週邊介面，提供強大的擴充性。



TAROKO 是以美國加州柏克萊大學(UC Berkeley)發展的 Telosb 為設計藍圖，因此完全相容於 TinyOS (<http://www.tinyos.net>)。TinyOS 是一套為無線感測網路所設計的開放原始碼的作業系統，同樣也是由美國加州柏克萊大學發展。TinyOS 的程式碼，只要編譯成 Telosb 所使用的執行檔，便可供 TAROKO 使用。

## 主要特點

- TI MSP430 微處理器，具有 10KB RAM、48KB Flash
- 250kbps 2.4GHz IEEE 802.15.4 Radio
- 內建 PCB 天線
- USB 介面
- 超低耗電量
- 週邊包括 ADC、DAC、UART、SPI、I2C 等
- 溫度感測器、濕度感測器及光感測器（選購）

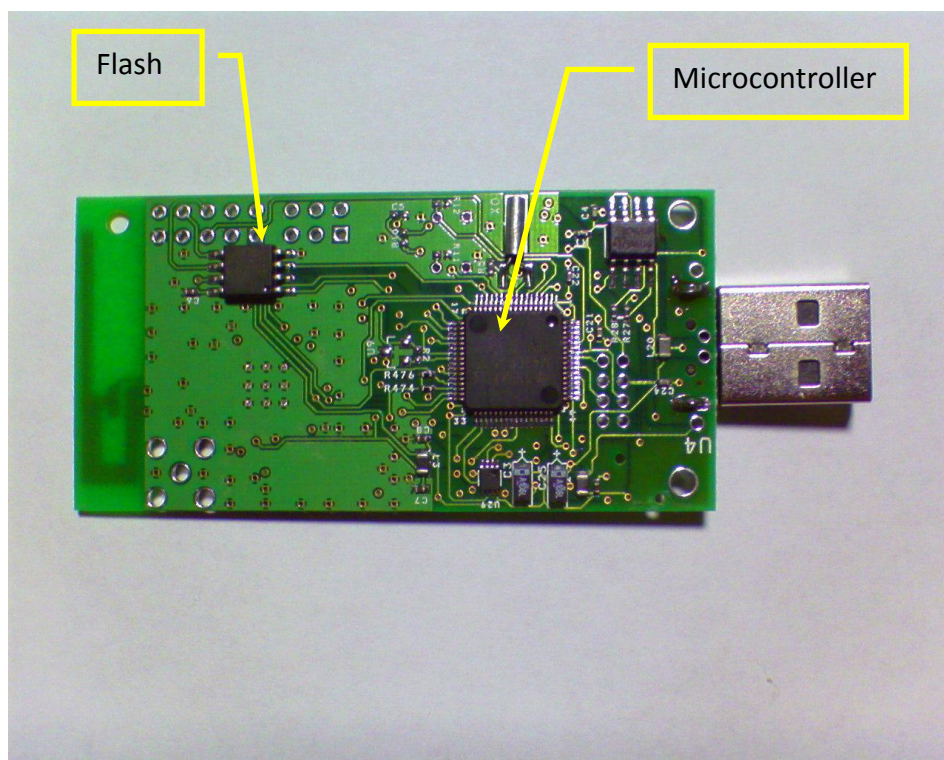
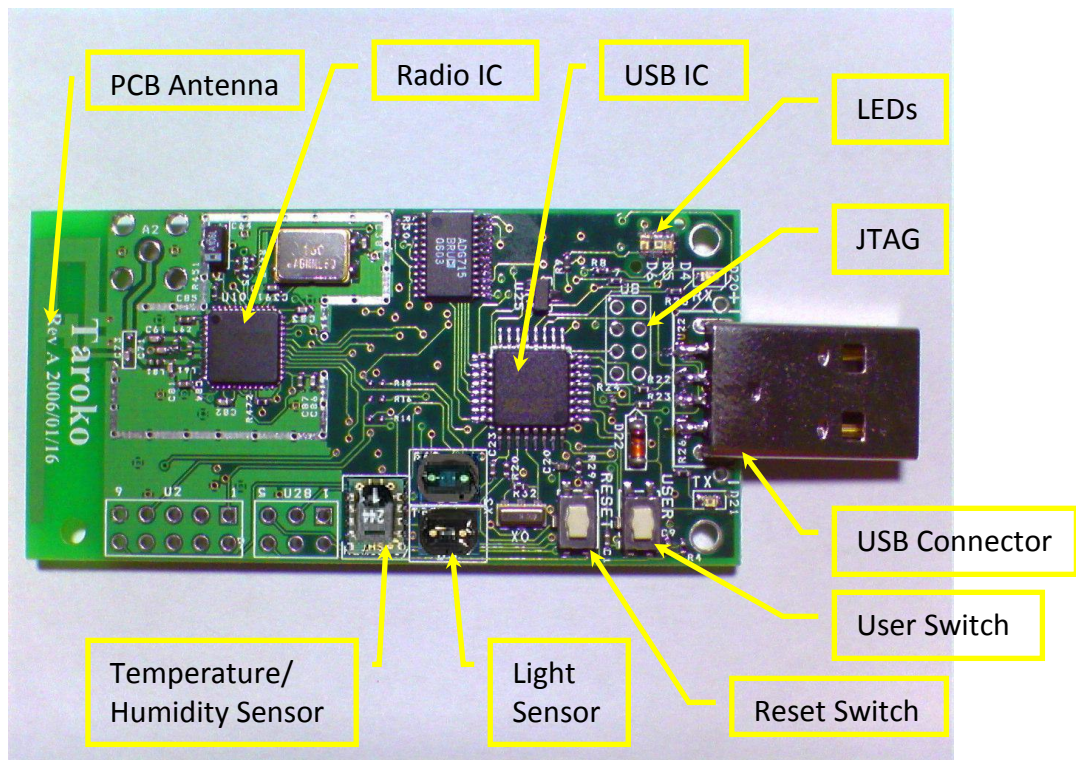


## b、技術規格

Microcontroller	
Architecture	16-bit RISC
Bus Speed	8 MHz
Program Flash	48K Bytes
RAM	10K Bytes
External Storage Flash	1024K Bytes
Serial communication	UART, I2C, SPI, Digital I/O
Analog to Digital Converter	8-ch 12-bit (only 6 channel wired to expansion connector)
Digital to Analog Converter	2-ch 12-bit
Current consumption	1.8 mA (Active mode, without Radio)
	5.1 $\mu$ A (Sleep mode)
Radio	
Frequency	2400MHz to 2483MHz ISM Band
Data Rate	250kbps
RF Power	-24 dBm to 0 dBm
Range	75m to 100m Outdoor
	20m to 30m Indoor
Current Consumption	19.7mA (Receive mode)
	17.4mA (Transmit mode, 0 dBm)
	426 $\mu$ A (Idle mode)
	1 $\mu$ A (Sleep mode)
Electromechanical	
Power supply	3.3 V (via power connector)
	5 V (via USB connector)
Size	65 x 31 x 6 (mm)
Weight	23 (grams)
Sensors (Optional)	
Visible Light Sensor Range	320nm to 730nm
Visible Light Sensor Range	320nm to 1100nm
Humidity Sensor	Range: 0 to 100% RH
	Accuracy: $\pm$ 3.5% RH
Temperature Sensor	Range: -40 °C to 123.8 °C
	Accuracy: $\pm$ 0.5 °C



# c、 各部位圖解



## d、 主要硬體元件

### 微處理器

TAROKO 平台使用德州儀器(TI)的 MSP430F1611 做為運算核心。MSP430F1611 是一個以低耗電量為設計目標的 16 位元 RISC 架構微處理器(MCU)，具有超低耗電量的優點。MSP430F1611 具有 10KB 的 RAM、48KB 的 Flash，最高的工作時脈可達 8MHz。除此之外，MSP430F1611 還具有高效能的硬體週邊如 ADC、DAC、USART、I2C、Timers、DMA 等，提供強大的擴充功能。關於 MSP430F1611 的詳細資料，可參考 TI 的網站 <http://www.ti.com/>。

#### 相關文件

MSP430F1611 Data Sheet

<http://www-s.ti.com/sc/ds/msp430f1611.pdf>

MSP430xlxx User Guide

<http://focus.ti.com/lit/ug/slau049f/slau049f.pdf>

### 無線傳輸

TAROKO 使用 Chipcon 的 CC2420 晶片做為無線射頻收發器。CC2420 是相容於 IEEE 802.15.4 協定標準的射頻收發器，運作於 2.4GHz 的 ISM 頻帶。**CC2420 具有低耗電量的優點，並有硬體 MAC 加密與驗證(AES-128)的功能。**MSP430 微處理器經由 SPI 介面控制 **CC2420 的收發動作。**CC2420 的資料可於 Chipcon 的網站上獲得 <http://www.chipcon.com/>

#### 相關文件

CC2420 Data Sheet

[http://www.chipcon.com/files/CC2420\\_Data\\_Sheet\\_1\\_4.pdf](http://www.chipcon.com/files/CC2420_Data_Sheet_1_4.pdf)

### USB 介面

TAROKO 具有 USB 介面，可連接到 PC，直接由 PC 寫入程式或者收集資料，不需經過其他的 Gateway。此 USB 的功能是由 FTDI Chip 公司的 FT232 晶片，將微處理器收發的資料，轉換為 USB 的資料格式傳送到 PC。使用 USB 的功能必需先安裝驅動程式，驅動程式的安裝方法於後面章節說明。

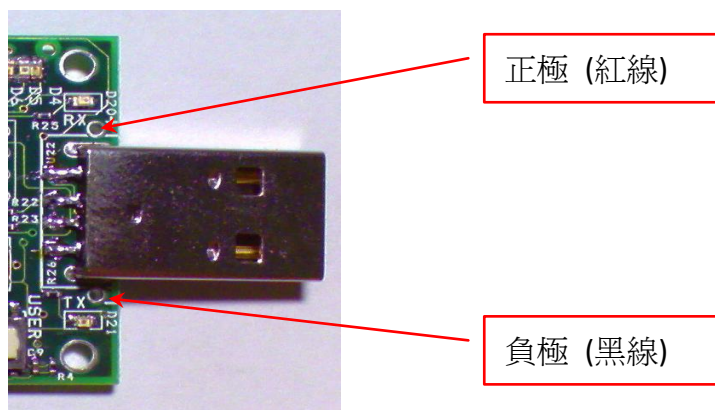
#### 相關文件

FT232BM Data Sheet

<http://www.ftdichip.com/Documents/DataSheets/ds232b18.pdf>

## e、 電源

TAROKO 的運作電壓範圍為 2.1V~3.6V，另外當 Flash 寫入時，至少需要 2.7V 的電壓。電源供應有兩個選項：一是由 USB 連結供電；二是外接兩顆 AA(或 AAA) 乾電池。因使用者的需求不盡相同，TAROKO 出貨時並未連接電池盒。如有需要，使用者可自行焊接電池盒，焊接方式如下圖。另外需注意的是，一般 Ni-MH 充電電池的額定電壓為 1.2V(與乾電池的 1.5V 不同)，因此兩顆 Ni-MH 充電電池的電壓為 2.4V。此電壓足夠讓系統運作，但無法提供足夠的電壓供 Flash 寫入。



## f、 感測器 (選購)

### 溫、濕度感測器

TAROKO 所使用的溫、濕度感測元件 SHT11 是由瑞士 Sensirion 公司所製造。  
SENSIRION 為全球第一家以 CMOS 技術開發出溫濕度雙功能感測元件的公司。

SHT11 將量測的類比訊號經由 14-bit 的類比數位轉換器(ADC)轉換為數位訊號傳送到微處理器上。以下為 SHT11 的主要規格：

#### SHT11 技術規格

濕度	量測範圍	0% ~ 100% RH
	精確度	+ - 3%RH (20%~80%RH)
	反應時間	≤4 秒
	重現性	+ - 0.1%RH
	解析度	0.03%RH
	操作溫度	-40°C ~ 120°C



溫度	量測範圍	-40°C ~ 120°C
	精確度	+/- 0.4°C @25°C +/- 0.9°C (0°C~40°C)
	反應時間	≤20 秒
	重現性	+/- 0.1°C
	解析度	+/- 0.01°C

### 相關文件

SHT11 Data Sheet

<http://www.sensirion.com/images/getFile?id=25>

### 光感測器

TAROKO 上共有兩顆光感測元件，型號分別為 S1087 及 S1087-01。感測元件的製造商為日商 HAMAMATSU。兩者之間的不同點為：S1087 量測可見光範圍，量測波長為 320nm ~ 730nm；S1087-01 可量測從可見光到紅外線的範圍，量測波長為 320nm ~ 1100nm。以下為相關規格：

■ Electrical and optical characteristics (Typ. Ta=25 °C, unless otherwise noted)

Type No.	Spectral response range $\lambda$ (nm)	Peak sensitivity wavelength $\lambda_p$ (nm)	Photo sensitivity S (A/W)			Infrared sensitivity ratio (%)	Short circuit current Isc 100 lx ( $\mu$ A)	Temp. coefficient of Isc (%/°C)	Dark current Id Vr=1 V Max. (pA)	Temp. coefficient of Id Tcid (times/°C)	Rise time tr Vr=0 V Rl=1 k $\Omega$ ( $\mu$ s)	Terminal capacitance Ct Vr=0 V f=10 kHz (pF)	Shunt resistance Rsh Vr=10 mV	
			$\lambda_p$	GaP LED 560 nm	He-Ne laser 633 nm								Min.	Typ.
													(G $\Omega$ )	(G $\Omega$ )
S1087	320 to 730	560	0.3	0.3	0.19	10	0.16	-0.01	10	1.12	0.5	200	10	250
S1087-01	320 to 1100	960	0.58	0.33	0.38	-	1.3	0.1			2.5	700		100
S1133	320 to 730	560	0.3	0.3	0.19	10	0.65	-0.01			0.5	200		50
S1133-01	320 to 1100	960	0.58	0.33	0.38	-	5.6	0.1	20					
S1133-14	320 to 1000	720	0.4		0.37	-	3.4							

### 相關文件

S1087/S1087-01 Data Sheet

[http://sales.hamamatsu.com/assets/pdf/parts\\_S/S1087\\_etc.pdf](http://sales.hamamatsu.com/assets/pdf/parts_S/S1087_etc.pdf)



## 2. 安裝 TinyOS 1.x

以下敘述安裝 TinyOS 1.x 於 Windows XP 作業系統之步驟。預設為作業系統並未安裝任何版本的 Cygwin 或 TinyOS。如已安裝，請選擇升級方式或先移除 Cygwin 及 TinyOS。相關升級或移除方法請參考 TinyOS 網站

<http://www.tinyos.net>。

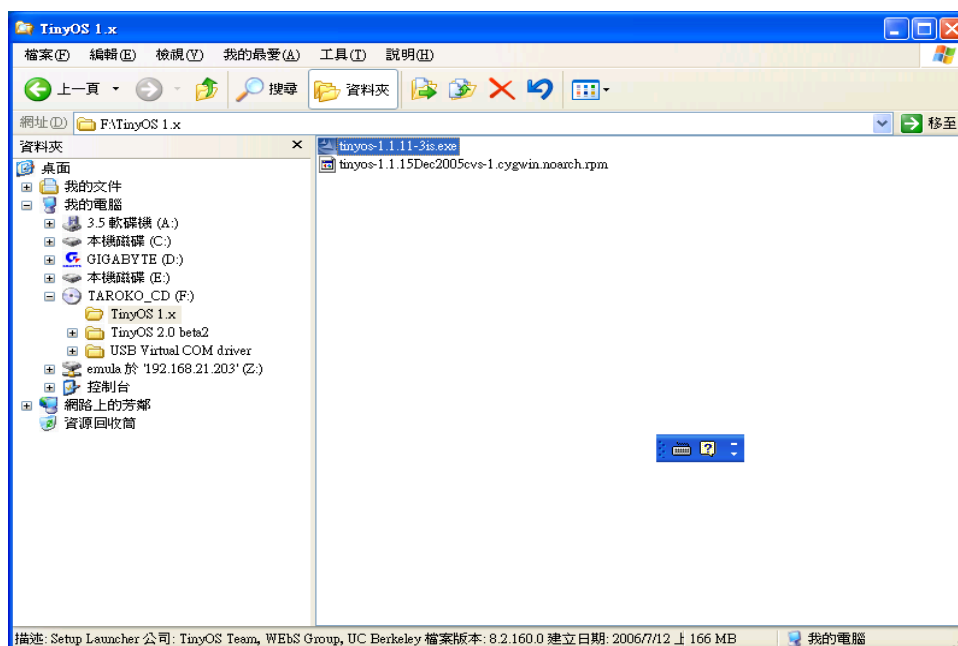
TAROKO CD 內亦包含 TinyOS 2.0 Beta 2 的安裝檔。TinyOS 2.0 目前仍在測試階段，詳細安裝方法請參考此網頁

<http://www.tinyos.net/tinyos-2.x/doc/html/install-tinyos.html>。

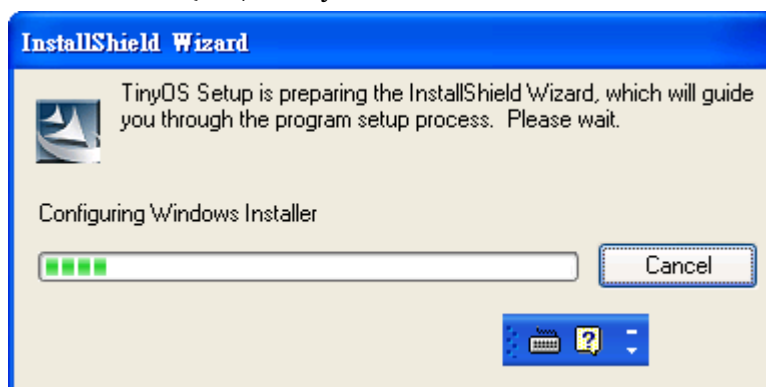
如果安裝上有任何問，可於 TinyOS 網站(<http://www.tinyos.net>)及 TinyOS 的 Mailing List 上取得協助

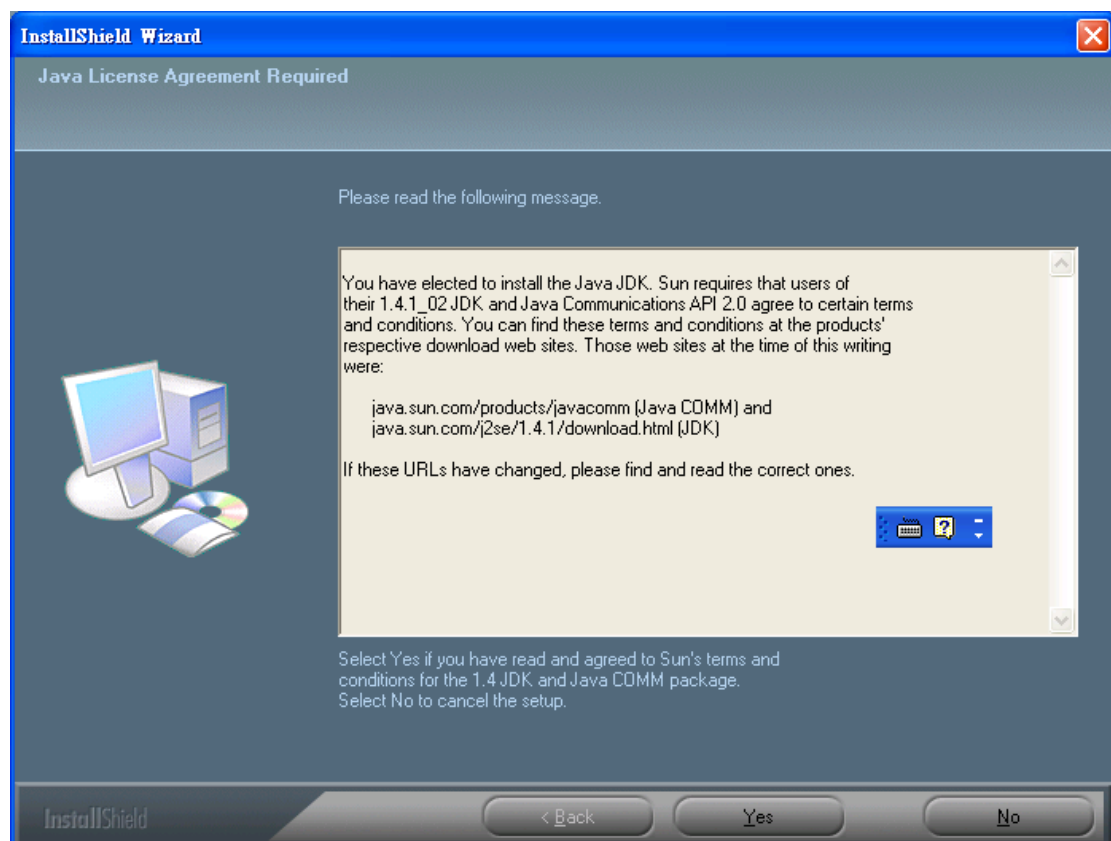
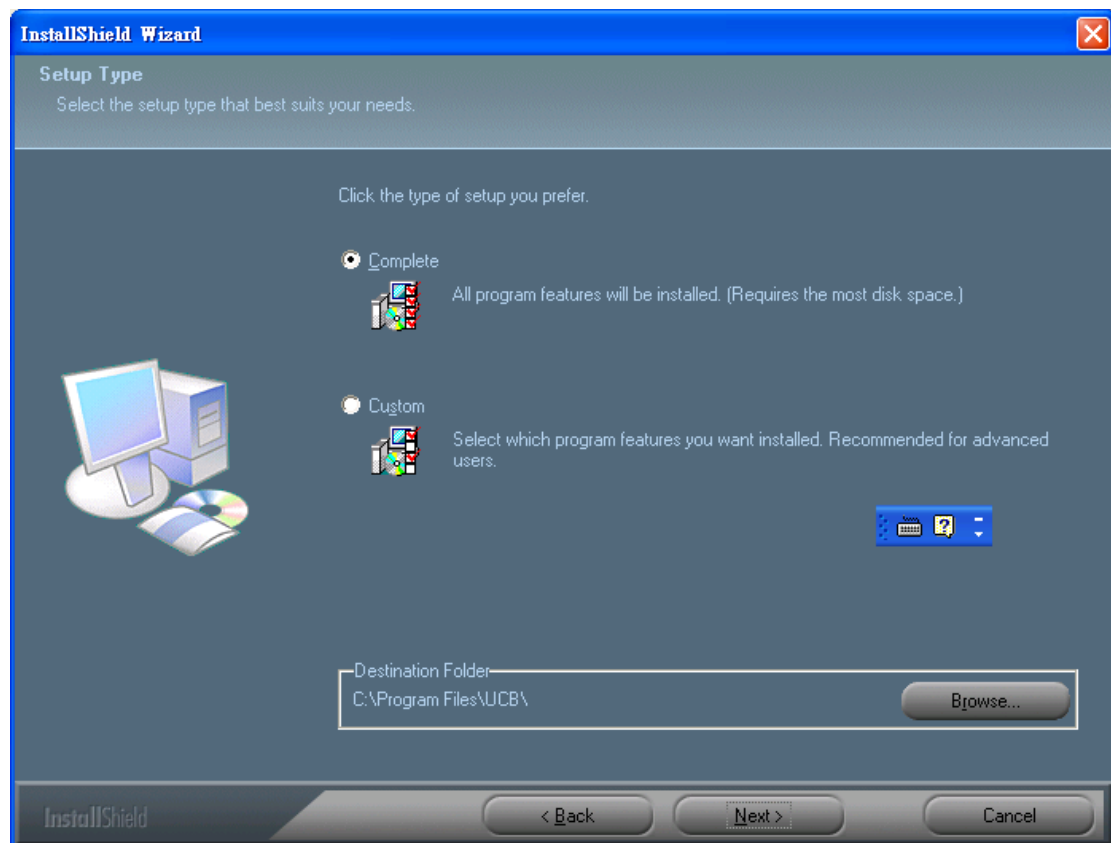
(<http://www.tinyos.net/scoop/special/support#mailing-lists>)。

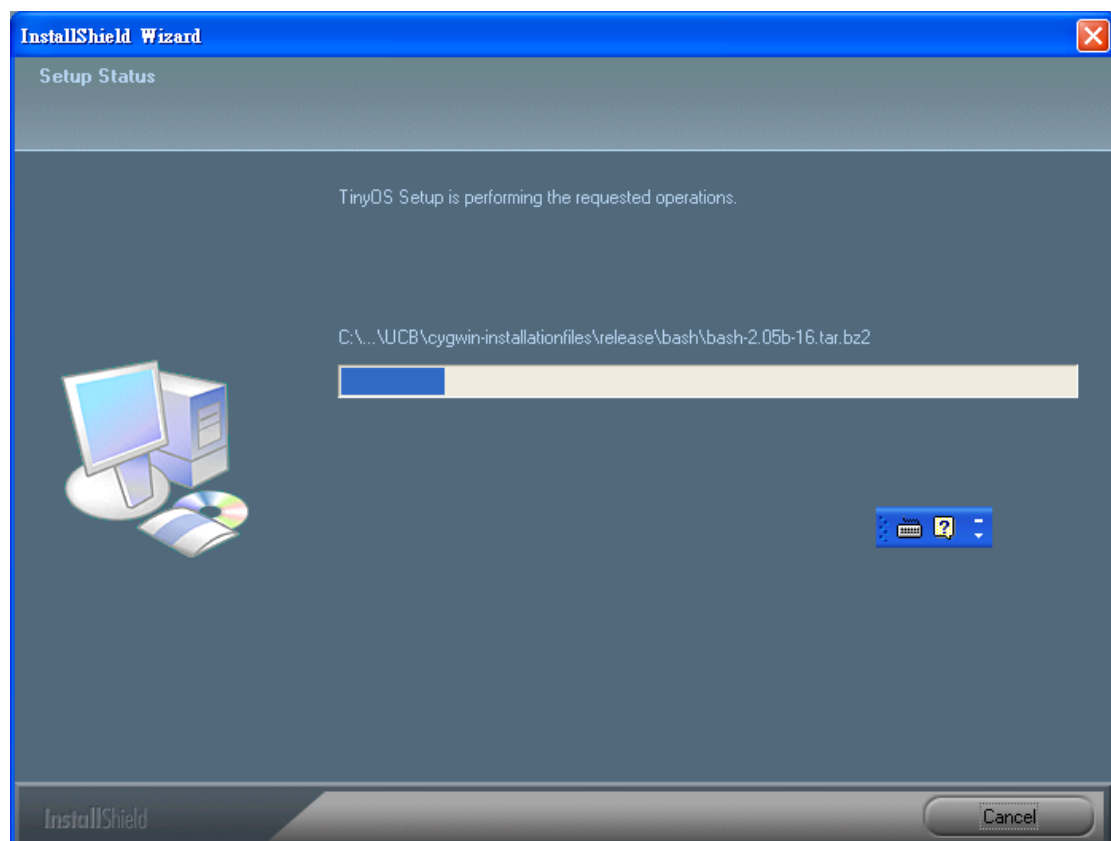
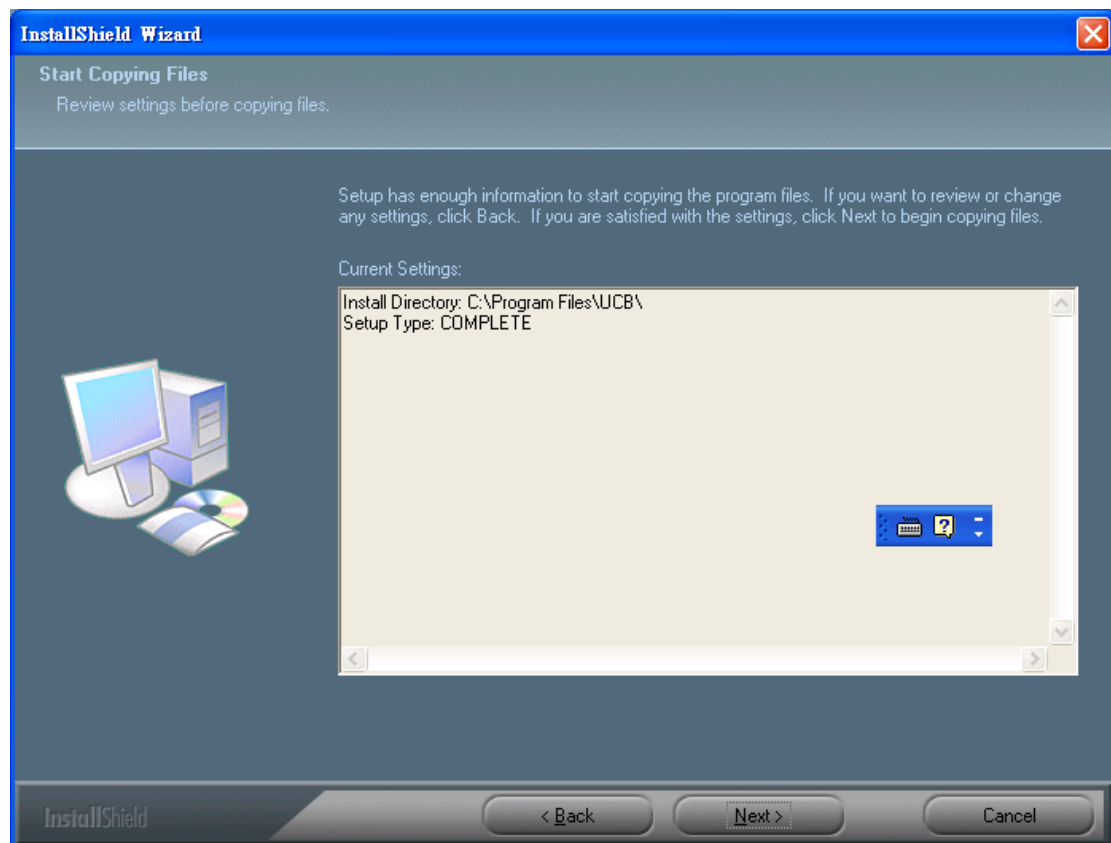
a、放入 TAROKO CD，打開 TinyOS 1.x 資料夾，開啓 tinyos-1.1.11-3is.exe。

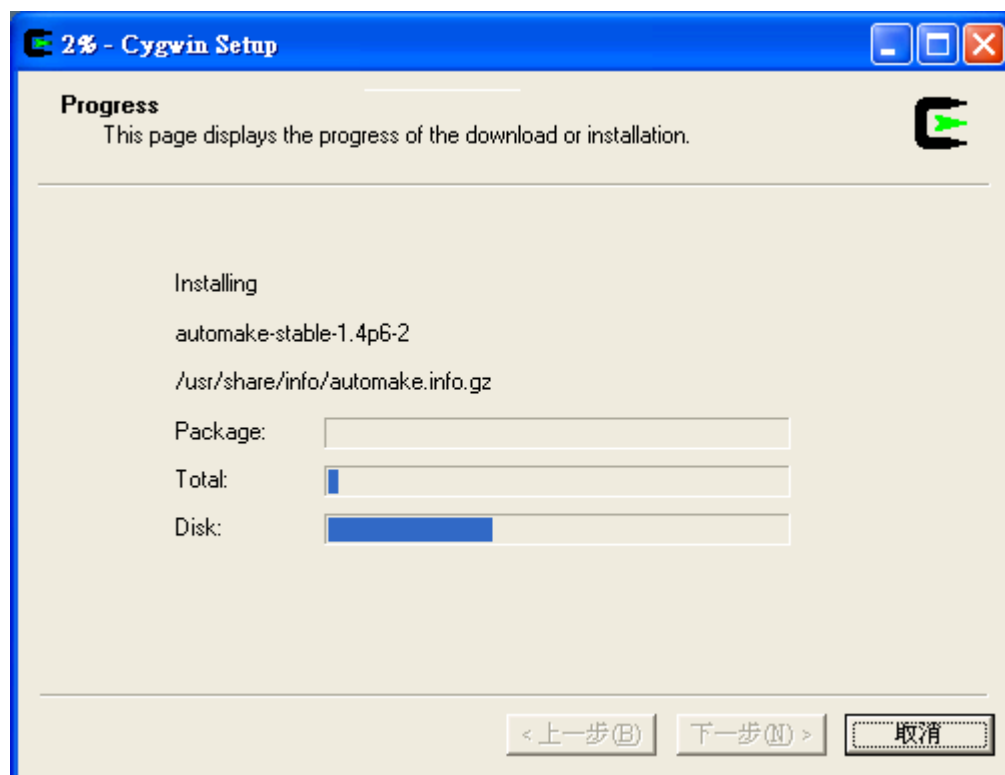
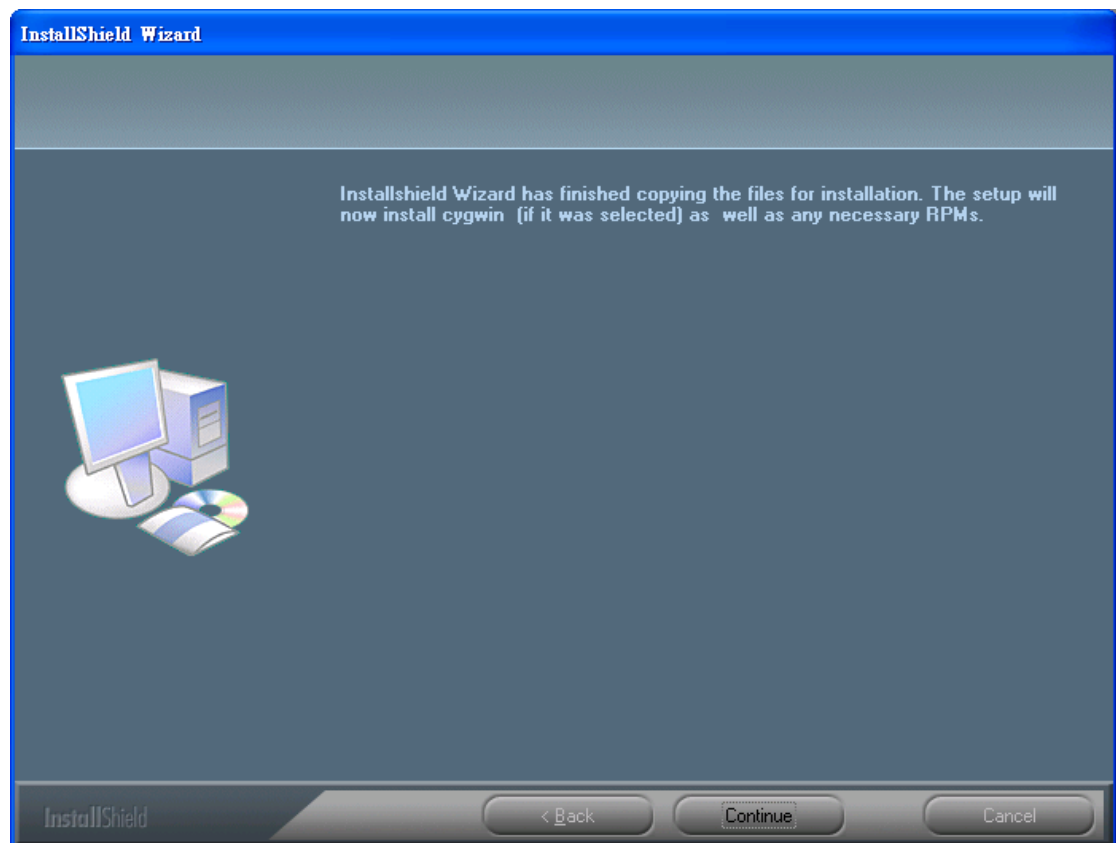


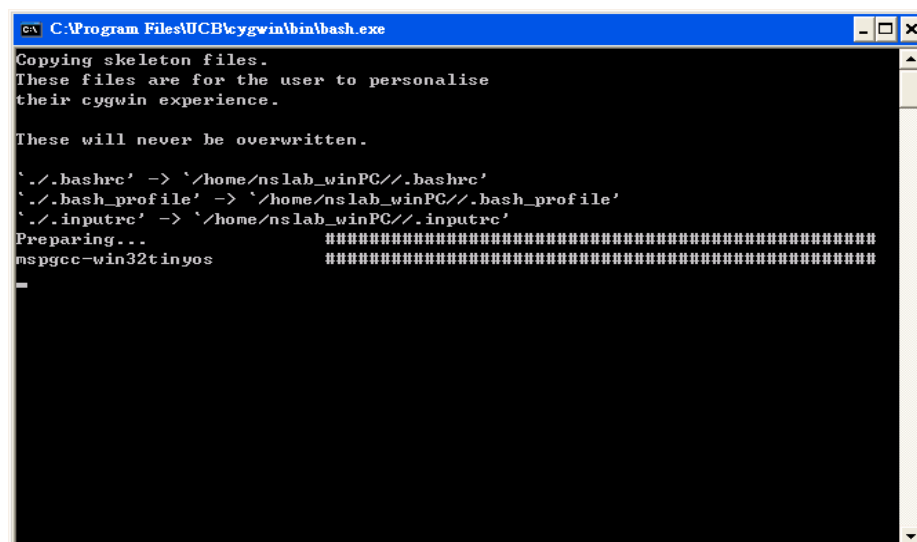
b、依照指示完成安裝 TinyOS 1.1.11。







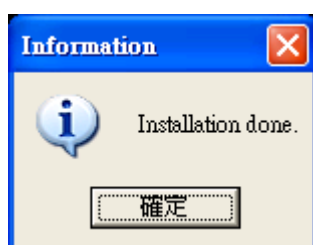




```
C:\Program Files\UCB\cygwin\bin\bash.exe
Copying skeleton files.
These files are for the user to personalise
their cygwin experience.

These will never be overwritten.

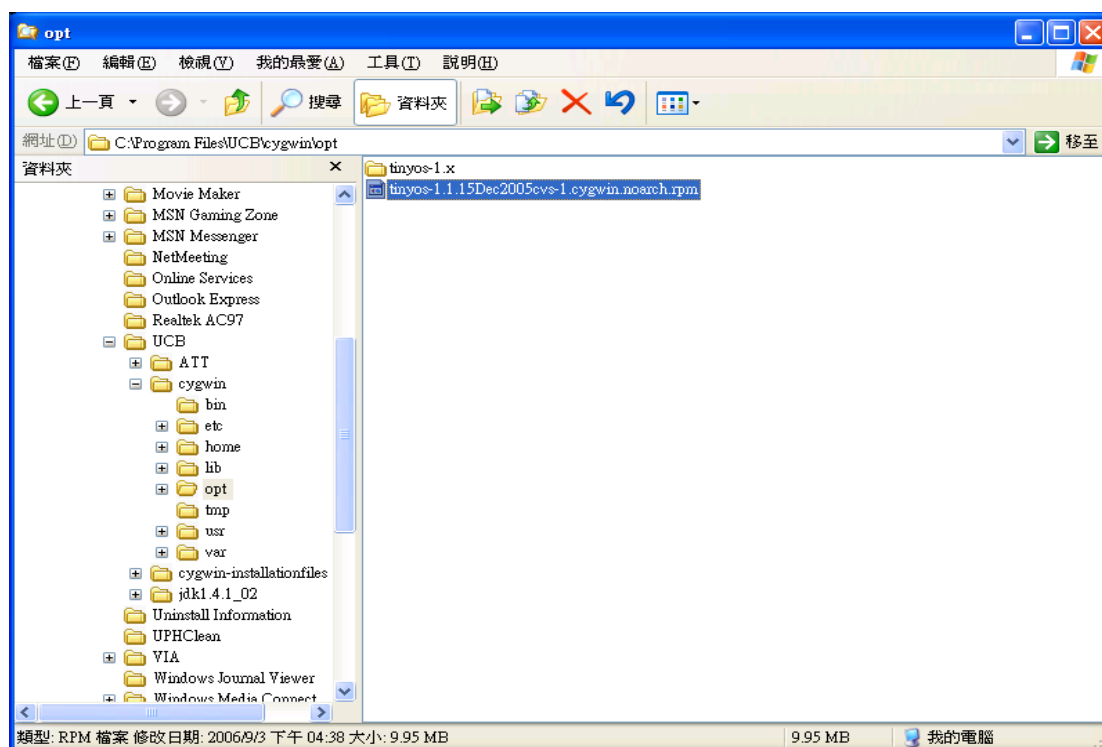
'./bashrc' -> '/home/ns1ab_winPC//.bashrc'
'./bash_profile' -> '/home/ns1ab_winPC//.bash_profile'
'./inputrc' -> '/home/ns1ab_winPC//.inputrc'
Preparing...
mspgcc-win32tinyos
```



c、接下來將 TinyOS 升級到 1.1.15 版本。

將 tinyos-1.1.15Dec2005cvs-1.cygwin.noarch.rpm 複製到

C:\Program Files\UCB\cygwin\opt 之下。

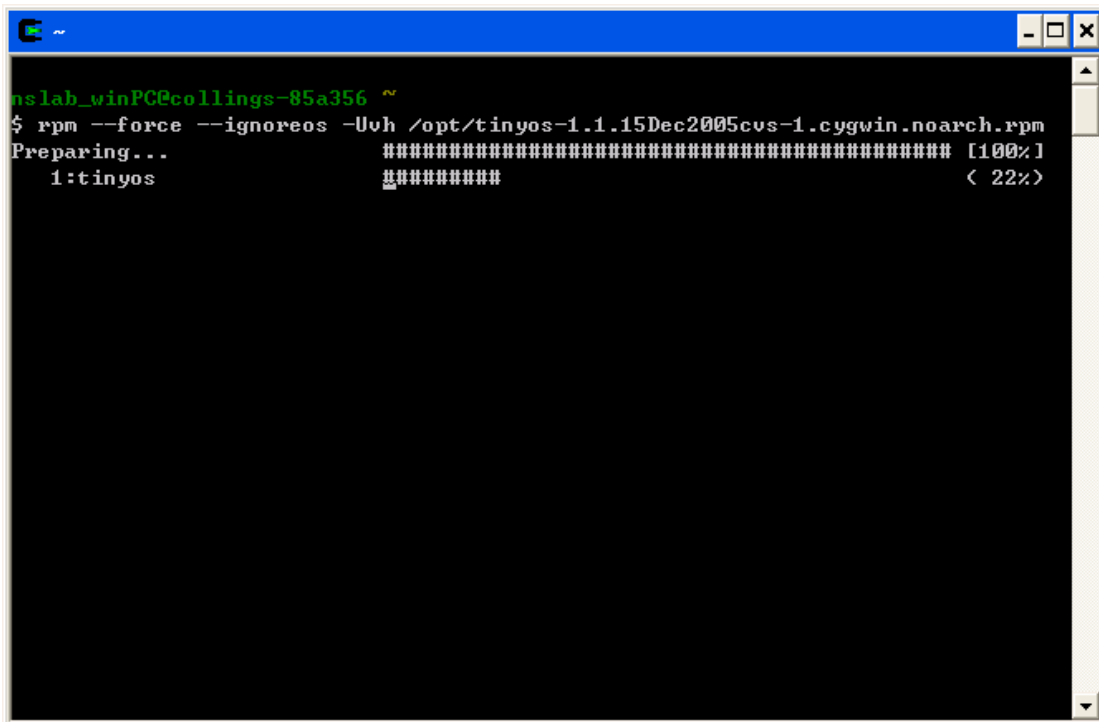




d、打開 Cygwin，執行以下指令。

```
$ rpm --force --ignoreos -Uvh
```

```
/opt/tinyos-1.1.15Dec2005cvs-1.cygwin.noarch.rpm
```

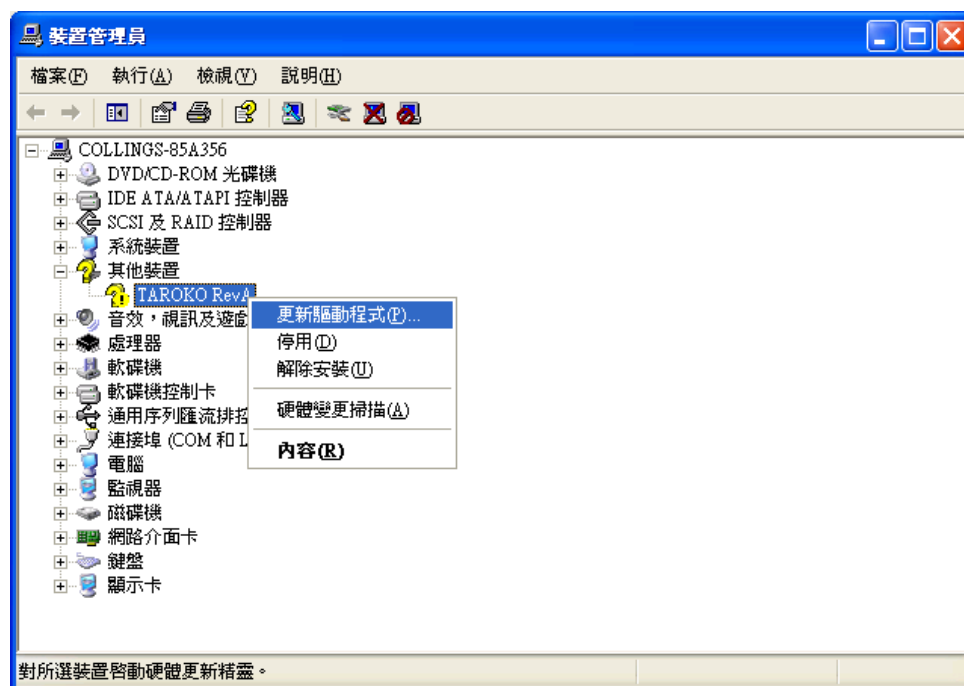


```
ns1ab_winPC@collings-85a356 ~  
$ rpm --force --ignoreos -Uvh /opt/tinyos-1.1.15Dec2005cvs-1.cygwin.noarch.rpm  
Preparing... ##### [100%]  
1:tinyos ##### < 22%
```

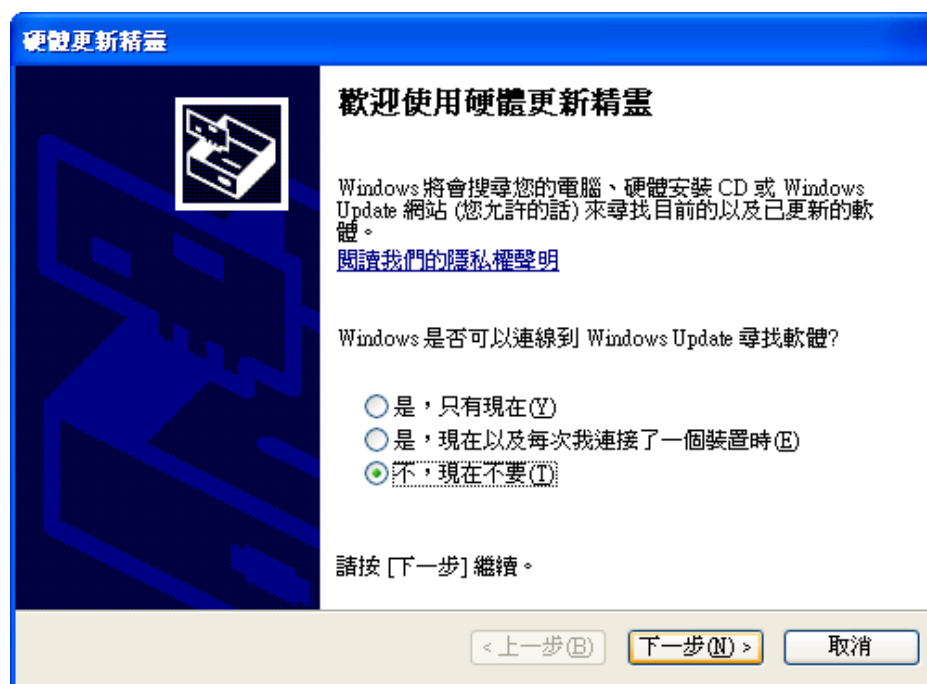
### 3. 安裝 USB 驅動程式

接下來以 Windows XP 作業系統為例，安裝 USB 的驅動程式

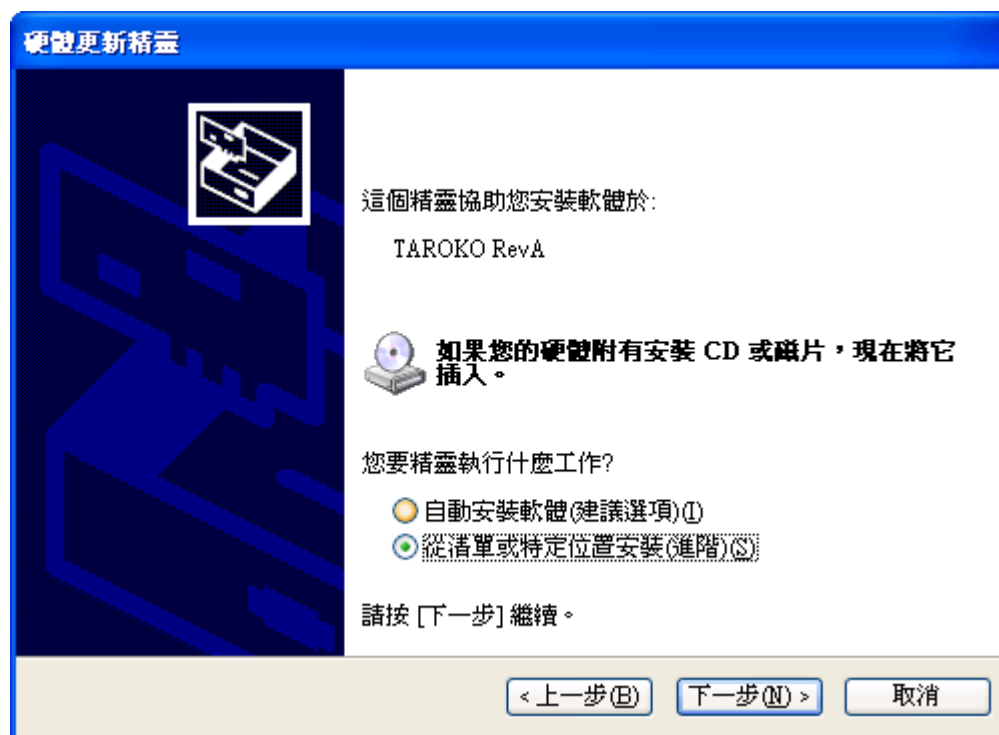
- a、將 TAROKO 插入電腦的 USB port，打開裝置管理員，會出現 TAROKO RevA 的不明硬體。按滑鼠右鍵選"更新驅動程式"。



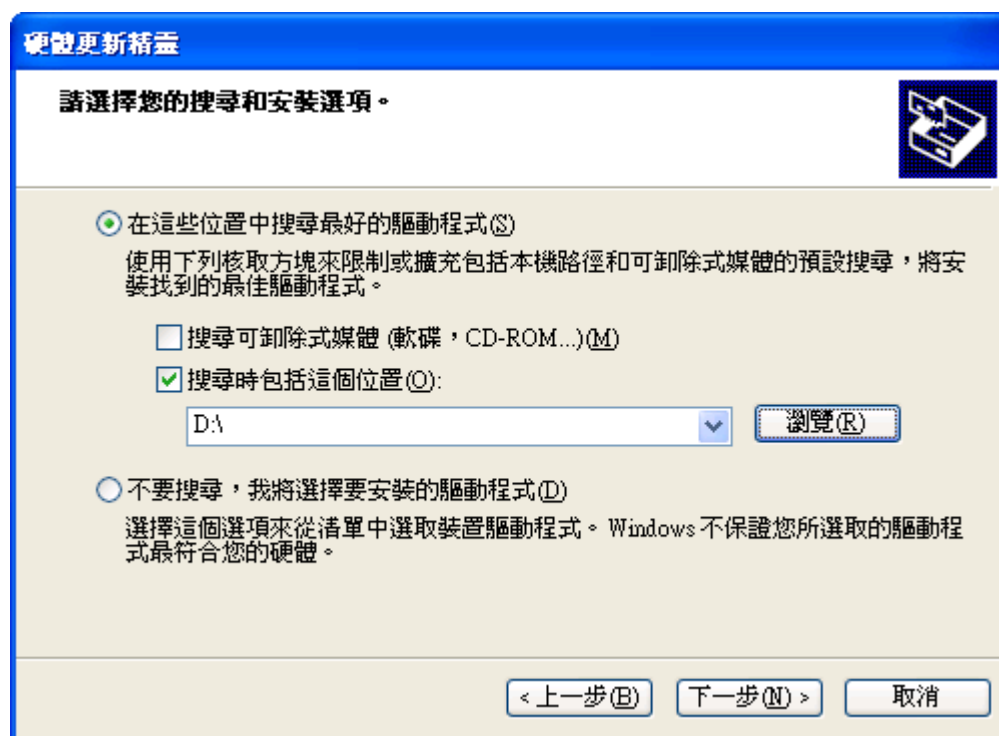
- b、選"不，現在不要"，按下一步。



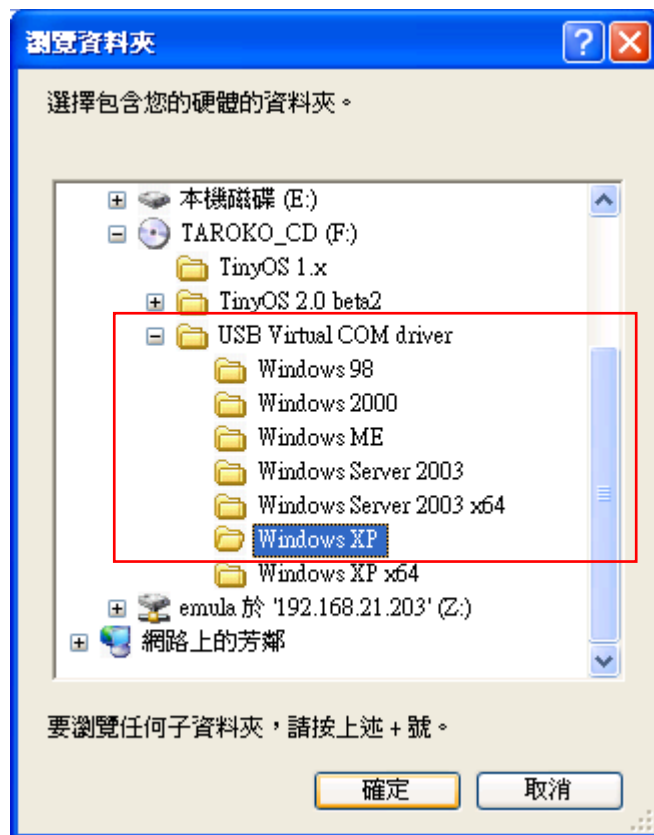
c、選"從清單或特定裝置安裝(進階)"，按下一步。



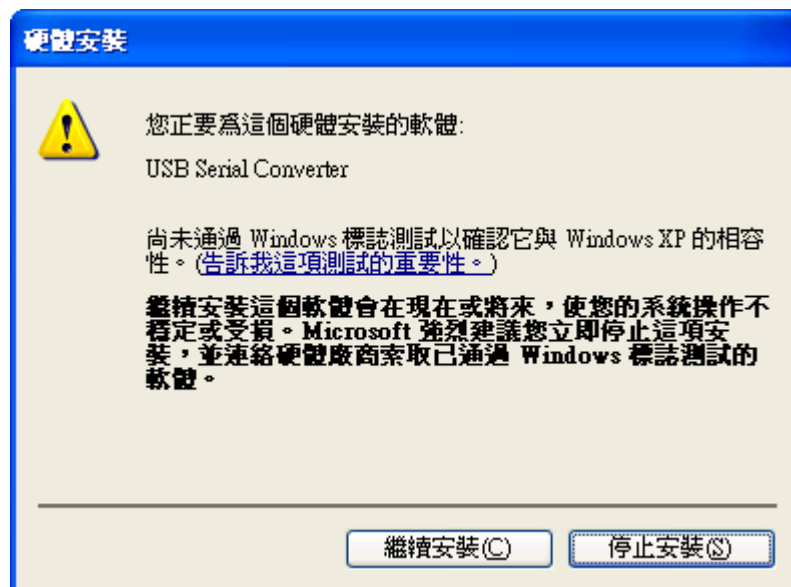
d、勾選"搜尋時包括這個位置"，按"瀏覽"。



- e、選擇 TAROKO CD 中"USB Virtual COM driver"之下的"Windows XP"資料夾，按確定，再按下一步。

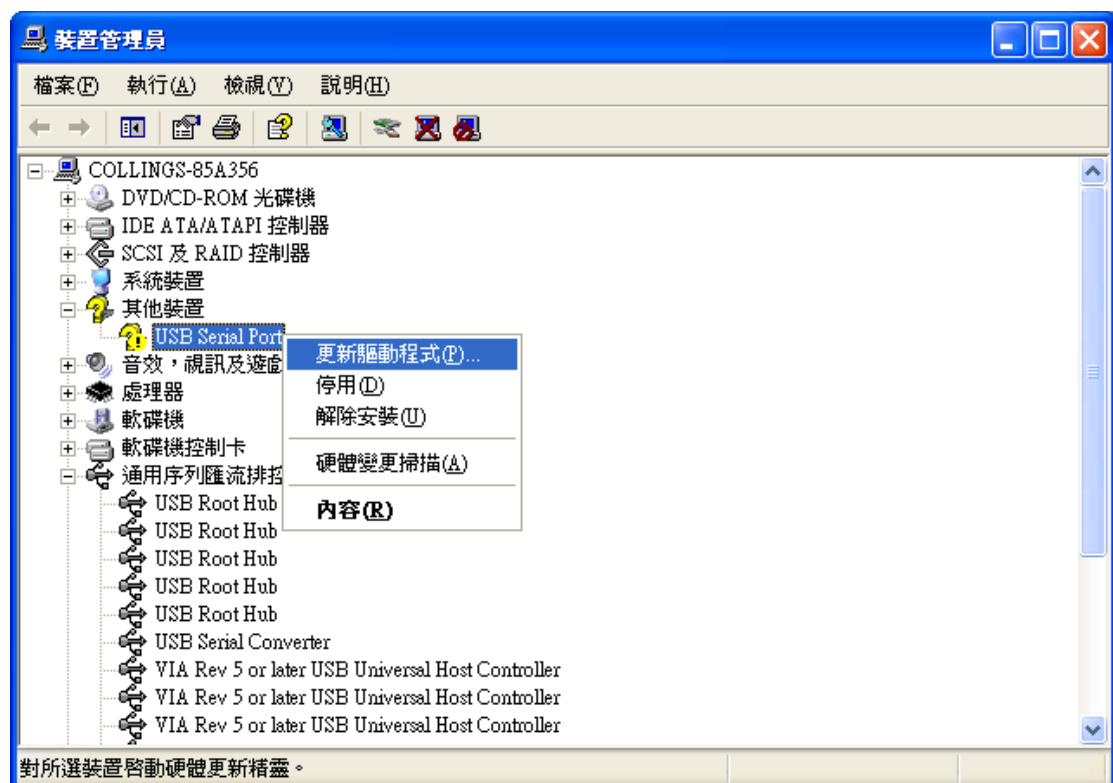


- f、跳出以下畫面時選"繼續安裝"，安裝完成之後按"完成"。

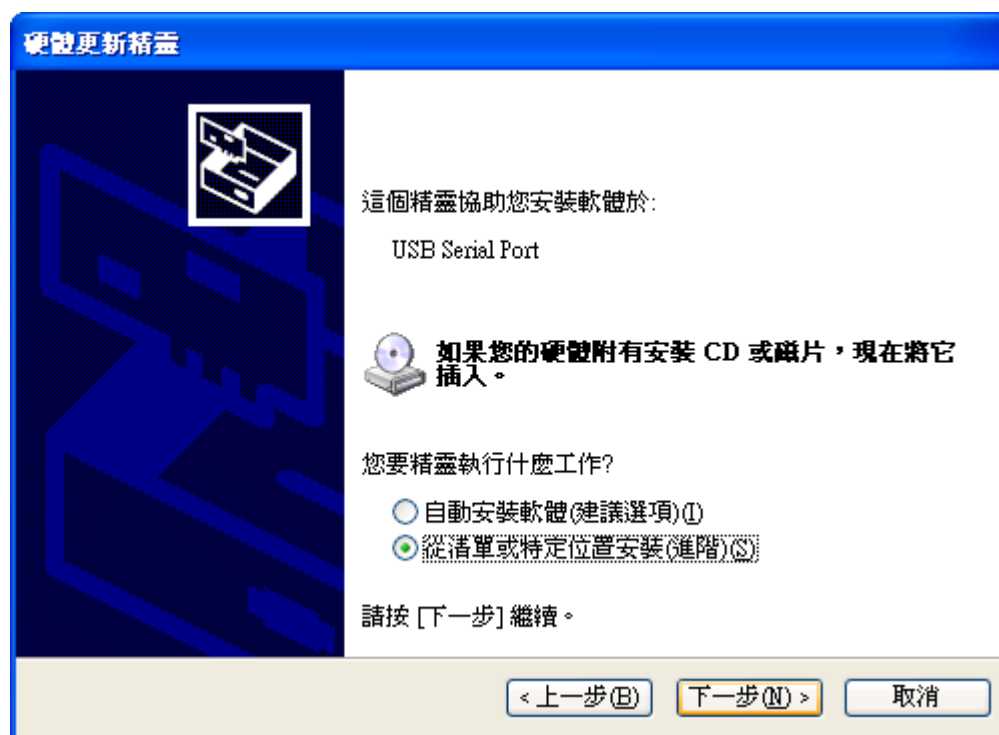
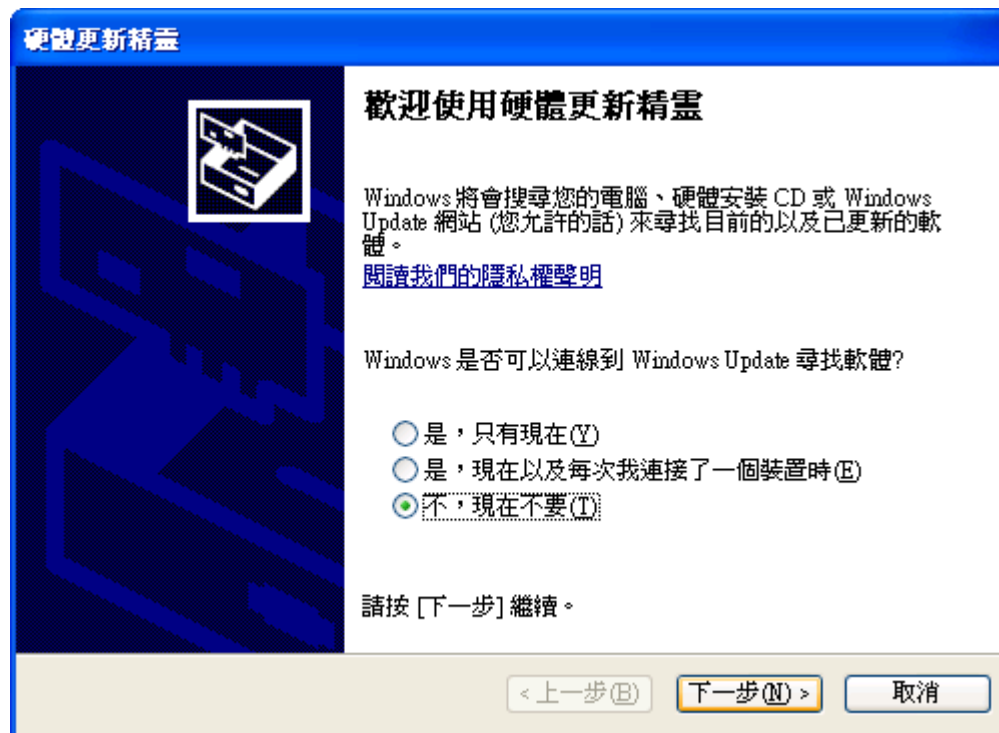


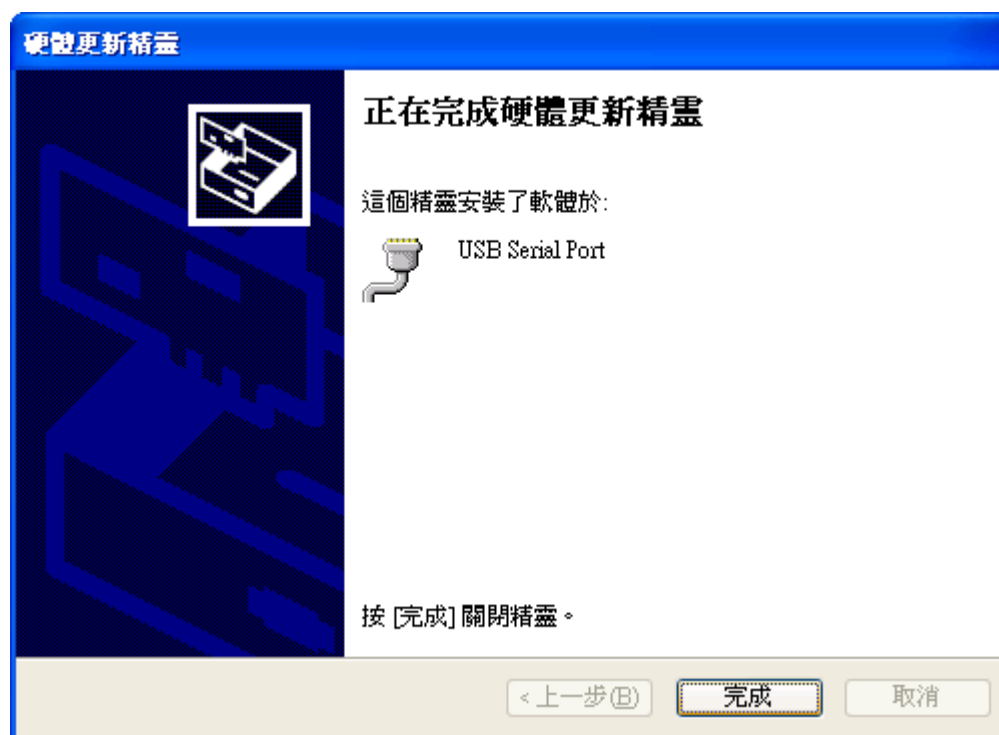
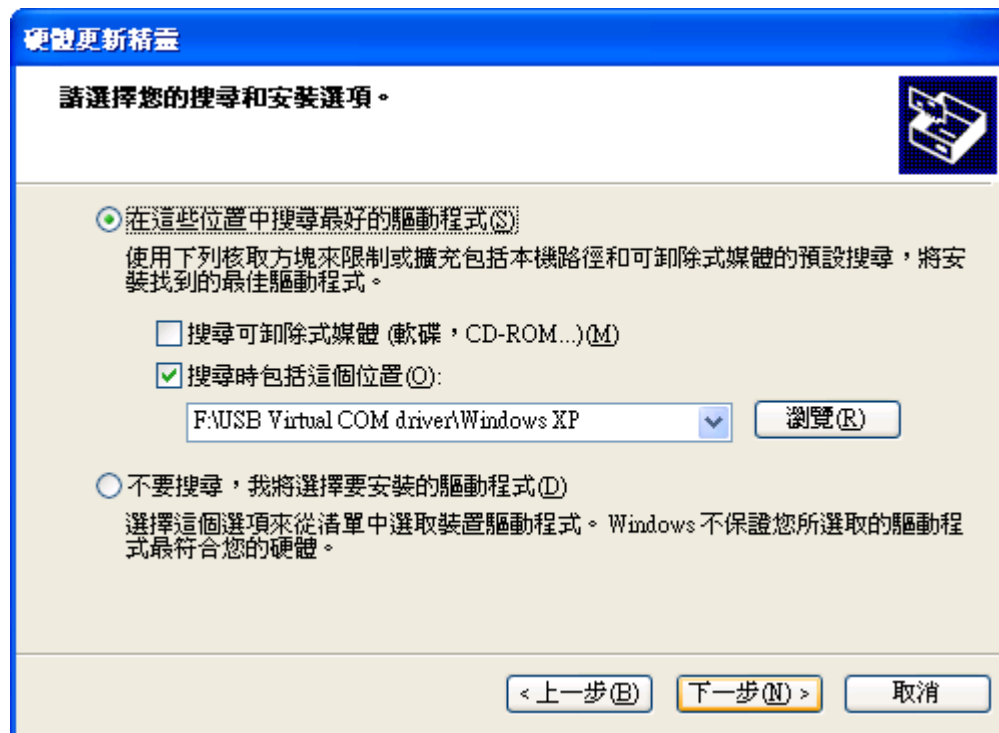


- g、此時裝置管理員會出現"USB Serial Port"的不明硬體，重覆以上相類似的步驟。

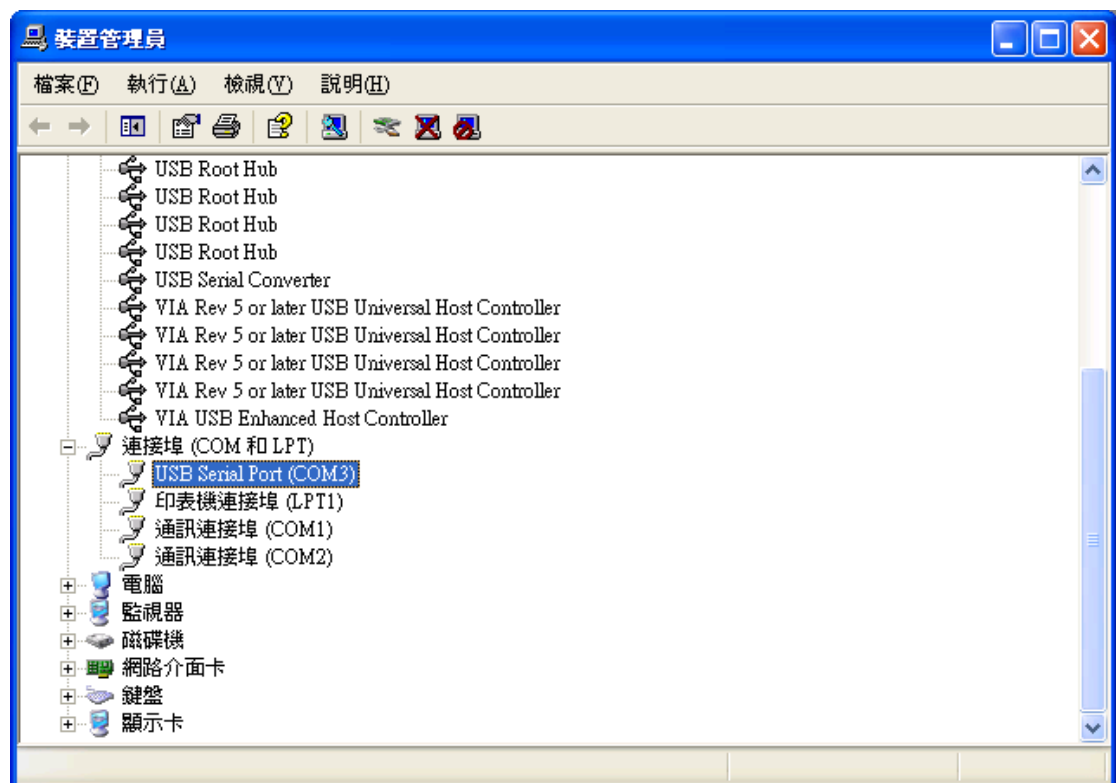








h、安裝完成後裝置管理員會出現"USB Serial Port (COMxx)"的硬體。



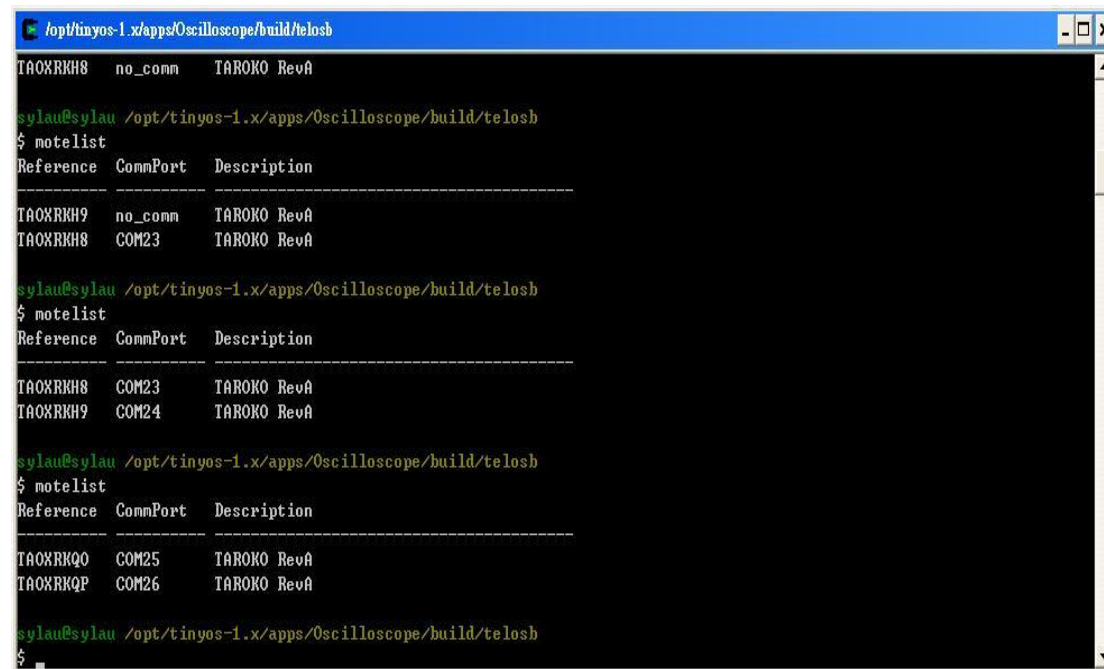
## 4. 使用 TinyOS

這裡將以 TinyOS 內建的範例程式：CntToLedsAndRfm 及 RfmToLeds，示範如何使用 TinyOS 編譯程式並安裝到 TAROKO 上。此程式分別位於 /opt/tinyos-1.x/apps/CntToLedsAndRfm 及 /opt/tinyos-1.x/apps/RfmToLeds 資料夾內。CntToLedsAndRfm 這支程式是在系統內執行一個計數器(Counter)，以 4Hz 的頻率遞增。系統根據現在的數值閃動不同的 LED，將數值通過無線傳送出去。RfmToLeds 則是接收前面程式所傳送出來的數值，根據該數值閃動 LED。因此把程式分別安裝到兩顆 TAROKO 上並接上電源，則兩顆 TAROKO 的 LED 會同步閃動。

另外本章也會示範如何設定 TAROKO 的無線傳輸頻道(Radio Channel)及傳輸功率(Transmission Power)。關於 TinyOS 程式的撰寫方法及範例程式的詳細資料，請參考 TinyOS 網站(<http://www.tinyos.net>)及相關範例程式資料夾內的 README 檔。

### 安裝 CntToLedsAndRfm

- a、打開 Cygwin，輸入 "motelist" 指令，會顯示出所有連接到 PC 的 TAROKO。從以下畫面可看出有兩個 TAROKO 分別連接到 COM25 及 COM26



```
/opt/tinyos-1.x/apps/Oscilloscope/build/telosh
TA0XRXH8 no_comm TAROKO RevA

sylvan@sylvan /opt/tinyos-1.x/apps/Oscilloscope/build/telosh
$ motelist
Reference  CommPort  Description
-----
TA0XRXH9   no_comm    TAROKO RevA
TA0XRXH8   COM23      TAROKO RevA

sylvan@sylvan /opt/tinyos-1.x/apps/Oscilloscope/build/telosh
$ motelist
Reference  CommPort  Description
-----
TA0XRXH8   COM23      TAROKO RevA
TA0XRXH9   COM24      TAROKO RevA

sylvan@sylvan /opt/tinyos-1.x/apps/Oscilloscope/build/telosh
$ motelist
Reference  CommPort  Description
-----
TA0XRXQ0    COM25      TAROKO RevA
TA0XRXQF    COM26      TAROKO RevA

sylvan@sylvan /opt/tinyos-1.x/apps/Oscilloscope/build/telosh
$
```

## b、切換至 CntToLedsAndRfm 資料夾

指令: \$ cd /opt/tinyos-1.x/apps/CntToLedsAndRfm

## c、將此程式安裝到 TARAKO 上，因 TAROKO 與 Telosb 完全相容，只要把平台指定為 Telosb 就可以供 TAROKO 使用。

指令: \$ make telosb install, A bsl, B

A 為指定給該 Taroko 的 local address, 範圍從 0~255

B 為連接該 Taroko 的 COM port 減 1, 假設 motelist 顯示為 COM25, 則 B=24

Example: \$ make telosb install, 1 bsl, 24

```

/opt/tinyos-1.x/apps/CntToLedsAndRfm
sylvan@sylvan ~
$ cd /opt/tinyos-1.x/apps/CntToLedsAndRfm/

sylvan@sylvan /opt/tinyos-1.x/apps/CntToLedsAndRfm
$ make telosb install, 1 bsl, 24
mkdir -p build/telosb
compiling CntToLedsAndRfm to a telosb binary
ncc -o build/telosb/main.exe -O -I~/lib/Counters -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnes
c-cfile=build/telosb/app.c -board= -I~/lib/Deluge -Wl,--section-start=.text=0x4800,--defsym=_reset_vector_=0x4000 -DID
ENT_PROGRAM_NAME="\CntToLedsAndRfm\" -DIDENT_USER_ID="\sylvan\" -DIDENT_HOSTNAME="\sylvan\" -DIDENT_USER_HASH=0x0191745fL
-DIDENT_UNIX_TIME=0x44ba3b70L -DIDENT_UID_HASH=0x0699849cL -mdisable-hwmu1 -I/opt/tinyos-1.x/tos/lib/CC2420Radio CntToLe
dsAndRfm.nc -lm
C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/CC2420Radio/CC2420RadioM.nc:116: warning: 'Send.sendDone' called asynchron
ously from 'sendFailed'
compiling CntToLedsAndRfm to build/telosb/main.exe
12088 bytes in ROM
373 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
writing TOS image
/opt/tinyos-1.x/tools/make/msp/set-mote-id --objcopy msp430-objcopy --objdump msp430-objdump --target ihex build/telosb/
main.ihex build/telosb/main.ihex.out-1 1
installing telosb bootloader using bsl
msp430-bsl --telosb -c 24 -r -e -I -p C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/Deluge/TOSBoot/build/telosb/main.ih
e
x
MSP430 Bootstrap Loader Version: 1.39-telos-7

```

## 安裝 RfmToLeds

## a、切換到 apps/RfmToLeds

指令: \$ cd /opt/tinyos-1.x/apps/RfmToLeds

## b、將此程式安裝到 TARAKO 上

指令: \$ make telosb install, 2 bsl, 25

```

/opt/tinyos-1.x/apps/RfmToLeds
sylvan@sylvan /opt/tinyos-1.x/apps/CntToLedsAndRfm
$ cd /opt/tinyos-1.x/apps/RfmToLeds/

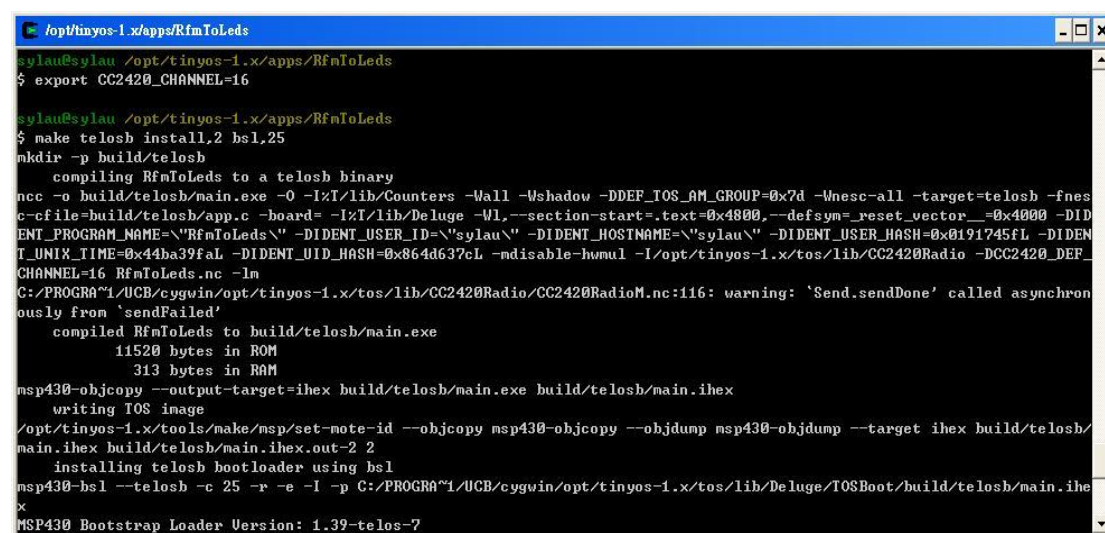
sylvan@sylvan /opt/tinyos-1.x/apps/RfmToLeds
$ make telosb install, 2 bsl, 25
mkdir -p build/telosb
compiling RfmToLeds to a telosb binary
ncc -o build/telosb/main.exe -O -I~/lib/Counters -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnes
c-cfile=build/telosb/app.c -board= -I~/lib/Deluge -Wl,--section-start=.text=0x4800,--defsym=_reset_vector_=0x4000 -DID
ENT_PROGRAM_NAME="\RfmToLeds\" -DIDENT_USER_ID="\sylvan\" -DIDENT_HOSTNAME="\sylvan\" -DIDENT_USER_HASH=0x0191745fL -DIDEN
T_UNIX_TIME=0x44ba3cf2L -DIDENT_UID_HASH=0x2b3294afL -mdisable-hwmu1 -I/opt/tinyos-1.x/tos/lib/CC2420Radio RfmToLeds.nc
-lm
C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/CC2420Radio/CC2420RadioM.nc:116: warning: 'Send.sendDone' called asynchron
ously from 'sendFailed'
compiling RfmToLeds to build/telosb/main.exe
11520 bytes in ROM
313 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
writing TOS image
/opt/tinyos-1.x/tools/make/msp/set-mote-id --objcopy msp430-objcopy --objdump msp430-objdump --target ihex build/telosb/
main.ihex build/telosb/main.ihex.out-2 2
installing telosb bootloader using bsl
msp430-bsl --telosb -c 25 -r -e -I -p C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/Deluge/TOSBoot/build/telosb/main.ih
e
x
MSP430 Bootstrap Loader Version: 1.39-telos-7

```



## 設定無線傳輸頻道(Radio Channel)

- a、TAROKO 總共有 16 個可用的無線傳輸頻道(Radio Channel)，設定範圍從 11~26。TinyOS 預設的頻道是 11。設定方法為先輸入以下指令"export CC2420\_CHANNEL=X"(其中 X 為欲使用的頻道)，再執行編譯及安裝程式。
- 指令：`$ export CC2420_CHANNEL=16`
- 指令：`$ make telosb install,2 bsl,25`



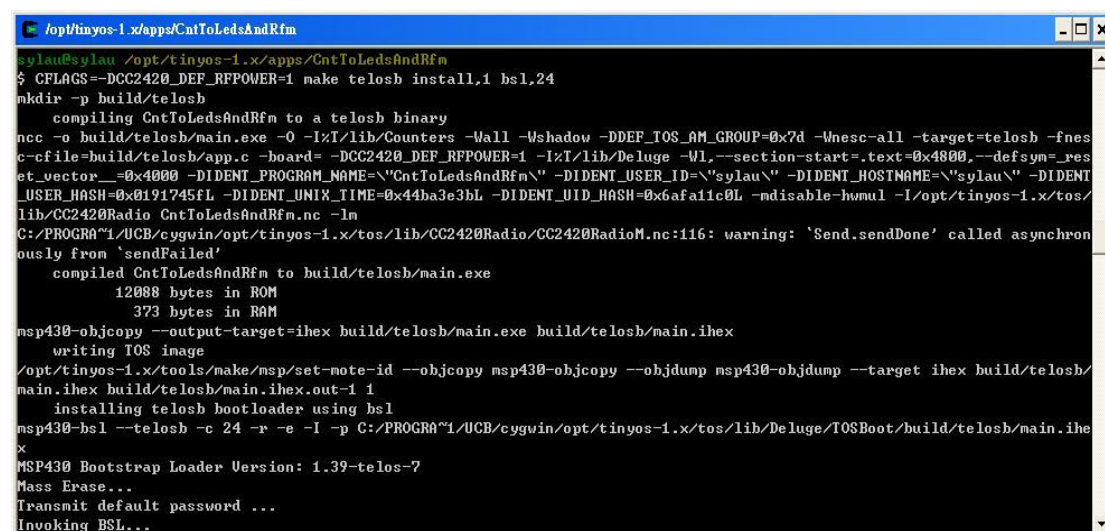
```

/opt/tinyos-1.x/apps/RfmToLeds
sylvan@sylvan /opt/tinyos-1.x/apps/RfmToLeds
$ export CC2420_CHANNEL=16

sylvan@sylvan /opt/tinyos-1.x/apps/RfmToLeds
$ make telosb install,2 bsl,25
mkdir -p build/telosb
compiling RfmToLeds to a telosb binary
ncc -o build/telosb/main.exe -O -I/I/lib/Counters -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnes
c-cfile=build/telosb/app.c -board= -I/I/lib/Deluge -Wl,--section-start=.text=0x4000,--defsym=_reset_vector_=0x4000 -DID
ENT_PROGRAM_NAME="RfmToLeds" -DIDENT_USER_ID="sylvan" -DIDENT_HOSTNAME="sylvan" -DIDENT_USER_HASH=0x0191745fL -DIDENT
T_UNIX_TIME=0x44ba39fL -DIDENT_UID_HASH=0x864d637cL -mdisable-hwmu1 -I/opt/tinyos-1.x/tos/lib/CC2420Radio -DCC2420_DEF
CHANNEL=16 RfmToLeds.nc -lm
C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/CC2420Radio/CC2420RadioM.nc:116: warning: 'Send.sendDone' called asynchron
ously from 'sendFailed'
compiled RfmToLeds to build/telosb/main.exe
11520 bytes in ROM
313 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
writing TOS image
/opt/tinyos-1.x/tools/make/msp/set-mote-id --objcopy msp430-objcopy --objdump msp430-objdump --target ihex build/telosb/
main.ihex build/telosb/main.ihex.out-2 2
installing telosb bootloader using bsl
msp430-bsl --telosb -c 25 -r -e -I -p C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/Deluge/TOSBoot/build/telosb/main.ihe
x
MSP430 Bootstrap Loader Version: 1.39-telos-7
  
```

## 設定傳輸功率(Transmission Power)

- a、傳輸功率(Transmission Power)的設定範圍:31~1(0 dBm ~ -25 dBm)，數字越大 power 越高，預設為 31(即 0 dBm)。設定方法為在 make 之前加上 "CFLAGS=-DCC2420\_DEF\_RFPOWER=Y"(其中 Y 為欲設定的功率)。
- 指令：`$ CFLAGS=-DCC2420_DEF_RFPOWER=1 make telosb install,2 bsl,25`



```

/opt/tinyos-1.x/apps/CntToLedsAndRfm
sylvan@sylvan /opt/tinyos-1.x/apps/CntToLedsAndRfm
$ CFLAGS=-DCC2420_DEF_RFPOWER=1 make telosb install,1 bsl,24
mkdir -p build/telosb
compiling CntToLedsAndRfm to a telosb binary
ncc -o build/telosb/main.exe -O -I/I/lib/Counters -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnes
c-cfile=build/telosb/app.c -board= -DCC2420_DEF_RFPOWER=1 -I/I/lib/Deluge -Wl,--section-start=.text=0x4000,--defsym=_res
et_vector_=0x4000 -DIDENT_PROGRAM_NAME="CntToLedsAndRfm" -DIDENT_USER_ID="sylvan" -DIDENT_HOSTNAME="sylvan" -DIDENT
_USER_HASH=0x0191745fL -DIDENT_UNIX_TIME=0x44ba3e3bL -DIDENT_UID_HASH=0x6afa11c0L -mdisable-hwmu1 -I/opt/tinyos-1.x/tos/
lib/CC2420Radio CntToLedsAndRfm.nc -lm
C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/CC2420Radio/CC2420RadioM.nc:116: warning: 'Send.sendDone' called asynchron
ously from 'sendFailed'
compiled CntToLedsAndRfm to build/telosb/main.exe
12088 bytes in ROM
373 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
writing TOS image
/opt/tinyos-1.x/tools/make/msp/set-mote-id --objcopy msp430-objcopy --objdump msp430-objdump --target ihex build/telosb/
main.ihex build/telosb/main.ihex.out-1 1
installing telosb bootloader using bsl
msp430-bsl --telosb -c 24 -r -e -I -p C:/PROGRAM~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/Deluge/TOSBoot/build/telosb/main.ihe
x
MSP430 Bootstrap Loader Version: 1.39-telos-7
Mass Erase...
Transmit default password ...
Invoking BSL...
  
```

## 5. 收集資料

Sensor 的資料必須收集到電腦上以供分析及統計。TAROKO 具有 USB 介面，任何一顆 TAROKO 只要安裝上 TOSBase 的程式就可以當做 Gateway 收集資料，傳送到電腦上。TOSBase 是 TinyOS 內的一支基地台(Base Station)程式，功能是接收其他 sensor node 發送過來的資料，經由 USB 介面傳送到電腦上。

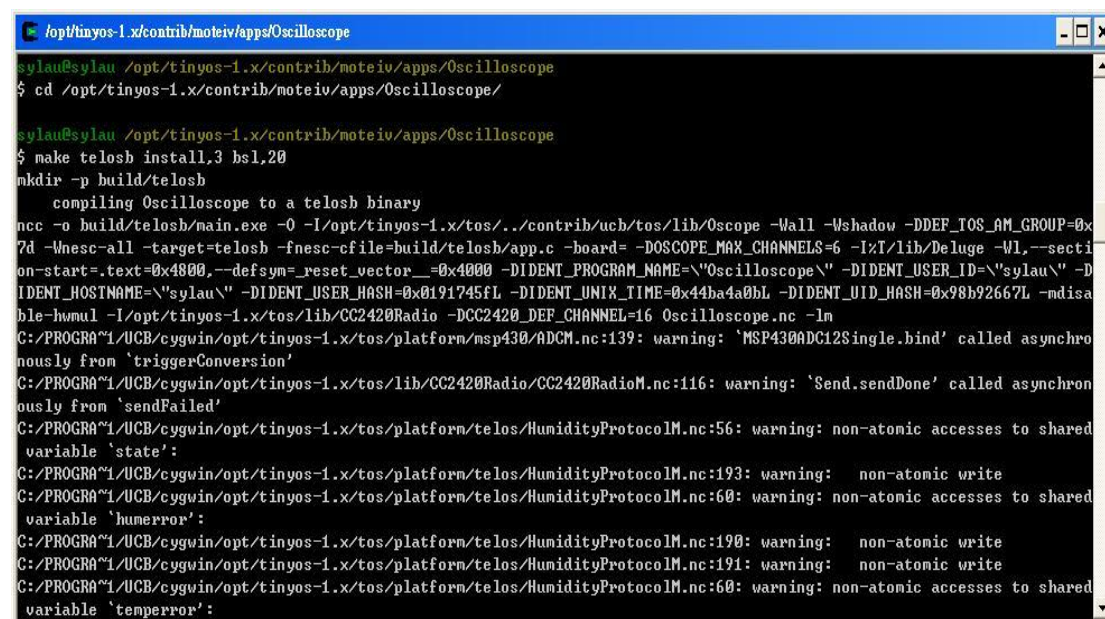
在這裡我們將在一顆裝有溫濕度感測器及光感測器的 TAROKO 上，安裝一支 Oscilloscope 的程式。Oscilloscope 的功能是收集 TAROKO 上各種感測器所量測到的數值，經由無線傳送出去。另外也將示範如何利用 TinyOS 內建的圖型化介面來顯示感測資料。

### 安裝 Oscilloscope

a、將 Oscilloscope 程式安裝到具有感測器的 TAROKO 上。

切換資料夾，指令：`$ cd /opt/tinyos-1.x/contrib/moteiv/apps/Oscilloscope`

安裝程式，指令：`$ make telosb install,3 bsl,20`



```
/opt/tinyos-1.x/contrib/moteiv/apps/Oscilloscope
syla@syla /opt/tinyos-1.x/contrib/moteiv/apps/Oscilloscope
$ cd /opt/tinyos-1.x/contrib/moteiv/apps/Oscilloscope/
syla@syla /opt/tinyos-1.x/contrib/moteiv/apps/Oscilloscope
$ make telosb install,3 bsl,20
mkdir -p build/telosb
compiling Oscilloscope to a telosb binary
ncc -o build/telosb/main.exe -O -I/opt/tinyos-1.x/tos/./contrib/uch/tos/lib/Oscope -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnesc-cfile=build/telosb/app.c -board= -DOSCOPE_MAX_CHANNELS=6 -IzT/lib/Deluge -Wl,--section-start=.text=0x4800,--defsym=_reset_vector_=0x4000 -DIDENT_PROGRAM_NAME="Oscilloscope" -DIDENT_USER_ID="syla" -DIDENT_HOSTNAME="syla" -DIDENT_USER_HASH=0x0191745fL -DIDENT_UNIX_TIME=0x44ba4a0bL -DIDENT_UID_HASH=0x98b92667L -mdisable-hwmu1 -I/opt/tinyos-1.x/tos/lib/CC2420Radio -DCC2420_DEF_CHANNEL=16 Oscilloscope.nc -lm
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/msp430/ADCM.nc:139: warning: 'MSP430ADC12Single.bind' called asynchronously from 'triggerConversion'
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/lib/CC2420Radio/CC2420RadioM.nc:116: warning: 'Send.sendDone' called asynchronously from 'sendFailed'
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/telos/HumidityProtocolM.nc:56: warning: non-atomic accesses to shared variable 'state':
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/telos/HumidityProtocolM.nc:193: warning: non-atomic write
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/telos/HumidityProtocolM.nc:60: warning: non-atomic accesses to shared variable 'humerror':
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/telos/HumidityProtocolM.nc:190: warning: non-atomic write
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/telos/HumidityProtocolM.nc:191: warning: non-atomic write
C:/PROGRA~1/UCB/cygwin/opt/tinyos-1.x/tos/platform/telos/HumidityProtocolM.nc:60: warning: non-atomic accesses to shared variable 'temperror':
```

## 安裝 TOSBase

a、將 TOSBase 程式安裝到另一個 TAROKO 上。

切換資料夾，指令：`$ cd /opt/tinvos-1.x/apps/TOSBase`

安裝程式，指令：`$ make telosb install.254 bsl.23`

```

/opt/tinyos-1.x/apps/TOSBase

sylvau@sylvau ~
$ cd /opt/tinyos-1.x/apps/TOSBase/

sylvau@sylvau /opt/tinyos-1.x/apps/TOSBase
$ make telosb install.254 bs1.23
mkdir -p build/telosb
    compiling TOSBase to a telosb binary
ncc -o build/telosb/main.exe -O -Wall -Wshadow -DDEF_IOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnesc-cfile=build/telosb/app.c -board=-DTOSH_MAX_TASKS_LOG2=8 -I../lib/Deluge -Wl,--section-start=.text=0x4800,--defsym=_reset_vector_=0x4000 -DIDENT_PROGRAM_NAME="\TOSBase\" -DIDENT_USER_ID="\sylvau\" -DIDENT_HOSTNAME="\sylvau\" -DIDENT_USER_HASH=0x0191745fL -DIDENT_UNIX_TIME=0x44ba067L -DIDENT_UID_HASH=0x52d0c7e1L -mdisable-hwmu1 -I/opt/tinyos-1.x/tos/lib/CC2420Radio -DCC2420_DEF_CHANNEL=16 TOSBase.nc -lm
C:/PROGRAM"1/UCB/cygwin/opt/tinyos-1.x/tos/lib/CC2420Radio/CC2420RadioM.nc:116: warning: 'Send.sendDone' called asynchronously from 'sendFailed'
    compiled TOSBase to build/telosb/main.exe
        11066 bytes in ROM
        1848 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
    writing TOS image
/opt/tinyos-1.x/tools/make/msp/set-mote-id --objcopy msp430-objcopy --objdump msp430-objdump --target ihex build/telosb/main.ihex build/telosb/main.ihex.out-254 254
    installing telosb bootloader using bs1
msp430-bs1 --telosb -c 23 -r -e -I -p C:/PROGRAM"1/UCB/cygwin/opt/tinyos-1.x/tos/lib/Deluge/TOSBoot/build/telosb/main.ihex
x
MSP430 Bootstrap Loader Version: 1.39-telos-7

```

打開 Listen

a、Listen 為 TinyOS 內建的 Java 程式，可接收 TOSBase 傳送過來的資料並將 Raw Data 顯示於 Cygwin 的視窗上。其指令為

```
"MOTECOM=serial@COMXX:telos java net.tinyos.tools.Listen"
```

其中 XX 為安裝 TOSBase 的 TAROKO 所連結的 COM port。

指令：`$ MOTECOM=serial@COM24:telos java net.tinyos.tools.Listen`

[illegible]

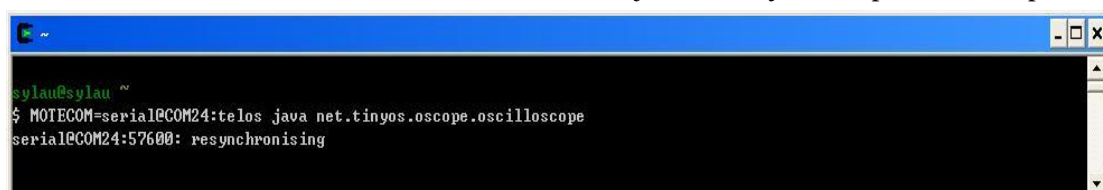


## 打開 Oscilloscope 圖型化介面

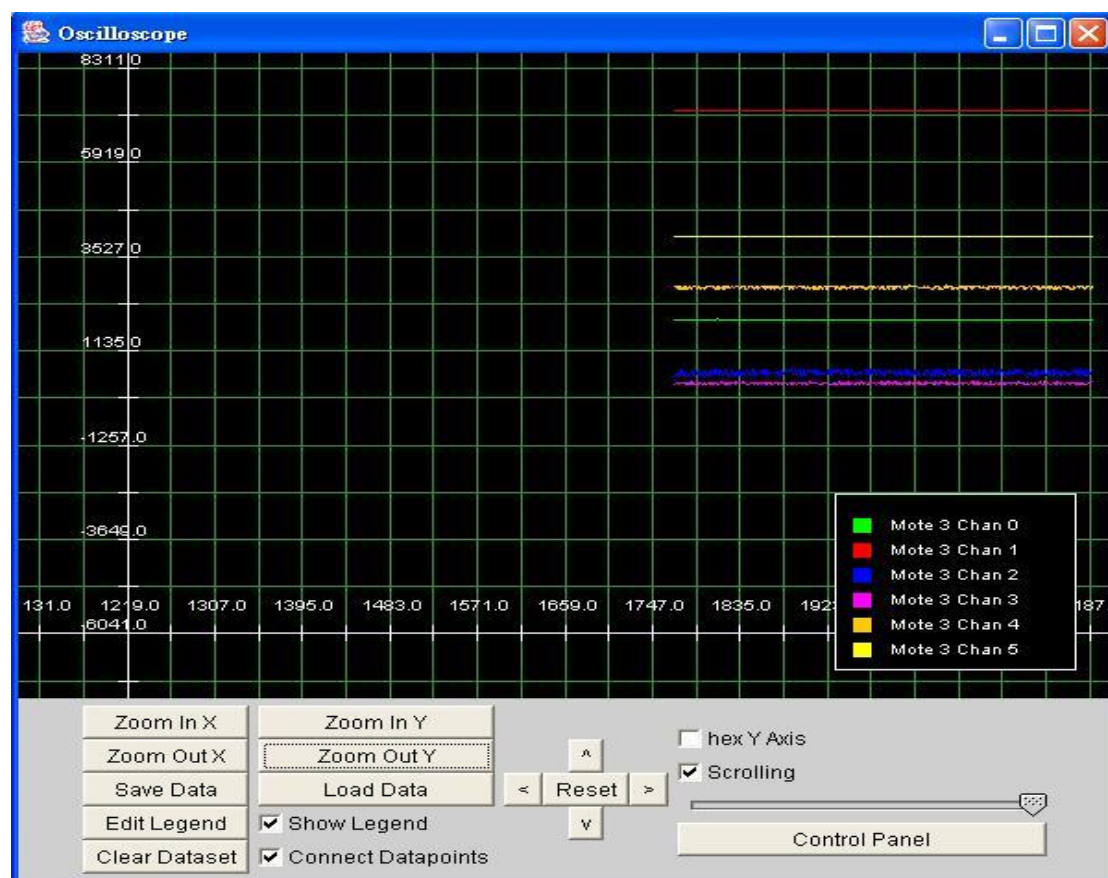
- a、從 Listen 所得到的資料只是一整串卦包的數值，TinyOS 內另有一支圖型化介面的 Java 程式，叫做 Oscilloscope，可將這些資料以圖型化的方式顯示出來。其指令為

"MOTECOM=serial@COMXX:telos java net.tinyos.oscope.oscilloscope"，其中 XX 為安裝 TOSBase 的 TAROKO 所連結的 COM port。

指令：\$ MOTECOM=serial@COM24:telos java net.tinyos.oscope.oscilloscope



- b、跳出以下畫面後，將 Scrolling 打勾。如果看不到資料，點幾下 Zoom out Y，以便顯示資料。



## Sensor 資料解讀

a、Oscilloscope 所顯示的數值為 ADC 輸出的 raw data，必須經過轉換才是正確讀值。

b、Oscilloscop 共輸出 6 項資料

- Humidity
- Temperature
- TSR (Total Solar Radiation)
- PAR (Photosynthetically Active Radiation)
- Internal Voltage
- Internal Temperature

c、Channel 0 (綠色)：濕度

- RawData: 12bit value
- 轉換程式：轉換為相對濕度
  - 先求出未經溫度補償的濕度
  - $Hum (4) = (0.405 * RawData) * (2.8e^{-6} * RawData^2)$
  - 再求出經過溫度補償後的正確濕度
  - $Hum_{Compensated} = (目前溫度^{\circ}C - 25^{\circ}C) * (0.01 - 0.0008 * RawData) + Hum$

d、Channel 1 (紅色)：溫度

- RawData: 14bit value
- 轉換程式：轉換為  $^{\circ}C$
- $Temperature = 39.6 - 0.01 * RawData$

e、Channel 2 (藍色)：TSR

- Sensor: S1087-01
- Total Solar Radiation: 量測包括紅外線在內的所有光線
- 波長範圍: 320~1100 nm
- RawData: 12bit value
- 轉換程式：轉換為照度 (Lux)
- $TSR = 10^8 * ((RawData / 4096) * 1.5) / 10^5$

f、Channel 3 (粉紅色)：PAR

- Sensor: S1087
- Photosynthetically Active Radiation: 量測可見光(與光合作用有關的光線)
- 波長範圍: 320~730 nm
- RawData: 12bit value
- 轉換程式：轉換為照度 (Lux)
- $PAR = 10^9 * ((RawData / 4096) * 1.5) / 10^5$



g、Channel 4 (橘色): Internal Temperature

- 量測 Microcontroller 內部溫度
- RawData: 12bit value
- 轉換程式: 轉換為 °C
- $In\_Temperature = (((RawData / 4096) * 1.5) - 0.986) / 0.00355$

h、Channel 5 (黃色): Internal Voltage

- 量測 Microcontroller supply voltage
- RawData: 12bit value
- 轉換程式: 轉換為 V
- $SupplyVoltage = ((RawData / 4096) * 1.5) * 2$