

EBox-AIO-002

Compact Multi-I/O Control Computer

User Manual

Ver. : V1.02

Table of Contents

1.	Product Introduction	1
1.1.	Product Description	1
1.2.	Introduction to the appearance.....	1
2.	Product Specifications	2
2.1.	Hardware specifications.....	2
2.2.	LED Indicator description	4
2.3.	Pin definition of each communication and control interface	5
2.4.	Software specifications	6
2.5.	Embedded-Linux File System	6
2.6.	Software definition of each communication and control interface	7
3.	Console	8
3.1.	Factory default setting	8
3.2.	Test environment recommendations	8
3.3.	Ethernet console connection	11
3.4.	RS-232 console connection	25
4.	Install the Development Environment	31
4.1.	Environmental requirements.....	31
4.2.	Install Linux Tool Chain	31
4.3.	Install Windows Tool Chain	32
5.	Execute the First Program – Hello (By Linux development environment as an example)	34
5.1.	Source code.....	34
5.2.	Produce execution file	34
5.3.	Upload to EBox-AIO-002.....	34
5.4.	Test the execution program.....	35
6.	Commonly Basic Operation	36
6.1.	Change the network settings	36
6.2.	Initialize the boot settings and application	37
6.3.	Plugin USB and SD memory	38
6.4.	Change the system time	38
6.5.	Web server basic setting	39
6.6.	Change console greeting.....	41
6.7.	LLD dedicated tool for service	42

EBox-AIO-002 Compact Multi-I/O Control Computer

1. Product Introduction

1.1. Product Description

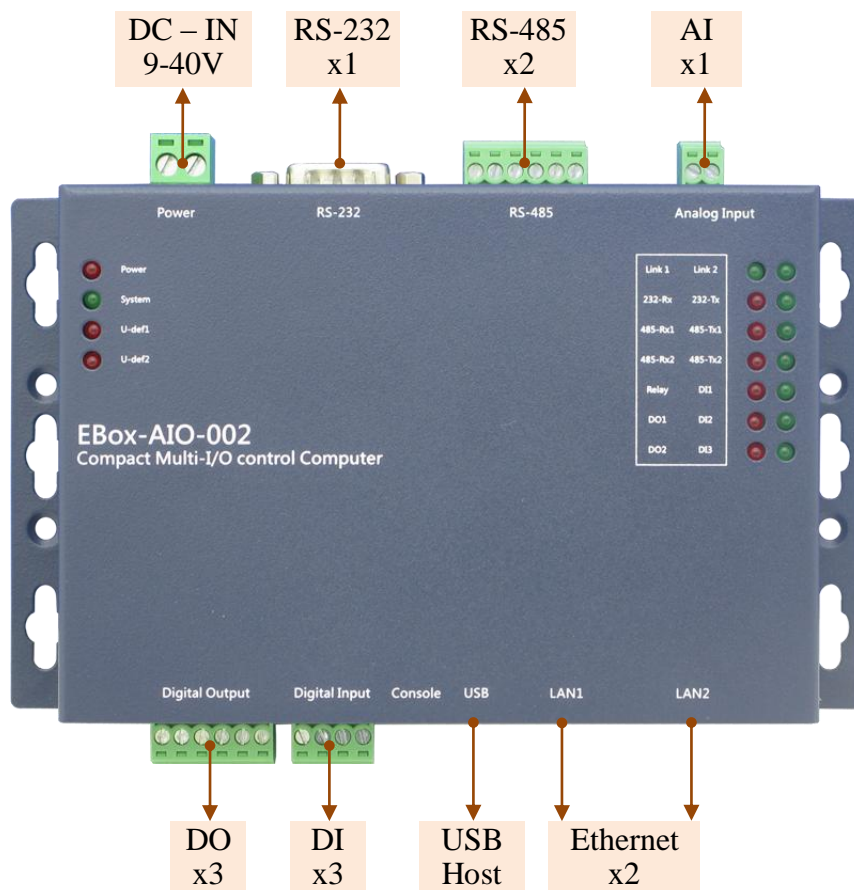
EBox-AIO-002 is a network type signal controller of ARM-based embedded architecture. It can be applied to real-time monitoring of the site equipment or instruments, with the SQL data base for data collection. And through the TCP/IP network interface and remote monitoring host connection, to achieve the application of remote monitoring.

EBox-AIO-002 with built-in Linux operating system is an open platform. User can through open GNU software development tools to install free embedded ARM-Linux operating system's C/C++ program compiler and Lib to provide user a tool for the secondary system development to let EBox-AIO-002 become a dedicated controller and gateway.

EBox-AIO-002 contains multifunctional communication and control function. It is featured Ethernet LAN and WAN port to connect with Internet and backend server system as well as possess multiple serial communication interfaces (e.g. RS-232 and RS-485). So that EBox-AIO-002 can connect more different kinds of equipment or instruments to fully play a good device monitoring or data processing role. Simultaneously, EBox-AIO-002 provides Digital I/O function which can perform on-site switching control without additional external modules. In addition, there is one channel Analog Input interface to execute relevant data collection by connecting to corresponding sensor, such as temperature, humidity, pressure, wind speed and other common sensors.

Fanless, industrial grade design and low power consumption are applicable to all kinds of long-term uninterrupted monitoring applications. Metal case design can strengthen the box and be installed by wall mounting or in DIN rail. In order to let installer do wiring construction more convenience, it carries lots of common connector (ex: RJ45、DB9、Terminal Block etc.).

1.2 Introduction to the appearance



2. Product Specifications

2.1. Hardware specifications

System Core

- ▶ CPU : ATMEL AT9G20 400MHz ARM9 @400MHz
- ▶ Memory : 64MB SDRAM, 128MB NAND FLASH

Network Interface

- ▶ Quantity : 2
- ▶ Type : 10/100BaseT Ethernet
- ▶ Connector : RJ45

Analog Input

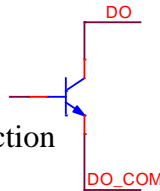
- ▶ Quantity : 1
- ▶ Signal Type : 0~20mA or 0-10VDC (by switch)
- ▶ Resolution : 12-bit
- ▶ Frequency : 10Hz
- ▶ Connector : 3.5mm Terminal Block

Relay Output

- ▶ Quantity : 1
- ▶ Signal Type : SPDT Relay , N.O./ N.C./ COM
- ▶ Input voltage : 1A@120VAC / 1A@24VDC
- ▶ Protection : 2000Vrms optically isolated protection
- ▶ Connector : 3.5mm Terminal Block

Digital Output

- ▶ Quantity : 2
- ▶ Signal Type : Open Collector
- ▶ Driving ability : 200mA
- ▶ Protection : 2000Vrms optically isolated protection
- ▶ Connector : 3.5mm Terminal Block



Digital Input

- ▶ Quantity : 3
- ▶ Input voltage : 5~24VDC
- ▶ Signal Type : Sink mode
- ▶ Protection : 2000Vrms optically isolated protection
- ▶ Connector : 3.5mm Terminal Block

RS-232 port

- ▶ Quantity : 1
- ▶ RS-232 Signal : Tx, Rx, RTS, CTS, DTR, DSR, DCD, GND
- ▶ Protection : 15KV ESD Static , 400W Surge Protection
- ▶ Connector : DB9 Male

RS-485 port

- ▶ Quantity : 2
- ▶ RS-485 Signal : Data+, Data-, GND (support Auto Data Direction Control)
- ▶ Multi-Drop Nodes : 400
- ▶ Built-in Terminal Resistor : 120Ω , Setting by Jumper (Need to open the case)
- ▶ Protection : 2KV isolated protection, 15KV ESD Static, 400W Surge Protection
- ▶ Connector : 3.5mm Terminal Block

Serial Port Parameter

- ▶ 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 Interface

- ▷ Quantity : 1
- ▷ Type : USB2.0 Compatible
- ▷ Connector : Single (Type A)

SD Expansion Interface

- ▷ Quantity : 1 (Need to open the case)
- ▷ Connector : Micro SD Slot

Console

- ▷ Quantity : 1
- ▷ Signal : RS-232 (TxD, RxD, GND)
- ▷ Terminal Type : 115,200 bps, VT-100
- ▷ Connector : 2.54mm PIN block

Mechanism

- ▷ Material/Dimensions : galvanized steel sheet / 151 x 95 x 31mm

Power

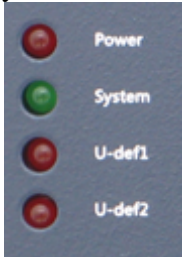
- ▷ Working Voltage : DC 9~24VDC
- ▷ Power Connector : 5.00mm Terminal Block
- ▷ Power Consumption : < 24W (exclude USB device)

Others

- ▷ Real Time Clock : 1
- ▷ Buzzer : 1
- ▷ LED indicator : power, network, serial port, control signal, user defined (DO * 2 more)
- ▷ Applicable Temperature : 0~50°C
- ▷ Applicable Humidity : 20%~80% RHG
- ▷ Certification : CE, FCC

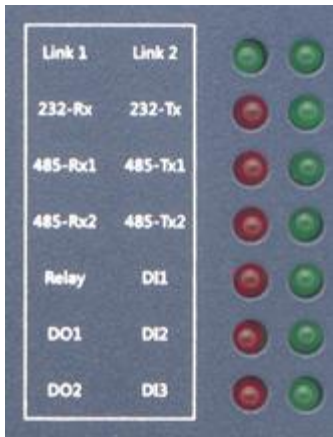
2.2. LED Indicator description

System attributes



Power	Power indicator Connect the operating voltage correctly.
System	System indicator Linux completed boot normally. (about 15 to 20 seconds to start the light)
U-Def	User-defined control lights User can control these 2 lights by Digital Output

Communication and control attributes



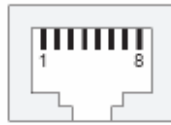
Link	Network indicator on : Connect the network cable correctly flash : Network data transmission
Tx	RS-232 and RS-485 data transmission indicator flash : Data transmission
Rx	RS-232 and RS-485 data receiving indicator flash : Data receiving
Relay	Relay status indicator on : GPIO becomes True, the relay switches to the NO position off : GPIO becomes False, the relay switches to the NC position
DO	DO status indicator on : GPIO becomes True , transistor is opened
DI	DI status indicator on : When detection of external voltage is high level (5~24V DC) , GPIO Interpreted as True

2.3. Pin definition of each communication and control interface

LAN1、LAN2

Pin	Signal
1	ETX+
2	ETX-
3	ERX+
6	ERX-

RJ45



Working Voltage Input

Pin	Signal
1	DC +
2	GND

● DC Range: 9~24V DC

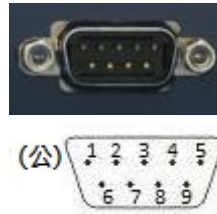
5.00mm TB



RS-232

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

DB9



Digital Output

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

3.50mm TB



RS-232 Console

Pin	Signal
1	TxD
2	RxD
3	GND

2.54mm PB



Digital Input

Pin	Signal
1	DI1
2	DI2
3	DI3
4	GND

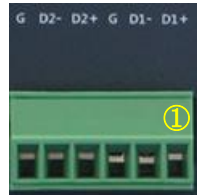
3.50mm TB



RS-485

Pin	Signal
1	D1+
2	D1-
3	GND
4	D2+
5	D2-
6	GND

3.50mm TB



Analog Input

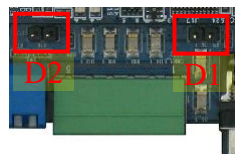
Pin	Signal
1	AI+
2	AI- (COM)

3.50mm TB



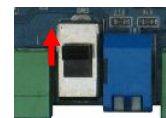
RS-485 Terminal Resistor Setting

Short	Yes
Open	No



AI Mode - (Need to open the case)

Up ↑	0-10V DC
Down	0-20 mA



2.4. Software specifications

Core

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

Pre-installed Utilities

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

Protocol stack

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

Daemons

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

Tool Chain

- ▷ Linux/Windows Operating System
- ▷ GCC: C/C++ PC cross compiler for Linux
- ▷ GLIBC: POSIX Library
- ▷ Examples
- ▷ Integrated Developing Environment (IDE) for Windows

USB expansion interface support driver

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

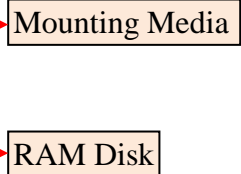
2.5. Embedded-Linux File System

There are three zones in the EBox-AIO-002 File System , Except the most important on-board Flash , there is also a RAM DISK and the user's plug-in SD or 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:/#

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. Software definition of each communication and control interface

Interface Type	Marked	Software definition
Network	LAN1	eth0
	LAN2	eth1
RS-232	RS-232	/dev/ttyS2
RS-485	RS-485 D1	/dev/ttyS1
	RS-485 D2	/dev/ttyS4
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
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
Analog Input	AI	liblldai.so.1 (User-Def. Lib)

External Memory	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. Console

3.1. Factory default setting

A. Login password (case sensitive)

Guest account	Login	guest
	Password	guest
Root account	Login	root
	Password	root

B. Master port communication parameters

Network	Ethernet 1 (LAN1)	IP address : 192.168.2.127
	Ethernet 2 (LAN2)	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. Test environment recommendations

A. Test requires equipment :

- a. EBox – AIO – 002 *1
- b. DC 9~24V power supply *1(supplied power consumption >10W)
(Remarks: Self-made adapter cable for general cable)
- c. Ethernet cross cable

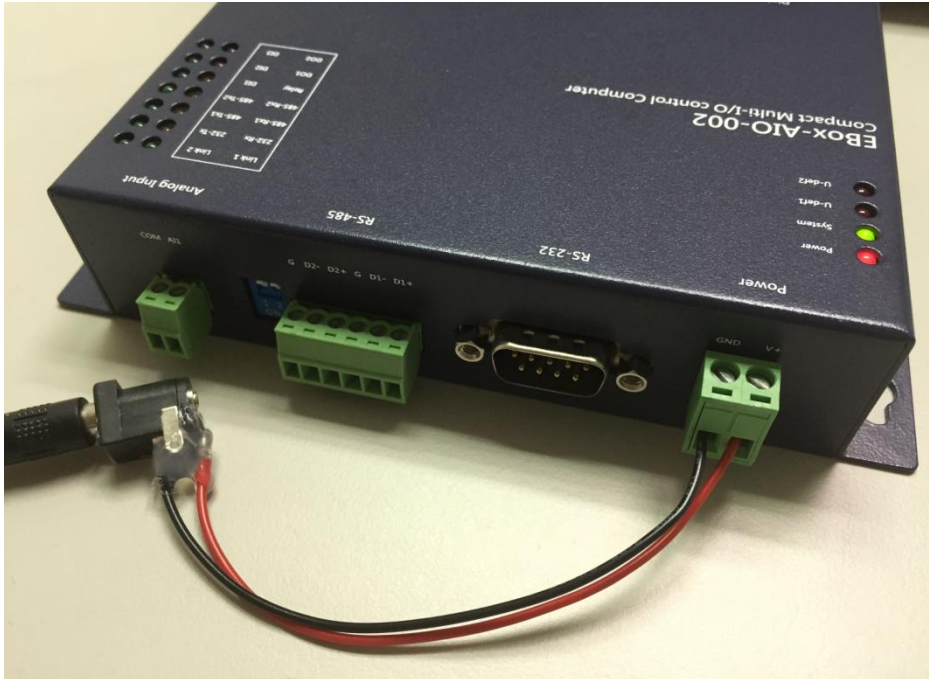
Other Suggested :

- a. DB9 male to female extension cable *1
(Remarks: If you are using a laptop , prepare a USB to RS-232 cable)
- b. EBox-AIO-002 proprietary RS-232 Console cable (product no.: CB-AIOCON-10)



B. Connection method

a. Power



b. Network Cable



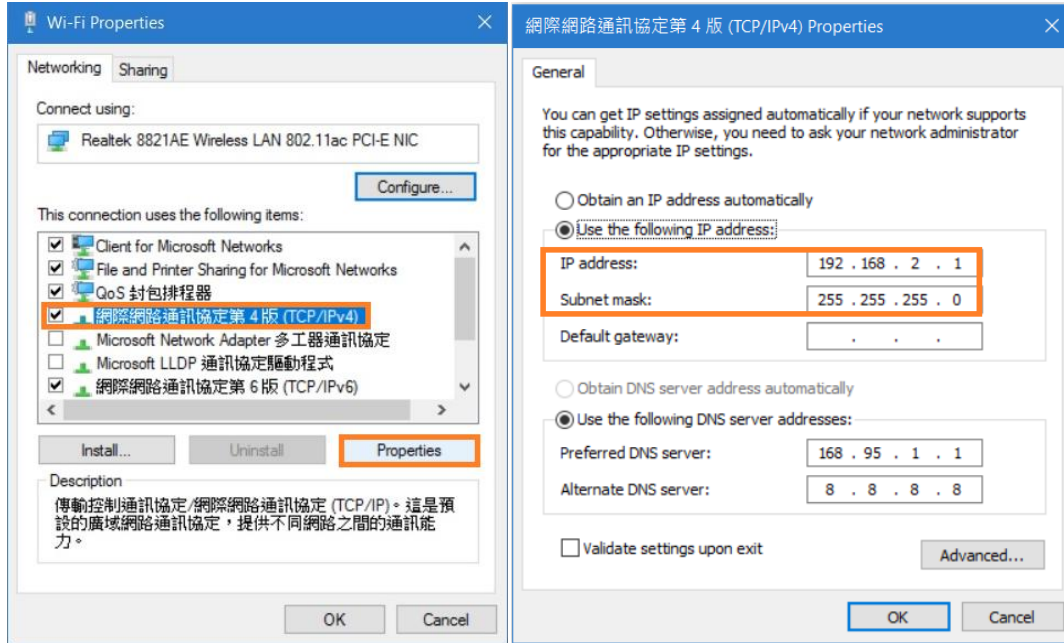
3.3.Ethernet console connection

A. Ethernet Console Introduction

- Through Ethernet cable connected to EBox–AIO-002, we can control and operate files on the system
- Connect computer network port with EBox–AIO-002 LAN port by Ethernet cable.
 - Notice : Sometimes, some computers may not automatically turn ordinary network into crossover , If the ordinary network lead the operation problem , can change into crossover testing.
- Set the computer IP to the same network segment as EBox–AIO-002
EBox–AIO-002 default IP is 192.168.2.127 Netmask is 255.255.255.0

➤ WINDOWS XP ENVIRONMENT SETTING

Step.01a My network places →right-click- Properties →Ethernet→ right-click-Properties ; Or hit Control panel.

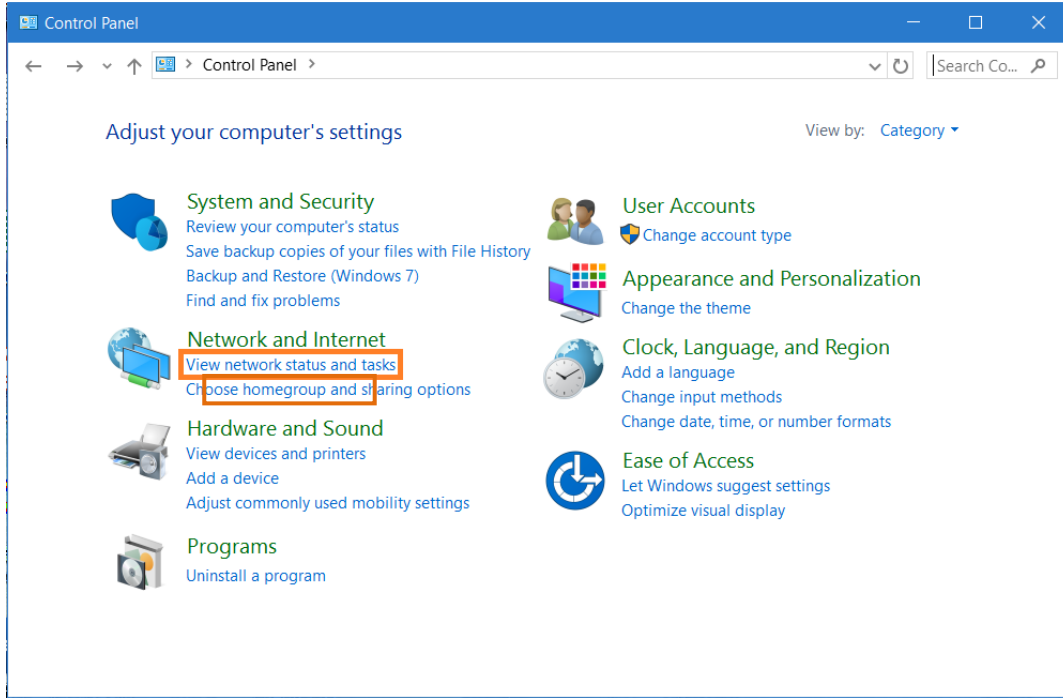


- Notice: DO set IP address to 192.168.2.X. The Range of X is 1~255. Also, do not use the same X number as EBox–AIO-002.

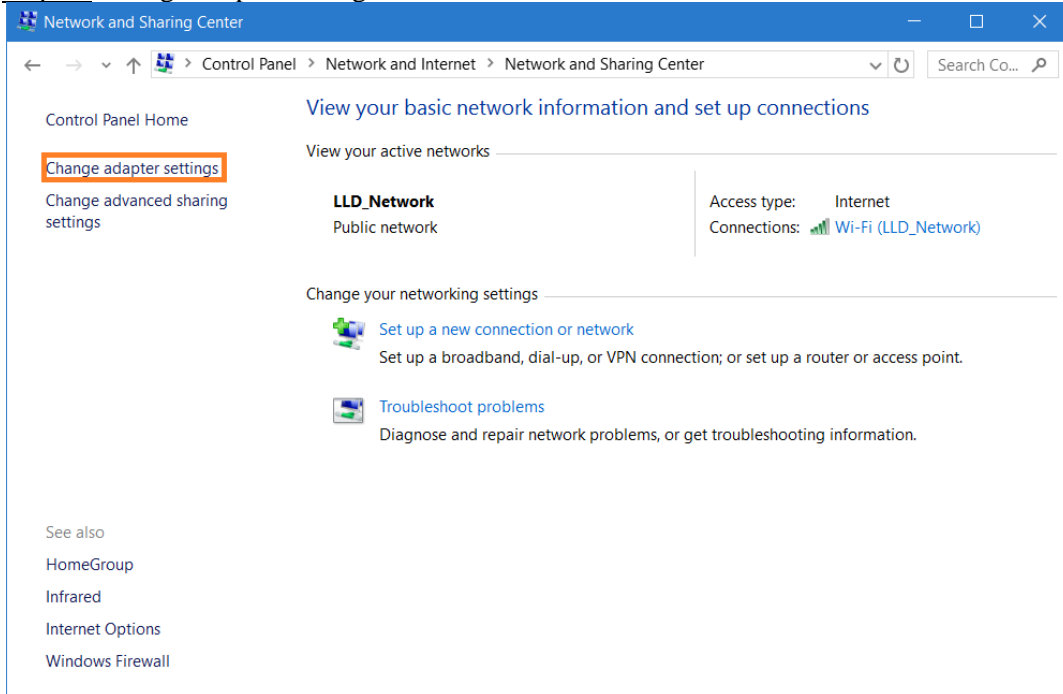


➤ WINDOWS 7 ENVIRONMENT SETTING

Step.01b Start → Control Panel → Network and Internet-View network status and tasks.

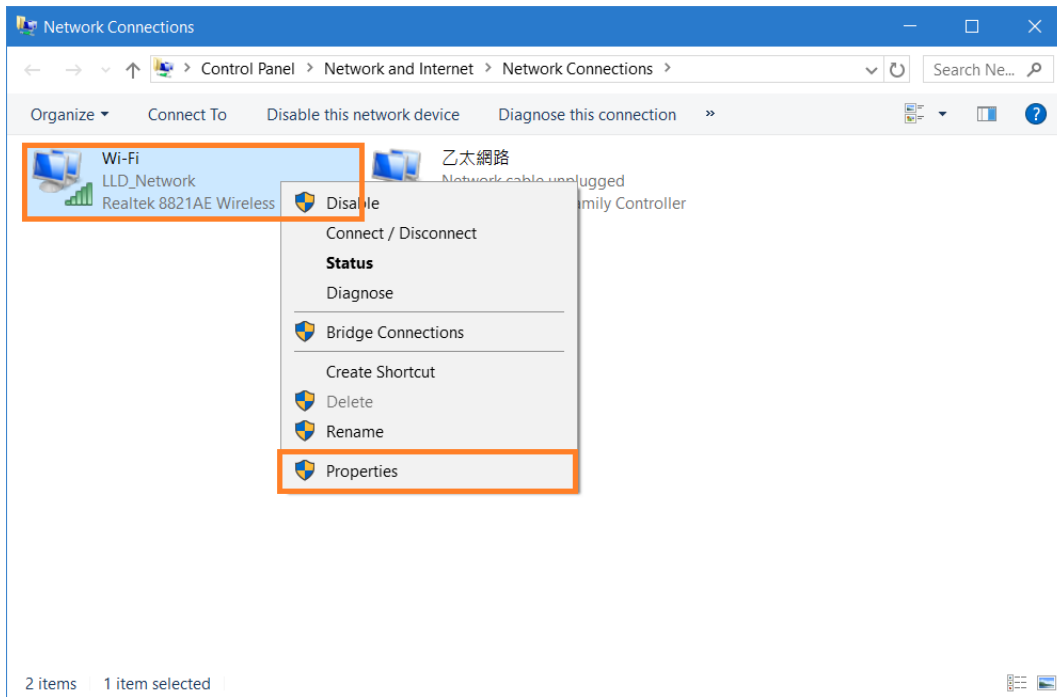


Step.02 Change adapter settings.

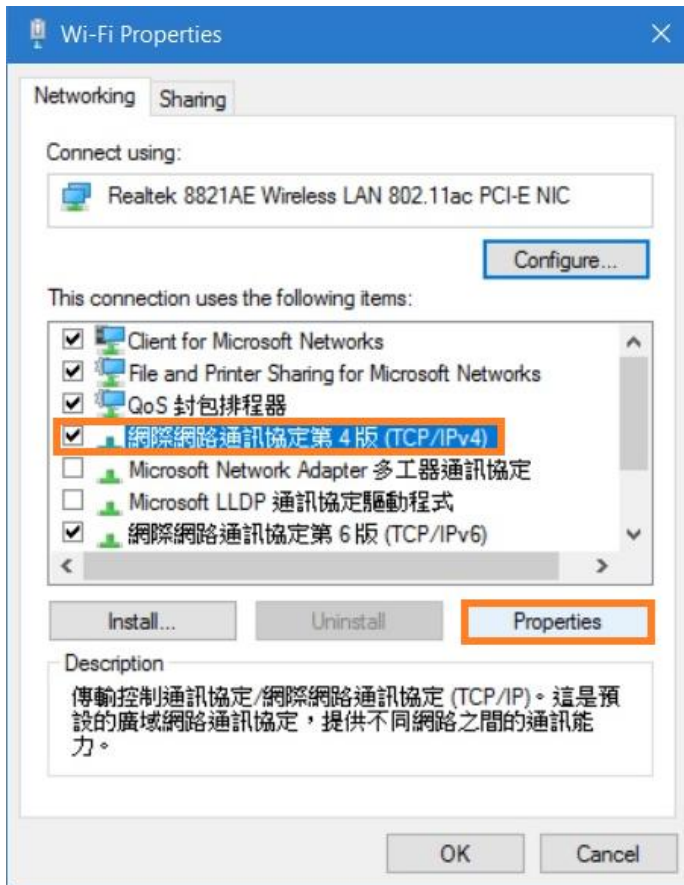




Step.03 Ethernet(Wi-Fi) →right-click Properties.

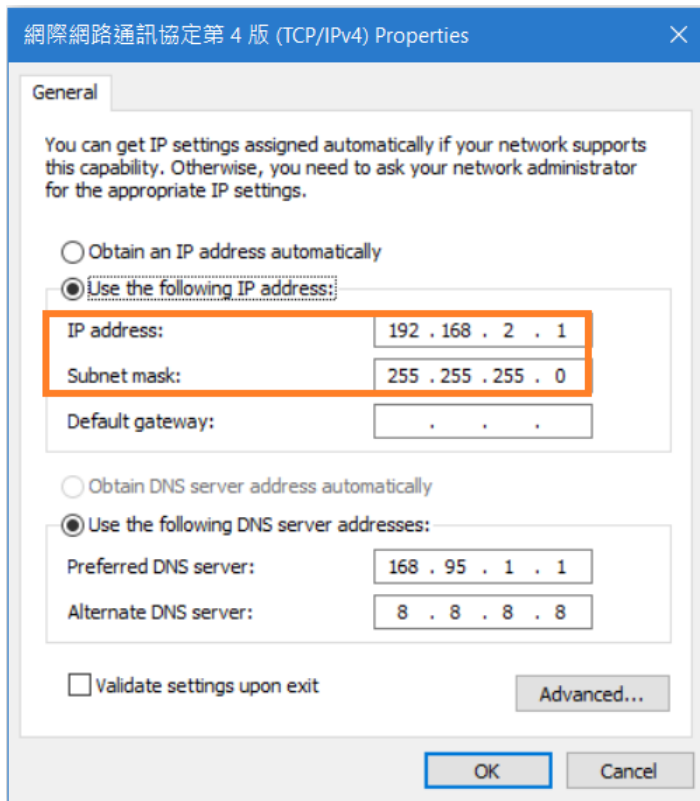


Step.04 網際網路通訊協定第 4 版(TCP/IPv4) → Properties.

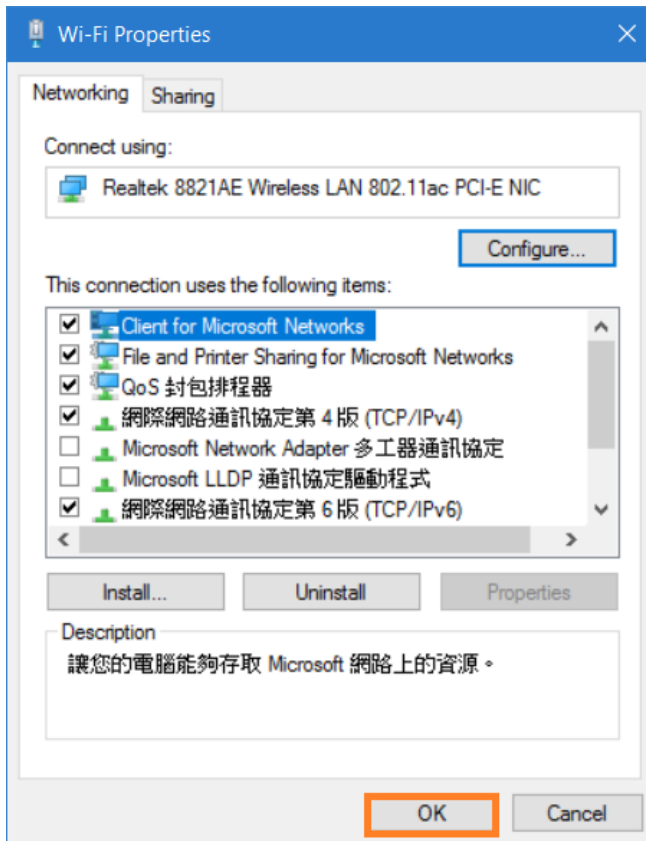




Step.05 IP address : 192.168.2.1 ; Subnet mask : 255.255.255.0 → OK.



Step.06 Click OK button when the setting is completed.



B. Telnet operating

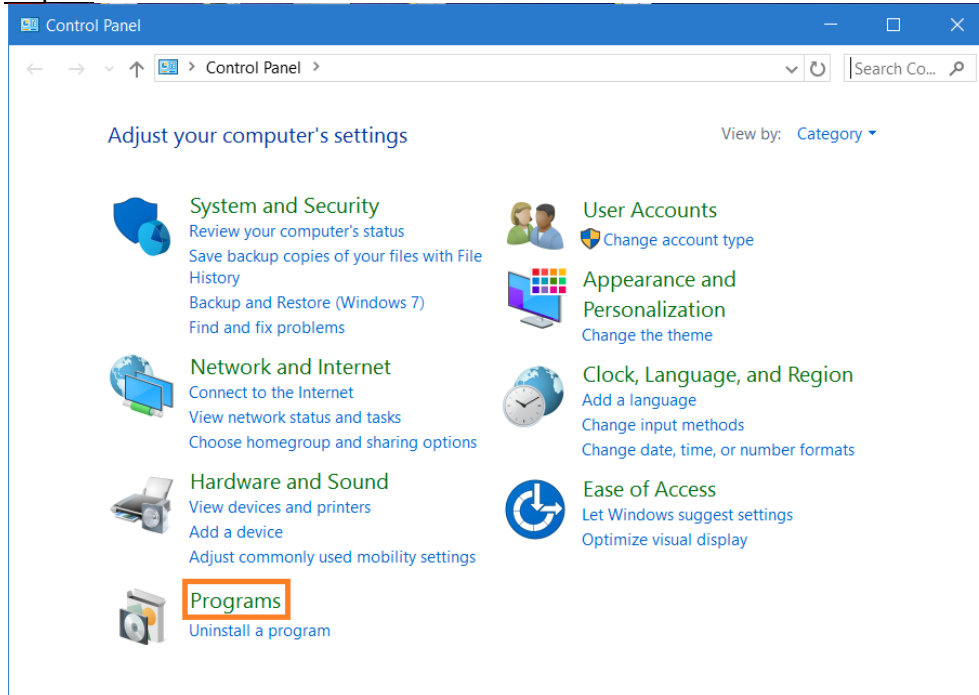
✓ WINDOWS XP ENVIRONMENT SETTING

Default already has telnet function.

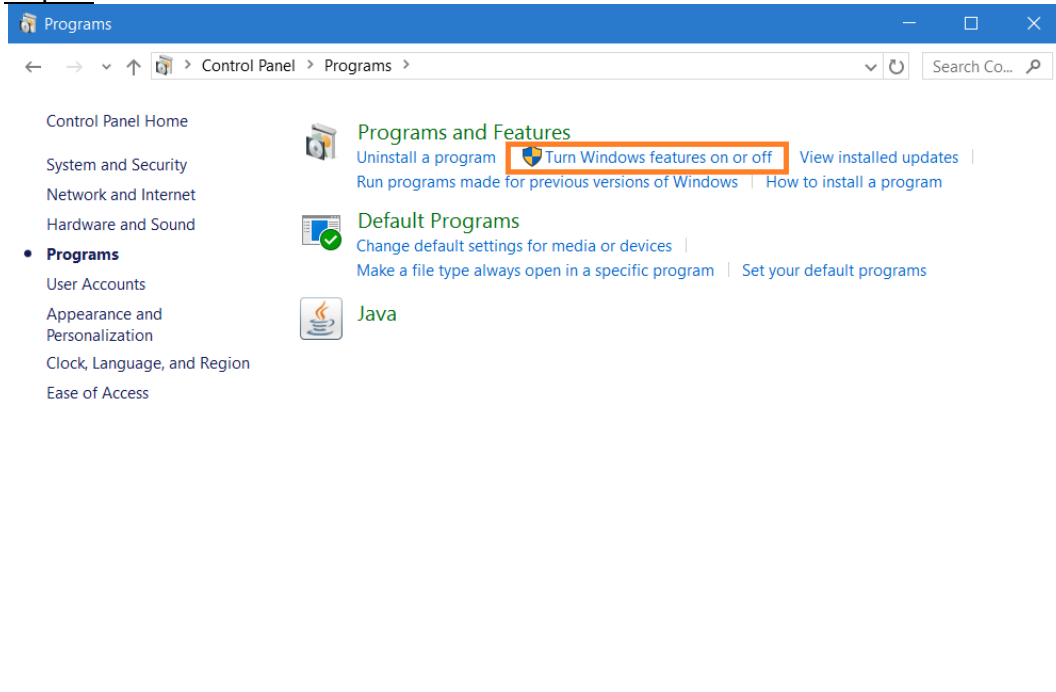
✓ WINDOWS 7 ENVIRONMENT SETTING

Windows 7 need to start telnet function

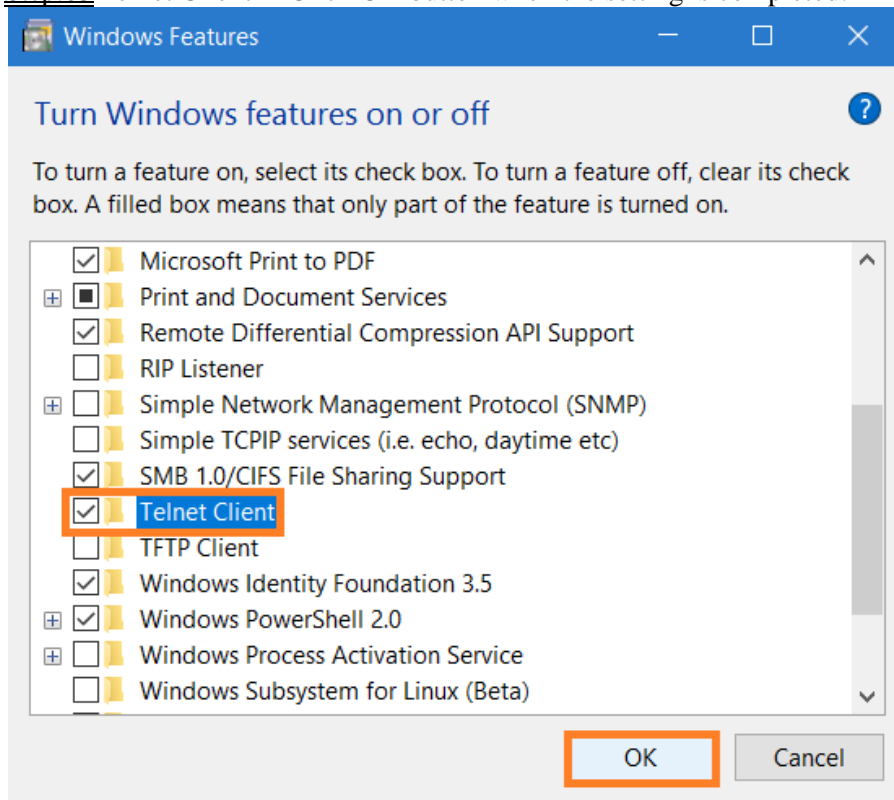
Step.01 Start → Control Panel



Step.02 Turn Windows features on or off

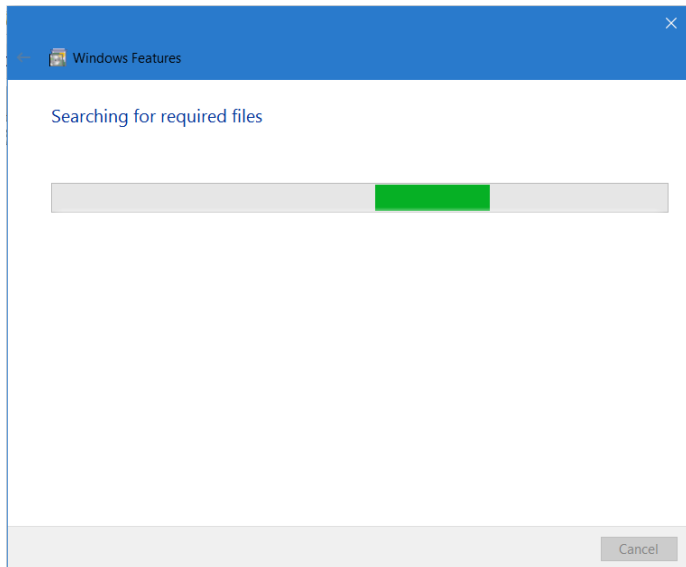


Step.03 Telnet Client → Click OK button when the setting is completed.





Step.04 Wait for a few minutes.



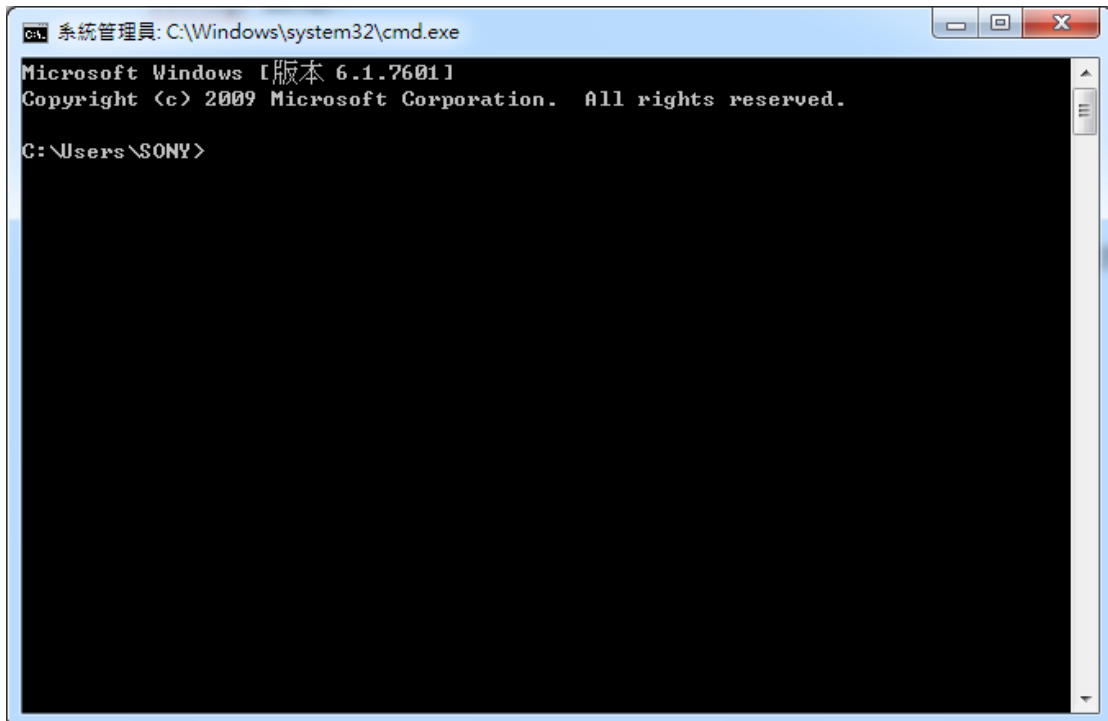
Telnet Login

➤ **WINDOWS XP ENVIRONMENT SETTING**

Step.01a Start → Execute → key in cmd and press Enter ◦

➤ **WINDOWS 7 ENVIRONMENT SETTING**

Step.01b Search→ key in cmd and press Enter →When process is completed, enter the following screen.





Step.02 Please key in “ping 192.168.2.127” to check whether or not to communicate with the M502 , If could not communicate, please check the IP setup process again , press Ctrl+c to end cmd process.

```
cmd 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
回覆自 192.168.2.127: 位元組=32 time<1ms TTL=64
回覆自 192.168.2.127: 位元組=32 time<1ms TTL=64
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 Please input” telnet 192.168.2.127” , you will see the login screen if connect successfully.

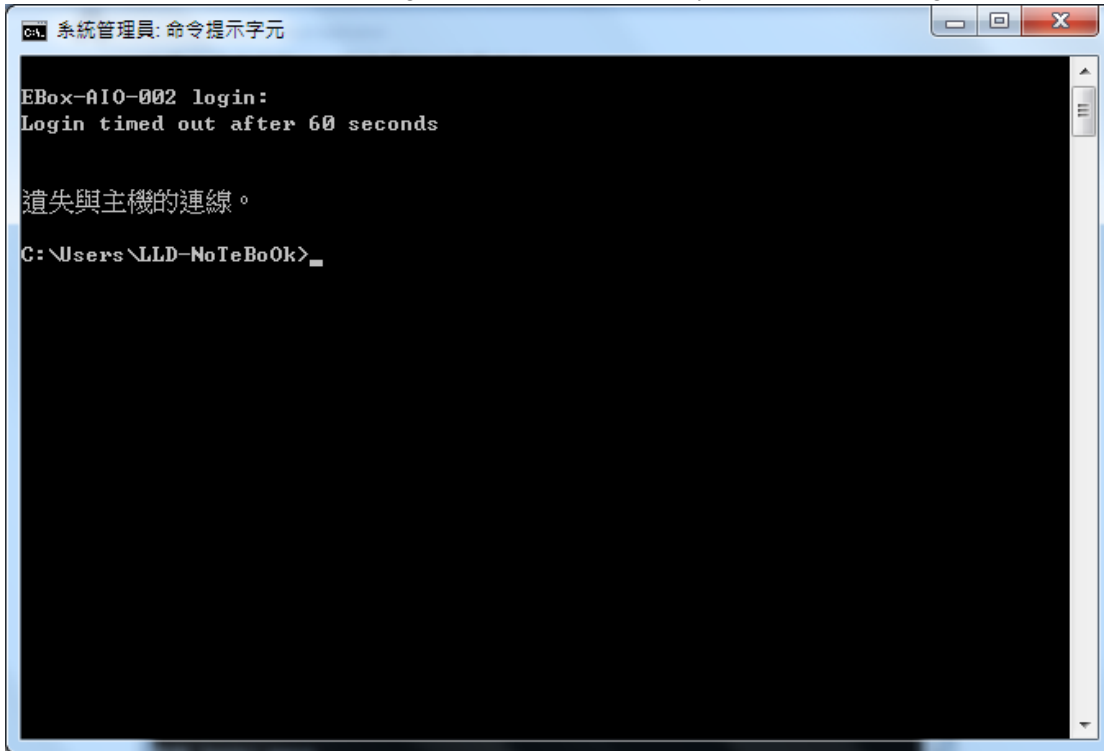
```
cmd Telnet 192.168.2.127

EBox-A10-002 login:
```

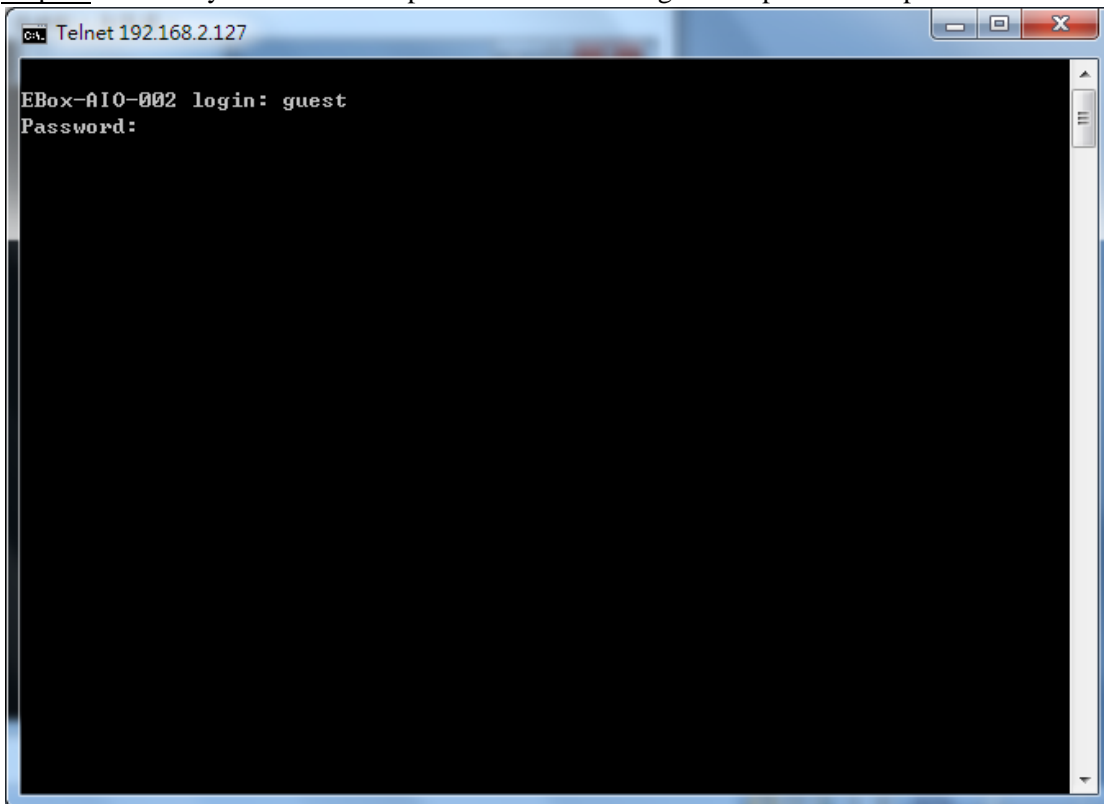




Notice : You will see the following screen if don't enter any instruction for long time !

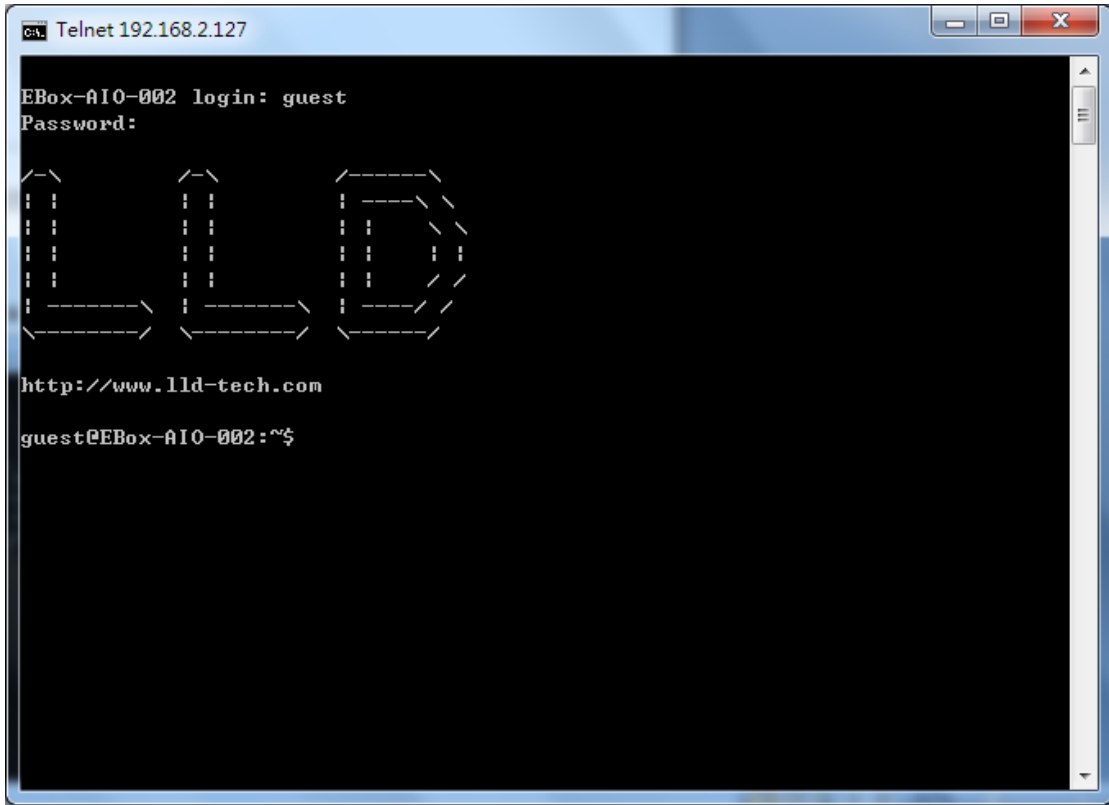


Step.04 Please key in account and password both with "guest" , password input field is hidden.

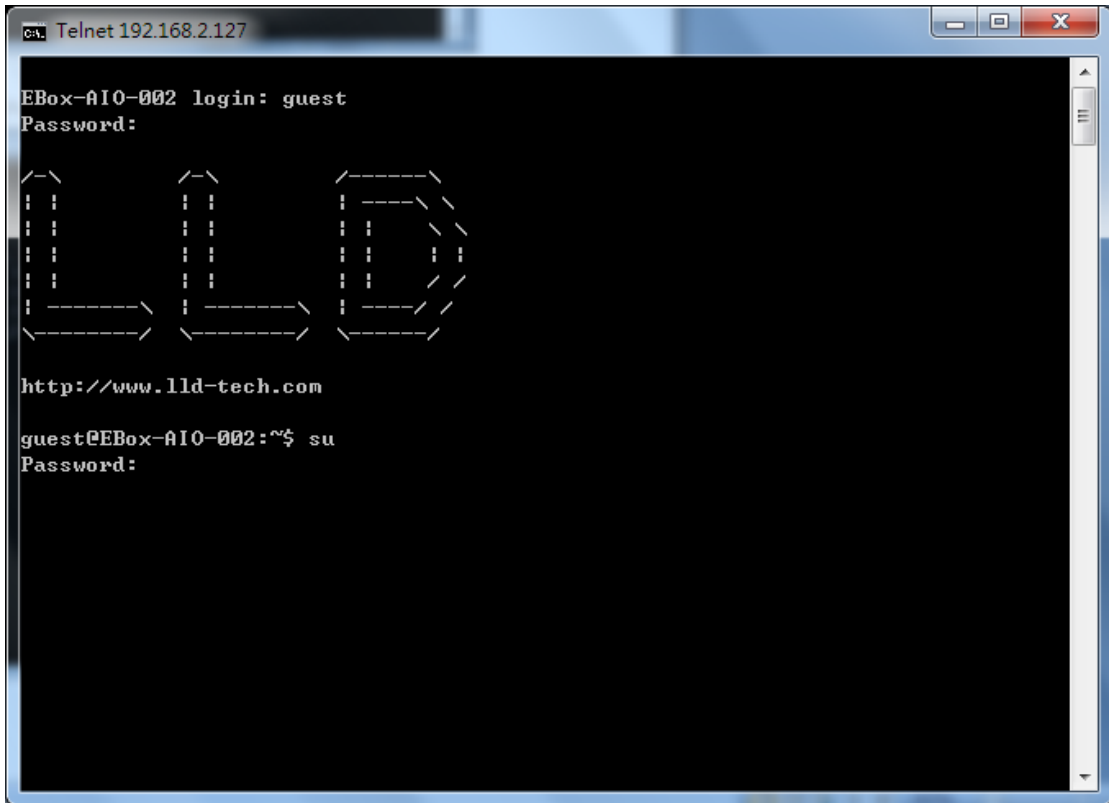




Step.05 You will see the login screen after you entered it successfully.



Step.06 Login to get root privileges , key in “su” , password : root , Password input field is hidden.





Step.07 Login successfully.

```
Telnet 192.168.2.127

EBox-A10-002 login: guest
Password:

LLD

http://www.lld-tech.com

guest@EBox-A10-002:~$ su
Password:
root@EBox-A10-002:~#
root@EBox-A10-002:~#
```

➤ Notice : If the password is incorrect, you will see the following screen !

```
Telnet 192.168.2.127

EBox-A10-002 login: guest
Password:

LLD

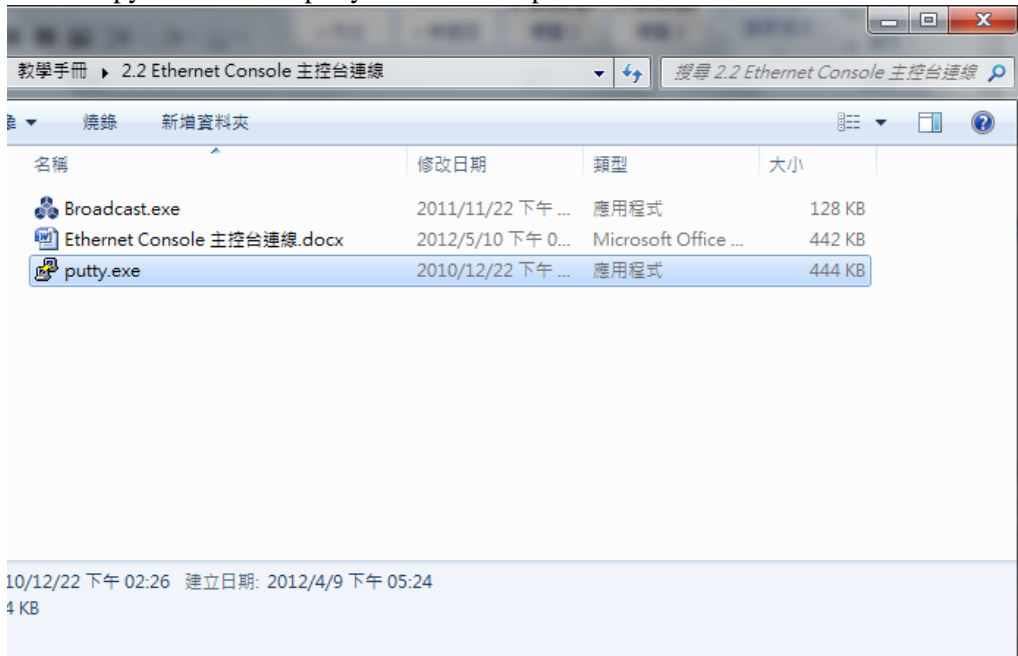
http://www.lld-tech.com

guest@EBox-A10-002:~$ su
Password:
su: incorrect password
guest@EBox-A10-002:~$
```



C. Putty operation

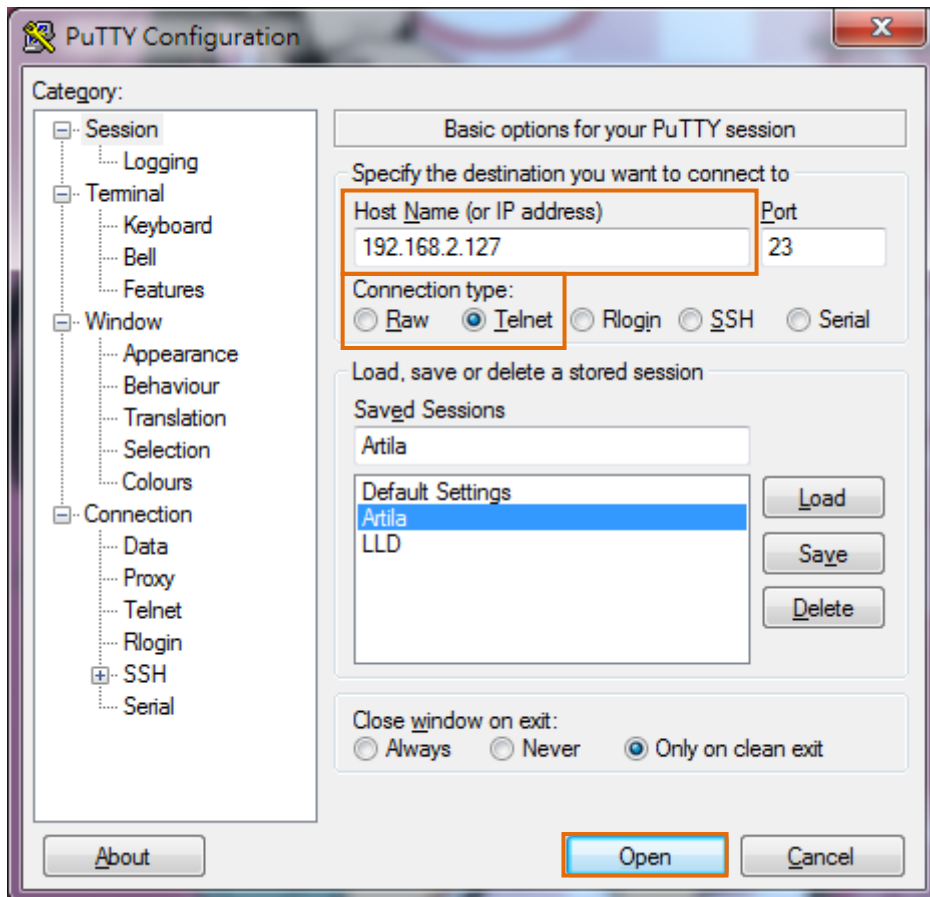
- ◆ Please copy attached file-putty.exe to Desktop and execute.



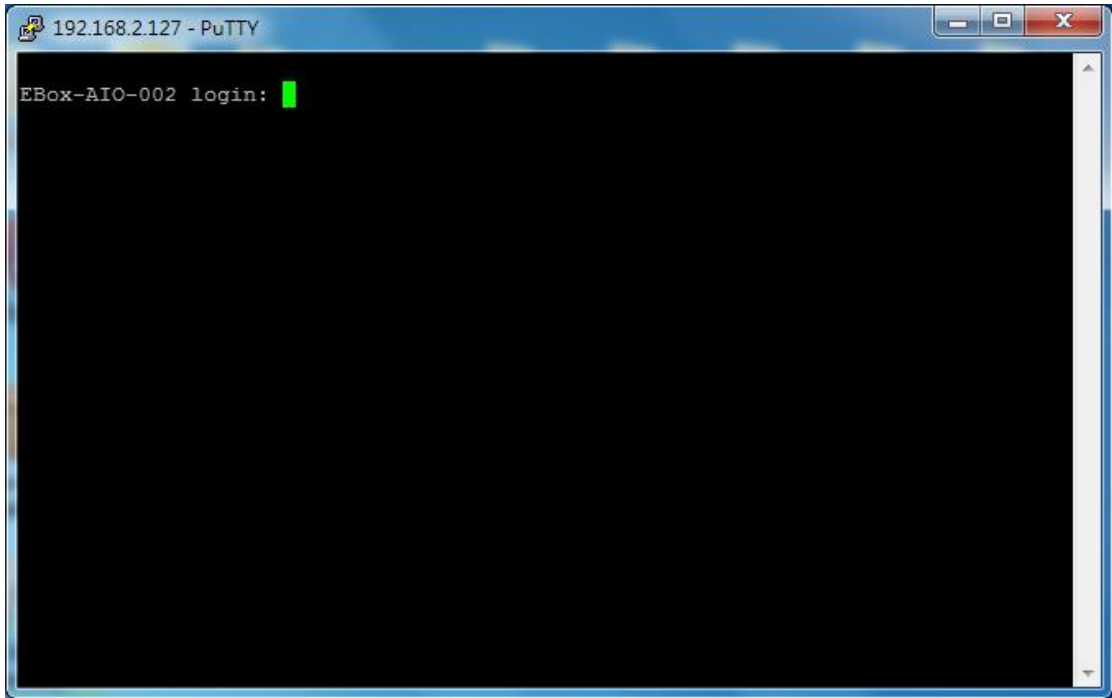
Step.01 Host Name (or IP address) key in 192.168.2.127

→ Connection type choose Telnet , Port will automatically change to 23

→ Press Open after the setting is completed.

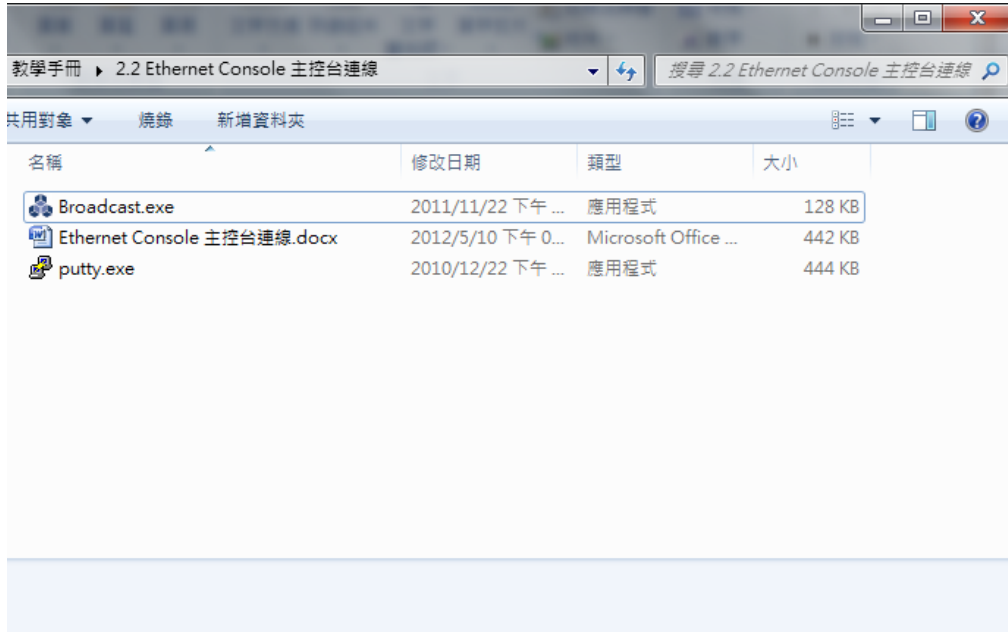


Step.02 Login operation. Please refer to Telnet login step 4.

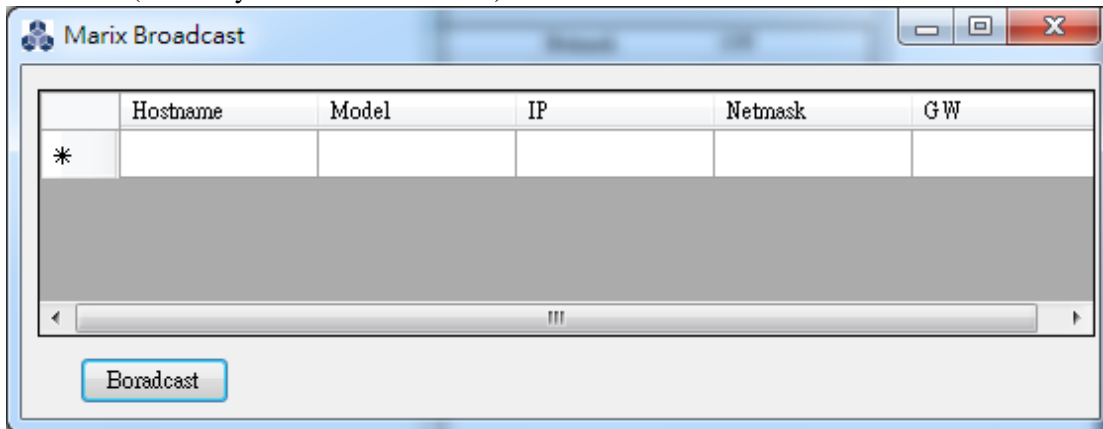


D. Broadcast tools : Broadcast.exe

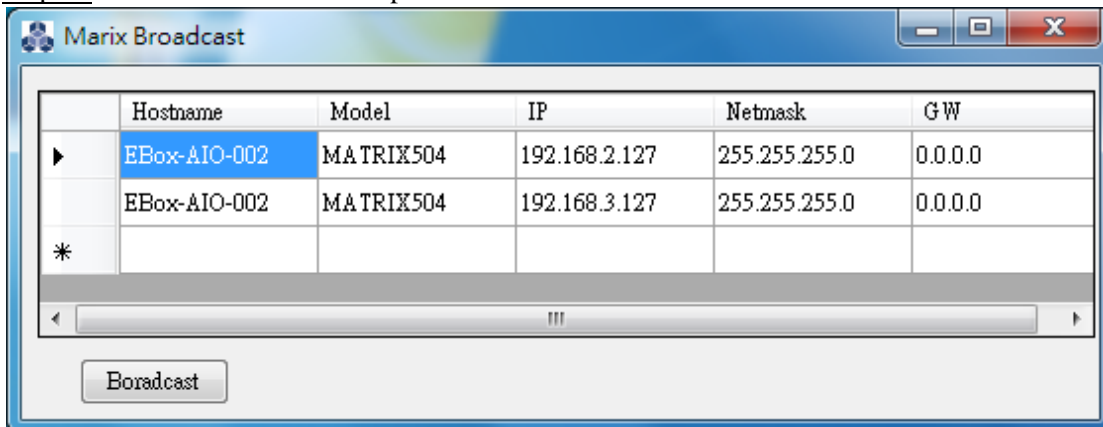
◆ IP search tools : search EBox-AIO-002 IP address



Step.01 Attached file-Broadcast.exe in folder , run Broadcast.exe need to install .Net (Basically installed on window).



Step.02 execute Broadcast.exe , press the Broadcast button .



3.4.RS-232 console connection

A. Serial Console Introduction

Through the RS-232 interface in computer to control and execute the file operation on EBox-AIO-002 embedded system .

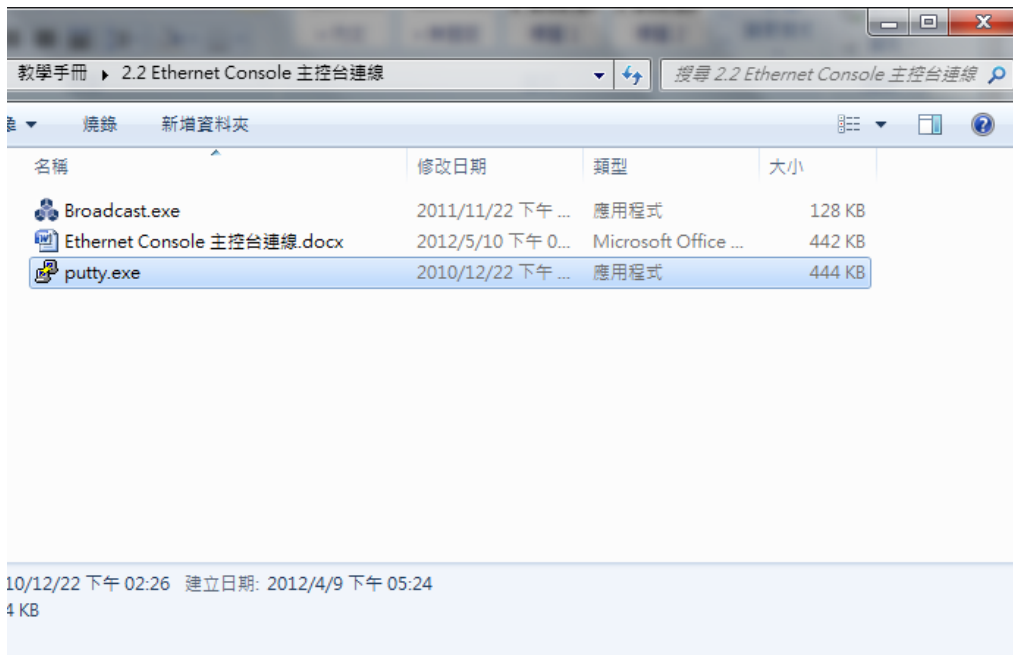
B. Putty operating

Step.01

Connect Console Port / RS-232(female) on EBox-AIO-002 with RS-232(male) on computer for each other.



◆ Please copy attached file putty.exe to Desktop and execute.



Step.02

Select the type of connection (Connection type → click Serial)

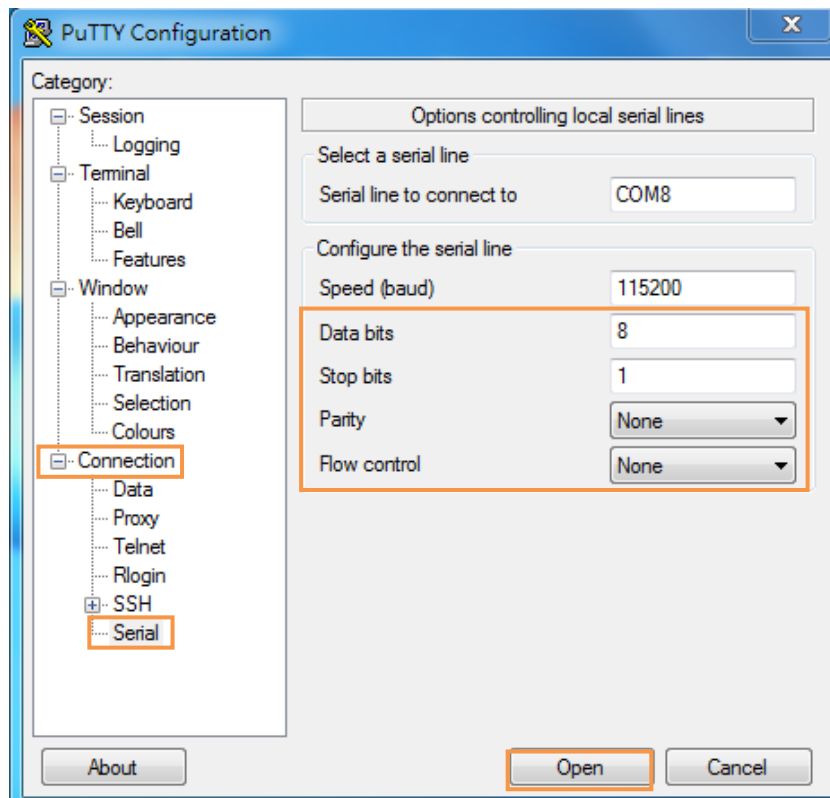
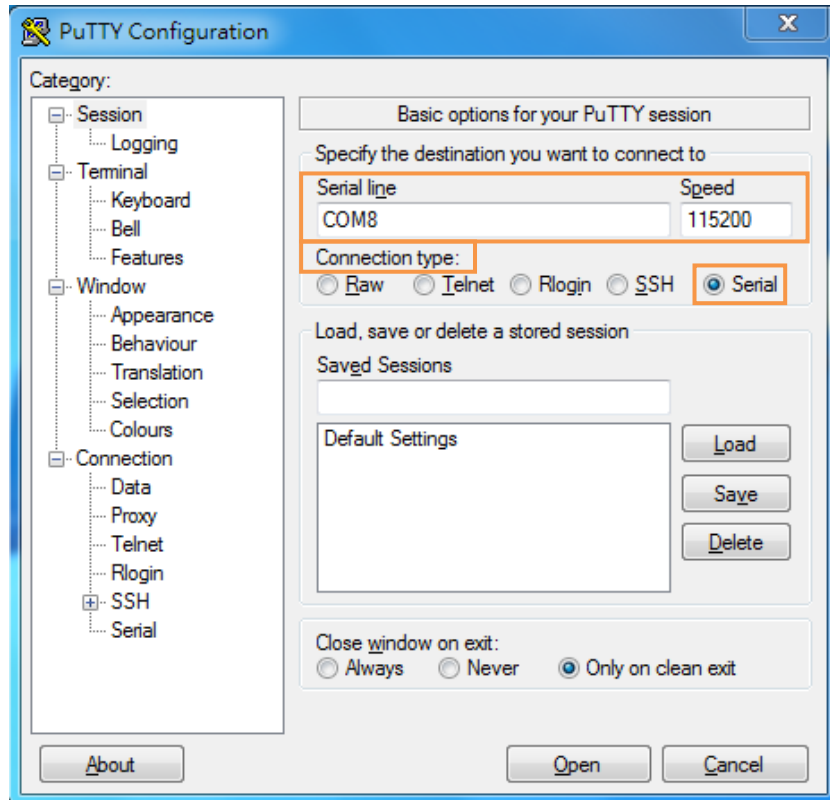
→Setting RS-232 Port (Serial line) and Speed 【Speed (baud) : 115200】 on computer.

【If you don't know port number. ◆Please search ComPort at Administrator】

→Setting Configure the serial line properties (Connection on left → choose Serial)

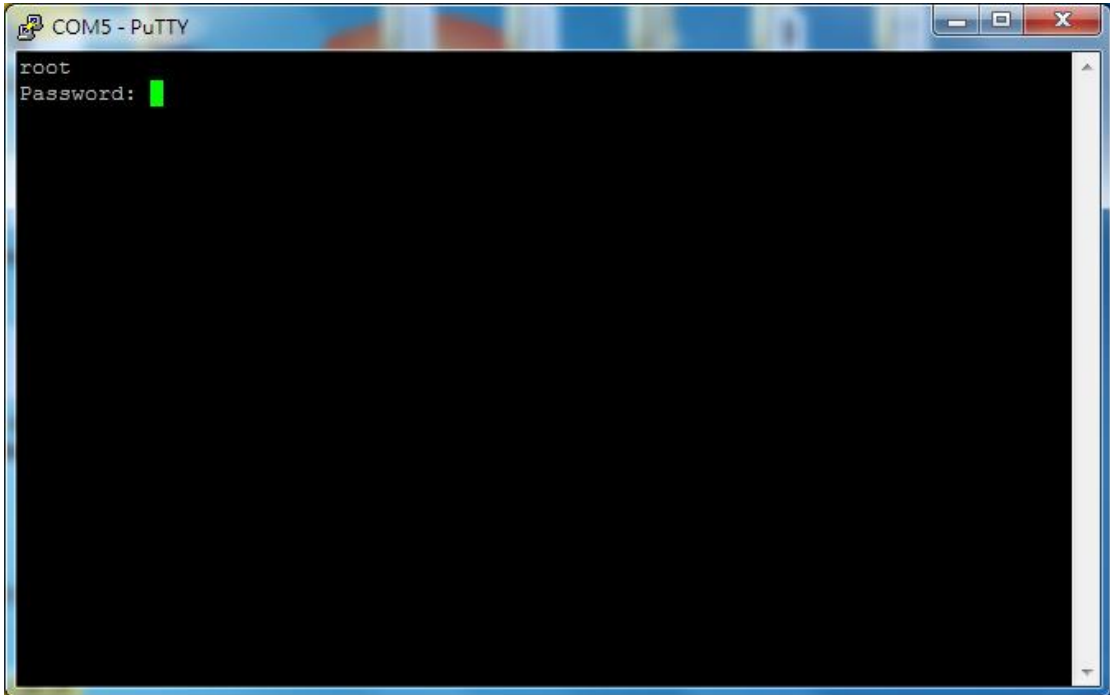
【Data bits : 8 、 Stop bits : 1 、 Parity : None 、 Flow control : None】

→Press Open after the setting is completed.



Step.03

Account / password default are both 'root'. But the password field is hidden.



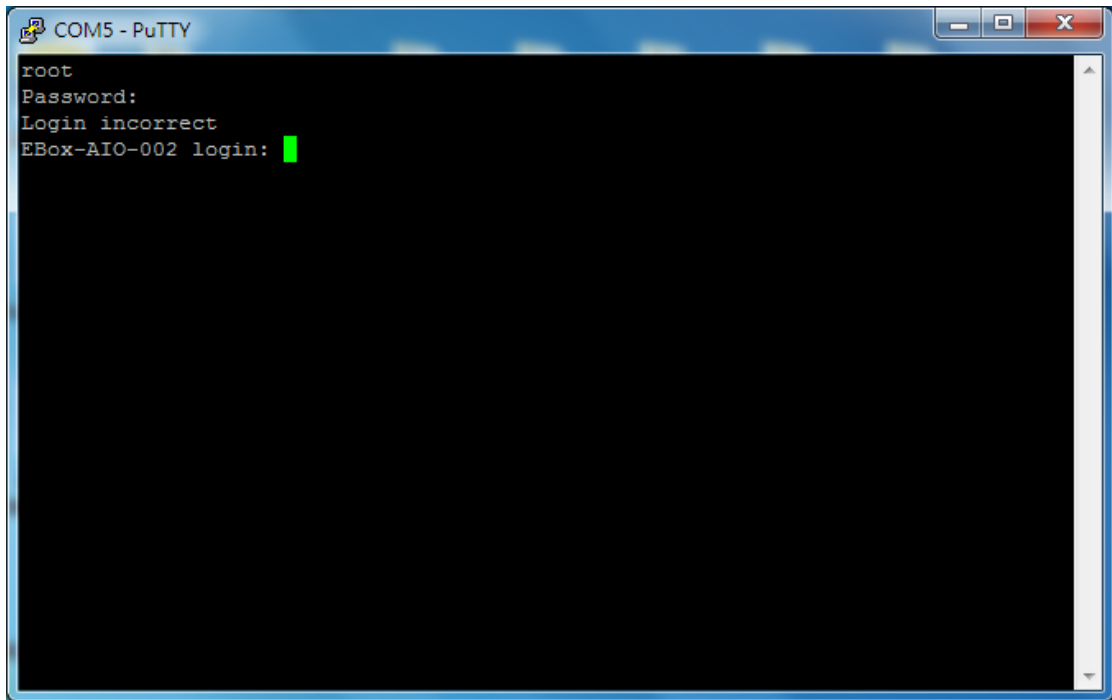
Step.04

Login Successfully.



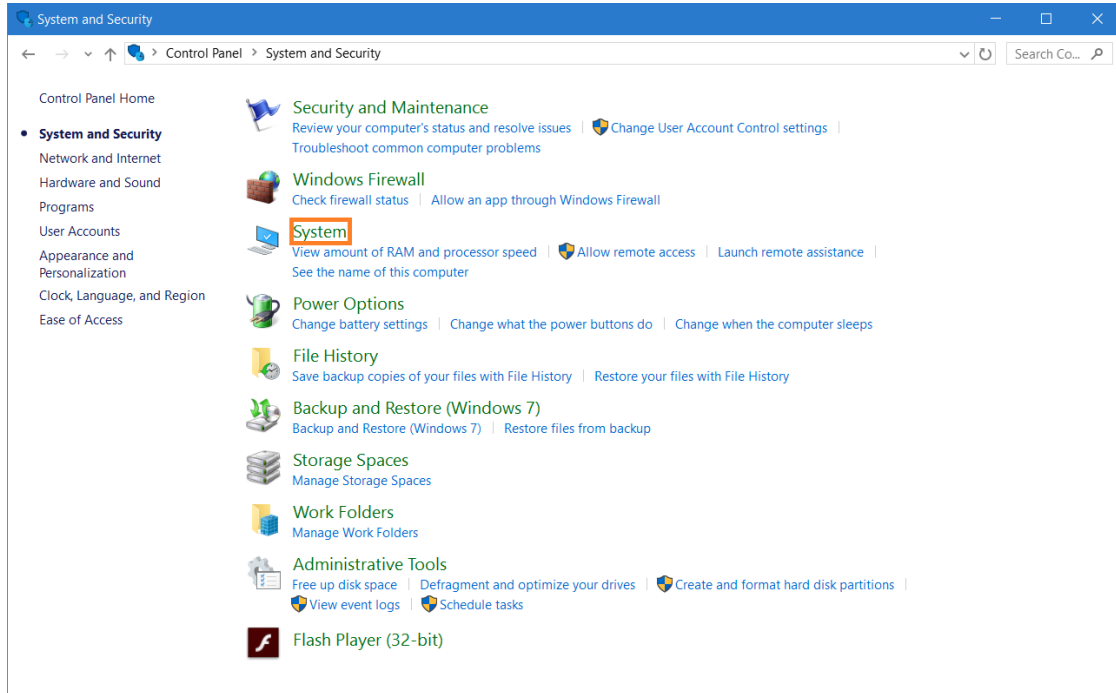
Step.05

If the input is wrong, need to enter the account and password again.

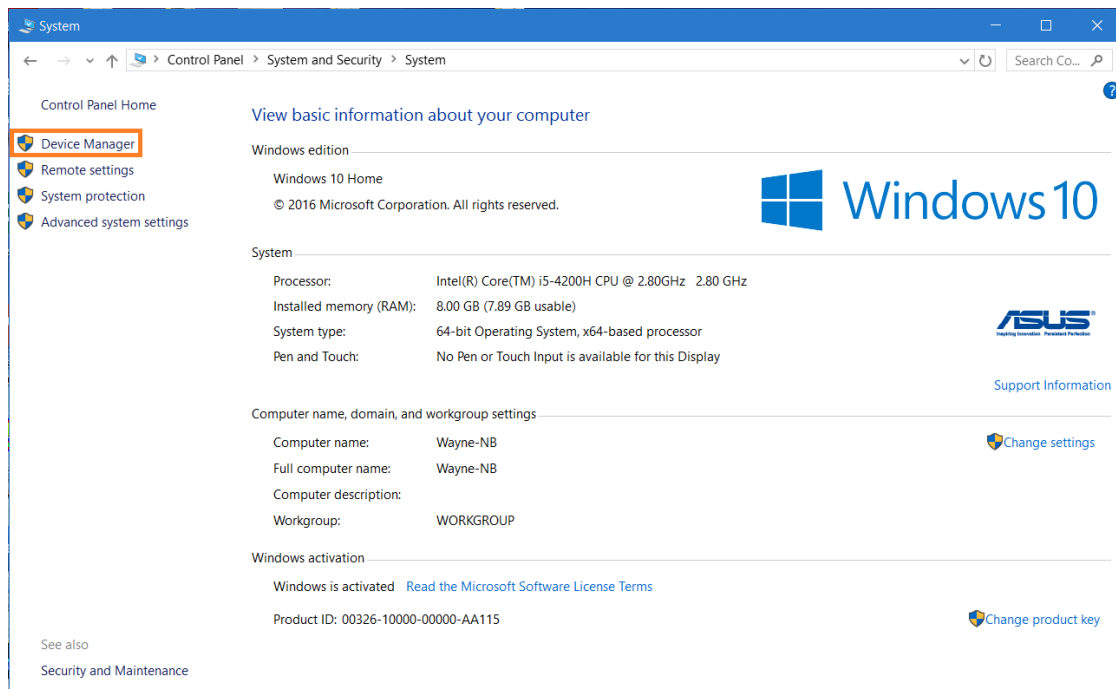


◆ Device Manager for Comport query

STEP.01 Press System in Control Panel

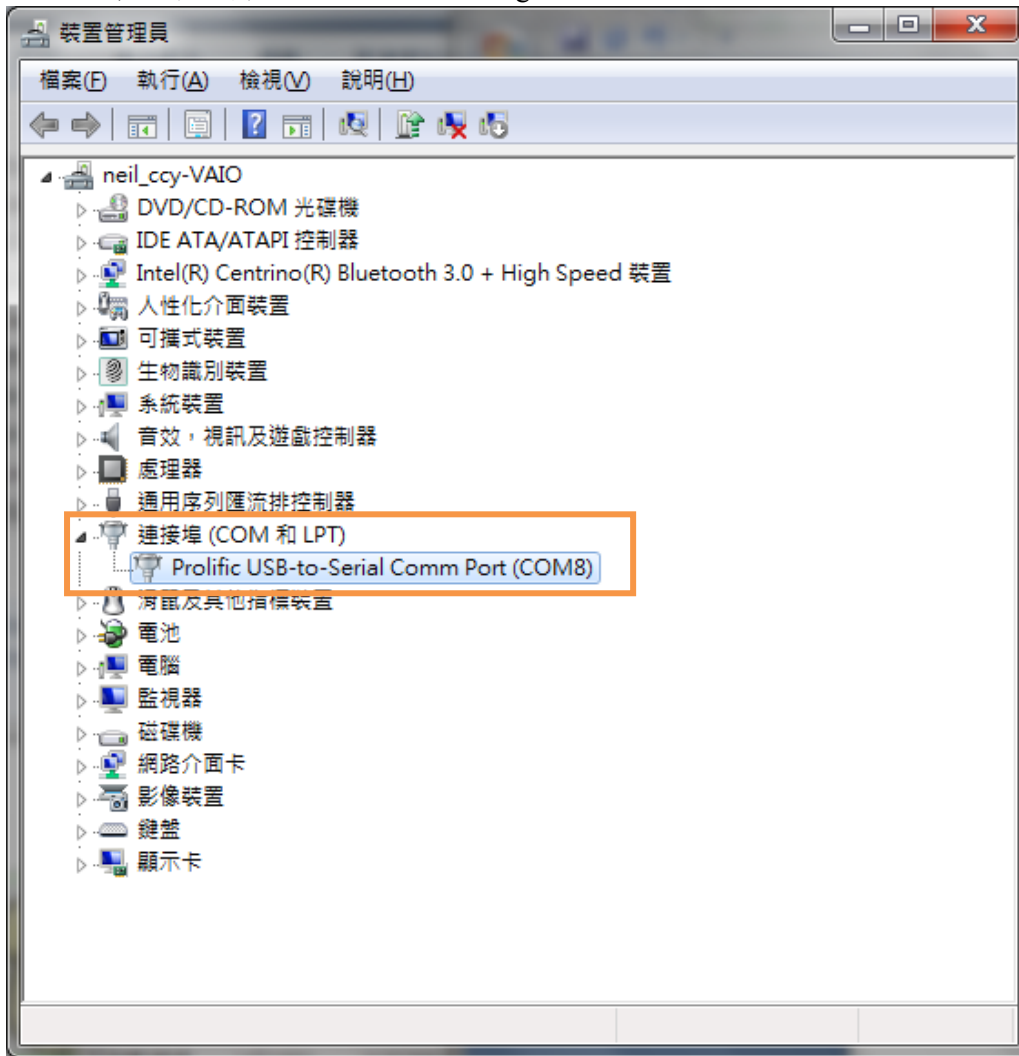


STEP.02 Click Device Manager

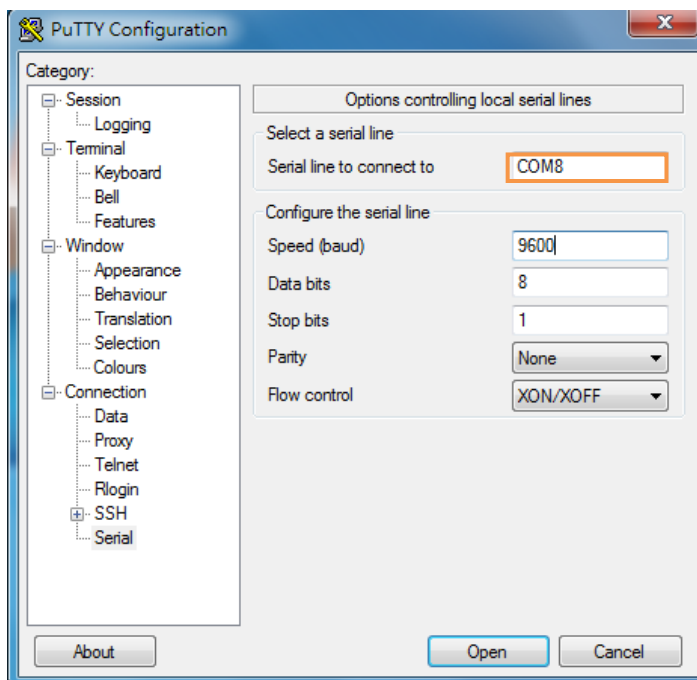


STEP.03 We can find Comm Port under option port (COM and LPT).

Notice: “裝置管理員” means Device Manager under.



STEP.04 Choose the active port.



4. Install the Development Environment

4.1.Environmental requirements

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

4.2.Install Linux Tool Chain

A. Acquire Tool Chain(Cross Compiler)

◆ **CD with the goods.**

Path screen

◆ **Download from the web**

screen

B. Install GNU Tool Chain

Installation steps :

◆ **Login as root user**

screen

◆ **Copy arm-linux-4.3.2.tar.gz to Root directory**

screen

◆ **Key in decompress the instruction when install Tool Chain**
#tar -xvfj arm-linux-4.3.3.tar.bz2

Decompress screen

Tool chain's file name:

arm-linux-gnueabi-gcc

arm-linux-gnueabi-g++

arm-linux-gnueabi-strip

Version: gcc 4.3.3, glibc 2.9, binutils 2.18

Execution of three instructions

4.3.Install Windows Tool Chain

A. Acquire SW

Windows's user, please download from <http://www.codesourcery.com>

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

B. Tool Chain

Eecute Tool Chain installation directly

screen

Tool chain's file name are

arm-linux-gnueabi-gcc

arm-linux-gnueabi-g++

arm-linux-gnueabi-strip

Version: gcc 4.3.3, glibc 2.8, binutils 2.19



Execution of three instructions



5. Execute the First Program – Hello

(By Linux development environment as an example)

5.1. Source code

Through the tool program "vi" edit a hello.c file or copy to the Linux host from CD with the goods

- ◆ hello.c ※reference source code

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

5.2. Produce execution file

- ◆ Through arm-linux-gnueabi-gcc command (.c file)

```
screen
```

- ◆ Through make command

```
Make content file and the results of execution
```

5.3.Upload to EBox-AIO-002

Through ftp , upload execution file a.out or hello to EBox-AIO-002

- ◆ ftp instruction
[ftp 192.168.2.127](ftp://192.168.2.127) (192.168.2.127 is EBox-AIO-002 default IP address)
Default account and password of manufacture :
Username : root
Password : root

```
ftp 192.168.2.127 screen
login
change mode
ftp
```

Note : Change the file transfer mode into binary.

5.4. Test the execution program

Refer to Chapter 3 , login EBox-AIO-002 Console

◆ Conversion format

Through chmod command (Change the file to an executable attribute)

```
chmod +x screen
```

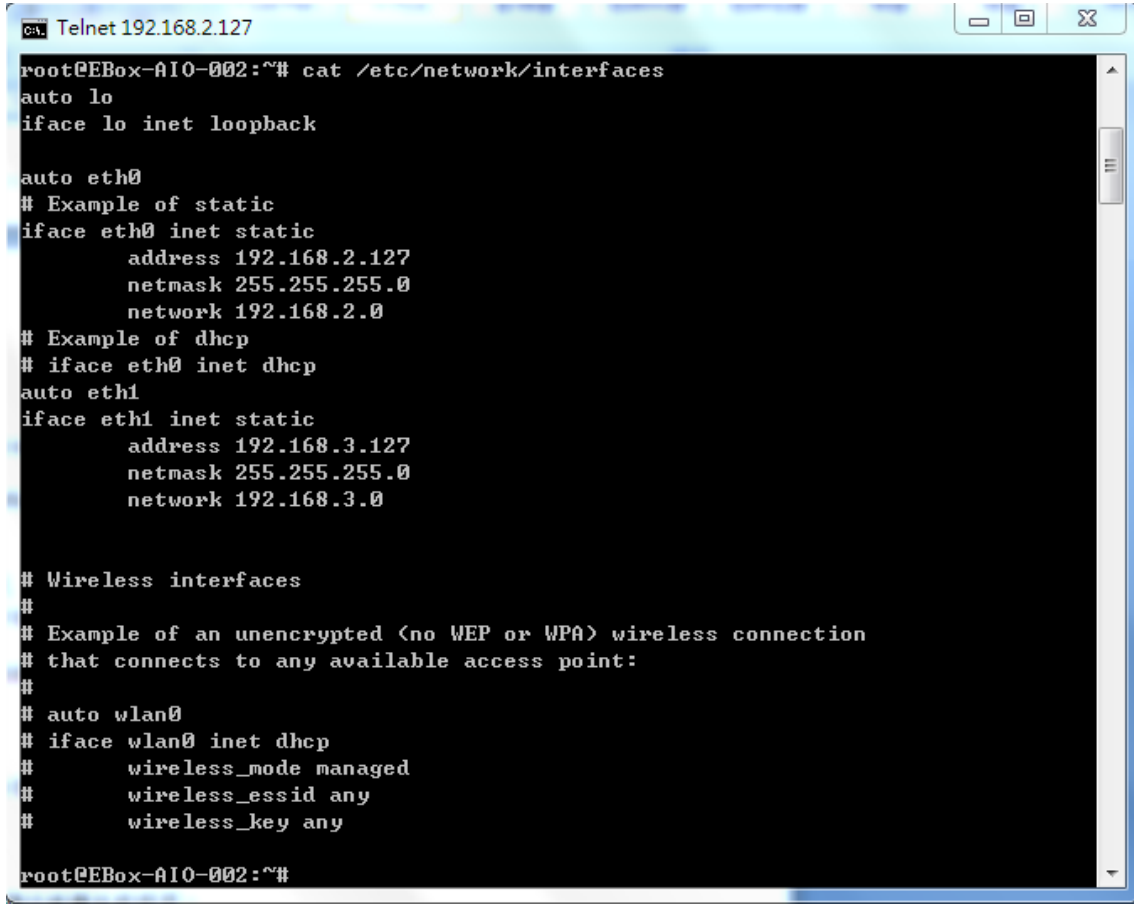
◆ Execute the program

```
Execute screen
```

6. Commonly Basic Operation

6.1. Change the network settings

STEP.01 Input command `【cat /etc/network/interfaces】` The screen is as follows.



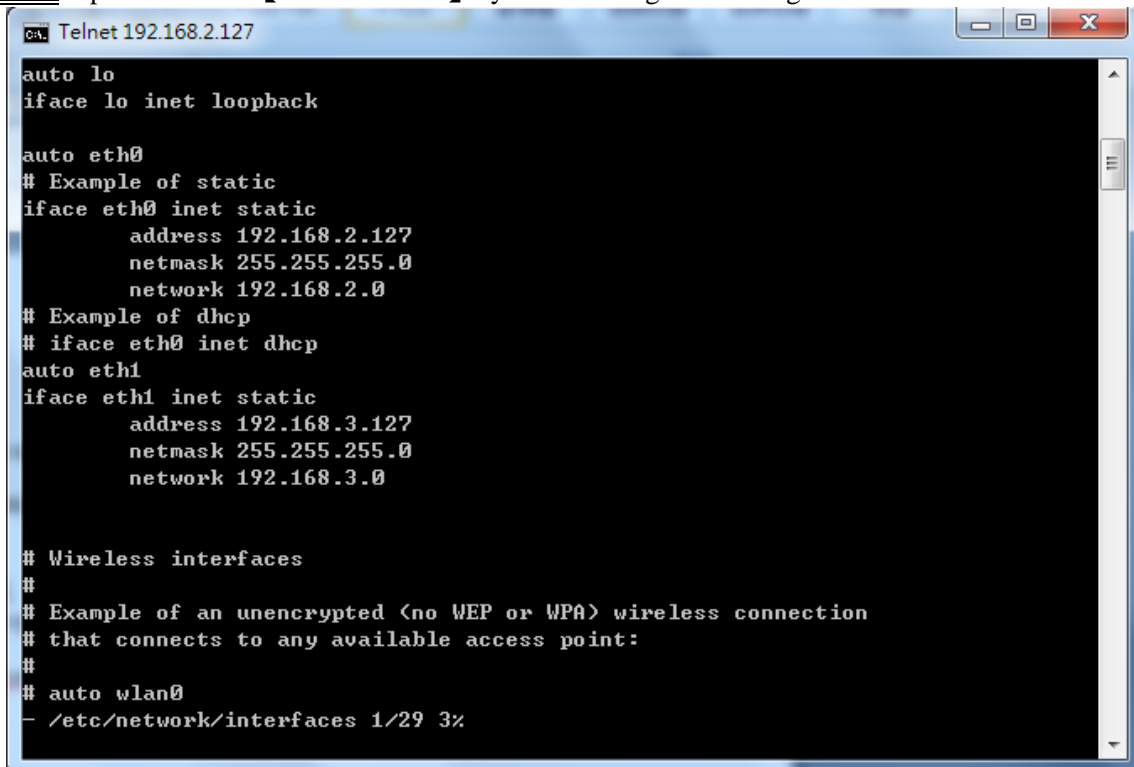
```
Telnet 192.168.2.127
root@EBox-A10-002:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

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 Input command `【vi interfaces】`, you can change the settings.



```
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. Initialize the boot settings and application

STEP.01 Input command **【cat /etc/rc.local】** The screen is as follows.

```
Telnet 192.168.2.127
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 Input command **【vi rc.local】** , you can change the settings.

```
Telnet 192.168.2.127
#!/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. Plugin USB and SD memory

STEP.01 Input command `【cat /etc/fstab】` , The screen is as follows.

```

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. Change the system time

STEP.01 Input command `【date】` , Show software current time.

```

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

```

STEP.02 Input command `【date MMDDhhmmYYYY】` , you can change the software time .

(MM = month(01~12) 、 DD = day(01~31) 、 hh = hour 、 mm = min 、 YYYY = year)

```

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

```





6.5. Web server basic setting

STEP.01 Input command `【cat /usr/www/index.html】`, display index.html as follows.

```

ca. 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 Input command `【vi /usr/www/index.html】`, You can change the settings.

```

ca. 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 Input command `【cat /etc/lighttpd.conf】` , display lighttpd.conf as follows.

```
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 Input command `【vi /etc/lighttpd.conf】` , You can change the settings.

```
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 #####

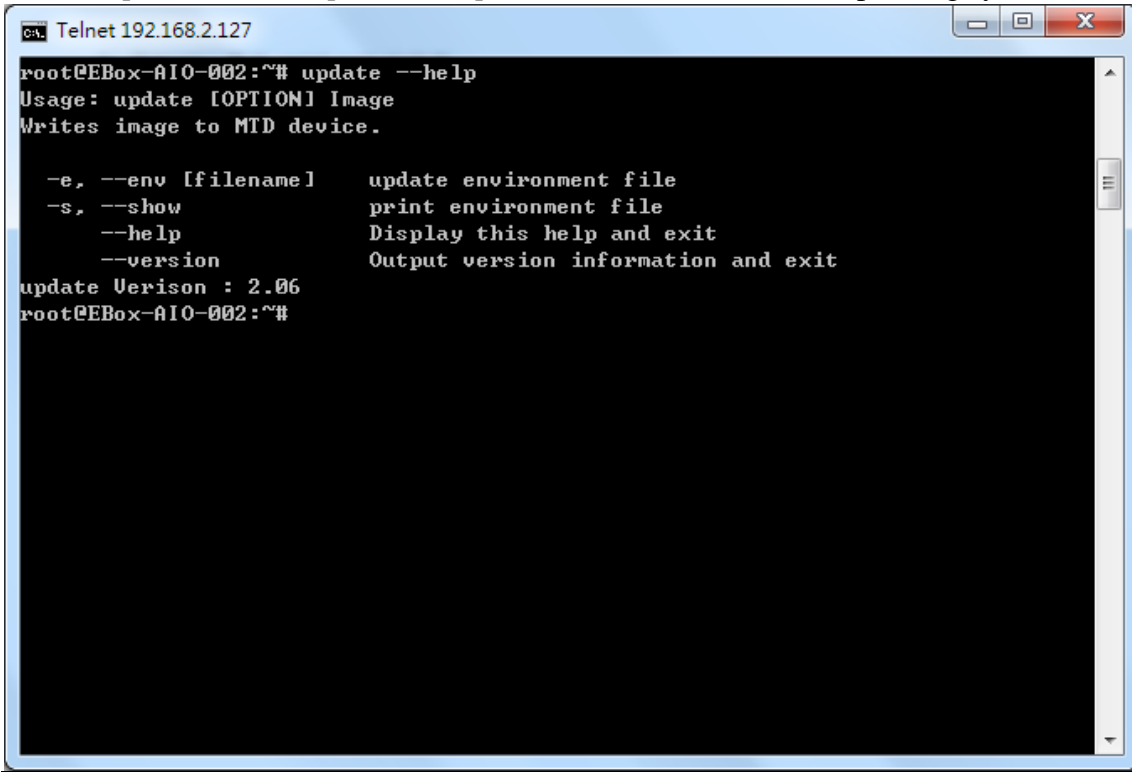
## modules to load
# at least mod_access and mod_accesslog should be loaded
# all other module should only be loaded if really necessary
# - saves some time
# - saves memory
server.modules          = <
#
#     "mod_rewrite",
#     "mod_redirect",
#     "mod_alias",
#     "mod_access",
#     "mod_cml",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



6.7. LLD dedicated tool for service

A. Update : Update loader 、 environment file 、 kernel image.

- Input command **【update --help】** , View the version of the operating system.

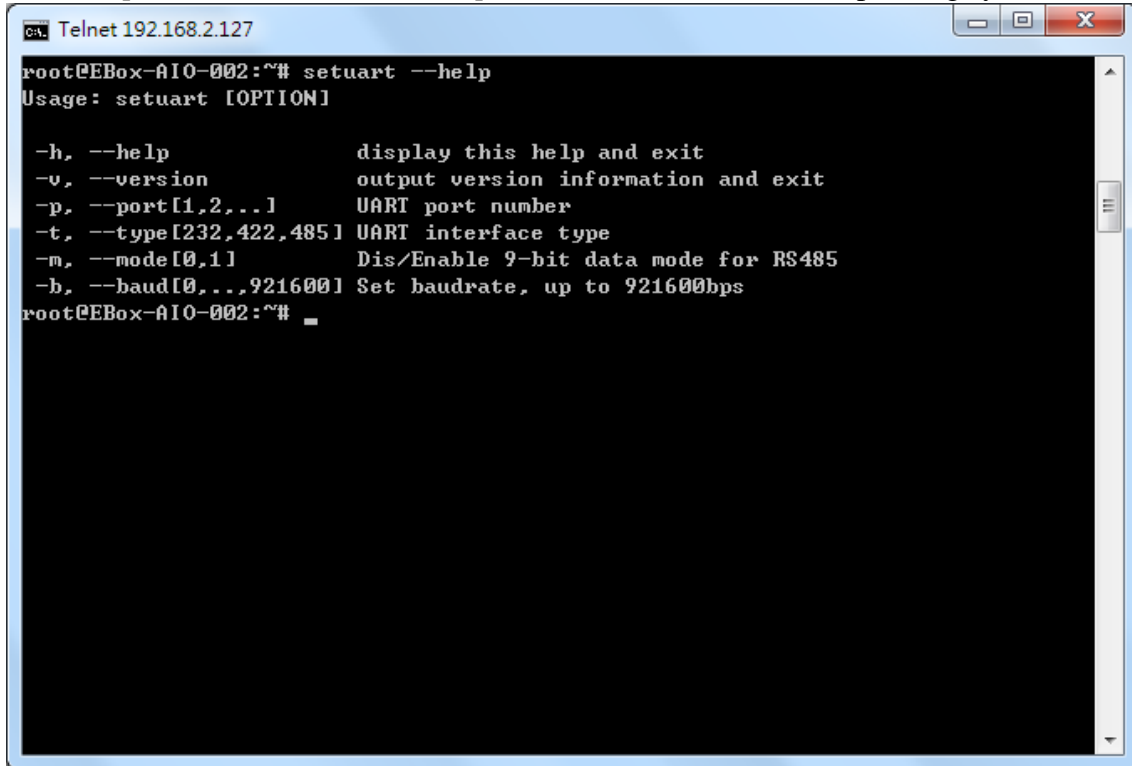


```
ca. Telnet 192.168.2.127
root@EBox-AIO-002:~# update --help
Usage: update [OPTION] Image
Writes image to MTD device.

-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 : change settings of serial port.

- Input command **【setuart --help】** , View the version of the operating system.



```
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 : View the version of the operating system.

```
ca. 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 : change GPIO

- Input command `【gpioctl --help】`, Show other related parameter.

```
ca. 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:~#
```

