

# EBox-AIO-008

## IIoT Gateway

操作手冊  
版本: V1.1

1	產品簡介.....	1
1.1	硬體規格.....	2
1.3	通訊及控制介面腳位元定義.....	4
1.4	LED 指示燈及定義說明.....	5
1.5	其他定義.....	6
1.6	各通訊及控制介面軟體定義.....	7
1.7	各擴充通訊晶片規格.....	8
2	第一次操作 EBox-AIO-008.....	9
2.1	測試環境建議.....	9
2.2	主控口(Console)簡介.....	11
2.3	出廠設定值.....	11
2.4	操作 console 主控口.....	11
3	EBox-AIO-008 基本操作.....	17
3.1	變更網路設定.....	17
3.2	下載新的 Linux 套件.....	18
3.3	GPS 操作.....	20
3.4	安裝 4G 行動通訊.....	21
3.5	安裝 SSD 硬碟.....	22
3.6	更新 RS-485 驅動程式(driver).....	23
3.7	FriendlyARM NanoPi NEO Module 核心模組參考技術資訊.....	24
附錄 A.	主機網路設定.....	25
附錄 B.	轉接線.....	28
附錄 C.	測試主機 COM 介面查詢.....	29



## 1 產品簡介

### 產品概述

EBox-AIO-008 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表的資料獲取、即時監控、記錄及與遠端或雲端的資料傳輸。

EBox-AIO-008 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-008 成為一款專用的物聯網應用控制器或閘道(Gateway)。

EBox-AIO-008 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，並可擴充 4G 行動通訊功能；還具備多路 RS-485 通訊介面，讓 EBox-AIO-008 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角色。EBox-AIO-008 同時也具備數位控制(GPIO)的功能，搭配專用的 Digital I/O 轉換模組與外接模組就可以進行現場端的開關控制。

EBox-AIO-008 更具備了 LoRa、GPS 及 SSD 等進階功能，LoRa 是目前無線物聯網常見的一款無線通訊功能，常用於小場域的分散監控。GPS 除可做定位外，更常用於時間的校正。而 SSD 主要可擴充 SSD Disk 讓 EBox-AIO-008 可增加更多的數據儲存容量。

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-008 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。



## 1.1 硬體規格

### 系統核心

- ▶ CPU：Allwinner H3 1.2GHz (Cortex®-A7)
- ▶ 記憶體：512MB DDR3 SDRAM、8GMB eMMC Flash

### 網路介面

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

### 數位 I/O(GPIO)

- ▶ 數量：16 點
- ▶ 信號種類：3.3V CMOS
- ▶ 2.54mm 牛角針座 x 10 GPIO
- ▶ DIP Switch x 2 GPIO
- ▶ LED x 3 GPIO
- ▶ Beeper x 1 GPIO
- ▶ 電壓範圍：0 / 3.3VDC

### RS-485 序列埠介面

- ▶ 數量：8 組
- ▶ RS-485 信號：Data+, Data-, GND (支援自動流向控制)
- ▶ Multi-Drop Nodes：128
- ▶ 信號終端電阻：120/600Ω可選 (DIP Switch)
- ▶ 信號拉平電阻：1K/10KΩ可選 (DIP Switch)
- ▶ 保護：2KV 隔離保護, 15KV ESD 靜電保護, 400W 突波保護
- ▶ 接頭：5.00mm 可插拔端子座

### 序列埠通訊參數

- ▶ Baud Rate：300 ~ 460,800 bps
- ▶ Parity：None, Even, Odd, Mark, Space
- ▶ Data Bits：5, 6, 7, 8
- ▶ Stop Bit：1, 2 bits

### 控制台串列埠(Console)

- ▶ 數量：1 組
- ▶ 信號：UART-TTL (TxD, RxD, GND)
- ▶ 終端樣式：115,200 bps, VT-100
- ▶ 接頭：2.54mm 排針座 (需開機殼)

### Wireless 無線擴充介面

- ▶ 數量：1 組(需開殼)
- ▶ 適用模組：4G、5G、Wi-Fi、NB-IoT
- ▶ 擴充信號：USB、UART、GPIO
- ▶ 接頭：Mini-PCIe 插槽

### GPS

- ▶ 數量：1 組

▷ 介面：UART

▷ 信號：BDS/GPS/GLONASS

### LoRa 無線通訊

▷ 數量：1 組

▷ 介面：UART

▷ 規格：410~525 MHz

▷ 發設功率：0.1Watt

### SSD 電子盤擴充介面

▷ 數量：1 組

▷ 介面：USB

▷ 規格：SATA III

▷ 接頭：M.2 2242

### Audio 輸出功能

▷ 數量：1 組

▷ 介面：3.5MM 身歷聲介面

### 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(於 NanoPi 模組，需開殼)

▷ 接頭：Micro SD 插槽

### 機構

▷ PCB 尺寸：鍍鋅鋼板 / 245 x 140 x 35 mm

### 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

▷ Beeper : 1 組

▷ LED 指示燈：電源，網路，序列埠，控制信號，4G、GPS、用戶自訂

▷ 適用溫度：-20~70°C

▷ 適用濕度：20%~80% RHG

▷ 認證：CE, FCC

### 1.3 通訊及控制介面腳位元定義

#### 工作電壓輸入

腳位	信號
右	DC+
左	GND

- DC Range: 9~24V DC

5.00mm 端子座

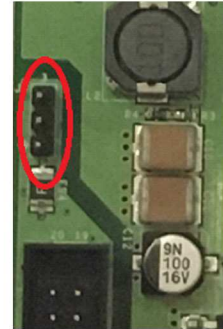


#### 5VDC 電壓輸出(for 外接散熱風扇)

需開蓋-零件編號: J2

腳位	信號
3	X
2	GND
1 (下)	5V+

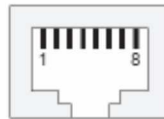
2.54mm 排針座



#### LAN 乙太網路介面(Ethernet)

腳位	信號
1	ETX+
2	ETX-
3	ERX+
6	ERX-

RJ45 座



#### UART Console

腳位	信號
1 (左)	RxD
2	TxD
3	GND

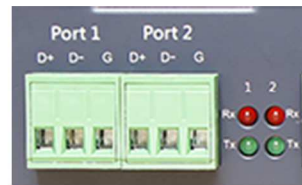
2.54mm 排針座



#### RS-485 (port 1~8)

腳位	信號
左 1	D+
左 2	D-
左 3	GND

5.00mm 端子座



### 1.4 LED 指示燈及定義說明

上側面



DC	U-def1 *	U-def2 *	LorA RxD
Status *	4G	GPS	LorA TxD

\* 為軟體可控制的 LED

下側面



紅燈：RS-485 RxD

綠燈：RS-485 TxD

Programmable LED

#### Status LED

- 以 GPIO 控制 (output)
- 對應核心定義：PIO\_A06
- 動作

GPIO status	LED status
Low	Light ON
High	Off

#### U-def1 LED

- 以 GPIO 控制 (output)
- 對應核心定義：PIO\_C00
- 動作

GPIO status	LED status
Low	Light ON
High	Off

#### U-def2 LED

- 以 GPIO 控制 (output)
- 對應核心定義：PIO\_C01
- 動作

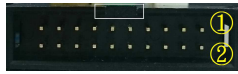
GPIO status	LED status
Low	Light ON
High	Off

## 1.5 其他定義

### 1.5.1 GPIO 定義

2.54mm

2x10 牛角排針座



零件編號: CON14

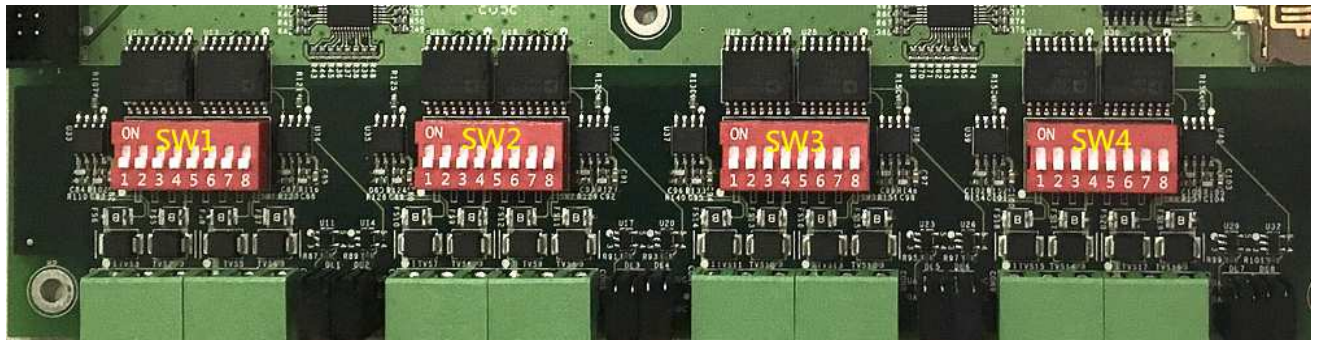
對應核心定義	定義	Pin#	Pin#	定義	對應核心定義
PIO_G08	GPIO-01	01	02	GPIO-02	PIO_G09
PIO_A12	GPIO-03	03	04	GPIO-04	PIO_A11
PIO_C03	GPIO-05	05	06	GPIO-06	PIO_A17
PIO_A07	GPIO-07	07	08	GPIO-08	PIO_L11
PIO_A16	GPIO-09	09	10	GPIO-10	PIO_A15
	X	11	12	X	
	X	13	14	X	
	X	15	16	X	
GND	GND	17	18	GND	GND
	X	19	20	X	

- 單雙號腳位”上、下”並排

### 1.5.2 RS-485 電位電阻設定

需打開機殼

8-pin dip switch



零件編號: SW1~SW4

腳位	RS-485	ON 定義	OFF 定義	備註
1	SW1 → Port1 SW2 → Port3 SW3 → Port5 SW4 → Port7	data-腳位 1KΩ 電位拉低	data-腳位 10KΩ 電位拉低	腳位 1、2 需 同時切換
2		data+腳位 1KΩ 電位拉高	data+腳位 10KΩ 電位拉高	
3	SW1 → Port2 SW2 → Port4 SW3 → Port6 SW4 → Port8	120Ω 終端電阻	無終端電阻	腳位 3、4 不 可同時切換
4		600Ω 終端電阻	無終端電阻	
5	SW1 → Port2 SW2 → Port4 SW3 → Port6 SW4 → Port8	data-腳位 1KΩ 電位拉低	data-腳位 10KΩ 電位拉低	腳位 1、2 需 同時切換
6		data+腳位 1KΩ 電位拉高	data+腳位 10KΩ 電位拉高	
7	SW1 → Port2 SW2 → Port4 SW3 → Port6 SW4 → Port8	120Ω 終端電阻	無終端電阻	腳位 3、4 不 可同時切換
8		600Ω 終端電阻	無終端電阻	



### 1.5.3 其它

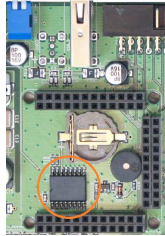
#### DIP Switch



- 以 GPIO 控制 (input)
- 對應核心定義：PIO\_A02、PIO\_A03
- 動作

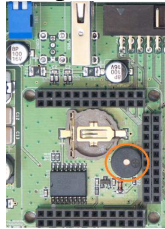
GPIO status	Switch status
Low	ON
High	OFF

#### RTC



- 以 I2C 控制
- 對應核心定義：I2C\_2
- RTC Battery size：CR1220

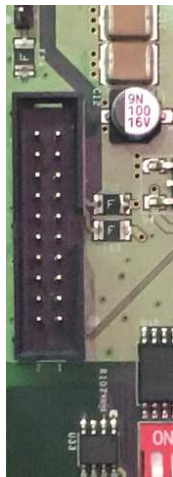
#### Beeper



- 以 GPIO 控制 (output)
- 對應核心定義：PIO\_G11
- 動作

GPIO status	Beeper status
Low	OFF
High	ON

#### 備用擴充介面



- 腳位及功能定義

定義	功能	Pin #	Pin #	功能	定義
UART3_TxD	UART	01	02	UART	UART3_RxD
		03	04		
PIO_G08	GPIO	05	06	GPIO	PIO_G09
PIO_A12	GPIO	07	08	GPIO	PIO_A11
PIO_C03	GPIO	09	10	GPIO	PIO_A17
PIO_A07	GPIO	11	12	GPIO	PIO_L11
PIO_A16	GPIO	13	14	GPIO	PIO_A15
USB D+	USB	15	16	USB	USB D-
5V	DC out	17	18	DC out	3.3V
GND	GND	19	20	GND	GND

### 1.6 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
	LAN0	eth1
RS-485	Port 1	/dev/ttyXRUSB0
	Port 2	/dev/tty XRUSB1
	Port 3	/dev/tty XRUSB2



	Port 4	/dev/tty XRUSB3
	Port 5	/dev/tty XRUSB4
	Port 6	/dev/tty XRUSB5
	Port 7	/dev/tty XRUSB6
	Port 8	/dev/tty XRUSB7
Console	(UART / TTL)	/dev/ttyS0
GPS	(UART / TTL)	/dev/ttyS1
LoRa	(UART / TTL)	/dev/ttyS2
擴充槽	(UART / TTL)	/dev/ttyS3
外掛記憶體	USB Disk	/dev/sda?、/dev/sdb? (依和 SSD 設備先後插入順序有所不同)
	SSD Disk	/dev/sda?、/dev/sdb? (依和 USB 設備先後插入順序有所不同)
	SD	/dev/mmc0

### 1.7 各擴充通訊晶片規格

功能	介面	晶片規格
LAN2	USB	RTL8152B / RealTek
RS-485	USB	XR21V1414M48 / Maxim
4G	USB	
M.2 SSD Disk	USB	JMS578 / JMicron
GPS	USB	ATGM332D / icofchina
LoRa	UART	E32-400T20S / eByte

## 2 第一次操作 EBox-AIO-008

### 2.1 測試環境建議

#### 2.1.1. 測試所需器材：

- EBox-AIO-008 一台
- DC 9~24V 電源供應器 一個(輸出功耗需大於>24W)  
(如電源供應器接頭與 EBox-AIO-008 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
- EBox-AIO-008 專用 Console 線, USB 轉 UART 轉換線



### 2.1.2. 連接安裝

將網路與 Console 線都與測試主機連接，做為開發主機於 EBox-AIO-008 的通訊橋樑。



#### ■ Console 連接說明

◆ 依 console 線腳位定義連接 EBox-AIO-008 console 排針

EBox-AIO-008		Console 線
RxD (pin-1)	↔	TxD
TxD (pin-2)	↔	RxD
GND (pin-3)	↔	GND

## 2.2 主控口(Console)簡介

EBox-AIO-008 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 介面，系統開發者可以透過主控口對 EBox-AIO-008 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
UART (TTL)	Console	Baud Rate : 115,200 bps Data Format : N-8-1 Flow Control : None Terminal Type : VT-100

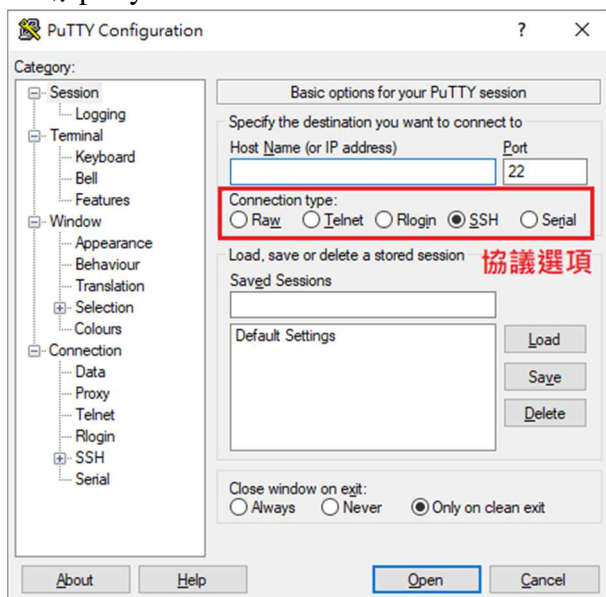
## 2.4 操作 console 主控口

### 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，常用連接於有主控介面的開發平臺，如 EBox-AIO-008

- 下載 putty  
<https://putty.org>

- 啟動 putty



### 2.4.2 進入 UART Console

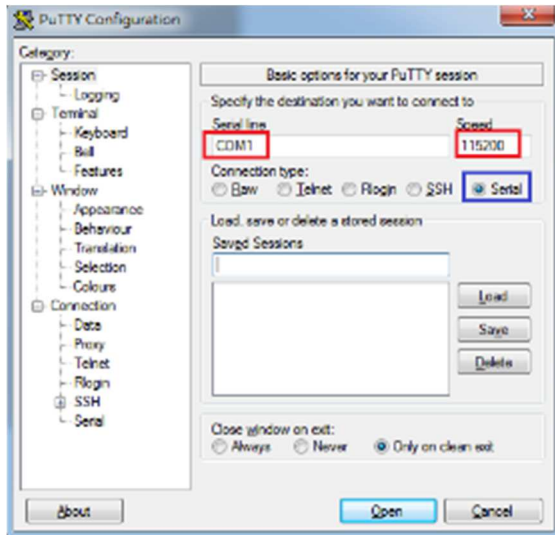
選擇 “Serial”通訊介面

輸入 “COM1” (注\*)及選擇通訊速度 ”115200”

注\* : COM#為主機串口(TTL、RS-232 介面)，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.”)

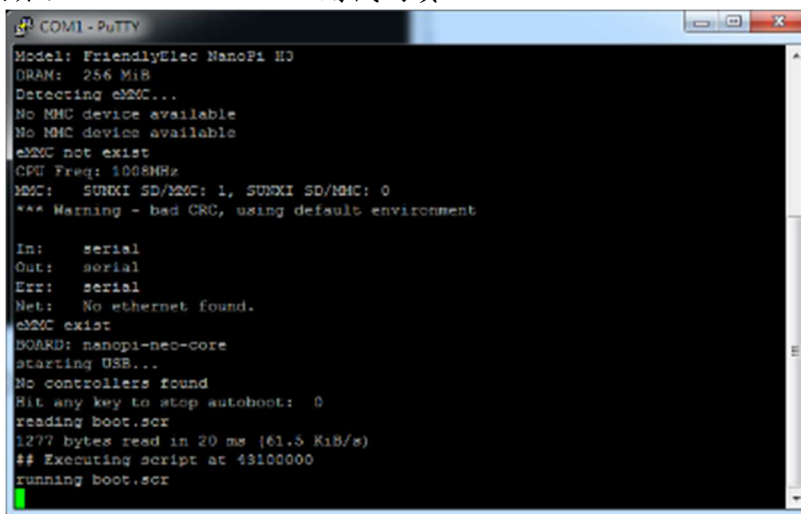


### ■ 啟動 putty



### ■ EBox-AIO-008 (通電)

顯示 Junminx-AIO-003 開機的資訊



```
COM1 - PuTTY
[ OK ] Started Hostname Service.
[ OK ] Started Network Manager Wait Online.
[ OK ] Reached target Network is Online.
Starting /etc/rc.local Compatibility...
Starting LSB: Start NTP daemon...
Starting LSB: disk temperature monitoring daemon...
[ 8.944321] rc.local[534]: ctp = 0
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started LSB: disk temperature monitoring daemon.
[ OK ] Started Getty on tty1.
[ OK ] Started Serial Getty on ttyS0.
[ OK ] Reached target Login Prompts.
[ OK ] Started LSB: Start NTP daemon.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Ubuntu 16.04.2 LTS NanoPi-NEO-Core ttyS0
NanoPi-NEO-Core login: pi (automatic login)
Last login: Thu Feb 11 16:28:05 UTC 2016 on tty1
```

### EBox-AIO-008 系統啟動完成

```
COM1 - PuTTY
[ OK ] Started Update UTMP about System Runlevel Changes.

Ubuntu 16.04.2 LTS NanoPi-NEO-Core ttyS0
NanoPi-NEO-Core login: pi (automatic login)
Last login: Thu Feb 11 16:28:05 UTC 2016 on tty1

          _ _ _ _ _
         | |_|_|_|_|
        / _/ _/ _/
       / _/ _/ _/
      / _/ _/ _/
     / _/ _/ _/
    / _/ _/ _/
   / _/ _/ _/
  / _/ _/ _/
 / _/ _/ _/
/_/ _/ _/

Welcome to Ubuntu 16.04.2 LTS 4.14.111
System load: 0.16          Up time: 10 sec
Memory usage: 15 % of 288Mb  IP:
CPU temp: 22.5°C
Usage of /: 58% of 1.1G

* Documentation: http://wiki.ubuntu.com/Ubuntu
* Forum: http://www.ubuntu.com/Forum/

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

注：出廠模式，透過 UART Console 連接進入 EBox-AIO-008 無需做密碼登入

### 2.4.3 進入 Ethernet console

因 EBox-AIO-008 的出廠網路設定為 DHCP mode，所以需將 EBox-AIO-008 連接到有 DHCP server 的局網(LAN)中，才可以分配到有效的網路 IP address，也才能夠進行 Ethernet console 的接入。

- 透過 UART console 查看網路設定

- 可透過指令 ifconfig

- ◆ EBox-AIO-008 未連接到有 DHCP Server 的網路

```
COM1 - PuTTY
pi@NanoPi-NEO-Core:~$
pi@NanoPi-NEO-Core:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:81:81:6c:91:bb
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          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:560 errors:0 dropped:0 overruns:0 frame:0
          TX packets:560 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:44576 (44.5 KB)  TX bytes:44576 (44.5 KB)

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

- ◆ EBox-AIO-008 連接到有 SHCP Server 的網路  
取得有效 IP address : 192.168.0.178

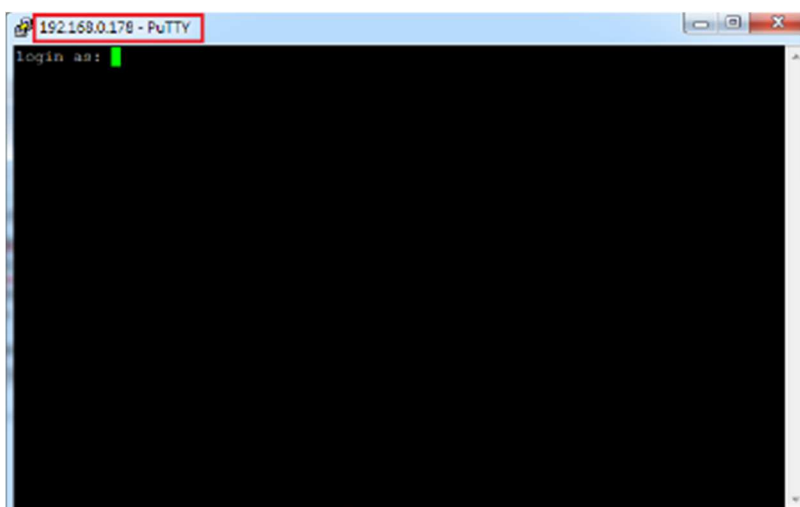
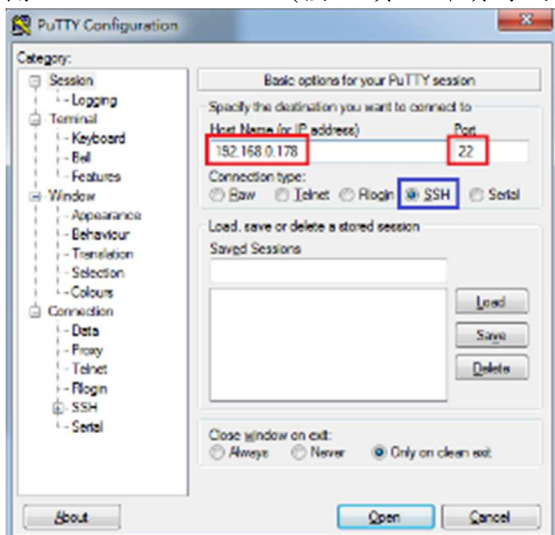
```

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:255.255.255.0
          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-008 IP address)”

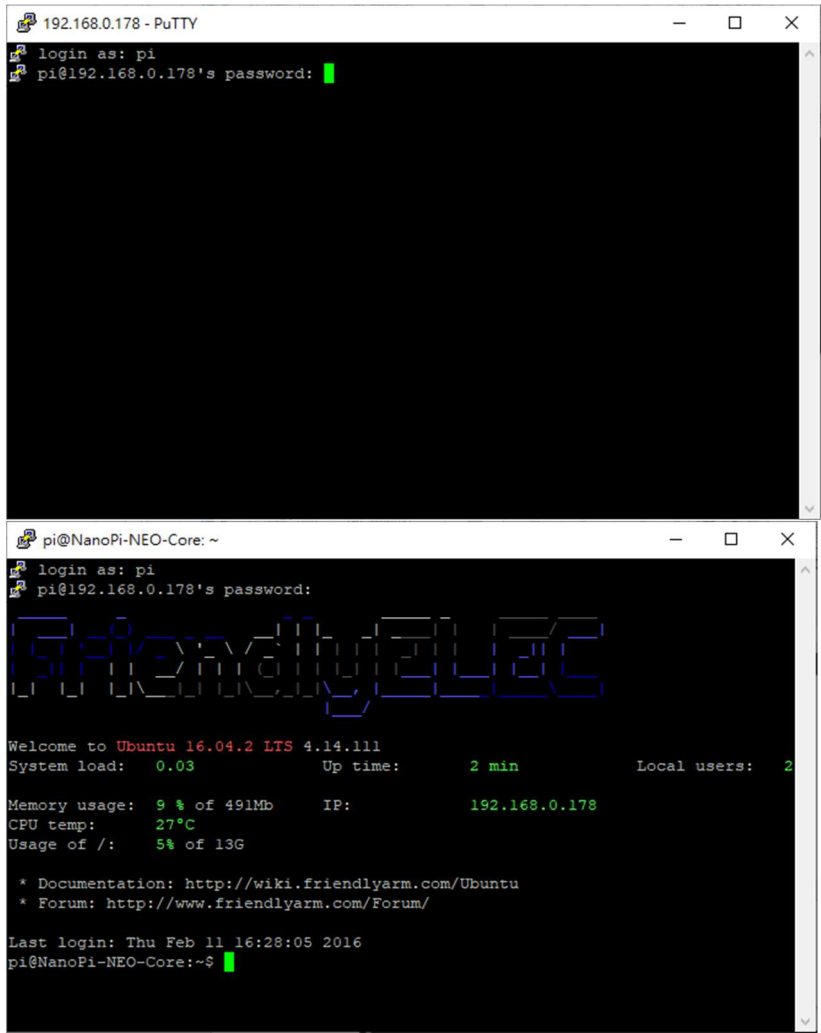


- 進入 EBox-AIO-008 Ethernet console  
輸入 EBox-AIO-008 出廠的一般用戶名及密碼

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

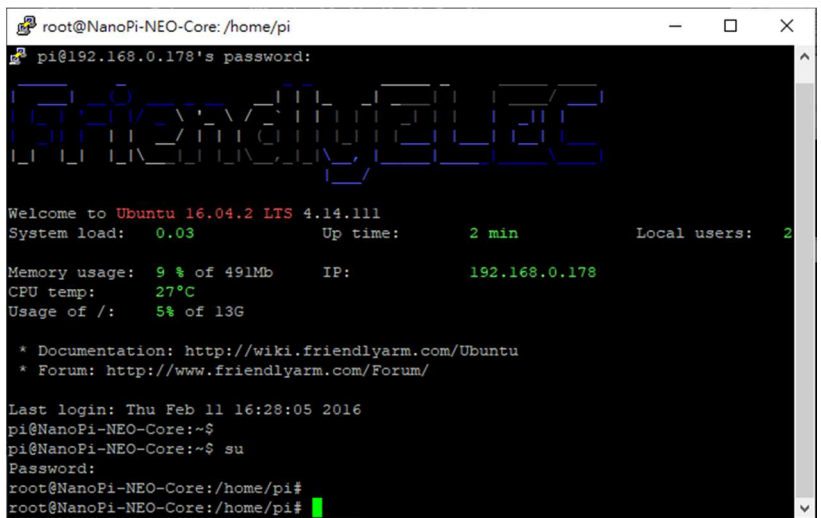


Password	pi
----------	----

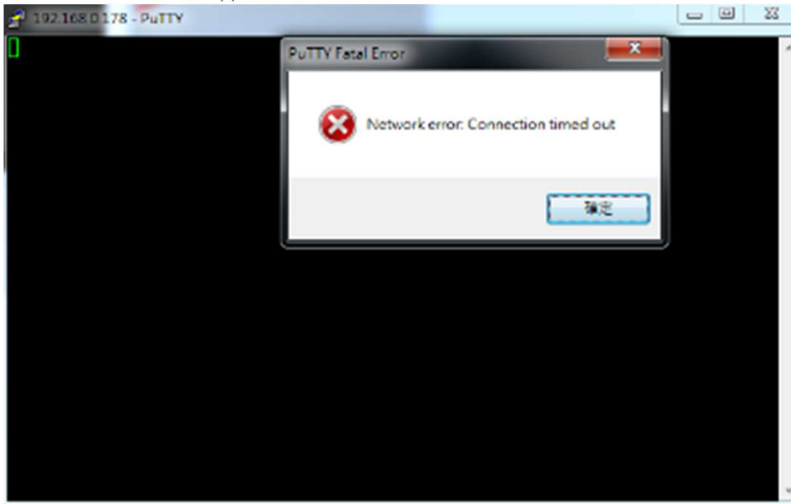


如需升級為管理層級，需執行指令”su”，並輸入 root 的密碼  
輸入 EBox-AIO-008 出廠的一般用戶名及密碼

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



- 無法連入 EBox-AIO-008 Ethernet console



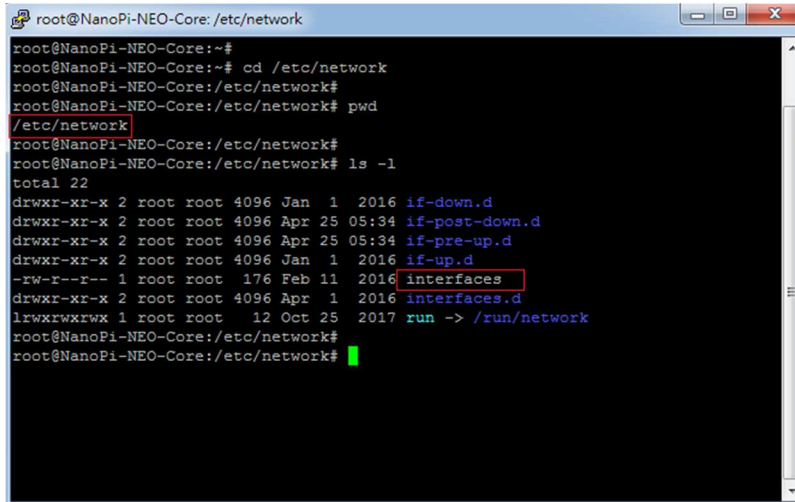
此狀況主要發生的原因常見有 3:

1. 輸入錯誤的 EBox-AIO-008 的 IP address
2. EBox-AIO-008 的網路有問題
  - EBox-AIO-008 的網路線沒接好
  - 不知 EBox-AIO-008 的網路 IP address 設定
3. EBox-AIO-008 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-008 在同一 IP 網段，可參考 ”附錄 A”

### 3 EBox-AIO-008 基本操作

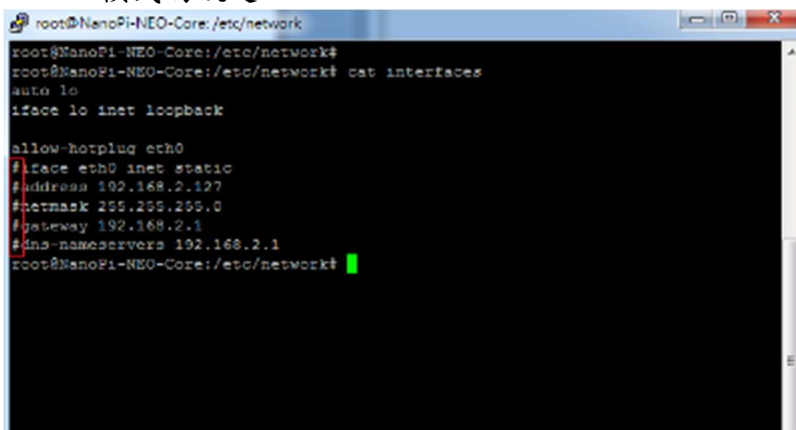
#### 3.1 變更網路設定

EBox-AIO-008 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-008，如要將 EBox-AIO-008 變更為自訂的固定 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  2016 if-post-down.d
drwxr-xr-x 2 root root 4096 Apr 25  2016 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



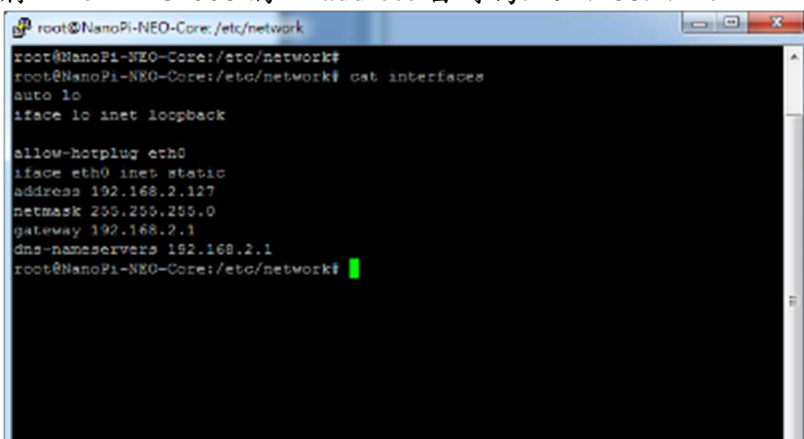
```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# cat interfaces
auto lo
iface lo inet loopback

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-008 的 IP address 自訂為: 192.168.2.127



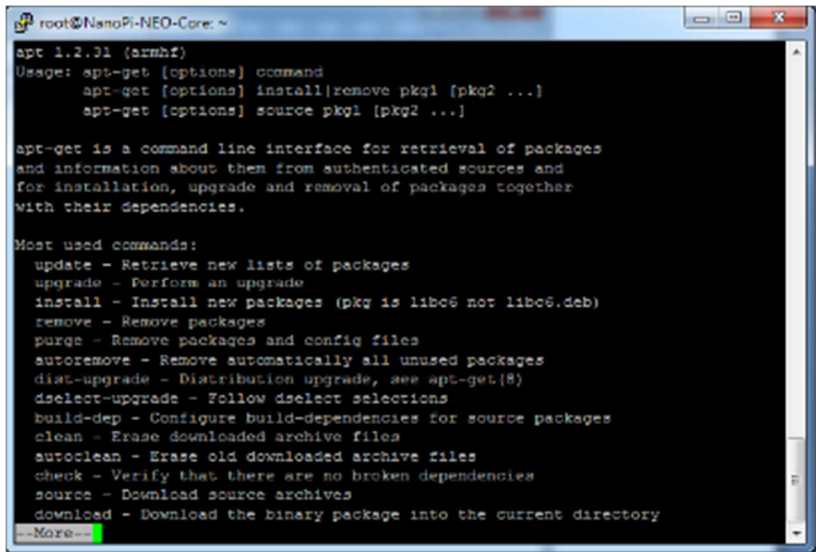
```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# cat interfaces
auto lo
iface lo inet loopback

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 下載新的 Linux 套件

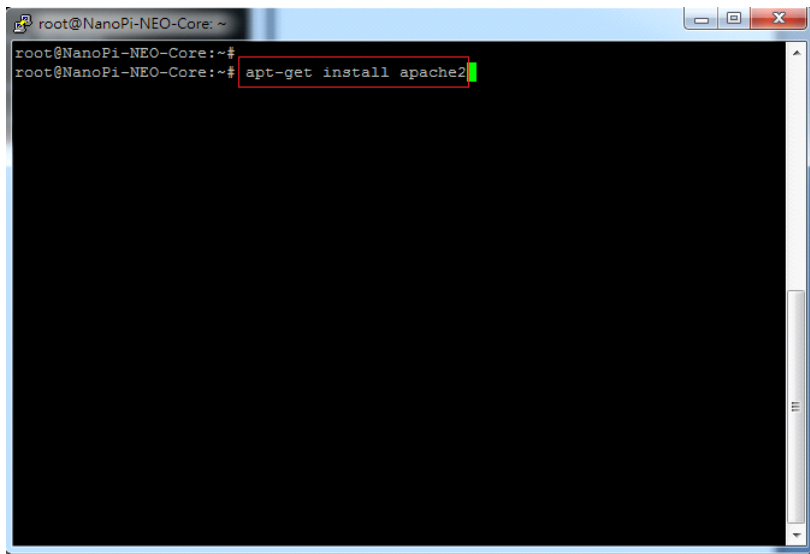
EBox-AIO-008 目前搭載的是 Linux 作業系統，開發或使用 EBox-AIO-008 可能需要安裝其它的開發套件或應用服務。

目前最常使用的方式，就是將 EBox-AIO-008 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 Apache 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:~#

```

### 3.3 GPS 操作

#### ■ 安裝 GPS 天線

- ◆ 機殼內部天線轉接線長度不可短於 15mm
- ◆ 需用 GPS 專用天線  
使用時天線需放置於”室外”，以接收衛星訊號



- ◆ 天線建議安裝於機殼下部右邊開孔



#### ■ 接收 GPS 信息

- ◆ 測試時可用串列埠傳輸工具程式 minicom  
可透過”apt-get install minicom” 安裝完成
- ◆ GPS 模組參數：
  - 串列埠：/dev/ttyS1
  - Baudrate：9600 bps
  - 格式：N-8-1 (None parity check, 8-databit, 1-stopbit)
  - 流量控制：無
- ◆ GPS 狀態

信號會不斷的傳入串列埠，只需進行接收

- GPS 尚位定位完成
  - GPS LED 恆亮
  - 接收資料參考圖示(minicom)：衛星信號尚未取得

```

COM99 - PuTTY
$GNVTG,,,,,,,,,N*2E
$GNZDA,,,,,,,,*56
$GPTXT,01,01,01,ANTENNA OK*35
$GNGGA,,,,,0,00,25.5,,,,,*64
$GNGLL,,,,,V,N*7A
$GPGSA,A,1,,,,,,,,,25.5,25.5,25.5*02
$BDGSA,A,1,,,,,,,,,25.5,25.5,25.5*13
$GPGSV,1,1,00*79
$BDGSV,1,1,00*68
$GNRMC,V,,,,,,,,,N*4D
$GNVTG,,,,,,,,,N*2E
$GNZDA,,,,,,,,*56
$GPTXT,01,01,01,ANTENNA OK*35
$GNGGA,,,,,0,00,25.5,,,,,*64
$GNGLL,,,,,V,N*7A
$GPGSA,A,1,,,,,,,,,25.5,25.5,25.5*02
$BDGSA,A,1,,,,,,,,,25.5,25.5,25.5*13
$GPGSV,1,1,00*79
$BDGSV,1,1,00*68
$GNRMC,V,,,,,,,,,N*4D
$GNVTG,,,,,,,,,N*2E
$GNZDA,,,,,,,,*56
$GPTXT,01,01,01,ANTENNA OK*35
    
```

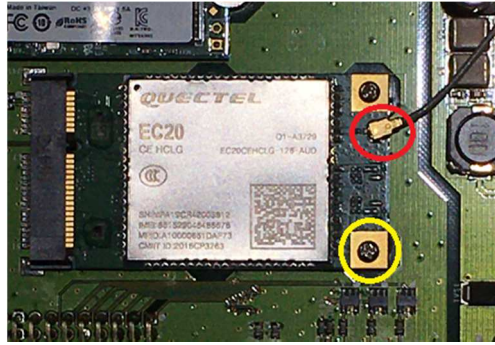
- GPS 定位完成
  - GPS LED 閃亮
  - 接收資料參考圖示(minicom)：衛星信號取得

```

COM99 - PuTTY
$GNGLL,2500.26984,N,12129.65680,E,084931.000,A,A*42
$GPGSA,A,3,07,16,26,27,31,194,195,,,,,1.8,1.3,1.2*38
$BDGSA,A,3,29,45,,,,,,1.8,1.3,1.2*21
$GPGSV,3,1,09,07,24,321,19,08,69,271,20,09,33,270,26,16,36,049,41*78
$GPGSV,3,2,09,26,19,077,21,27,58,021,28,31,06,135,34,194,65,062,35*46
$GPGSV,3,3,09,195,73,095,29*7E
$BDGSV,1,1,03,29,26,039,38,36,05,120,34,45,51,095,40*5C
$GNRMC,084931.000,A,2500.26984,N,12129.65680,E,0.00,0.00,290121,,,A*7C
$GNVTG,0.00,T,,M,0.00,N,0.00,K,A*23
$GNZDA,084931.000,29,01,2021,00,00*44
$GPTXT,01,01,01,ANTENNA OK*35
$GNGGA,084932.000,2500.26974,N,12129.65667,E,1,09,1.3,68.5,M,0.0,M,,*42
$GNGLL,2500.26974,N,12129.65667,E,084932.000,A,A*47
$GPGSA,A,3,07,16,26,27,31,194,195,,,,,1.8,1.3,1.2*38
$BDGSA,A,3,29,45,,,,,,1.8,1.3,1.2*21
$GPGSV,3,1,09,07,24,321,19,08,69,271,20,09,33,270,30,16,36,049,42*7C
$GPGSV,3,2,09,26,19,077,21,27,58,021,28,31,06,135,34,194,65,062,35*46
$GPGSV,3,3,09,195,73,095,29*7E
$BDGSV,1,1,03,29,26,039,38,36,05,120,34,45,51,095,40*5C
$GNRMC,084932.000,A,2500.26974,N,12129.65667,E,0.00,0.00,290121,,,A*79
$GNVTG,0.00,T,,M,0.00,N,0.00,K,A*23
$GNZDA,084932.000,29,01,2021,00,00*47
$GPTXT,01,01,01,ANTENNA OK*35
    
```

### 3.4 安裝 4G 行動通訊

- 安裝 4G 模組及 SIM 卡
  - ◆ 安裝 4G 模組
    - 建議使用：Questel EC20  
<https://www.quectel.com/product/ec20r21minipc1e.htm>
    - 注意固定螺絲規格：M2 (黃圈處)
    - 注意天線安裝 (紅圈處)



◆ 天線安裝

- 天線轉接線長度不可短於 10mm
- 建議安裝於機殼下部左邊開孔



◆ 安裝 SIM 卡

SIM 卡插槽於 PCB 背面，需先開啟機殼，卸下 PCB，完成 SIM 卡安裝後，再裝回 PCB 及機殼



■ 啟用 4G 模組

詳細步驟請參考

[http://wiki.friendlyarm.com/wiki/index.php/NanoPi\\_NEO\\_Core/zh#FriendlyCore.E4.B8.8B.E4.BD.BF.E7.94.A84G.E6.A8.A1.E5.9D.97EC20](http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Core/zh#FriendlyCore.E4.B8.8B.E4.BD.BF.E7.94.A84G.E6.A8.A1.E5.9D.97EC20)

### 3.5 安裝 SSD 硬碟

■ 安裝 M.2 SATA SSD

- ◆ 需為 2242 的長度規格
- ◆ 注意固定螺絲規格：M2”





- 在 console 介面透過指令 `lsusb` 確認 SATA 轉換芯片能運作  
安裝 SSD 模組後，應該會增加一個 usb device

```
root@NanoPi-NE0-Core:~# lsusb
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 003: ID 152d:0562 JMicron Technology Corp. / JMicron USA Techno
gy Corp.
Bus 005 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 004: ID 04e2:1414 Exar Corp.
Bus 004 Device 003: ID 04e2:1414 Exar Corp.
Bus 004 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
```

- 透過指令”fdisk”查看新增 SSD 的詳細可用空間信息

```
root@NanoPi-NE0-Core:~# fdisk -l
Disk /dev/mmcblk0: 3.7 GiB, 3909091328 bytes, 7634944 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x374a96c8

Device            Boot    Start      End Sectors  Size Id Type
/dev/mmcblk0p1                49152   131071    81920    40M 83 Linux
/dev/mmcblk0p2                131072 2492415 2361344    1.1G 83 Linux
/dev/mmcblk0p3                2492416 7634943 5142528    2.5G 83 Linux

Disk /dev/mmcblk0boot1: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mmcblk0boot0: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sda: 111.8 GiB, 120034123776 bytes, 234441648 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 250D609B-2C9F-4884-8A50-9D7BA2AFFEA0

Device      Start      End Sectors  Size Type
/dev/sda1   34         32767    32734    16M Microsoft reserved
/dev/sda2   32768     234438655 234405888 111.8G Microsoft basic data

Partition 1 does not start on physical sector boundary.
```

### 3.6 更新 RS-485 驅動程式(driver)

因部份核心板內建的 USB 轉 UART(for RS-485)驅動程式版本會啟動硬體流量控制(HW Flow Control)，會間接導致 RS-485 串列埠無法發出資料(Transmit)，建議替換驅動程式或重新編譯系統核心(compile kernel)



■ USB-UART 驅動原始碼：

<https://www.maxlinear.com/product/interface/uarts/usb-uarts/xr21b1411#pills-tools>

■ 替代用驅動程式：xr\_usb\_serial\_common.ko

◆ 可至連立達官網下載：xr\_usb\_serial\_common.ko

◆ 不一定適用每一版本，仍建議自行重新編譯(compile)

### 3.7 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

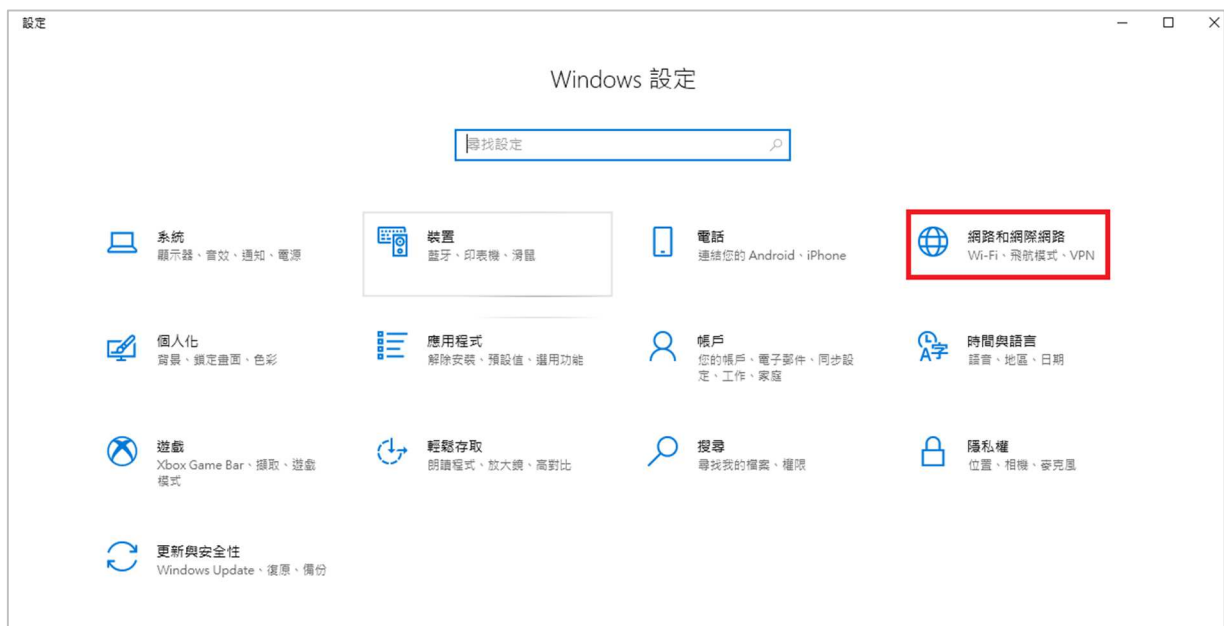
NanoPi NEO Core : [http://wiki.friendlyarm.com/wiki/index.php/NanoPi\\_NEO\\_Core](http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Core)

NanoPi NEO Core2 : [http://wiki.friendlyarm.com/wiki/index.php/NanoPi\\_NEO\\_Core2](http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Core2)

## 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。

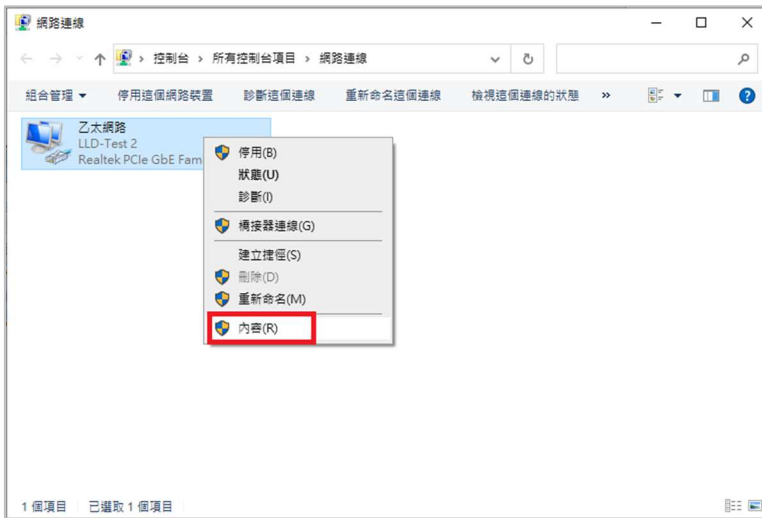


**Step.02** 變更介面卡設定。

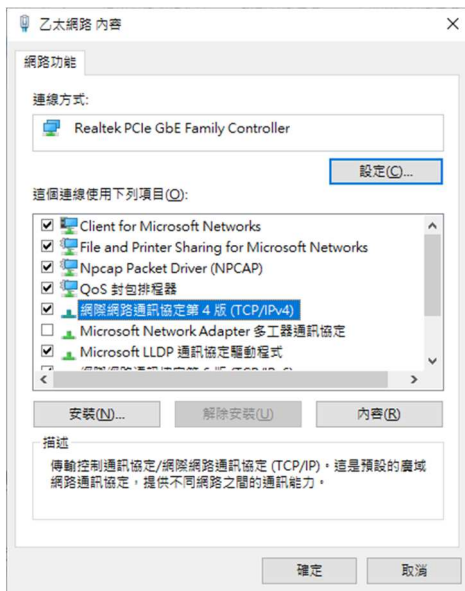




### Step.03 區域連線 → 按右鍵 內容。

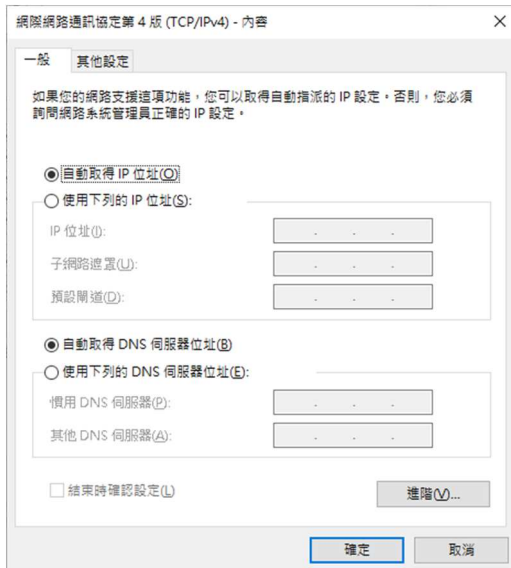


### Step.04 網際網路通訊協定第 4 版(TCP/IPv4) → 內容。



### Step.05 IP 地址：

確認測試主機和 EBox-AIO-008 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

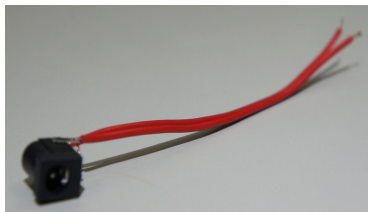
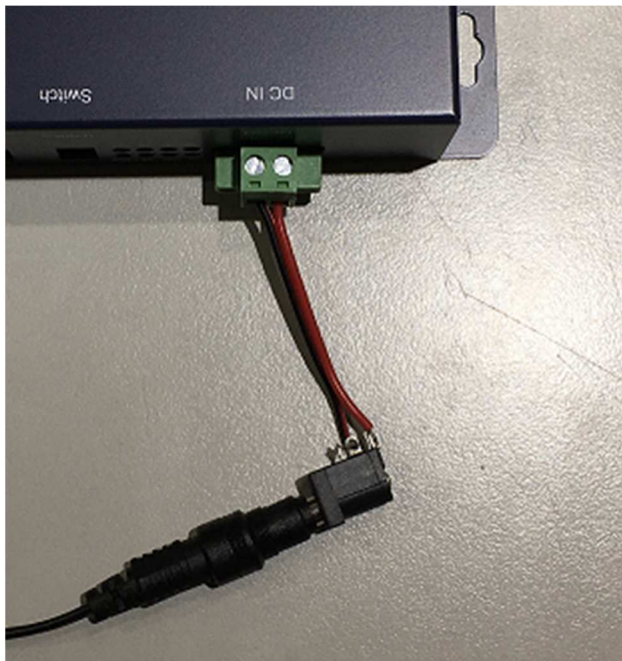


USB Converter		UART Console
Tx (白)	↔	RxD (pin-1)
Rx (綠)	↔	TxD (pin-2)
GND (黑)	↔	GND (pin-3)
DC (紅)	X	

- 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  $\Phi 2.1\text{mm}$  powerjack

### 附錄 C. 測試主機 COM 介面查詢

