

EBox-AIO-002

多功能信號控制器

操作手冊

版本: V1.02

目錄

1.	產品簡介	1
1.1.	產品概述	1
1.2.	外觀簡介	1
2.	產品規格	2
2.1.	硬體規格	2
2.2.	LED 指示燈說明	4
2.3.	各通訊及控制介面腳位定義	5
2.4.	軟體規格說明	6
2.5.	Embedded-Linux 檔案系統	6
2.6.	各通訊及控制介面軟體定義	8
3.	主控口(Console).....	9
3.1.	出廠設定值	9
3.2.	測試環境建議	9
3.3.	Ethernet Console 主控台連線.....	12
3.4.	RS-232 Console 主控台連線.....	26
4.	安裝開發環境	32
4.1.	環境需求	32
4.2.	安裝 Linux Tool Chain	32
4.3.	安裝 Windows Tool Chain	33
5.	執行第一隻程式 – Hello (以 Linux 開發環境為例).....	35
5.1.	原程式碼	35
5.2.	產生執行檔	35
5.3.	上傳到 EBox-AIO-002	35
5.4.	測試執行程式	36
6.	常用基本操作	37
6.1.	變更網路設定	37
6.2.	初始化開機設定及應用程式	38
6.3.	外掛 USB 及 SD 記憶體	39
6.4.	變更系統時間	39
6.5.	Web Server 基本設定.....	40
6.6.	變更主控台(console)歡迎詞	42
6.7.	LLD 專用工具程式.....	43

1. 產品簡介

1.1. 產品概述

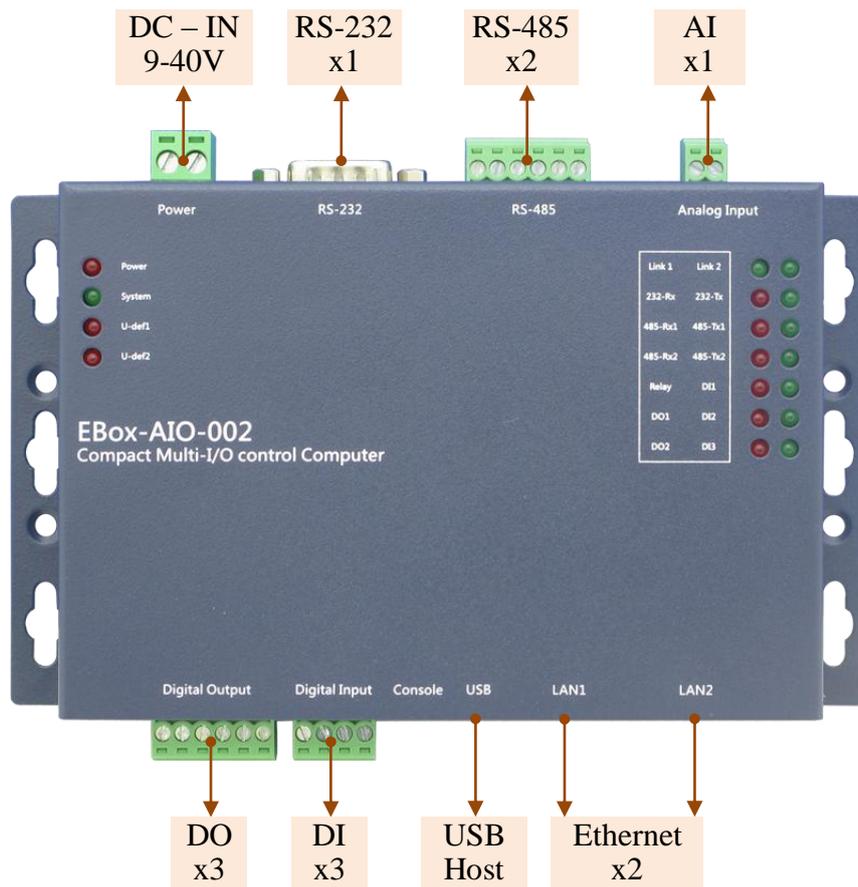
EBox-AIO-002 是一款 ARM-based 嵌入式架構的網路型信號控制器，主要可應用於現場端設備或儀表的即時監控，並搭配 SQL 資料庫進行資料採集及暫存，並透過 TCP/IP 網路介面與遠端監控主機連接，實現遠端監控的應用。

EBox-AIO-002 內建的 Linux 作業系統是一種開放式的架構，使用者可以利用公開的 GNU 軟體開發工具，安裝免費的 ARM-Linux 嵌入式系統專用的 C/C++ 程式編譯器(Compiler)及函數庫(Lib)後，進行二次應用程式的開發，使 EBox-AIO-002 成為一款專用的控制器或閘道器(Gateway)。

EBox-AIO-002 具備多元的通訊及控制功能。有乙太網路介面，用來連接網路(LAN 或 WAN)與後台系統連接；還具備多組串列通訊介面(RS-232 及 RS-485)，讓 EBox-AIO-002 可連接更多不同類別的設備或儀表。充份扮演好一個設備監控或資料處理的閘道器角色。EBox-AIO-002 同時也具備數位控制(Digital I/O)的功能，無需透過額外的外接模組就可以進行現場端的開關控制。另外還有一路類比介面(Analog Input)，只要連接對應的感測器(sensor)就可以進行相關的資料採集，如溫濕度、壓力、風速等常見的感應器。

工業級的設計，無風扇、低功耗，適用於各類長時間不間斷的監控應用。採用金屬外殼，除了增加產品的強度外，壁掛及導軌安裝的設計，讓 EBox-AIO-002 可輕易的安裝使用在各類的場域。RJ45、DB9、可插拔的端子(Terminal Block)，都是常見的接頭，種種的設計，都是易於現場施工人員的配線。

1.2 外觀簡介



2. 產品規格

2.1. 硬體規格

系統核心

- ▷ CPU：ATMEL AT9G20 400MHz ARM9 @400MHz
- ▷ 記憶體：64MB SDRAM, 128MB NAND FLASH

網路介面

- ▷ 數量：2 組
- ▷ 類型：10/100BaseT 乙太網路(Ethernet)
- ▷ 接頭：RJ45

類比輸入控制(Analog Input)

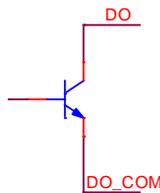
- ▷ 數量：1 組
- ▷ 信號種類：0~20mA or 0-10VDC(by switch)
- ▷ 解析度：12-bit
- ▷ 頻率：10Hz
- ▷ 接頭：3.5mm 端子座

繼電器數位輸出控制(Digital Output)

- ▷ 數量：1 組
- ▷ 信號種類：SPDT 繼電器，N.O./ N.C./ COM
- ▷ 輸入電壓範圍：1A@120VAC / 1A@24VDC
- ▷ 保護：2000Vrms 光隔離保護
- ▷ 接頭：3.5mm 可插拔端子座

數位輸出控制(Digital Output)

- ▷ 數量：2 組
- ▷ 信號種類：Open Collector
- ▷ 推動能力：200mA
- ▷ 保護：2000Vrms 光隔離保護
- ▷ 接頭：3.5mm 可插拔端子座



數位輸入控制(Digital Input)

- ▷ 數量：3 組
- ▷ 輸入電壓範圍：5~24VDC
- ▷ 信號種類：濕接點(sink mode)
- ▷ 保護：2000Vrms 光隔離保護
- ▷ 接頭：3.5mm 可插拔端子座

RS-232 串列埠介面

- ▷ 數量：1 組
- ▷ 信號：TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND
- ▷ 保護：15KV ESD 靜電保護，400W 突波保護
- ▷ 接頭：DB9 公頭

RS-485 串列埠介面

- ▷ 數量：2 組
- ▷ RS-485 信號：Data+, Data-, GND (支援自動流向控制)
- ▷ Multi-Drop Nodes：400
- ▷ 內建終端電阻：120Ω，可透過 Jumper 設定
- ▷ 保護：2KV 隔離保護，15KV ESD 靜電保護，400W 突波保護
- ▷ 接頭：3.5mm 可插拔端子座

串列埠通訊參數

- ▷ Baud Rate : 300 ~ 921,600 bps
- ▷ Parity : None, Even, Odd, Mark, Space
- ▷ Data Bits : 5, 6, 7, 8
- ▷ Stop Bit : 1, 1.5, 2 bits
- ▷ Flow Control(RS-232 only) : RTS/CTS, XON/XOFF, None

USB 介面

- ▷ 數量 : 1 組
- ▷ 類型 : USB2.0 相容
- ▷ 接頭 : Single (Type A)

SD 擴充介面

- ▷ 數量 : 1 組(需開殼)
- ▷ 接頭 : Micro SD 插槽

主控台串列埠(Console)

- ▷ 數量 : 1 組
- ▷ 信號 : RS-232 (TxD, RxD, GND)
- ▷ 終端樣式 : 115,200 bps, VT-100
- ▷ 接頭 : 2.54mm 排針座

機構

- ▷ 材質/尺寸 : 鍍鋅鋼板 / 151 x 95 x 31mm

電源

- ▷ 工作電壓 : DC 9~24VDC
- ▷ 電源接頭 : 5.00mm 端子座
- ▷ 功耗 : <24W (不含 USB device)

其它

- ▷ Real Time Clock : 1 組
- ▷ Buzzer : 1 組
- ▷ LED 指示燈 : 電源, 網路, 串列埠, 控制信號, 使用者自定義
- ▷ 適用溫度 : 0~50°C
- ▷ 適用濕度 : 20%~80% RHG
- ▷ 認證 : CE, FCC

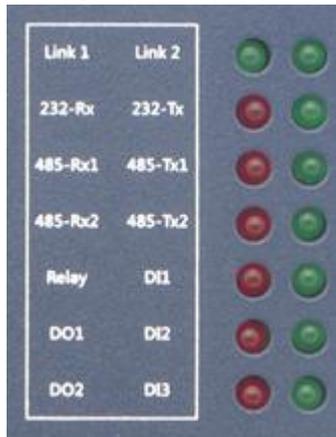
2.2.LED 指示燈說明

系統屬性



Power	電源指示燈 正確接上工作電壓
System	系統指示燈 Linux 系統正常完成開機，約 15~20 秒亮起
U-Def	使用者控制燈號 使用者以 DO(Digital Output)的方式控制

通訊及控制屬性



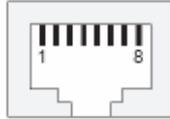
Link	網路燈號 燈亮：正確連接網路線 燈閃：網路資料傳輸中
Tx	RS-232 及 RS-485 資料傳送燈號 燈閃：資料傳輸中
Rx	RS-232 及 RS-485 資料接收燈號 燈閃：資料接收中
Relay	繼電器狀態指示燈號 燈亮：GPIO 變為 True 的狀態，繼電器切換到 NO 位置 燈滅：GPIO 變為 False 的狀態，繼電器切換到 NC 位置
DO	一般 DO 狀態指示燈號 燈亮：GPIO 變為 True 的狀態，電晶體導通
DI	一般 DI 狀態指示燈號 燈亮：偵測外部電壓為高準位(5~24V DC)，對應 GPIO 判讀為 True 狀態

2.3.各通訊及控制介面腳位定義

LAN1、LAN2 乙太網路介面(Ethernet)

腳位	信號
1	ETX+
2	ETX-
3	ERX+
6	ERX-

RJ45



工作電壓輸入

腳位	信號
1	DC +
2	GND

- DC Range: 9~24V DC

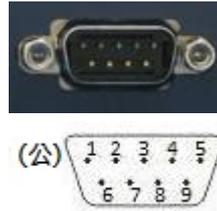
5.00mm 端子座



RS-232

腳位	信號
1	DCD
2	RxD
3	TxD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	x

DB9



數位輸出控制(Digital Output)

腳位	信號	
1	Open Collector	DO1
2		DO2
3		COM
4	SPDT Relay	COM
5		N.C.
6		N.O.

3.50mm 端子座



RS-232 Console

腳位	信號
1	TxD
2	RxD
3	GND

3.50mm 端子座



數位輸入控制(Digital Input)

腳位	信號
1	DI1
2	DI2
3	DI3
4	GND

3.50mm 端子座



RS-485

腳位	信號
1	D1+
2	D1-
3	GND
4	D2+
5	D2-
6	GND

3.50mm 端子座



類比輸入(Analog Input)

腳位	信號
1	AI+
2	AI- (COM)

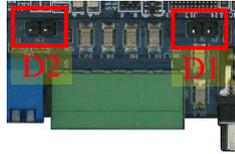
3.50mm 端子座



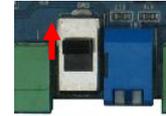
RS-485 終端電阻-需開外殼

類比輸入模式-需開外殼 (AI Mode)

Short (短接)	有 終端電阻
Open (沒接)	無 終端電阻



上 ↑	0-10V DC
下 ↓	0-20 mA



2.4.軟體規格說明

核心

- OS: Linux kernel 2.6.29
- Boot Loader: U-Boot 1.1.2
- File Systems: JFFS2, ETX2/ETX3, VFAT/FAT, NFS

預置工具(Pre-installed Utilities)

- bash, busybox, sysvinit, wget, ipkg, procps, psmics, lighttpd, vsftpd, iptable, ppp, ssh, wireless_tools, usbutils util-linux-mount/umount

網路協議(Protocol stack)

- IPv4, ICMP, ARP, DHCP, NTP, TCP, UDP, FTP, Telnet, HTTP, PPPoE, CHAP, PAP, SMTP, SSL, SSH

服務(Daemons)

- ssh, syslog/klogd, telnet server, ftp server, MySQL, PHP, Web server(lighttpd)

應用程式開發工具 (Tool Chain)

- Linux/Windows 作業系統可用
- GCC: C/C++ PC cross compiler for Linux
- GLIBC: POSIX Library
- 相關的範例程式(examples)
- 提供 Windows 環境整合式開發介面(IDE)

USB 擴充介面支援驅動程式

- Flash thumb disk
- IEEE-802.11b/g WiFi adapter
- 3G adapter
- Web CAM

2.5.Embedded-Linux 檔案系統

EBox-AIO-002 File System 中的空間共有三個區塊，除了最重要的 on-board Flash 外，還有一個 RAM DISK 及使用者外掛的 SD 或 USB Storage

```

root@EBox-AIO-002:/#
root@EBox-AIO-002:/# cd /
root@EBox-AIO-002:/#
root@EBox-AIO-002:/# ls -l
drwxrwxr-x  2 root  root           4416 Nov 12  2010 bin
drwxr-xr-x  7 root  root              0 Jun  4 18:35 dev
lrwxrwxrwx  1 root  root             10 Nov 12  2010 disk -> media/disk
drwxrwxr-x 29 root  root          5368 Mar  9 18:55 etc
drwxrwxr-x  4 root  root           288 Mar  9 19:42 home
drwxrwxr-x  4 root  root          2832 Dec 25 23:03 lib
drwxrwxr-x  6 mysql root           416 Nov 12  2010 media
lrwxrwxrwx  1 root  root             5 Nov 12  2010 mnt -> media
-rw-r--r--  1 root  root           76 Mar  9 22:38 msg.txt
dr-xr-xr-x 61 root  root              0 Jan  1 1970 proc
drwxrwxr-x  2 root  root          4112 Nov 12  2010 sbin
drwxr-xr-x 12 root  root              0 Jan  1 1970 sys
lrwxrwxrwx  1 root  root             8 Sep  2 2014 tmp -> /var/tmp
drwxrwxr-x  8 root  root           544 Nov 18  2009 usr
drwxrwxr-x 10 root  root          1000 May 11  2010 var
root@EBox-AIO-002:/#
root@EBox-AIO-002:/#

```

外掛區塊

RAM Disk



```
root@EBox-AIO-002:~# mount
rootfs on / type rootfs (rw)
ubi0:rootfs on / type ubifs (rw, sync)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
ramfs on /dev type ramfs (rw)
devpts on /dev/pts type devpts (rw, gid=5, mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
tmpfs on /var/volatile type tmpfs (rw, size=16384k)
root@EBox-AIO-002:~#
```



2.6.各通訊及控制介面軟體定義

介面型態	標示	軟體定義
網路	LAN1	eth0
	LAN2	eth1
RS-232	RS-232	/dev/ttyS2
RS-485	RS-485 D1	/dev/ttyS1
	RS-485 D2	/dev/ttyS4
數位輸入控制 (Digital Input)	DI-1	/dev/gpio No.= 00
	DI-2	/dev/gpio No.= 01
	DI-3	/dev/gpio No.= 02
	SW-1 (DIP Switch -01)	/dev/gpio No.= 08
	SW-2 (DIP Switch -02)	/dev/gpio No.= 09
數位輸出控制 (Digital Output)	DO-1	/dev/gpio No.= 03
	DO-2	/dev/gpio No.= 04
	Relay	/dev/gpio No.= 05
	User-def. LED 1	/dev/gpio No.= 06
	User-def. LED 2	/dev/gpio No.= 07
類比輸入 (Analog Input)	AI	liblldai.so.1 (User-Def. Lib)

外掛記憶體	USB Disk	/mnt/sda?
	SD	/mnt/mmc0

```

root@EBox-AIO-002:~# cat /etc/fstab
# stock fstab - you probably want to override this with a machine specific one

rootfs          /                auto            defaults        1 1
proc            /proc           proc            defaults        0 0
devpts          /dev/pts        devpts         mode=0620,gid=5 0 0
usbfs           /proc/bus/usb   usbfs          defaults        0 0
tmpfs           /var/volatile   tmpfs          defaults,size=16M 0 0

# mount dev
/dev/sda1       /media/sda1     auto            defaults,sync,noauto 0 0
/dev/sda        /media/sda1     auto            defaults,sync,noauto 0 0
/dev/sdb1       /media/sdb1     auto            defaults,sync,noauto 0 0
/dev/sdb        /media/sdb1     auto            defaults,sync,noauto 0 0
/dev/mmc        /media/mmc      auto            defaults,sync,noauto 0 0
/dev/mmc0       /media/mmc      auto            defaults,sync,noauto 0 0
/dev/mmcblk0    /media/mmc      auto            defaults,sync,noauto 0 0
/dev/mmcblk0p1  /media/mmc      auto            defaults,sync,noauto 0 0
root@EBox-AIO-002:~#

```



3. 主控口 (Console)

3.1. 出廠設定值

A. 登入密碼 (有大小寫之分)

基本帳戶 guest	Login	guest
	Password	guest
管理者帳戶 root	Login	root
	Password	root

B. 主控口通訊參數

網路	Ethernet 1 (LAN1)	IP address : 192.168.2.127
	Ethernet 1 (LAN1)	IP address : 192.168.3.127
RS-232	Console	Baud Rate : 115,200 bps Data Format : N-8-1 Flow Control : None Terminal Type : VT-100

3.2. 測試環境建議

A. 測試所需器材：

- a. EBox - AIO - 002 一台
- b. DC 9~24V 電源供應器 一個(功耗提供>10W)
(註:可自製轉接線轉換為一般接線)
- c. 網路交叉線
其他建議線材：
 - a. DB9 公對母延長線 一條
(註:若使用筆電，需準備 USB to RS-232 轉接線)
 - b. EBox-AIO-002 專用 RS-232 Console 線(品名: CB-AIOCON-10)



B. 連接方式

a. 電源



b. 網路線





c. RS – 232 Console 線



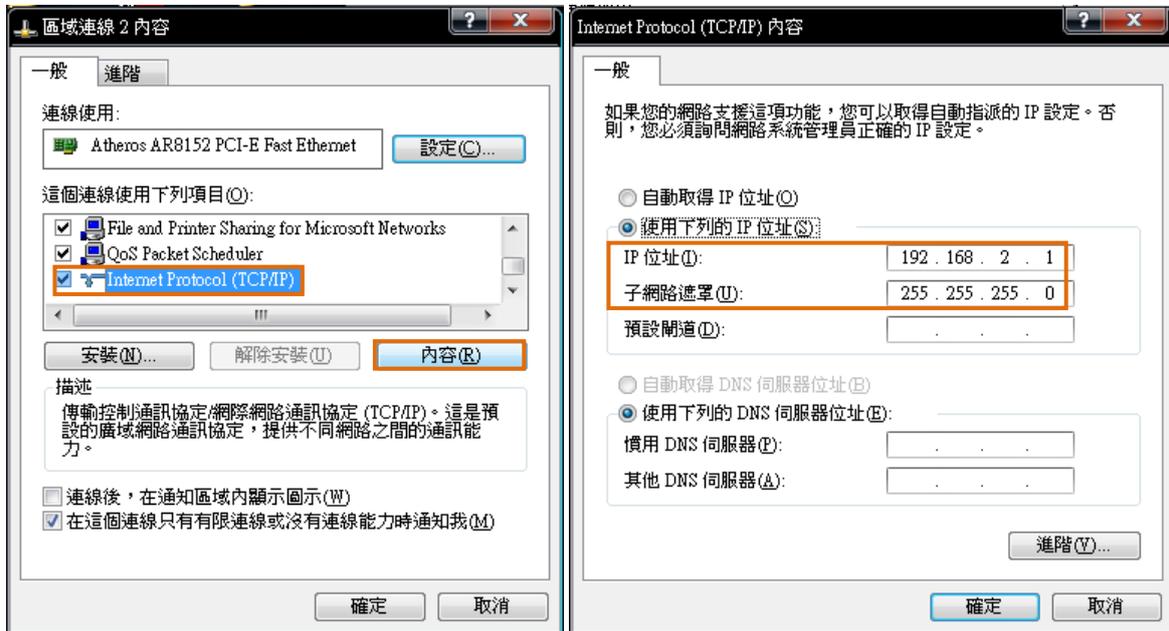
3.3.Ethernet Console 主控台連線

A. Ethernet Console 簡介

- a. 透過網路線與 EBox-AIO-002 連線並對系統做控制及檔案操作
- b. 將網路線一端接到電腦網路口 上另一端接到 EBox-AIO-002 KIT 網路口
- 注意：有些電腦不會自動將普通網路線轉為跳線，所以如果一般網路線導致之後操作有問題，可換跳線測試！
- c. 設定電腦 IP 與 EBox-AIO-002 相同網段
EBox-AIO-002 預設 IP 為 192.168.2.150 Netmask 為 255.255.255.0

➤ WINDOWS XP 的環境設定

Step.01a 網路上芳鄰 → 右鍵-內容 → 區域網路 → 右鍵-內容；或者是由控制台進入。



- 注意：將 IP 設定為非 192.168.2.X，而 X 不能與 EBox-AIO-002 相同，他們的範圍是 1~255！

➤ WINDOWS 7 的環境設定

Step.01b 開始 → 控制台 → 網路和網際網路-檢視網路狀態及工作。

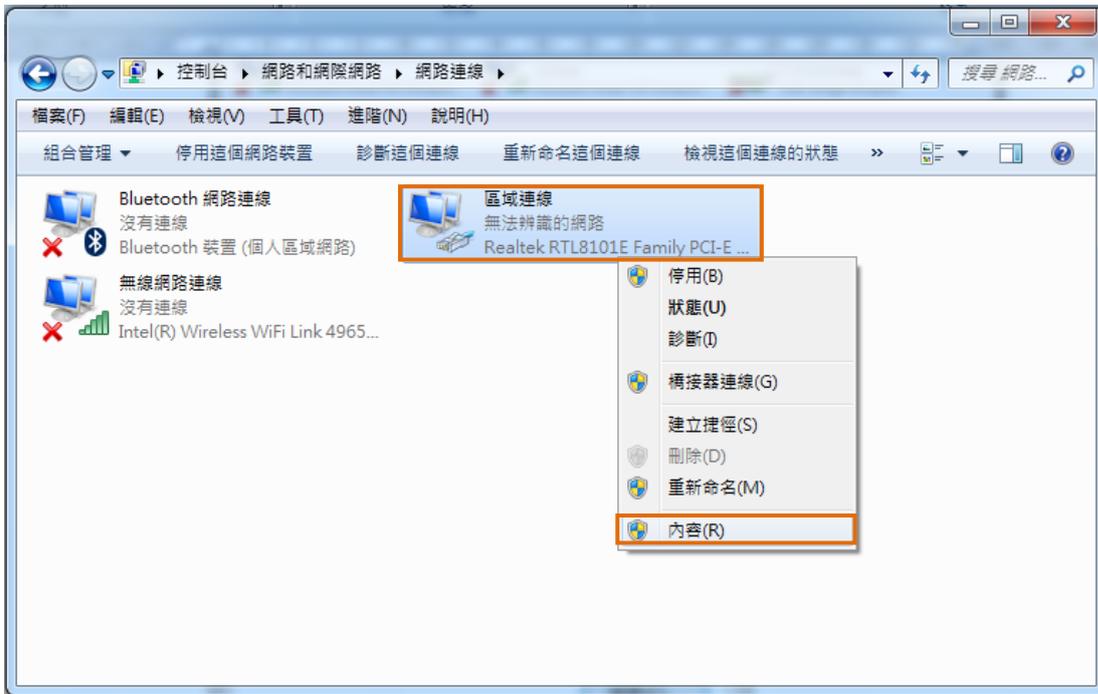


Step.02 變更介面卡設定。





Step.03 區域連線 →按右鍵 內容。

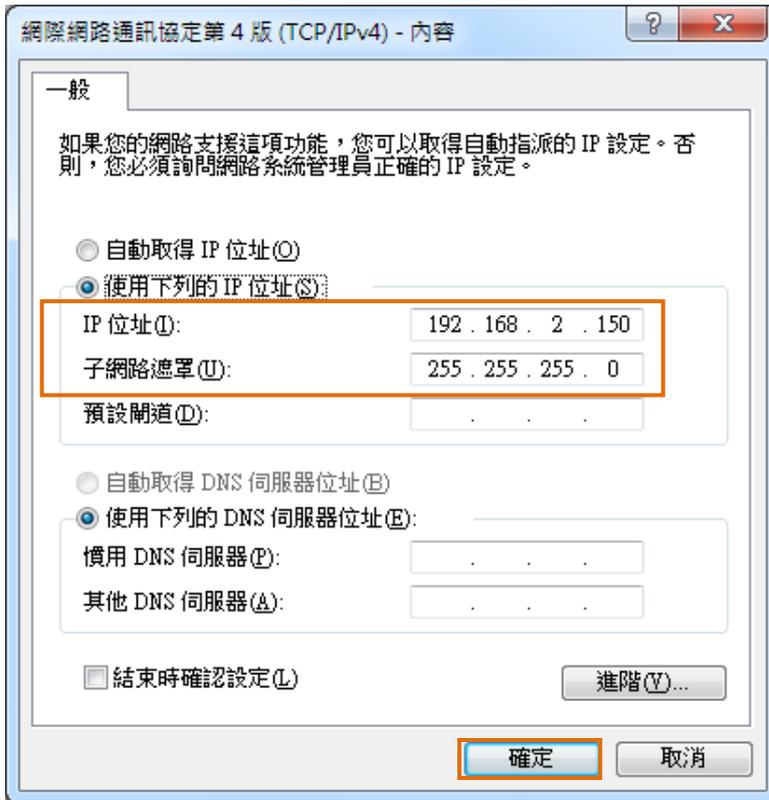


Step.04 網際網路通訊協定第 4 版(TCP/IPv4) → 內容。

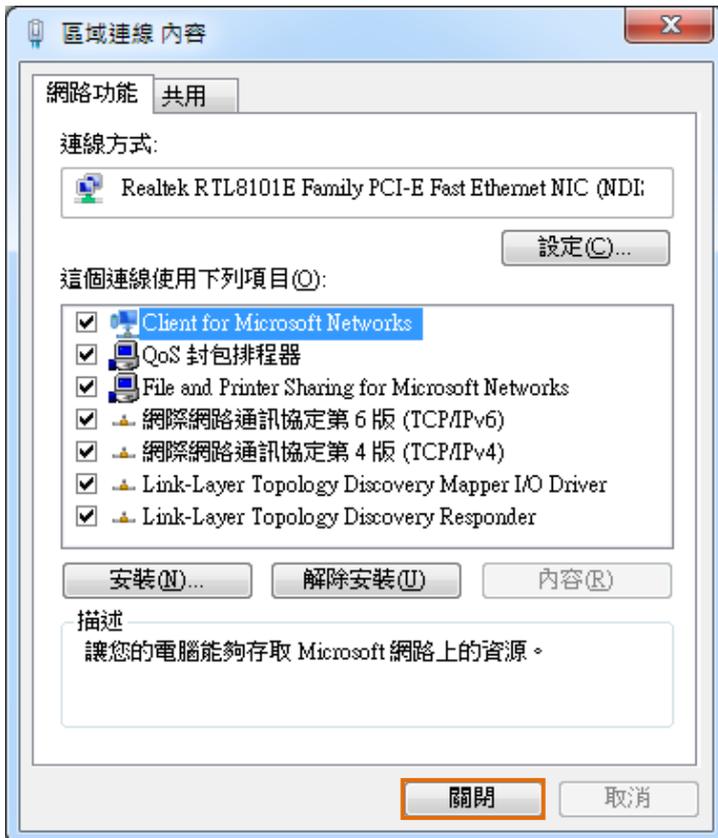




Step.05 IP 位址(I) : 192.168.2.150 ; 子網路遮罩(U) : 255.255.255.0 → 確定鍵。



Step.06 設定完畢點擊關閉鍵。



B. Telnet 操作

✓ WINDOWS XP 的環境設定

預設已有 telnet 功能

✓ WINDOWS 7 的環境設定

Windows7 需要啟動 telnet 功能

Step.01 開始 → 控制台

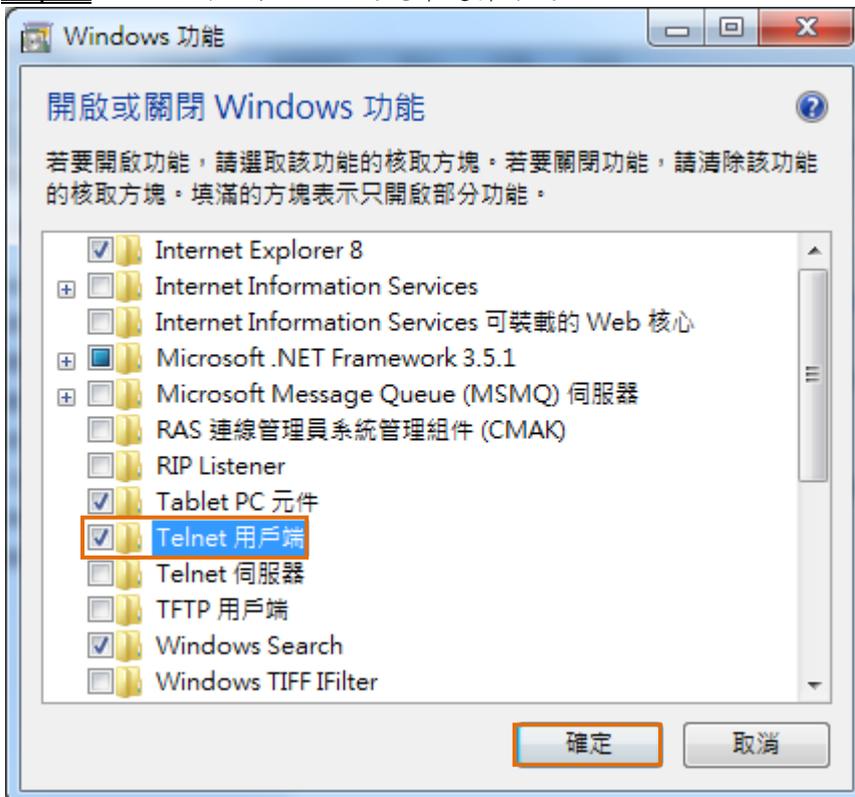




Step.02 開啟或關閉 Windows 功能

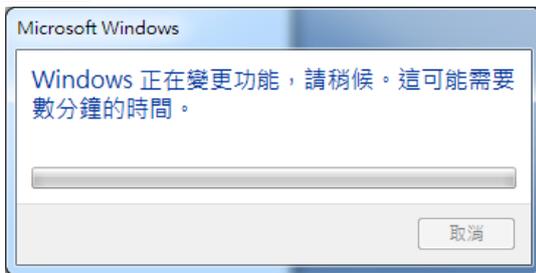


Step.03 Telnet 用戶端 → 設定完畢選擇確定。





Step.04 需要幾分鐘等待變更完畢。



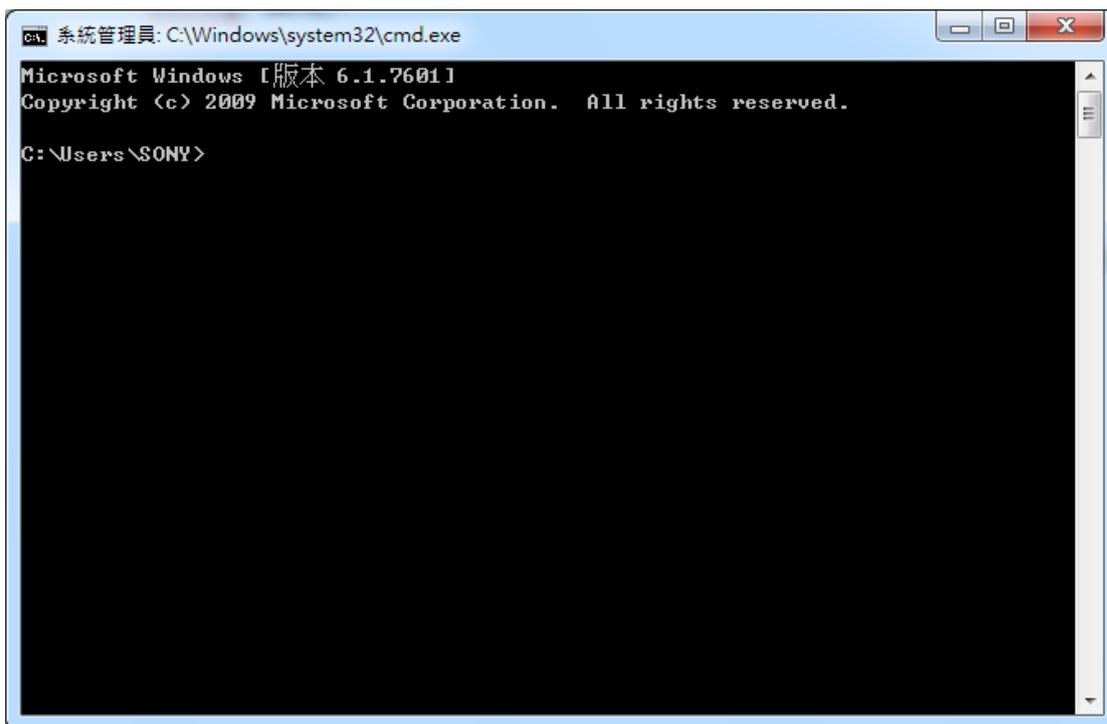
Telnet 登入

➤ **WINDOWS XP 的環境設定**

Step.01a 開始 → 執行 → 輸入 cmd 後確定。

➤ **WINDOWS 7 的環境設定**

Step.01b 開始的最下方-搜尋程式及檔案 → 輸入 cmd 後 Enter → 步驟完成後進入以下畫面。





Step.02 請輸入 ping 192.168.2.127 判斷是否與 M502 可進行通訊，若無法通訊請檢查 ip 設定流程，結束動作時，請按 Ctrl+c。

```
系統管理員: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\SONY>ping 192.168.2.127

Ping 192.168.2.127 <使用 32 位元組的資料>:
回覆自 192.168.2.127: 位元組=32 time<1ms TTL=64

192.168.2.127 的 Ping 統計資料:
    封包: 已傳送 = 4, 已收到 = 4, 已遺失 = 0 (0% 遺失),
    大約的來回時間 <毫秒>:
        最小值 = 0ms , 最大值 = 0ms , 平均 = 0ms

C:\Users\SONY>
```

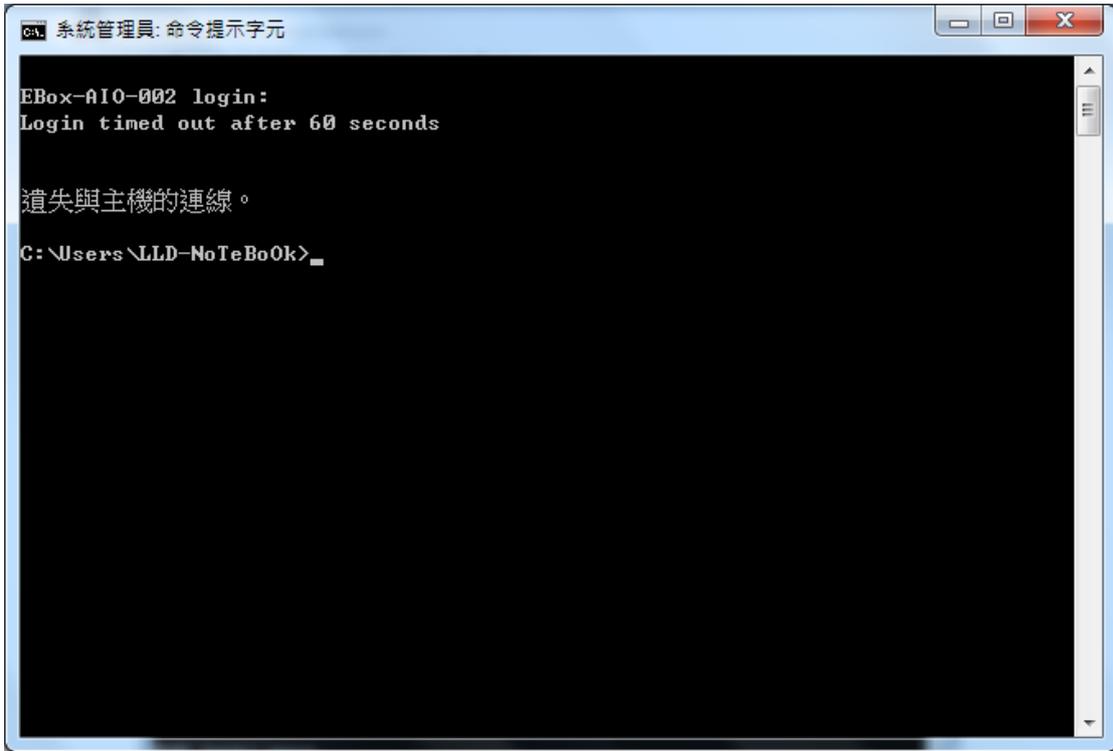
Step.03 請輸入 telnet 192.168.2.127，輸入正確後會看到登入畫面。

```
Telnet 192.168.2.127
EBox-A10-002 login:
```

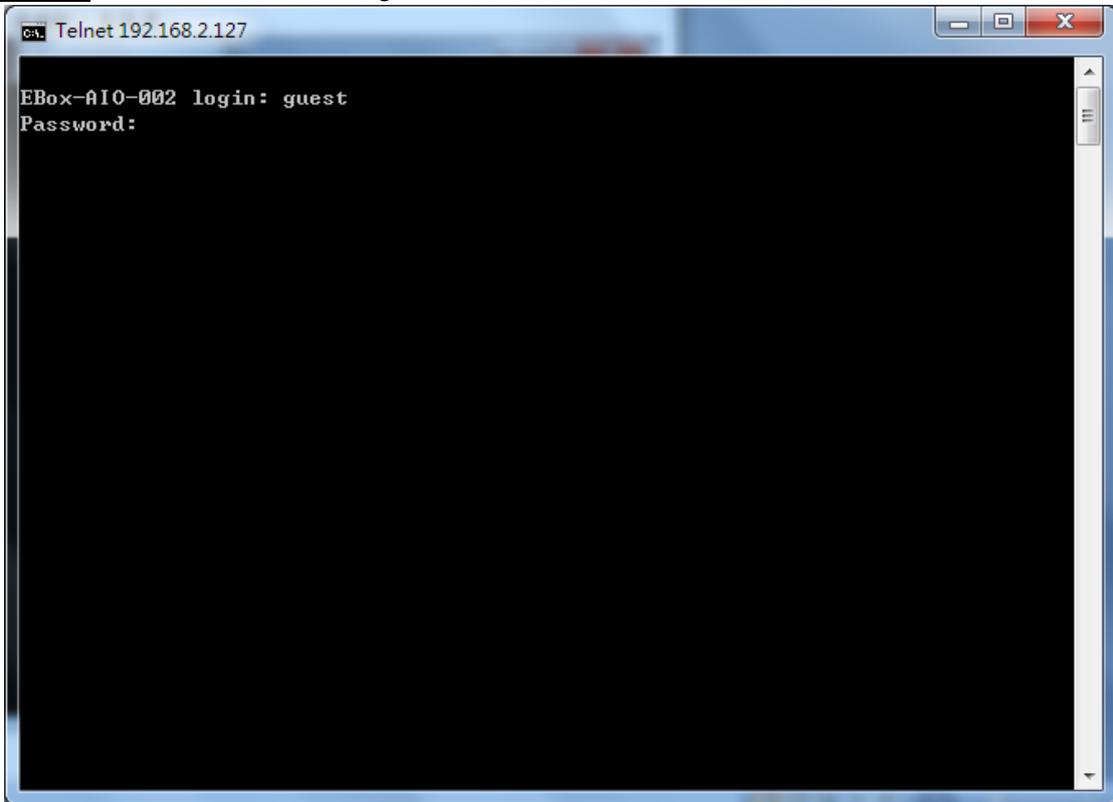




注意：太久未輸入，則會看到以下畫面！

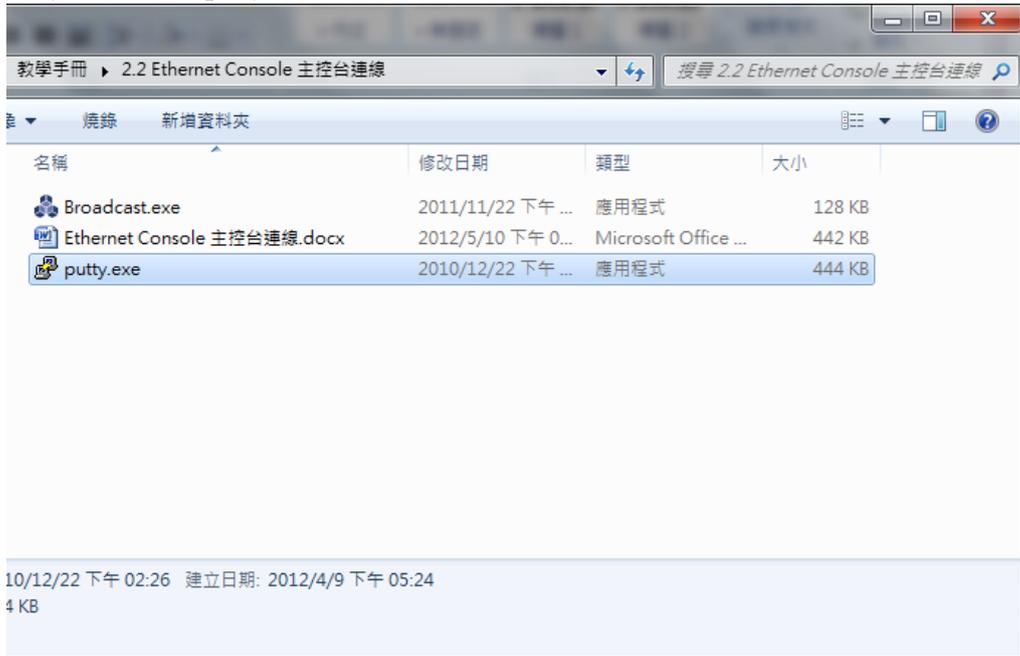


Step.04 請輸入帳號密碼皆為 guest，密碼輸入為隱藏。



C. Putty 操作

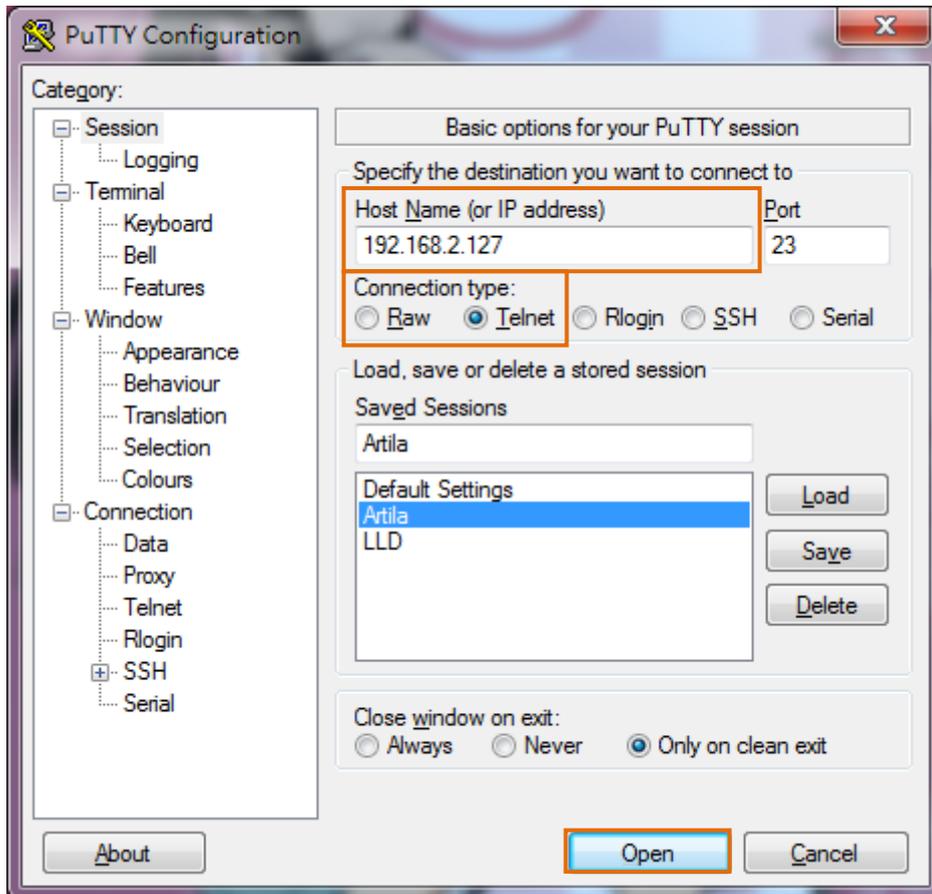
- ◆ Putty 請將附檔 putty.exe 複製到桌面後執行



Step.01 在 Host Name (or IP address)輸入 192.168.2.127

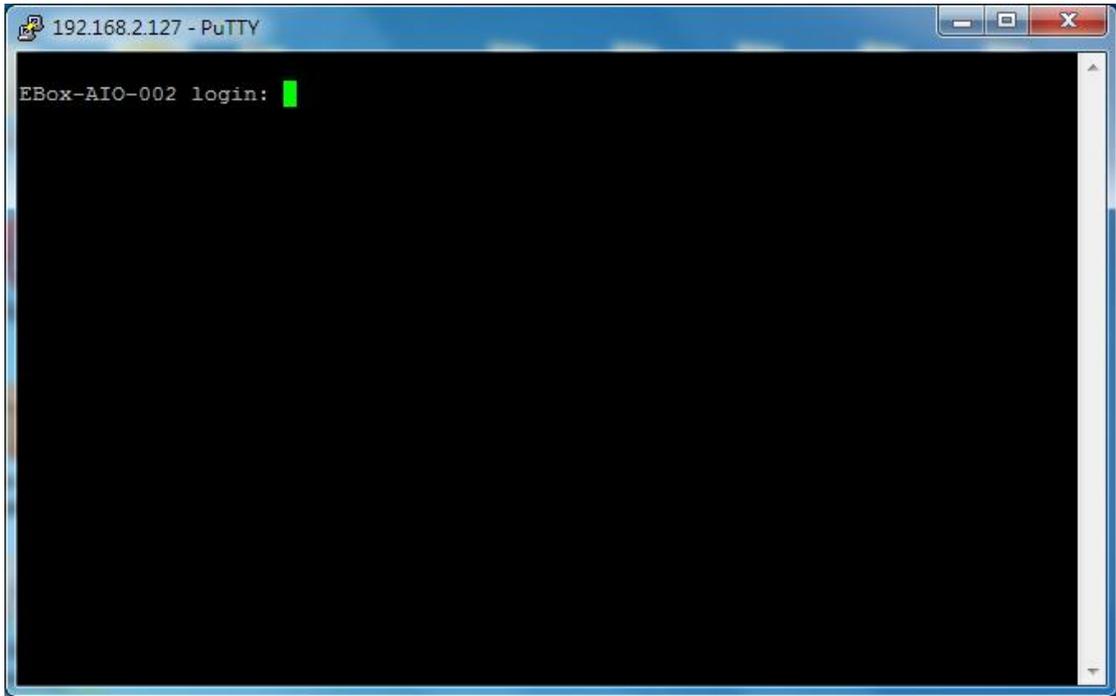
→在 Connection type 選擇 Telnet，Port 會自動變更為 23

→設定完成後按 Open



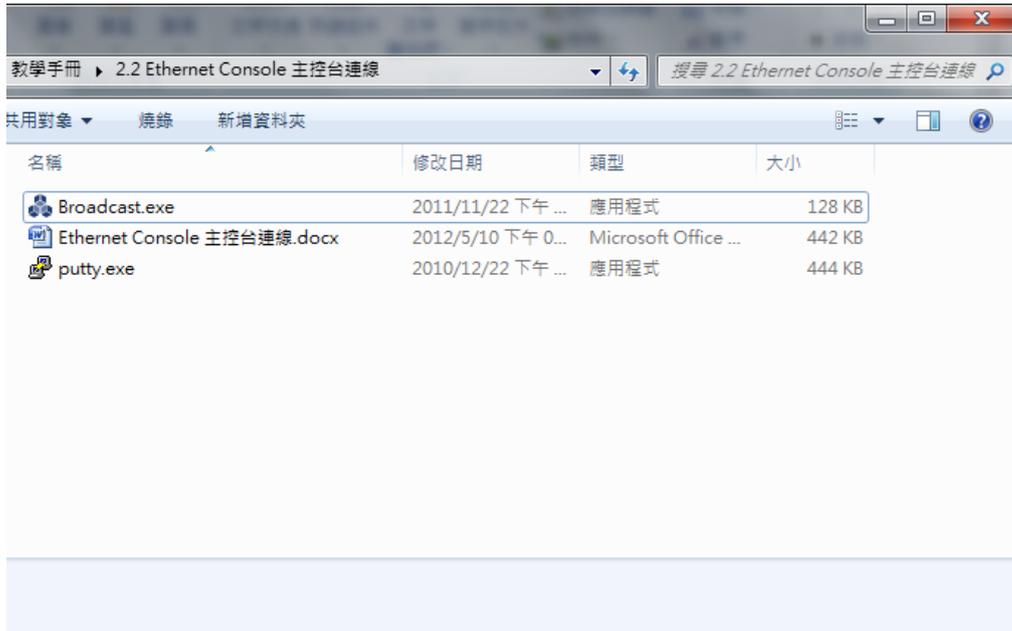


Step.02 登入操作請參考 Telnet 登入 step 4

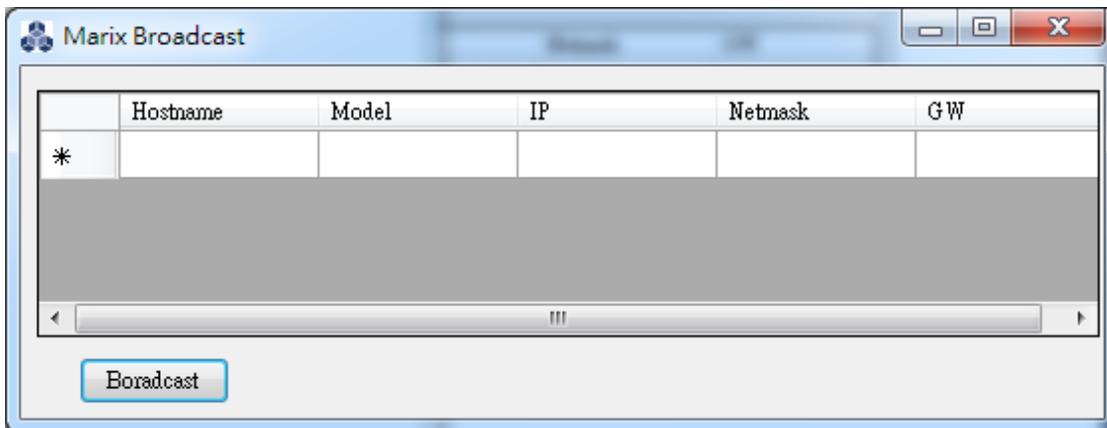


D. Broadcast 工具：Broadcast.exe

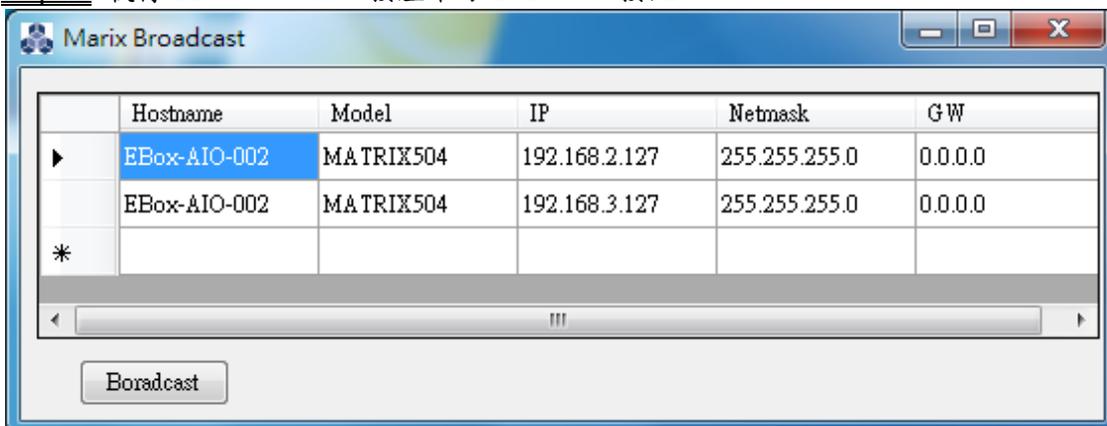
◆ IP search 工具 搜尋 EBox-AIO-002 IP 位置



Step.01 資料夾中附檔-Broadcast.exe，為 Broadcast.exe 需安裝.Net
(window 基本上已安裝)



Step.02 執行 Broadcast.exe，按左下方 Boradcast 按鈕



3.4.RS-232 Console 主控台連線

A. Serial Console 簡介

電腦端透過 RS-232 介面對 EBox-AIO-002 嵌入式系統控制及下命令進行檔案操作。

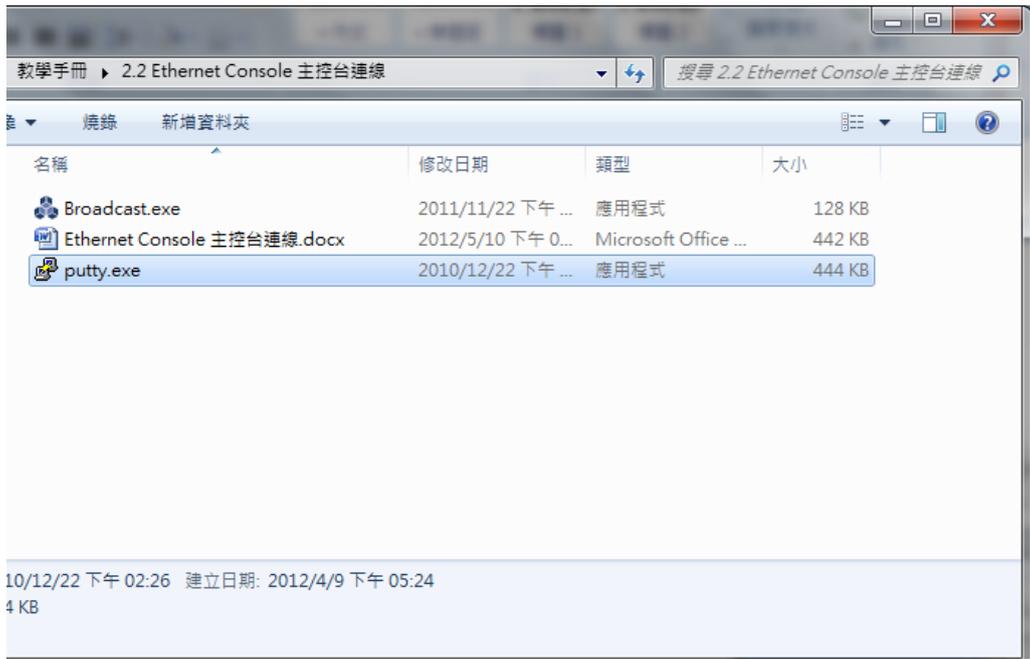
B. Putty 操作

Step.01

將 EBox-AIO-002 上 Console Port / RS-232(母) 與 電腦端 RS-232(公)連接。



◆ Putty 請將附檔 putty.exe 複製到桌面後執行



Step.02

選擇連線種類 (Connection type → 點選 Serial)

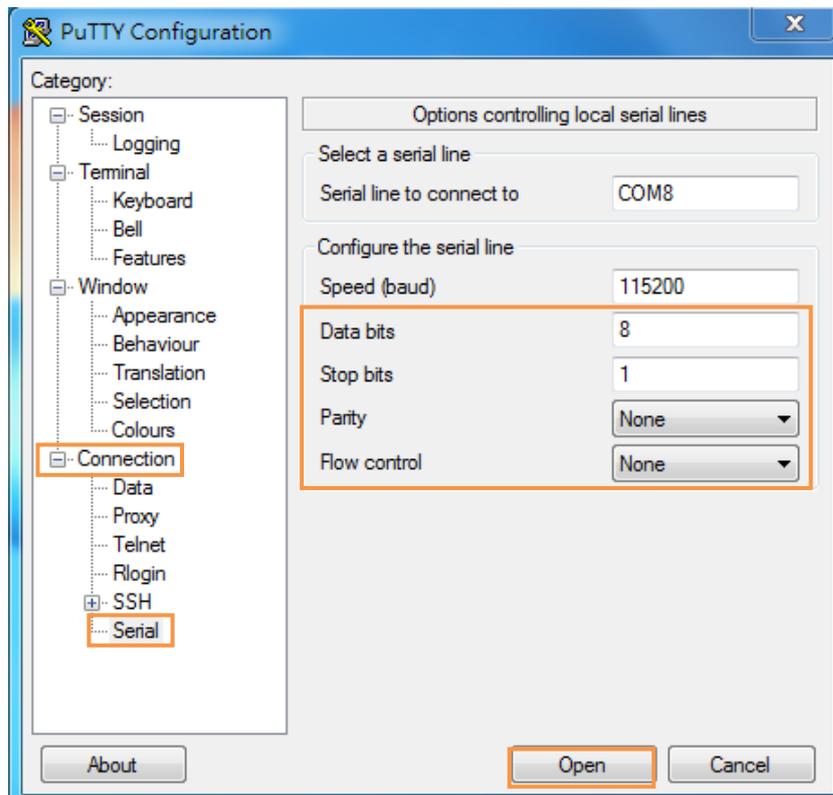
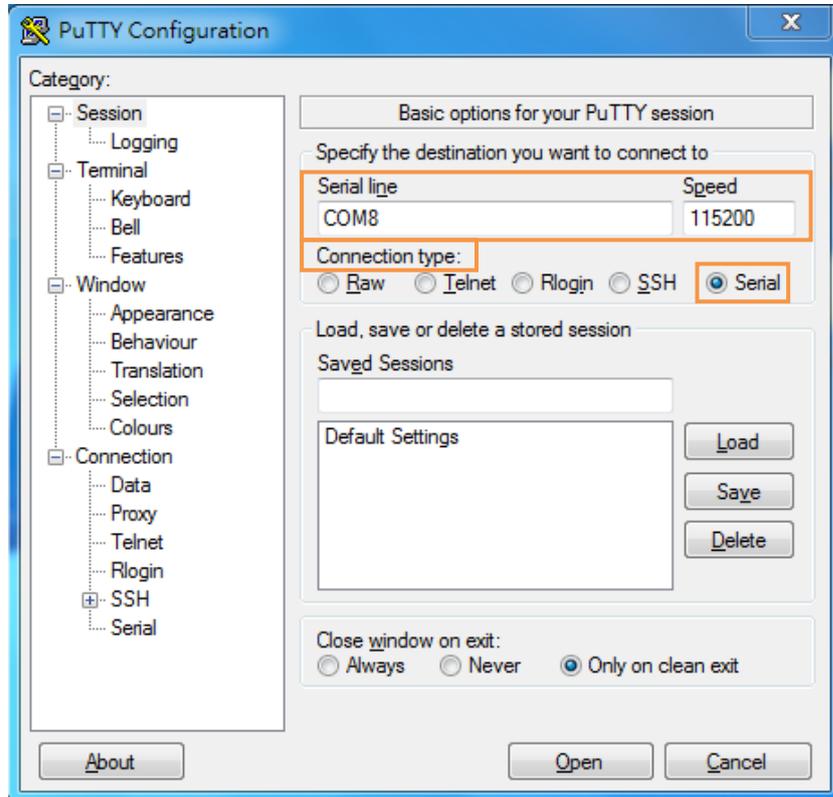
→設定電腦端的 RS-232 Port (Serial line)及 Speed 【Speed (baud)：115200】

【若不知道 Port 請到 ◆電腦管理員 ComPort 查詢，設定完畢點擊確定鍵。】

→設定 Configure the serial line 內容 (左欄 Connection → 點擊 Serial)

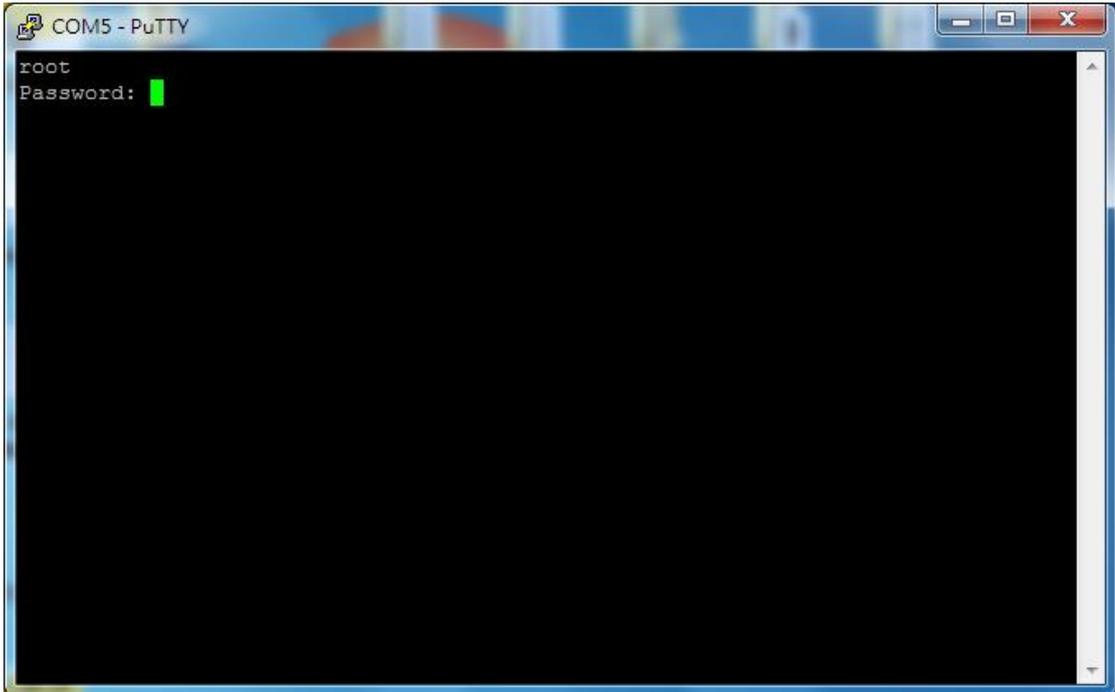
【Data bits：8、Stop bits：1、Parity：None、Flow control：None】

→設定完成後按 Open



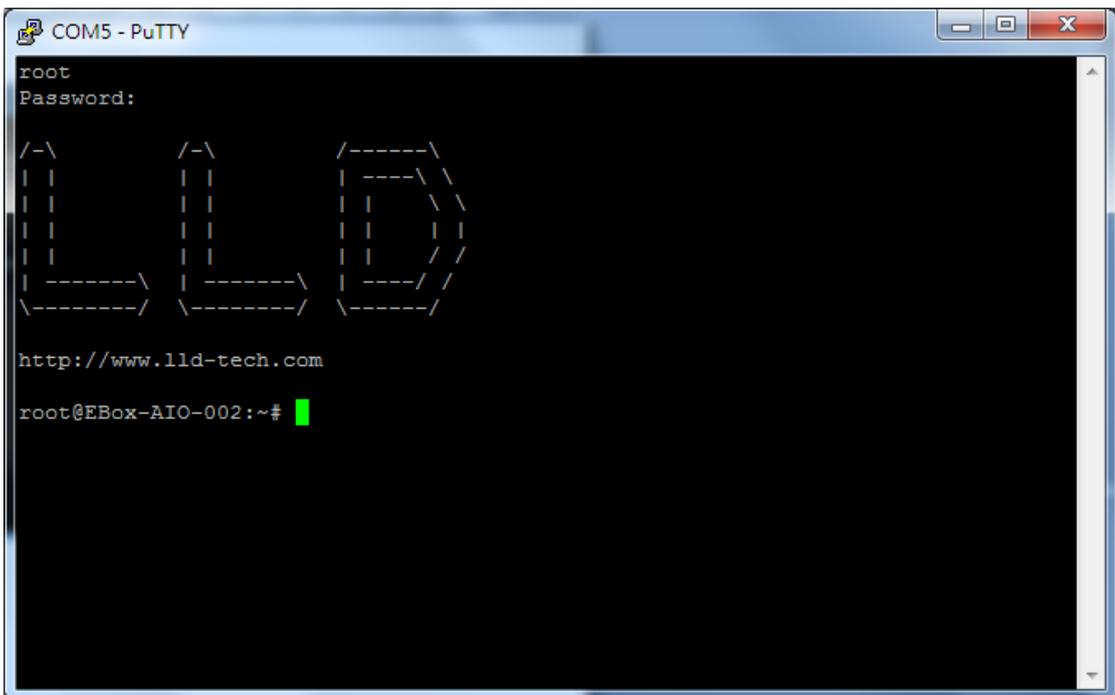
Step.03

帳號密碼預設皆為 root，但密碼的字為隱藏，不會顯示。



Step.04

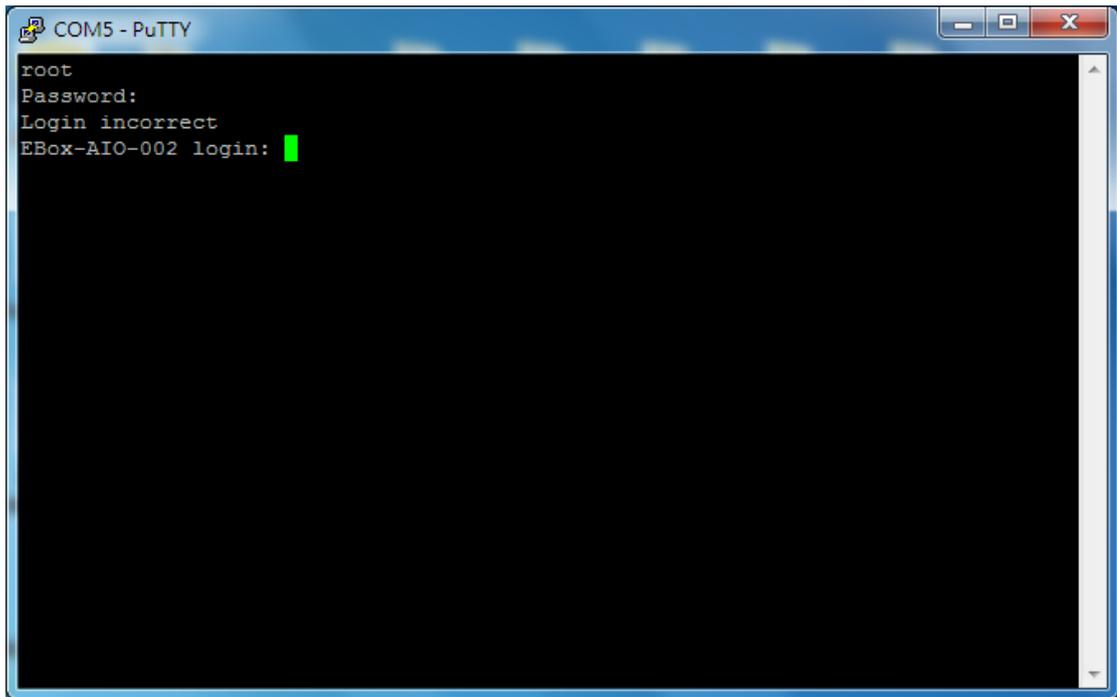
登入完成。





Step.05

如輸入有誤，需再輸入帳號與密碼。





◆ 電腦管理員 Comport 查詢

STEP.01 在開始中的電腦按右鍵，選擇內容

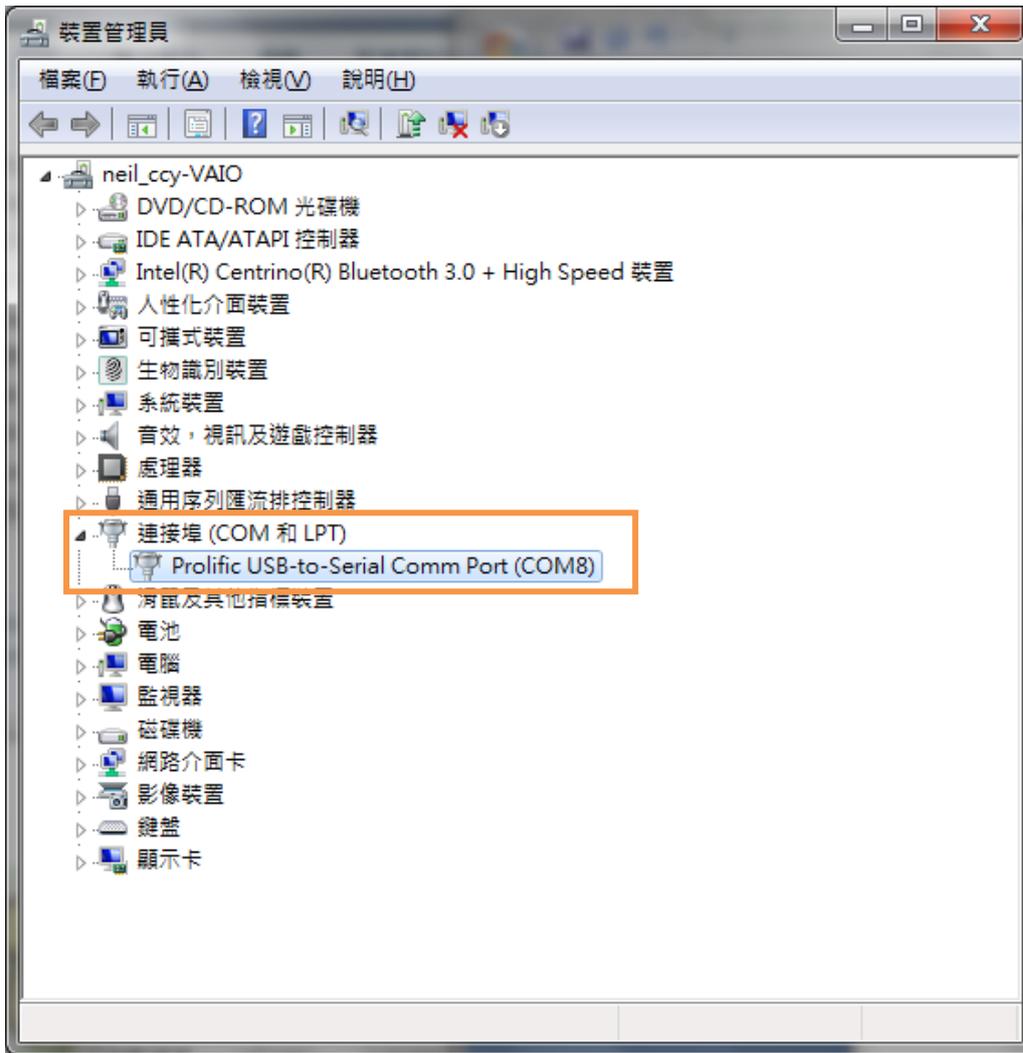


STEP.02 左邊的部分選擇『裝置管理員』

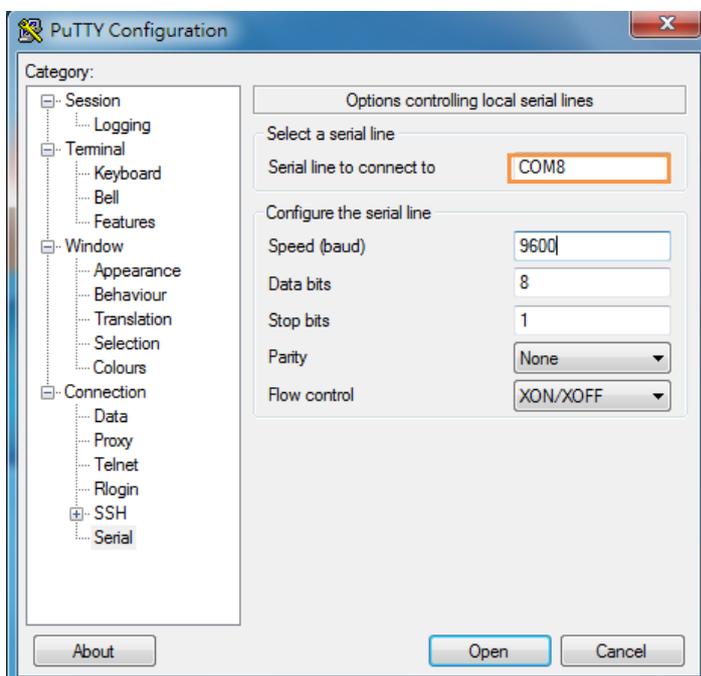




STEP.03 在選項 連接埠(COM 和 LPT)中，找到可使用的 Comm Port



STEP.04 使用連線(N)要選擇可使用的 Port。



4. 安裝開發環境

4.1.環境需求

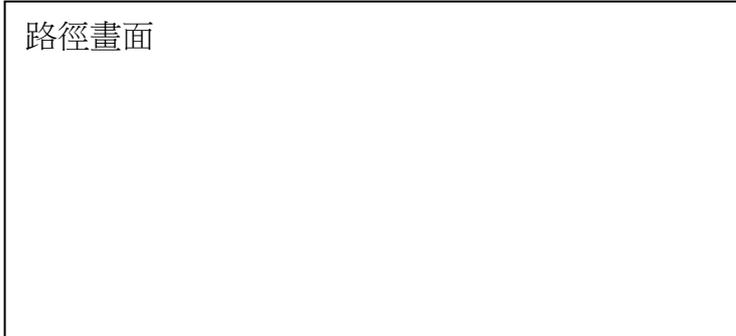
Linux：Fedora 7, ubuntu 7.04, OpenSUSE 10.2, Mandriva 2008, Debian 5.0, Centos(RedHat) 5

Windows：Windows XP、Windows 7

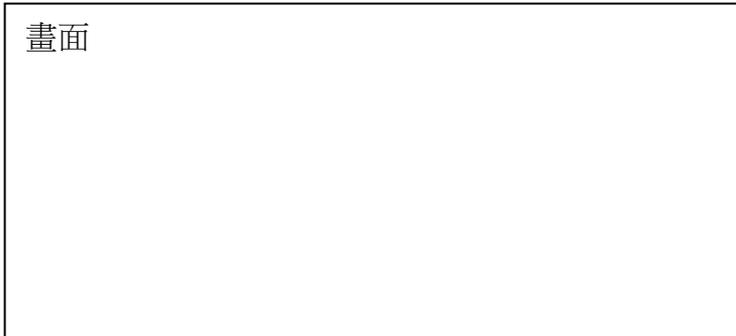
4.2.安裝 Linux Tool Chain

A. 取得 Tool Chain(Cross Compiler)

- ◆ 從隨貨 CD 取得



- ◆ 網路下載



B. 安裝 GNU Tool Chain

安裝步驟：

- ◆ 以 root 使用者身分登入



- ◆ 複製 arm-linux-4.3.2.tar.gz 到根目錄底下





◆ 輸入解壓縮指令安裝 Tool Chain

```
#tar -xvfj arm-linux-4.3.3.tar.bz2
```

解壓縮畫面

Tool chain 的檔案名稱為

arm-linux-gnueabi-gcc

arm-linux-gnueabi-g++

arm-linux-gnueabi-strip

版本: gcc 4.3.3, glibc 2.9, binutils 2.18

三個指令的執行畫面

4.3.安裝 Windows Tool Chain

A. 取得 SW

Windows的使用者，請先至<http://www.codesourcery.com>網站

(<http://www.codesourcery.com/sgpp/lite/arm/portal/package4547/public/arm-none-linux-gnueabi/arm-2009q1-203-arm-none-linux-gnueabi.exe>) 下載tool chain

B. Tool Chain

直接於執行下載的 Tool Chain 進行安裝

畫面

Tool chain 的檔案名稱為

arm-linux-gnueabi-gcc

arm-linux-gnueabi-g++

arm-linux-gnueabi-strip

版本: Version: gcc 4.3.3, glibc 2.8, binutils 2.19



三個指令的執行畫面

5. 執行第一隻程式 – Hello (以 Linux 開發環境為例)

5.1. 原程式碼

可透過工具程式”vi”，編輯一個 hello.c 的檔案或自隨貨光碟中 copy 到 Linux 主機

- ◆ hello.c 參考原始碼

```
<include stdio.h >
int main()
{
    printf (hello.c \n”);
}
```

5.2. 產生執行檔

- ◆ 透過指令 arm-linux-gnueabi-gcc (針對.c 檔)

畫面

- ◆ 透過 make 指令

Make 檔內容及執行結果畫面

5.3. 上傳到 EBox-AIO-002

透過 ftp 檔案傳輸協議，將執行檔 a.out 或 hello 傳到送 EBox-AIO-002 中

- ◆ ftp 指令
[ftp 192.168.2.127](http://192.168.2.127) (192.168.2.127 為 EBox-AIO-002 出廠 IP address)
出廠帳號及密碼：
Username : root
Password : root

[ftp 192.168.2.127](http://192.168.2.127) 畫面

login
change mode
ftp

Note：變更檔案傳輸模式為二進制(bin)

5.4.測試執行程式

參考第 3 章內容，登入到 EBox-AIO-002 Console，

◆ 轉換格式

透過命令 `chmod` (將檔案變更為可執行屬性)

```
chmod +x 畫面
```

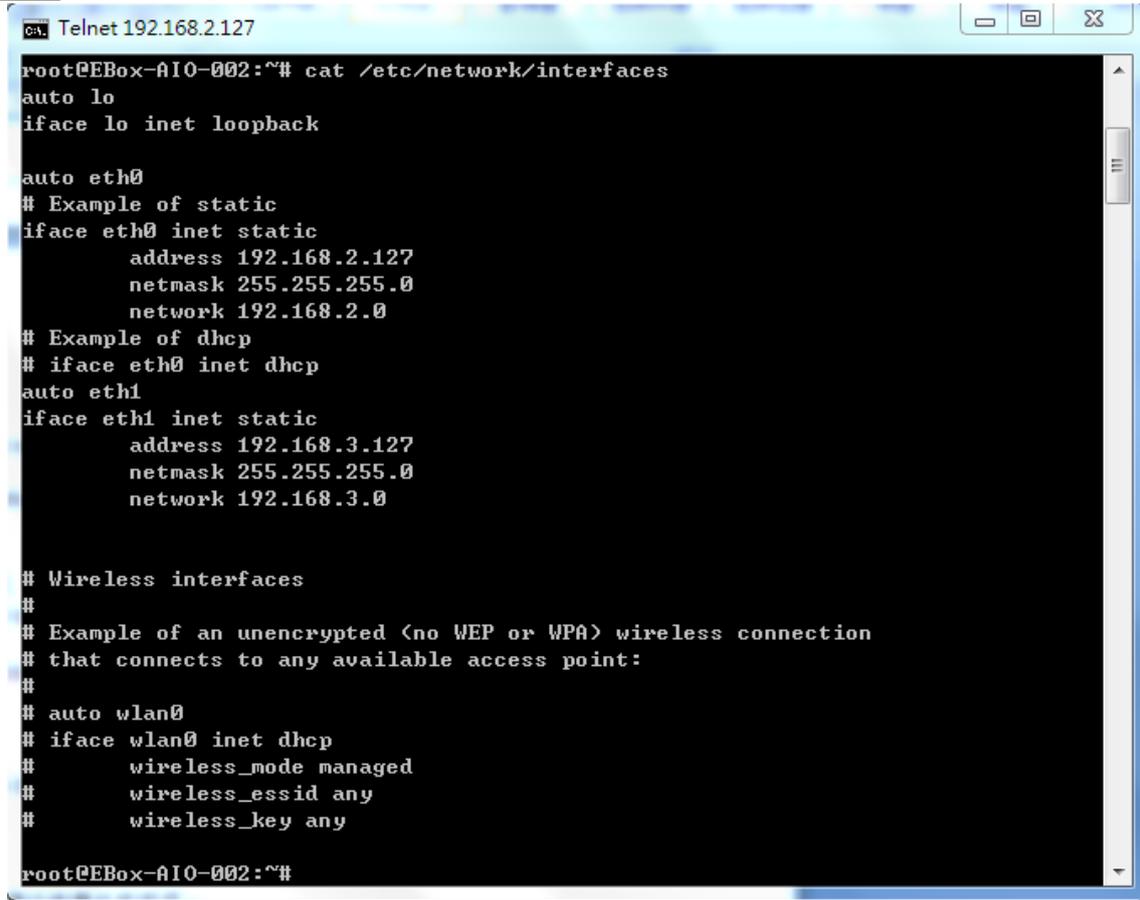
◆ 執行程式

```
執行畫面
```

6. 常用基本操作

6.1. 變更網路設定

STEP.01 輸入指令【`cat /etc/network/interfaces`】，顯示畫面。



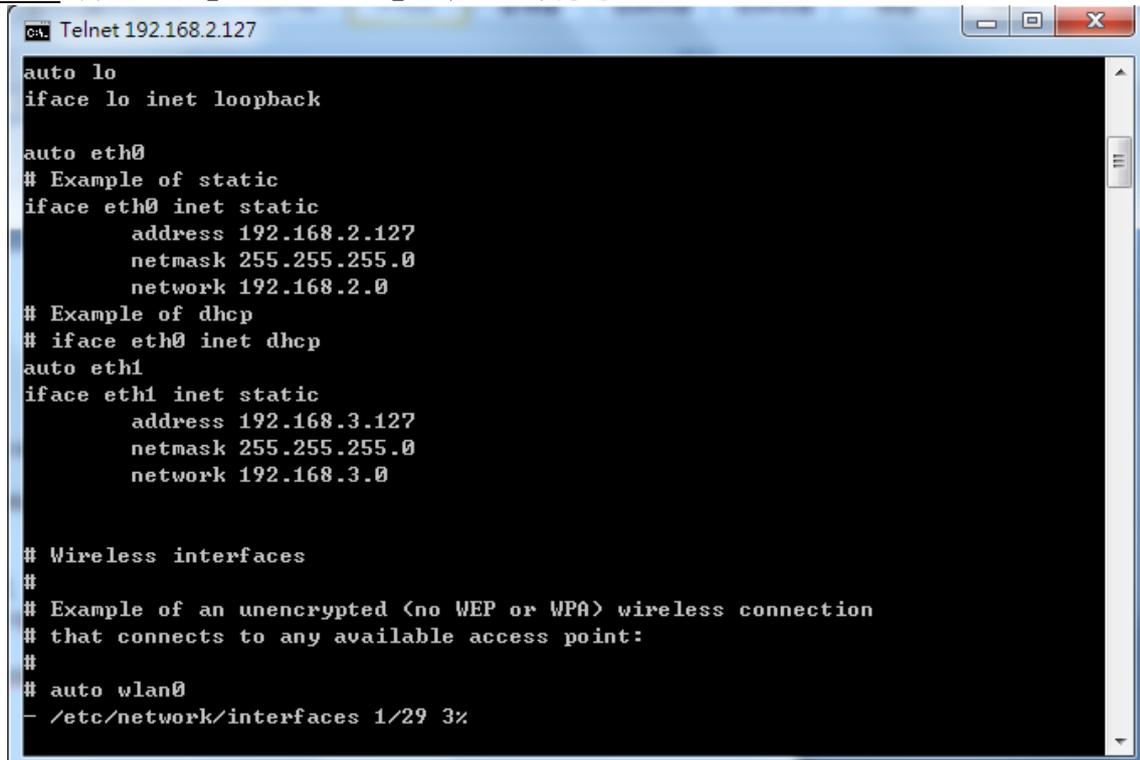
```
Telnet 192.168.2.127
root@EBox-A10-002:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
# Example of static
iface eth0 inet static
    address 192.168.2.127
    netmask 255.255.255.0
    network 192.168.2.0
# Example of dhcp
# iface eth0 inet dhcp
auto eth1
iface eth1 inet static
    address 192.168.3.127
    netmask 255.255.255.0
    network 192.168.3.0

# Wireless interfaces
#
# Example of an unencrypted (no WEP or WPA) wireless connection
# that connects to any available access point:
#
# auto wlan0
# iface wlan0 inet dhcp
#     wireless_mode managed
#     wireless_essid any
#     wireless_key any

root@EBox-A10-002:~#
```

STEP.02 輸入指令【`vi interfaces`】，即可進行變更設定



```
Telnet 192.168.2.127
auto lo
iface lo inet loopback

auto eth0
# Example of static
iface eth0 inet static
    address 192.168.2.127
    netmask 255.255.255.0
    network 192.168.2.0
# Example of dhcp
# iface eth0 inet dhcp
auto eth1
iface eth1 inet static
    address 192.168.3.127
    netmask 255.255.255.0
    network 192.168.3.0

# Wireless interfaces
#
# Example of an unencrypted (no WEP or WPA) wireless connection
# that connects to any available access point:
#
# auto wlan0
- /etc/network/interfaces 1/29 3%
```



6.2. 初始化開機設定及應用程式

STEP.01 輸入指令【cat /etc/rc.local】，顯示畫面。

```
root@EBox-A10-002:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

setuart -p1 -t485 -b19200 &
setuart -p4 -t485 -b19200 &
setuart -p2 -t232 -b19200 &

gpiocctl -i0 -m1 &
gpiocctl -i1 -m1 &
gpiocctl -i2 -m1 &
gpiocctl -i3 -m0 &
gpiocctl -i4 -m0 &
gpiocctl -i5 -m0 &
gpiocctl -i6 -m1 &
gpiocctl -i7 -m1 &
gpiocctl -i8 -m1 &
gpiocctl -i9 -m1 &

#mount /dev/mmc0
#/home/root/mysql_start.sh &

#mysqld &
#sleep 3

exit 0
root@EBox-A10-002:~#
```

STEP.02 輸入指令【vi rc.local】，即可變更設定內容

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

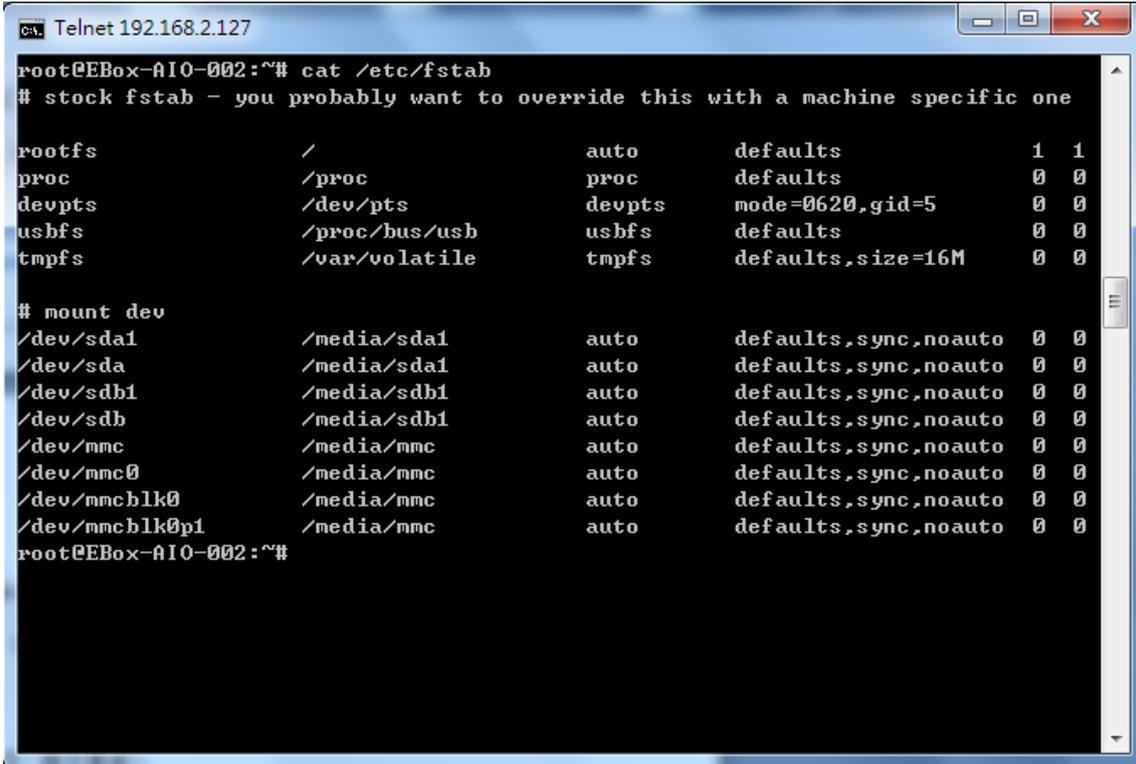
setuart -p1 -t485 -b19200 &
setuart -p4 -t485 -b19200 &
setuart -p2 -t232 -b19200 &

gpiocctl -i0 -m1 &
gpiocctl -i1 -m1 &
gpiocctl -i2 -m1 &
gpiocctl -i3 -m0 &
gpiocctl -i4 -m0 &
gpiocctl -i5 -m0 &
gpiocctl -i6 -m1 &
- /etc/rc.local 1/35 2%
```



6.3.外掛 USB 及 SD 記憶體

STEP.01 輸入指令【cat /etc/fstab】，顯示畫面。



```
ca. Telnet 192.168.2.127
root@EBox-A10-002:~# cat /etc/fstab
# stock fstab - you probably want to override this with a machine specific one

rootfs          /                    auto          defaults      1 1
proc            /proc               proc          defaults      0 0
devpts          /dev/pts            devpts        mode=0620,gid=5 0 0
usbfs           /proc/bus/usb       usbfs         defaults      0 0
tmpfs           /var/volatile       tmpfs         defaults,size=16M 0 0

# mount dev
/dev/sda1       /media/sda1         auto          defaults, sync, noauto 0 0
/dev/sda        /media/sda1         auto          defaults, sync, noauto 0 0
/dev/sdb1       /media/sdb1         auto          defaults, sync, noauto 0 0
/dev/sdb        /media/sdb1         auto          defaults, sync, noauto 0 0
/dev/mmc        /media/mmc          auto          defaults, sync, noauto 0 0
/dev/mmc0       /media/mmc          auto          defaults, sync, noauto 0 0
/dev/mmcblk0    /media/mmc          auto          defaults, sync, noauto 0 0
/dev/mmcblk0p1  /media/mmc          auto          defaults, sync, noauto 0 0
root@EBox-A10-002:~#
```

6.4.變更系統時間

STEP.01 輸入指令【date】，顯示軟體目前時間。



```
ca. Telnet 192.168.2.127
root@EBox-A10-002:~# date
Sat May 2 01:17:13 GST 2015
```

STEP.02 輸入指令【date MMDDhhmmYYYY】，即可變更軟體時間。

(MM =月(01~12)、DD =日(01~31)、hh =時、mm =分、YYYY =年)



```
ca. Telnet 192.168.2.127
root@EBox-A10-002:~# date 060516372015
Fri Jun 5 16:37:00 GST 2015
```



6.5.Web Server 基本設定

STEP.01 輸入指令【`cat /usr/www/index.html`】，顯示 index.html 內容。

```
Telnet 192.168.2.127
root@EBox-A10-002:~# cat /usr/www/index.html
<!doctype html>
<html lang="us">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"
/>
  <title>EBox-CGate-Modbus</title>
  <link rel="stylesheet" href="style/login.css">
  <script type="text/javascript" src="scripts/jquery-1.10.2.min.js"></scri
pt>
  <script type="text/javascript" src="scripts/jquery-ui.min.js"></script>
  <script type="text/javascript" src="scripts/myIndexJqy.js"></script>
  <script>
    $(function() {
      $(".button").hover(function() {
        $(this).attr("src", "images/ebox_loginbt_2.png");
      }, function() {
        $(this).attr("src", "images/ebox_loginbt_1.png");
      });
      function login_key(e) {
        if(e.keyCode == 13) {
          this.form.submit();
        }
      }
      $.fn.lang_init();
    });
  </script>
</head>
<body>
<form action="login.php" method="post">
<div align="center" style="margin-top:40px;">
  
  <div class="login" >
    <div class="lg_cnt" id="lgin">
```

STEP.02 輸入指令【`vi /usr/www/index.html`】，即可變更設定內容

```
Telnet 192.168.2.127
<!doctype html>^M
<html lang="us">^M
<head>^M
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"
  <title>EBox-CGate-Modbus</title>^M
  <link rel="stylesheet" href="style/login.css">^M
  <script type="text/javascript" src="scripts/jquery-1.10.2.min.js"></scri
  <script type="text/javascript" src="scripts/jquery-ui.min.js"></script>^M
  <script type="text/javascript" src="scripts/myIndexJqy.js"></script>^M
  <script>^M
    $(function() {^M
      $(".button").hover(function() {^M
        $(this).attr("src", "images/ebox_loginbt_2.png");^M
      }, function() {^M
        $(this).attr("src", "images/ebox_loginbt_1.png");^M
      });^M
      function login_key(e) {^M
        if(e.keyCode == 13) {^M
          this.form.submit();^M
        }^M
      }^M
      $.fn.lang_init();^M
    });^M
  </script>^M
- /usr/www/index.html 1/45 2%
```



STEP.03 輸入指令【`cat /etc/lighttpd.conf`】，顯示 `lighttpd.conf` 內容。

```
Telnet 192.168.2.127
# cml.memcache-hosts          = < "127.0.0.1:11211" >

##### variable usage:
## variable name without "." is auto prefixed by "var." and becomes "var.bar"
#bar = 1
#var.mystring = "foo"

## integer add
#bar += 1
## string concat, with integer cast as string, result: "www.foo1.com"
#server.name = "www." + mystring + var.bar + ".com"
## array merge
#index-file.names = <foo + ".php"> + index-file.names
#index-file.names += <foo + ".php">

##### include
#include /etc/lighttpd/lighttpd-inc.conf
## same as above if you run: "lighttpd -f /etc/lighttpd/lighttpd.conf"
#include "lighttpd-inc.conf"

##### include_shell
#include_shell "echo var.a=1"
## the above is same as:
#var.a=1
root@EBox-A10-002:~#
```

STEP.04 輸入指令【`vi /etc/lighttpd.conf`】，即可變更設定內容

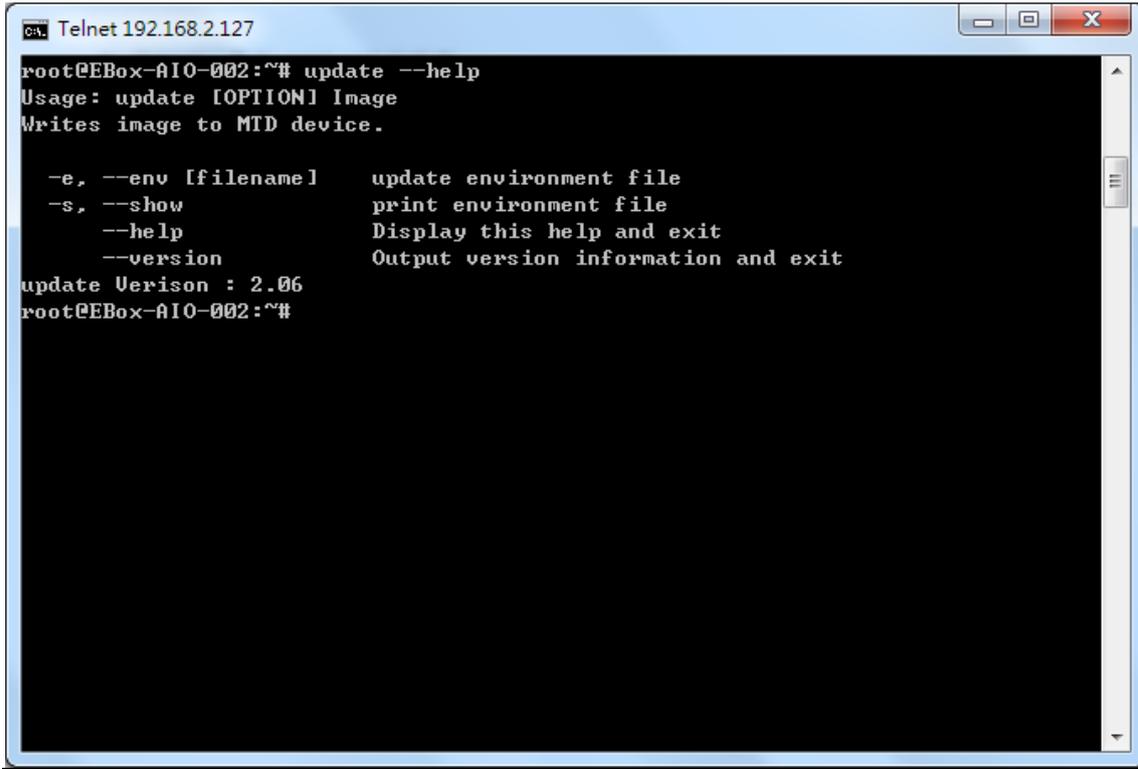
```
Telnet 192.168.2.127
# lighttpd configuration file
#
# use it as a base for lighttpd 1.0.0 and above
#
# $Id: lighttpd.conf,v 1.7 2004/11/03 22:26:05 weigon Exp $

##### Options you really have to take care of #####

## modules to load
# at least mod_access and mod_accesslog should be loaded
# all other module should only be loaded if really necessary
# - saves some time
# - saves memory
server.modules          = <
#
#         "mod_rewrite",
#         "mod_redirect",
#         "mod_alias",
#         "mod_access",
#         "mod_cml",
#         "mod_trigger_b4_dl",
#         "mod_auth",
#         "mod_status",
#         "mod_setenv",
#         "mod_fastcgi",
- /etc/lighttpd.conf 1/341 0%
```


6.7.LLD 專用工具程式

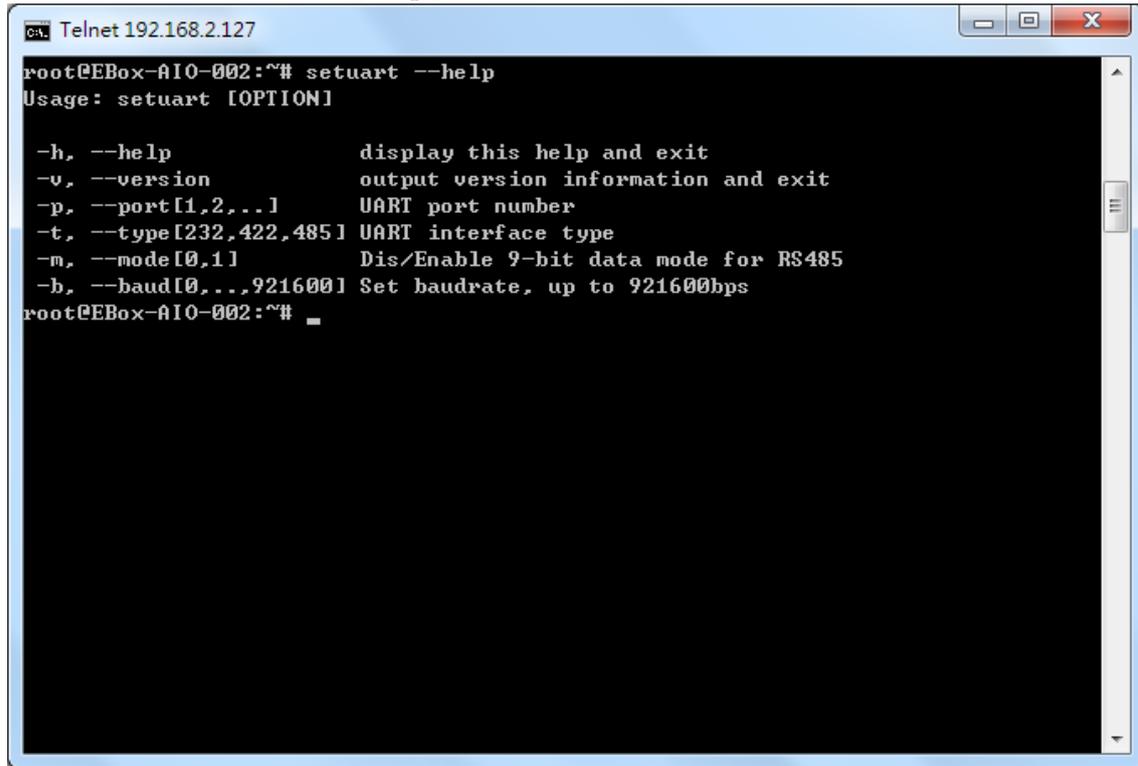
- A. Update：更新 loader、environment file、kernel image。
 - 輸入指令【update --help】，顯示其他相關的指令。



```
ca. Telnet 192.168.2.127
root@EBox-AIO-002:~# update --help
Usage: update [OPTION] Image
Writes image to MTD device.

-e, --env [filename]  update environment file
-s, --show            print environment file
--help              Display this help and exit
--version           Output version information and exit
update Verison : 2.06
root@EBox-AIO-002:~#
```

- B. Setuart：更改 serial port 的設定。
 - 輸入指令【setuart --help】，顯示其他相關的指令。



```
ca. Telnet 192.168.2.127
root@EBox-AIO-002:~# setuart --help
Usage: setuart [OPTION]

-h, --help          display this help and exit
-v, --version       output version information and exit
-p, --port[1,2,..] UART port number
-t, --type[232,422,485] UART interface type
-m, --mode[0,1]    Dis/Enable 9-bit data mode for RS485
-b, --baud[0,..,921600] Set baudrate, up to 921600bps
root@EBox-AIO-002:~#
```



C. Version：查看作業系統的版本

```
Telnet 192.168.2.127
root@EBox-AIO-002:~# version
EBox-AIO-002 Firmware Verison.<Linux 2.6.29.4>
Loader      : 2.0.13-64M
Kernel      : build #25 PREEMPT Wed Oct 29 15:42:49 CST 2014
Filesystem  : build #121 PREEMPT Fri Nov 12 14:33:56 CST 2010
root@EBox-AIO-002:~#
```

D. Gpioctl：更改 GPIO

- 輸入指令【`gpioctl --help`】，顯示其他相關的指令。

```
Telnet 192.168.2.127
root@EBox-AIO-002:~# gpioctl --help
Usage: gpioctl [OPTION]

-h, --help            display this help and exit
-v, --version          output version information and exit
-i, --io[0,1,2,..]   GPIO number
-s, --state[0,1]     GPIO state, 1:HIGH, 0:LOW
-m, --mode[0,1]      GPIO mode, 1:INPUT , 0:OUTPUT
-a, --all             Show all GPIO state and mode
root@EBox-AIO-002:~#
```