

EBox-AIO-006

IIoT Gateway

操作手冊
版本: V1.1

1	產品簡介.....	1
	硬體規格.....	2
1.1	通訊及控制介面腳位定義	3
1.3	LED 指示燈說明	4
1.4	其他定義	4
1.5	各通訊及控制介面軟體定義.....	5
2	第一次操作 EBox-AIO-006	6
2.1	測試環境建議.....	6
2.2	主控口(Console)簡介	8
2.3	出廠設定值	8
2.4	操作 console 主控口	8
3	EBox-AIO-006 基本操作	14
3.1	變更網路設定.....	14
3.2	變更系統時間.....	15
3.3	下載新的 Linux 套件.....	15
3.4	FriendlyARM NanoPi NEO Module 核心模組參考技術資訊	17
附錄 A.	主機網路設定	18
附錄 B.	轉接線.....	22
附錄 C.	測試主機 COM 接口查詢	23

1 產品簡介

產品概述

EBox-AIO-006 是一款針對做為 IoT 物聯網應用的多功能通訊網關(Gateway)，主要可應用於現場端設備或儀錶的資料獲取、即時監控、記錄及與遠端或雲端的資料傳輸。

EBox-AIO-006 搭配的 Cortex-A7®及內建的 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 軟體發展工具，安裝免費的嵌入式系統專用的 C/C++ 程式編譯器(Compiler)及函式程式庫(Lib)後或更高階的開發語言如 Python、PHP...，進行二次應用程式的開發，使 EBox-AIO-006 成為一款專用的物聯網應用控制器或閘道(Gateway)。

EBox-AIO-006 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)與後臺系統連接；還具備 RS-485 通訊介面，讓 EBox-AIO-006 可連接更多不同類別的設備或儀錶。充份扮演好一個設備監控或資料處理的閘道角色。

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-006 可輕易的安裝使用在各類的場域。RJ45、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。



硬體規格

系統核心

- ▶ CPU：Allwinner H3 1.2GHz (Cortex®-A7)
- ▶ 記憶體：256MB DDR3 SDRAM、16GB (MicroSD)

網路介面

- ▶ 數量：1 組
- ▶ 類型：10/100BaseT 乙太網路(Ethernet)
- ▶ 接頭：RJ45

數位 I/O(GPIO)

- ▶ 數量：4 點
- ▶ 信號種類：3.3V CMOS
- ▶ DIP Switch x 2 GPIO
- ▶ Beeper x 1 GPIO
- ▶ LED x 1 GPIO

RS-485 串列埠介面

- ▶ 數量：1 組
- ▶ RS-485 信號：Data+, Data-, GND
- ▶ Multi-Drop Nodes：128(1/4 Load)
- ▶ 內建終端電阻：120Ω,可透過 Jumper 設定
- ▶ Pull High/Low 電阻：1KΩ
- ▶ 保護：2KV ESD 靜電保護, 400W 突波保護
- ▶ 接頭：5.00mm 可插拔端子座 x1

串列埠通訊參數

- ▶ Baud Rate：300 ~ 115.200 bps
- ▶ Parity：None, Even, Odd, Mark, Space
- ▶ Data Bits：5, 6, 7, 8
- ▶ Stop Bit：1, 1.5, 2 bits

主控台串列埠(Console)

- ▶ 數量：1 組
- ▶ 信號：UART-TTL (TxD, RxD, GND)
- ▶ 接頭：3-pin 2.54mm 排針座

USB 介面

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

SD 介面

- ▶ 數量：1 組(於 NanoPi 模組，需開殼)
- ▶ 接頭：Micro SD 插槽

人機介面

- ▶ LED：電源, 網路, 串列埠
- ▶ 蜂鳴器：1 組

機構

▷ 尺寸：75 x 75 x 26 mm (不含端子)

▷ 材質：鍍鋅鋼板

電源

▷ 工作電壓：DC 5VDC

▷ 電源接頭：5.00mm 端子座

▷ 功耗：<10W (不含 USB device)

其它

▷ Real Time Clock：1 組

▷ Real Time Clock 電池座：CR1220

▷ 適用溫度：-20~70°C

▷ 適用濕度：20%~80% RHG

▷ 認證：CE, FCC

1.1 通訊及控制介面腳位定義

工作電壓輸入

腳位	信號
左	DC+
右	GND

DC Range: **5VDC only**

5.00mm 端子座



UART Console

腳位	信號
1	RxD
2	TxD
3	GND

(需開機殼)

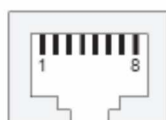
2.54mm 排針座



LAN 乙太網路介面(Ethernet)

腳位	信號
1	ETX+
2	ETX-
3	ERX+
6	ERX-

RJ45 座



蜂鳴器開關

排針	模式
開 (open)	啟用
關 (short)	停用

(需開機殼)

2.54mm 排針座



RS-485

腳位	信號
1	D+
2	D-
3	GND

5.00mm 端子座



1.3 LED 指示燈說明



Power	電源指示燈 正確接上工作電壓時就會亮起 (不可透過程式控制 LED)
Status	可做為系統狀態指示燈用 是透過 GPIO 進行控制
RxD	RS-485 資料"接收"燈號
TxD	RS-485 資料"傳送"燈號

1.4 其他定義

1.4.1 RS-485 終端電阻設定

2-pin jumper

零件編號: JP1





(需打開機殼)

排針/ jumper	模式
開(open)	啟用
關(short)	停用

1.4.2 其它

<p>RTC</p> <p>(需打開機殼、移除 CPU 模組)</p>	<ul style="list-style-type: none"> - 以 I2C 控制 - 對應核心定義：I2C_0 - RTC Battery size：CR1220 						
<p>Beeper</p> <p>(需打開機殼、移除 CPU 模組)</p>	<ul style="list-style-type: none"> - 以 GPIO 控制 (output) - 對應核心定義：PIO_G08 - 動作 <table border="1"> <thead> <tr> <th>GPIO status</th> <th>Beeper status</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>OFF</td> </tr> <tr> <td>High</td> <td>ON</td> </tr> </tbody> </table>	GPIO status	Beeper status	Low	OFF	High	ON
GPIO status	Beeper status						
Low	OFF						
High	ON						



<p>DIP Switch</p> 	<ul style="list-style-type: none"> - 以 GPIO 控制 (input) - 對應核心定義：PIO_G09、PIO_A03 - 動作 <table border="1" data-bbox="536 230 986 365"> <thead> <tr> <th>GPIO status</th> <th>Switch status</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>ON</td> </tr> <tr> <td>High</td> <td>OFF</td> </tr> </tbody> </table>	GPIO status	Switch status	Low	ON	High	OFF
GPIO status	Switch status						
Low	ON						
High	OFF						
<p>Programmable LED</p> 	<p>Status LED</p> <ul style="list-style-type: none"> - 以 GPIO 控制 (output) - 對應核心定義：PIO_A06 - 動作 <table border="1" data-bbox="536 589 986 723"> <thead> <tr> <th>GPIO status</th> <th>LED status</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>Light ON</td> </tr> <tr> <td>High</td> <td>Off</td> </tr> </tbody> </table>	GPIO status	LED status	Low	Light ON	High	Off
GPIO status	LED status						
Low	Light ON						
High	Off						

1.5 各通訊及控制介面軟體定義

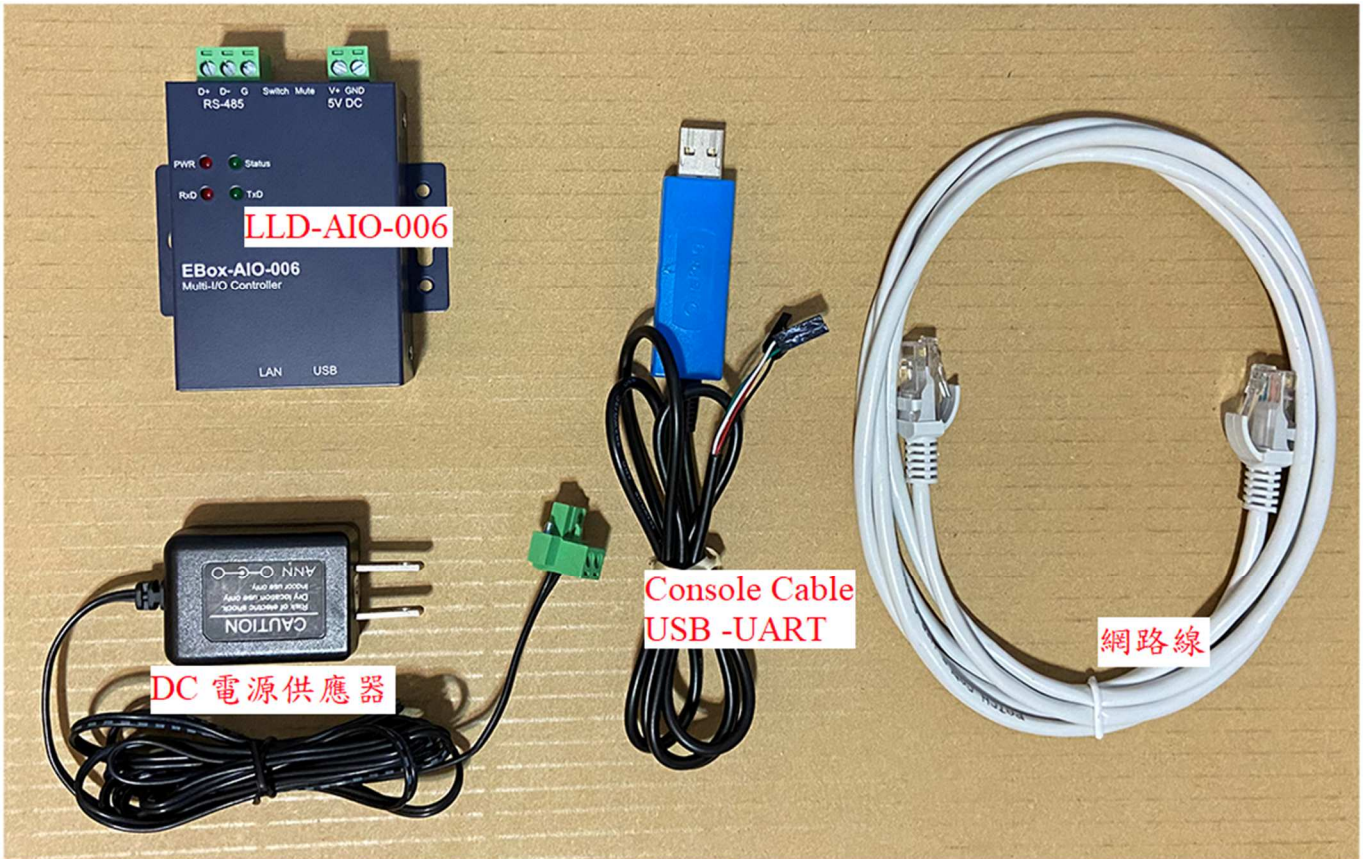
介面型態	標示	軟體定義
網路	LAN1	eth0
Console	X	/dev/ttyS0
RS-485	RS-485	/dev/ttyS1
外掛記憶體	USB Disk	/dev/sda? (依設備先後插入順序有所不同)

2 第一次操作 EBox-AIO-006

2.1 測試環境建議

2.1.1. 測試所需器材：

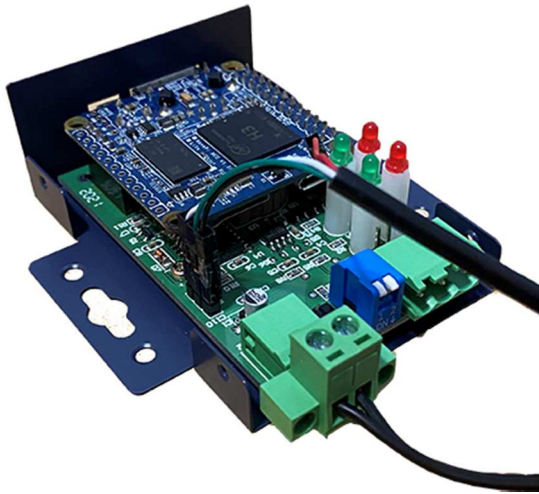
- EBox-AIO-006 一台
- DC 5V 電源供應器一個(輸出功耗需大於>10W)
(如電源供應器接頭與 EBox-AIO-006 不同,可自製轉接線轉換為一般接線,參考附錄 B. 轉接線參考)
- 網路線或網路交叉線一條
- USB to UART console 轉接線一條



2.1.2. 連接安裝

(console 連接)

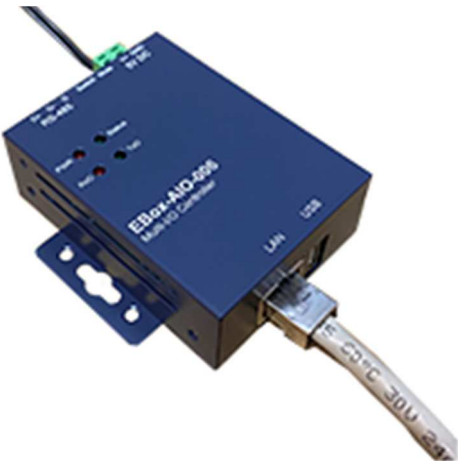
將 Console 線與測試主機連接，做為開發主機於 EBox-AIO-006 的通訊橋樑。



(參考附錄 B. 轉接線參考)

(網路連接)

將網路與測試主機連接，做為開發主機於 EBox-AIO-006 的通訊橋樑。



2.2 主控口(Console)簡介

EBox-AIO-006 出廠具備兩個主控口介面，一個是透過網路，另一個可透過 UART(RS-232)介面，系統開發者可以透過主控口對 EBox-AIO-006 進行完全的控制及開發。除了一般的參數變更外，更可以直接進行系統、程式開發或系統套件的安裝及更新。

2.3 出廠設定值

A. 登入密碼 (有大小寫之分)

管理者帳戶	Login Name	root
root	Password	fa

B. 主控口通訊參數

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

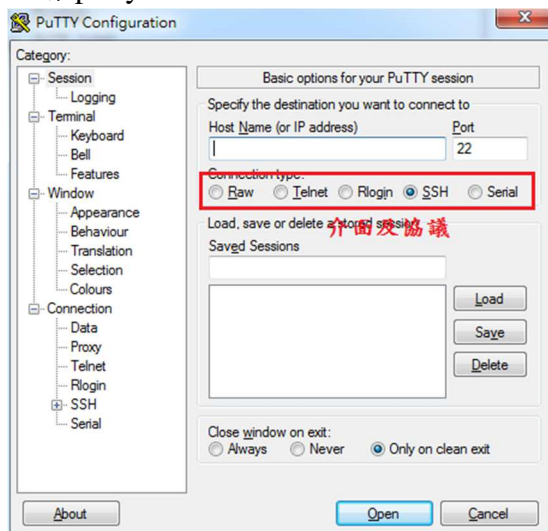
2.4 操作 console 主控口

2.4.1 console 工具程式-putty

putty 為 windows 作業系統中用來進行遠端通訊的一套工具軟體，常用連接於有主控接口的開發平台，如 EBox-AIO-006

- 下載 putty
<https://putty.org>

- 啟動 putty

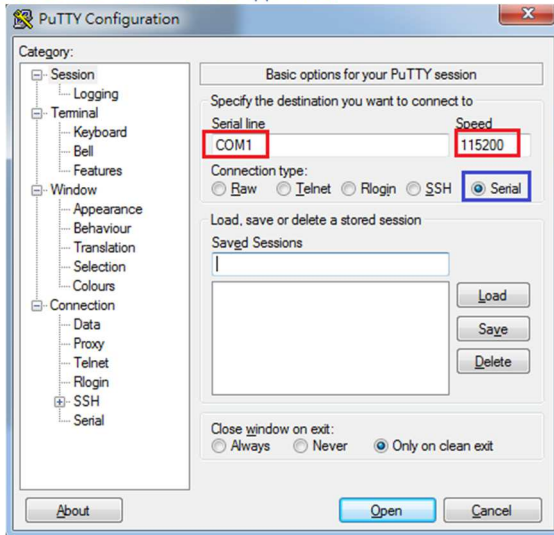


2.4.2 進入 UART Console

選擇 “Serial”通訊介面

輸入 “COM1” (註*)及選擇通訊速度 ”115200”

註*: COM#為主機 RS-232 接口，統稱 COM 口的編號，透過 USB 轉接線擴充的編號需自主機系統中查詢。(可參考“附錄 C.”)

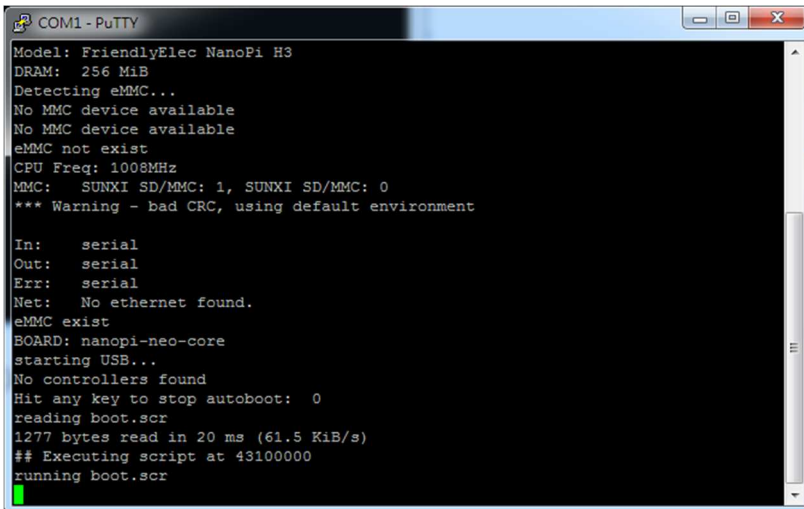


■ 啟動 putty



■ EBox-AIO-006 (通電)

顯示 Junminx-AIO-006 開機的信息



- ◆ EBox-AIO-006 連接到有 SHCP Server 的網路
取得有效 IP address : 192.168.0.178

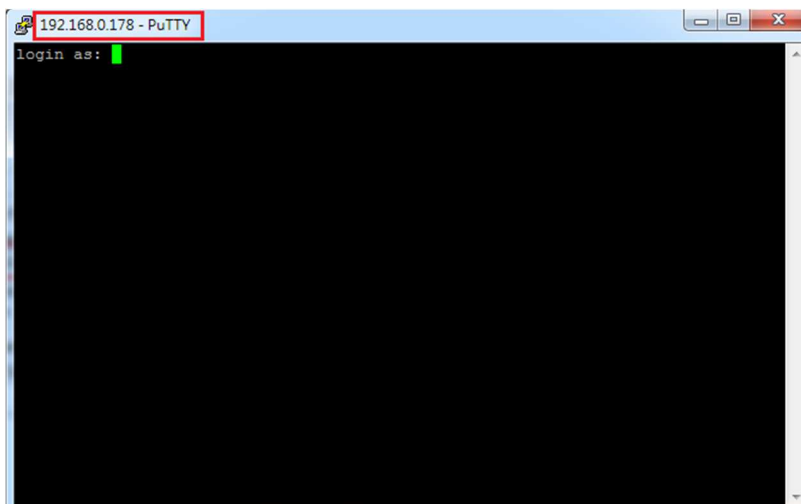
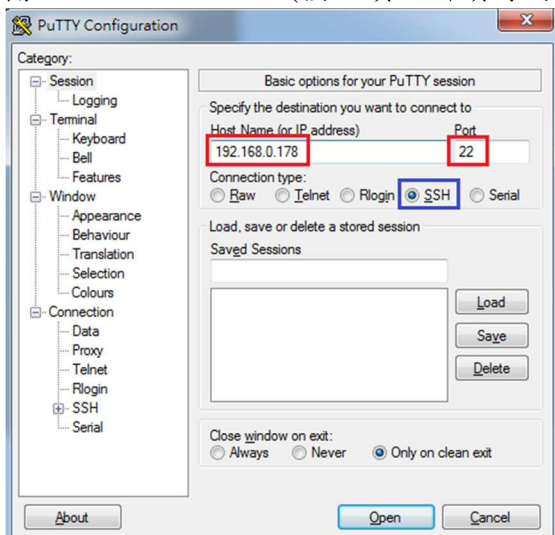
```

pi@NanoPi-NEO-Core:~$
pi@NanoPi-NEO-Core:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:81:81:6c:91:bb
          inet addr:192.168.0.178  Bcast:192.168.0.255  Mask:255.255.0
          inet6 addr: fe80::670a:c25c:8048:4c73/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:68  errors:0  dropped:0  overruns:0  frame:0
          TX packets:51  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6402 (6.4 KB)  TX bytes:4844 (4.8 KB)
          Interrupt:41

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:680  errors:0  dropped:0  overruns:0  frame:0
          TX packets:680  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:54402 (54.4 KB)  TX bytes:54402 (54.4 KB)

pi@NanoPi-NEO-Core:~$
    
```

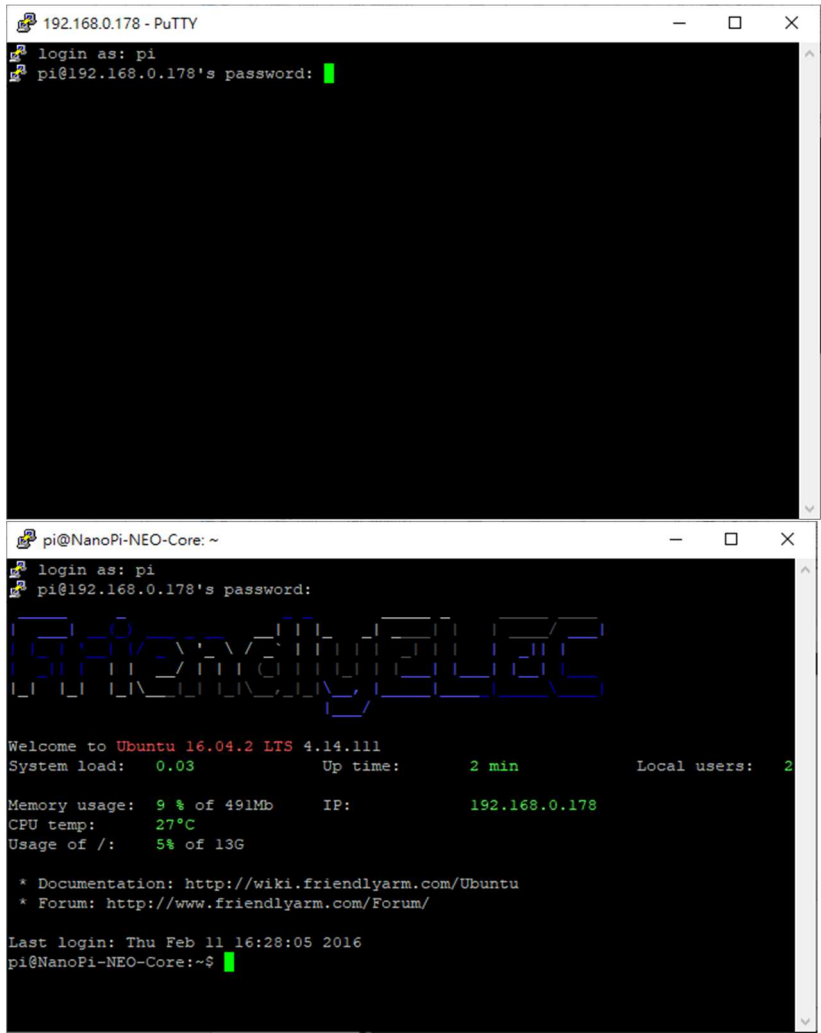
- 啟動 putty
選擇 “SSH”通訊介面
輸入 “192.168.0.178 (接上續、所讀得的 EBox-AIO-006 IP address)”



- 進入 EBox-AIO-006 Ethernet console
輸入 EBox-AIO-006 出廠的一般用戶名及密碼

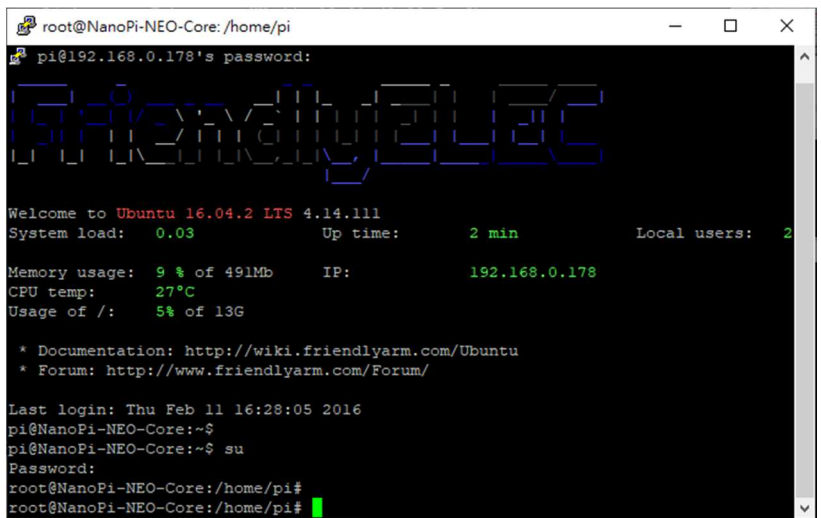
Login Name	pi
------------	----

Password	pi
----------	----



如需升級為管理層級，需執行指令”su”，並輸入 root 的密碼
輸入 EBox-AIO-006 出廠的一般用戶名及密碼

Password	fa
----------	----



- 無法連入 EBox-AIO-006 Ethernet console



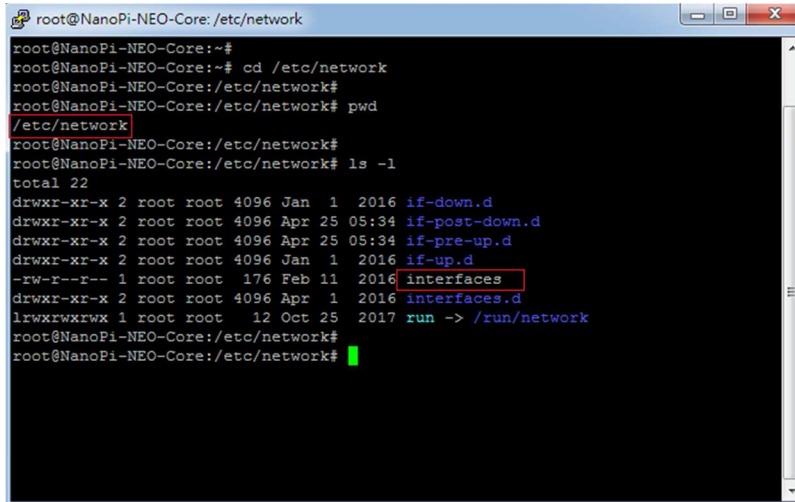
此狀況主要發生的原因常見有 3:

1. 輸入錯誤的 EBox-AIO-006 的 IP address
2. EBox-AIO-006 的網路有問題
 - EBox-AIO-006 的網路線沒接好
 - 不知 EBox-AIO-006 的網路 IP address 設定
3. EBox-AIO-006 和測試主機的網路設定在不同的網段
需查看主機的網路設定是否和 EBox-AIO-006 在同一 IP 網段，可參考”附錄 A”

3 EBox-AIO-006 基本操作

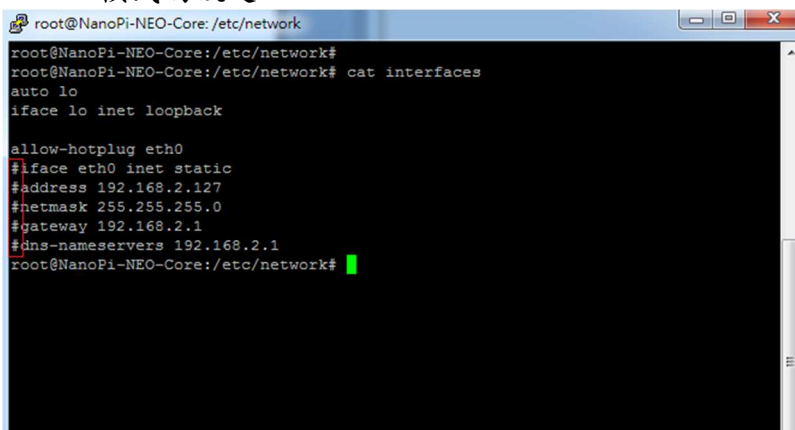
3.1 變更網路設定

EBox-AIO-006 的出廠設定是 DHCP mode，所連接的網路上需有一台 DHCP Server 功能的主機分配 IP address 給 EBox-AIO-006，如要將 EBox-AIO-006 變更為自定義的固定 IP 模式，需變更一系統設定檔 “interfaces”，所在路徑: "/etc/network"



```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:~#
root@NanoPi-NEO-Core:~# cd /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# pwd
/etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# ls -l
total 22
drwxr-xr-x 2 root root 4096 Jan  1 2016 if-down.d
drwxr-xr-x 2 root root 4096 Apr 25 05:34 if-post-down.d
drwxr-xr-x 2 root root 4096 Apr 25 05:34 if-pre-up.d
drwxr-xr-x 2 root root 4096 Jan  1 2016 if-up.d
-rw-r--r-- 1 root root 176 Feb 11 2016 interfaces
drwxr-xr-x 2 root root 4096 Apr  1 2016 interfaces.d
lrwxrwxrwx 1 root root  12 Oct 25 2017 run -> /run/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network#
```

■ DHCP 模式的設定



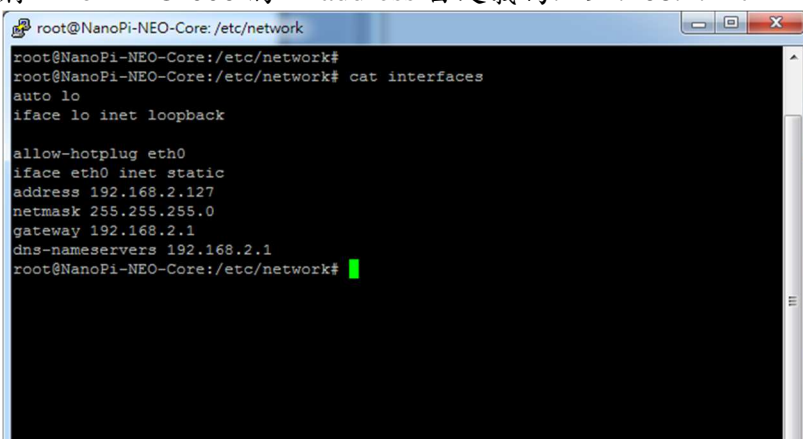
```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# cat interfaces
auto lo
iface lo inet loopback

allow-hotplug eth0
#iface eth0 inet static
#address 192.168.2.127
#netmask 255.255.255.0
#gateway 192.168.2.1
#dns-nameservers 192.168.2.1
root@NanoPi-NEO-Core:/etc/network#
```

設定命令列前端多了“#”，此命令列就不執行

■ 固定 IP Address 的設定

將 EBox-AIO-006 的 IP address 自定義為: 192.168.2.127




```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# cat interfaces
auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet static
address 192.168.2.127
netmask 255.255.255.0
gateway 192.168.2.1
dns-nameservers 192.168.2.1
root@NanoPi-NEO-Core:/etc/network#
```


3.2 變更系統時間

EBox-AIO-006 正常運行時，會有兩個時間同時運行，一個是系統時間，可透過“date”這個指令查看或變更，另一個是 HW RTC(實時時鐘)的時間，可透過”hwclock”這指令查看及變更。

系統時間易受元件老化、運行的環境溫度等等影響，較容易發生時間不準的狀況，通常可透過網路校時，如 NTP 協議或命令，另外就是透過與 EBox-AIO-006 本機的 HW RTC 進行時間同步。

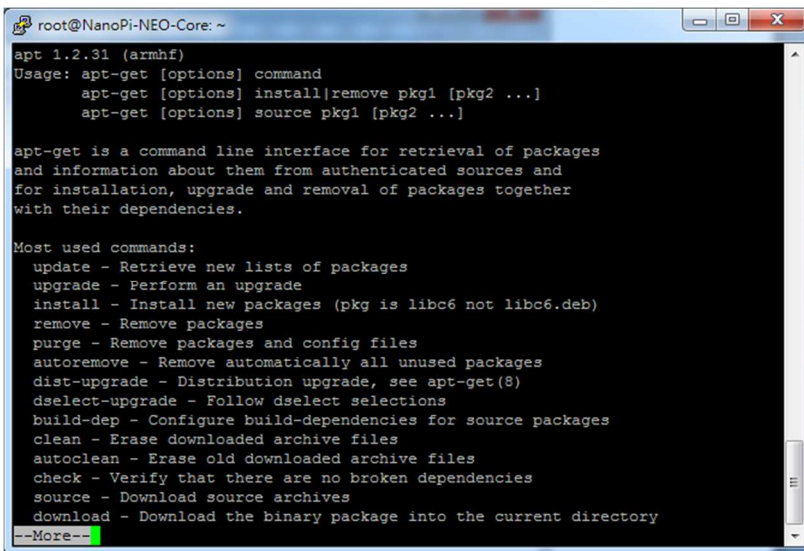


```
root@NanoPi-NEO-Core: ~  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~# date 查看系統時間  
Wed Aug 14 20:01:25 UTC 2019  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~# date 081421002019 變更系統時間  
Wed Aug 14 21:00:00 UTC 2019  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~# hwclock -s 自 HW RTC 讀回時間成系統時間  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~# date 驗證 HW RTC 取回的時間  
Wed Aug 14 20:02:02 UTC 2019  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~#
```

3.3 下載新的 Linux 套件

EBox-AIO-006 目前搭載的是 Linux 作業系統，開發或使用 EBox-AIO-006 可能需要安裝其它的開發套件或應用服務。

目前最常使用的方式，就是將 EBox-AIO-006 安裝在有連接到 Internet 的網路環境，透過”apt-get”這個安裝套件的工具程式進行軟體安裝。



```
root@NanoPi-NEO-Core: ~  
apt 1.2.31 (armhf)  
Usage: apt-get [options] command  
       apt-get [options] install|remove pkg1 [pkg2 ...]  
       apt-get [options] source pkg1 [pkg2 ...]  
  
apt-get is a command line interface for retrieval of packages  
and information about them from authenticated sources and  
for installation, upgrade and removal of packages together  
with their dependencies.  
  
Most used commands:  
update - Retrieve new lists of packages  
upgrade - Perform an upgrade  
install - Install new packages (pkg is libc6 not libc6.deb)  
remove - Remove packages  
purge - Remove packages and config files  
autoremove - Remove automatically all unused packages  
dist-upgrade - Distribution upgrade, see apt-get(8)  
dselect-upgrade - Follow dselect selections  
build-dep - Configure build-dependencies for source packages  
clean - Erase downloaded archive files  
autoclean - Erase old downloaded archive files  
check - Verify that there are no broken dependencies  
source - Download source archives  
download - Download the binary package into the current directory  
--More--
```

- 例：安裝 Apach Web Server
執行 apt-get install apache2

```
root@NanoPi-NEO-Core:~  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~# apt-get install apache2
```

安裝過程中，或許會有些提問要做確認

```
root@NanoPi-NEO-Core:~  
root@NanoPi-NEO-Core:~#  
root@NanoPi-NEO-Core:~# apt-get install apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert  
Suggested packages:  
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom ufw  
  openssl-blacklist  
The following NEW packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert  
0 upgraded, 10 newly installed, 0 to remove and 231 not upgraded.  
Need to get 1,411 kB of archives.  
After this operation, 4,974 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

不同的套件，安裝需要的時間不一樣

```
root@NanoPi-NEO-Core:~  
Setting up apache2-utils (2.4.18-2ubuntu3.10) ...  
Setting up apache2-data (2.4.18-2ubuntu3.10) ...  
Setting up apache2 (2.4.18-2ubuntu3.10) ...  
Enabling module mpm_event.  
Enabling module authz_core.  
Enabling module authz_host.  
Enabling module authn_core.  
Enabling module auth_basic.  
Enabling module access_compat.  
Enabling module authn_file.  
Enabling module authz_user.  
Enabling module alias.  
Enabling module dir.  
Enabling module autoindex.  
Enabling module env.  
Enabling module mime.  
Enabling module negotiation.  
Enabling module setenvif.  
Enabling module filter.  
Enabling module deflate.  
Enabling module status.  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.
```

```

root@NanoPi-NEO-Core: ~
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Setting up ssl-cert (1.0.37) ...
Processing triggers for libc-bin (2.23-0ubuntu7) ...
Processing triggers for systemd (229-4ubuntu16) ...
root@NanoPi-NEO-Core:~#

```

確認 Apache 已安裝，並且已運行

```

root@NanoPi-NEO-Core: ~
pi      733    554    0 19:29 tty1      00:00:00 -bash
pi      734    551    0 19:29 ttyS0    00:00:00 -bash
nobody  758    468    0 19:29 ?          00:00:00 /usr/sbin/dnsmasq --no-resolv --
root    848     2    0 19:30 ?          00:00:00 [kworker/0:1H]
root    1529    2    0 19:37 ?          00:00:00 [kworker/u8:1]
root    1556    2    0 19:37 ?          00:00:00 [kworker/2:0]
root    1559    519    0 19:37 ?          00:00:00 sshd: root@pts/0
root    1569    1    0 19:37 ?          00:00:00 /lib/systemd/systemd --user
root    1572    1569  0 19:37 ?          00:00:00 (sd-pam)
root    1652    1559  0 19:37 pts/0      00:00:00 -bash
root    1708    2    0 19:44 ?          00:00:00 [kworker/0:0]
root    1786    519    0 19:45 ?          00:00:00 sshd: root@pts/1
root    1888    1786  0 19:45 pts/1      00:00:00 -bash
root    2081    1888  0 19:58 pts/1      00:00:00 su
root    2086    2081  0 19:58 pts/1      00:00:00 bash
root    2192    2    0 20:02 ?          00:00:01 [kworker/0:1]
root    2219    2    0 20:14 ?          00:00:00 [kworker/u8:0]
root    2438    2    0 20:19 ?          00:00:00 [kworker/3:0]
root    2451    2    0 20:21 ?          00:00:00 [kworker/1:1]
root    2461    2    0 20:23 ?          00:00:00 [kworker/u8:2]
root    2478    2    0 20:26 ?          00:00:00 [kworker/1:0]
root    2895    1    0 20:26 ?          00:00:00 /usr/sbin/apache2 -k start
www-data 2898  2895  0 20:26 ?          00:00:00 /usr/sbin/apache2 -k start
www-data 2899  2895  0 20:26 ?          00:00:00 /usr/sbin/apache2 -k start
root    3019    2086  0 20:28 pts/1      00:00:00 ps -ef
root@NanoPi-NEO-Core:~#

```

3.4 FriendlyARM NanoPi NEO Module 核心模組參考技術資訊

如需更詳細的系統核心模組的相關資訊，可至系統核心模組的產品官網進行查閱

NanoPi NEO : https://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO

附錄 A. 主機網路設定

WINDOWS 10 的環境設定

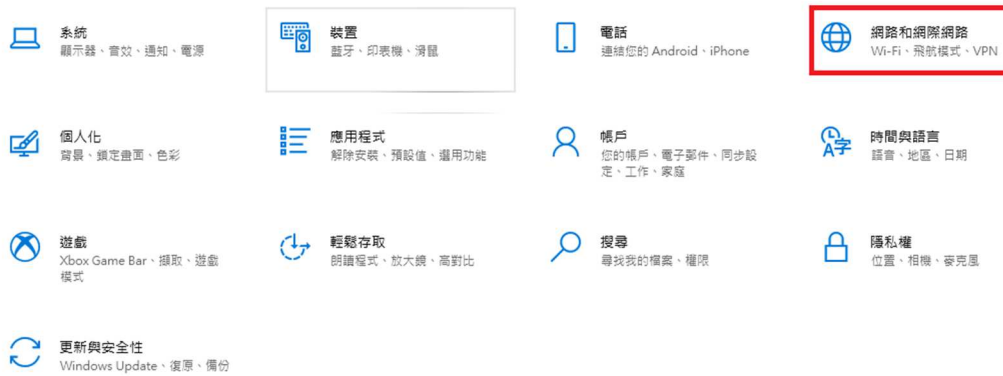
Step.01 開始 → 控制台 → 網路和網際網路-檢視網路狀態及工作。



設定

Windows 設定

尋找設定



Step.02 變更介面卡設定。



狀態

網路狀態



乙太網路
公用網路

您已連線到網際網路

若您使用的行動數據方案受限，可將此網路設為計量付費連線或變更其他內容。

乙太網路 51.4 GB
最近 30 天內

內容

數據使用量

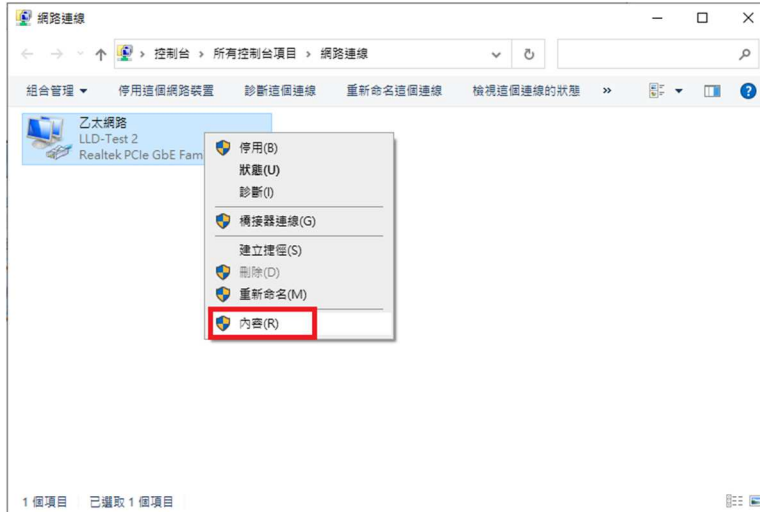
顯示可用網路
檢視您周圍的連線選項。

進階網路設定

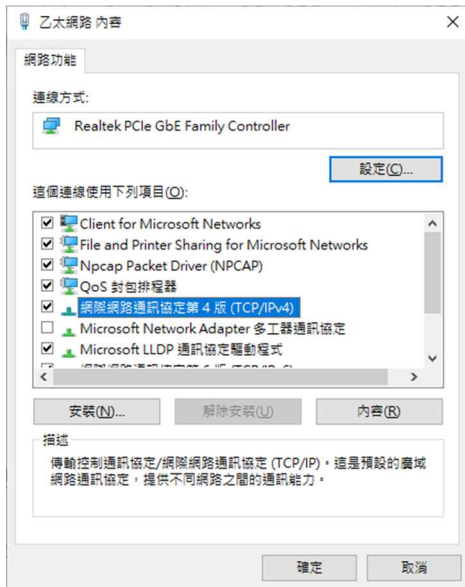
變更介面卡選項
檢視網路介面卡及變更連線設定。

網路和共用中心

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 地址：

確認測試主機和 EBox-AIO-009 一般，都是在 DHCP(被分配 IP)的模式下



Step.06 設定完畢點擊“確定”鍵。



附錄 B. 轉接線

B1. Console 線



USB Converter		Console (CON1)
Tx (白)	↔	RxD (pin-1)
Rx (綠)	↔	TxD (pin-2)
GND (黑)	↔	GND (pin-3)
DC (紅)	X	

- 不同廠家 USB 轉換器的腳位元定義和配色模式並非固定，需參考廠家的規格書

B2. 電源轉接線

DC Power Jack 轉 Terminal Block



內徑Φ2.1mm powerjack

附錄 C. 測試主機 COM 接口查詢

