

# EBox-AIO-009

## 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.....	5
1.6	各通訊及控制介面軟體定義.....	8
1.7	各擴充通訊晶片規格.....	8
2	第一次操作 EBox-AIO-009.....	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-009 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	安裝 4G 行動通訊.....	20
3.4	FriendlyARM NanoPi NEO Module 核心模組參考技術資訊.....	21
附錄 A.	主機網路設定.....	22
附錄 B.	轉接線.....	25
附錄 C.	外部配件安裝.....	26
附錄 D.	測試主機 COM 介面查詢.....	28



## 1 產品簡介

### 產品概述

EBox-AIO-009 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-009 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-009 成為一款專用的物聯網應用控制器或閘道(Gateway)。

EBox-AIO-009 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-009 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-009 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：6 點
- ▶ 信號種類：3.3V CMOS
- ▶ DIP Switch x 2 GPIO
- ▶ LED x 2 GPIO
- ▶ Beeper x 1 GPIO
- ▶ Button x 1 GPIO
- ▶ 電壓範圍：0 / 3.3VDC

### RS-485 序列埠介面

- ▶ 數量：2 組
- ▶ RS-485 信號：Data+, Data-, GND (支援自動流向控制)
- ▶ Multi-Drop Nodes：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：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：RS-232 (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、Wi-Fi、NB-IoT
- ▶ 擴充信號：USB、GPIO
- ▶ 接頭：Mini-PCIe 插槽

### 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(於核心模組，需開殼)

▷ 接頭：Micro SD 插槽

## 延伸擴充介面

▷ USB：3 組

▷ UART：1 組

▷ I2C：1 組

▷ SPI：2 組

▷ GPIO：4 組

▷ DC：3.3V、5V、24V 各 1 組

▷ 接頭：2x12 2.54mm 排針孔座

## 機構

▷ 材質：鍍鋅鋼板

▷ 尺寸：102.6 x 130.8 x 40 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

▷ LED 指示燈：電源, 網路, 序列埠, 4G、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
上(靠邊)	GND
下	DC+

DC Range: 9~24V DC



#### Console (RS-232)

腳位	信號
1 (左)	RxD
2	TxD
3	GND

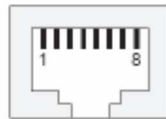
#### 2.54mm 排針座



#### LAN 乙太網路介面(Ethernet)

腳位	信號
1	ETX+
2	ETX-
3	ERX+
6	ERX-

#### RJ45 座



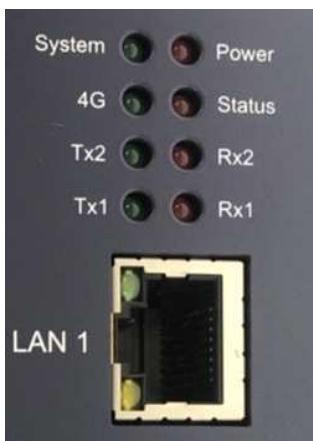
#### RS-485 (port 1~2)

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

	狀態指示燈*	System	Power	電源狀態
	無線通訊狀態	4G	Status	狀態指示燈*
	2 <sup>nd</sup> RS-485 Tx 狀態	Tx2	Rx2	2 <sup>nd</sup> RS-485 Rx 狀態
	1 <sup>st</sup> RS-485 Tx 狀態	Tx1	Rx1	1 <sup>st</sup> RS-485 Rx 狀態

\* 為軟體可控制的 LED  
Programmable LED

### System LED

- 以 GPIO 控制 (output)
- 對應核心定義：PIO\_G08
- 動作

GPIO status	LED status
Low	Light ON
High	Off

### Status LED

- 以 GPIO 控制 (output)
- 對應核心定義：PIO\_G09
- 動作

GPIO status	LED status
Low	Light ON
High	Off

## 1.5 其他定義

### 1.5.1 延伸擴充功能定義

- 2.54mm 2x12 座
- 零件編號: J5/(正) J6/(背)
- 單雙號腳位”左、右”並排
- 正反面直接 pin-pin 對應



(正面)



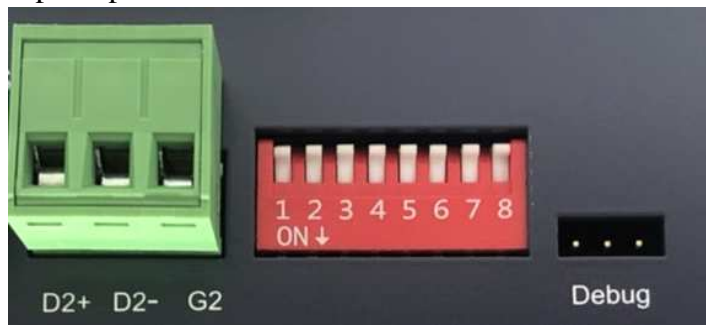
(背面)

腳位定義：正面

定義	腳位(左)	腳位(右)	定義
3.3V	01	02	3.3V
I2C-SDA (I2C2)	03	04	SPI-MOSI (SPI0)
I2C-SCL (I2C2)	05	06	SPI-MISO (SPI0)
USB-EXT-DP1 (USB Hub)	07	08	SPI-CLK (SPI0)
USB-EXT-DM1 (USB Hub)	09	10	SPI-CS (SPI0)
USB-EXT-DP3 (USB Hub)	11	12	USB-EXT-DP2 (USB Hub)
USB-EXT-DM3 (USB Hub)	13	14	USB-EXT-DM2 (USB Hub)
GND	15	16	GND
EXTIO-02 (GPIO-PA16/SPI1)	17	18	UART-Tx (UART3/SPI1)
EXTIO-03 (GPIO-PA02)	19	20	UART-Rx (UART3/SPI1)
EXTIO-04 (GPIO-PA03)	21	22	EXTIO-01 (GPIO-PA15/SPI1)
5VDC	23	24	5VDC

### 1.5.2 RS-485 電位電阻設定

8-pin dip switch

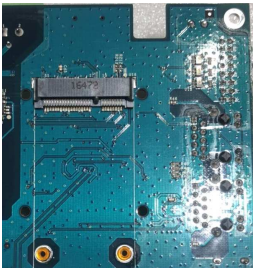



腳位	RS-485	ON 定義	OFF 定義	備註
1	P1	data-腳位 1K $\Omega$ 電位拉低	data-腳位 10K $\Omega$ 電位拉低	腳位 1、2 需 同時切換
2		data+腳位 1K $\Omega$ 電位拉高	data+腳位 10K $\Omega$ 電位拉高	
3		120 $\Omega$ 終端電阻	無終端電阻	腳位 3、4 不 可同時切換
4		600 $\Omega$ 終端電阻	無終端電阻	
5	P2	data-腳位 1K $\Omega$ 電位拉低	data-腳位 10K $\Omega$ 電位拉低	腳位 1、2 需 同時切換
6		data+腳位 1K $\Omega$ 電位拉高	data+腳位 10K $\Omega$ 電位拉高	
7		120 $\Omega$ 終端電阻	無終端電阻	腳位 3、4 不 可同時切換
8		600 $\Omega$ 終端電阻	無終端電阻	



### 1.5.3 其它





<p><b>DIP Switch &amp; Button</b></p> 	<p><b>DIP Switch</b></p> <ul style="list-style-type: none"> <li>- 以 GPIO 控制 (input)</li> <li>- 對應核心定義：PIO_L11、PIO_A17</li> <li>- 動作</li> </ul> <table border="1" data-bbox="534 264 986 398"> <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> <p><b>Button</b></p> <ul style="list-style-type: none"> <li>- 以 GPIO 控制 (input)</li> <li>- 對應核心定義：PIO_A07</li> <li>- 動作</li> </ul> <table border="1" data-bbox="534 600 986 734"> <thead> <tr> <th>GPIO status</th> <th>Switch status</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>Pushed</td> </tr> <tr> <td>High</td> <td>X</td> </tr> </tbody> </table>	GPIO status	Switch status	Low	ON	High	OFF	GPIO status	Switch status	Low	Pushed	High	X
GPIO status	Switch status												
Low	ON												
High	OFF												
GPIO status	Switch status												
Low	Pushed												
High	X												
<p><b>RTC</b></p>	<ul style="list-style-type: none"> <li>- 以 I2C 控制</li> <li>- 對應核心定義：I2C_2</li> <li>- RTC Battery size：CR1220</li> </ul>												
<p><b>Beeper</b></p>	<ul style="list-style-type: none"> <li>- 以 GPIO 控制 (output)</li> <li>- 對應核心定義：PIO_A061</li> <li>- 動作</li> </ul> <table border="1" data-bbox="534 1070 986 1205"> <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><b>4G Function</b></p>  	<ul style="list-style-type: none"> <li>- 可擴充 4G、Wi-Fi 模組</li> <li>- Mini-PCIe 插槽 (需開殼，在 PCB 背面)</li> <li>- USB 介面</li> <li>- 天線孔：2</li> <li>- SIM Card：nano card</li> </ul>												



<p>導軌配件固定孔</p> 	<ul style="list-style-type: none"> <li>- 2 個 <math>\Phi 3\text{mm}</math>，M3 螺絲孔位</li> <li>- 可安裝專用配件 DK-P02</li> <li>- 參考附錄 C</li> </ul>
<p>散熱片固定孔</p> 	<ul style="list-style-type: none"> <li>- 兩側面各 4 個 <math>\Phi 3\text{mm}</math>，M3 螺絲孔位</li> <li>- 可安裝專用配件 Heatsink-02</li> <li>- 參考附錄 C</li> </ul>

### 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/ttyS1
	Port 2	/dev/ttyS2
Console	(UART / TTL)	/dev/ttyS0
Audio		
擴充槽	(UART / TTL)	/dev/ttyS3
外掛記憶體	USB Disk	/dev/sda? (依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

## 2 第一次操作 EBox-AIO-009

### 2.1 測試環境建議

#### 2.1.1. 測試所需器材：

- EBox-AIO-009 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-009 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-009 專用 Console 線  
應該需多一條 USB 轉 RS-232 轉換線

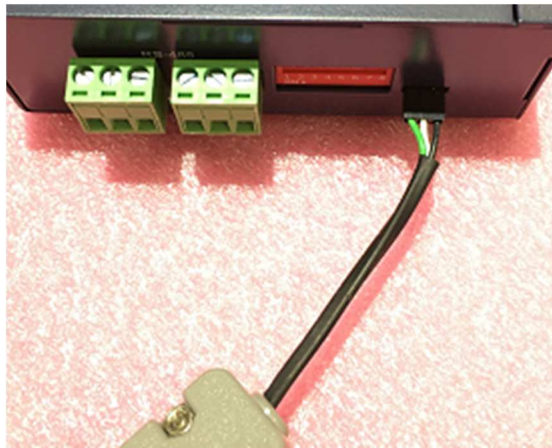


### 2.1.2. 連接安裝

將網路與 Console 線都與測試主機連接，做為開發主機於 EBox-AIO-009 的通訊橋樑。



#### ■ Console 連接說明



◆ 依 console 線腳位定義連接 EBox-AIO-009 console 排針

EBox-AIO-009		Console 線
RxD (pin-1)	↔	RxD
TxD (pin-2)	↔	TxD
GND (pin-3)	↔	GND

接線考參考附錄 B1

## 2.2 主控口(Console)簡介

EBox-AIO-009 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-009 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

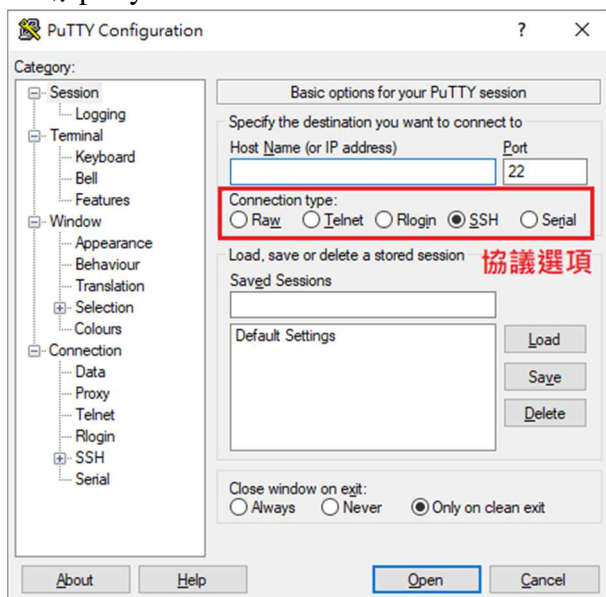
## 2.4 操作 console 主控口

### 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-009

- 下載 putty  
<https://putty.org>

- 啟動 putty



### 2.4.2 進入 UART Console

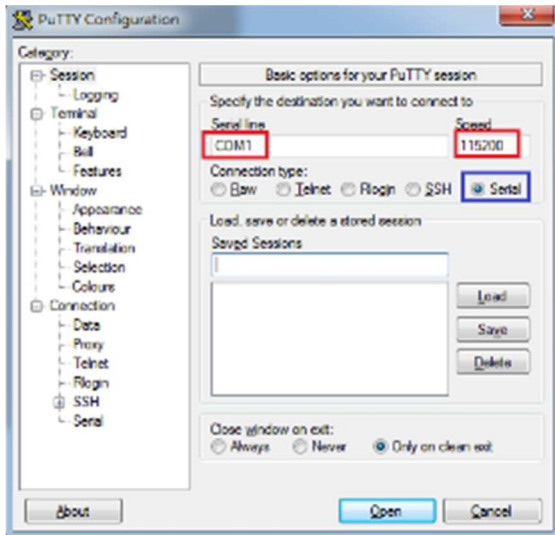
選擇 “Serial”通訊介面

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 D.”)

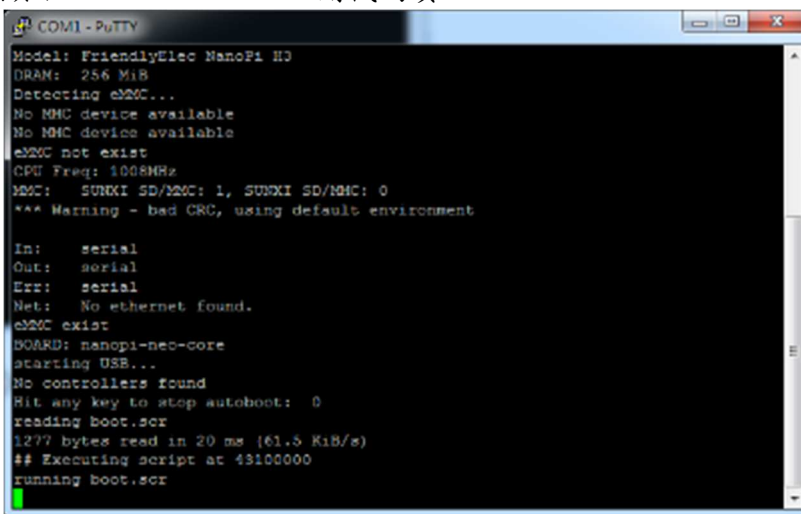


### ■ 啟動 putty



### ■ EBox-AIO-009 (通電)

顯示 Junminx-AIO-003 開機的資訊





- ◆ EBox-AIO-009 連接到有 SHCP Server 的網路  
取得有效 IP address : 192.168.0.178

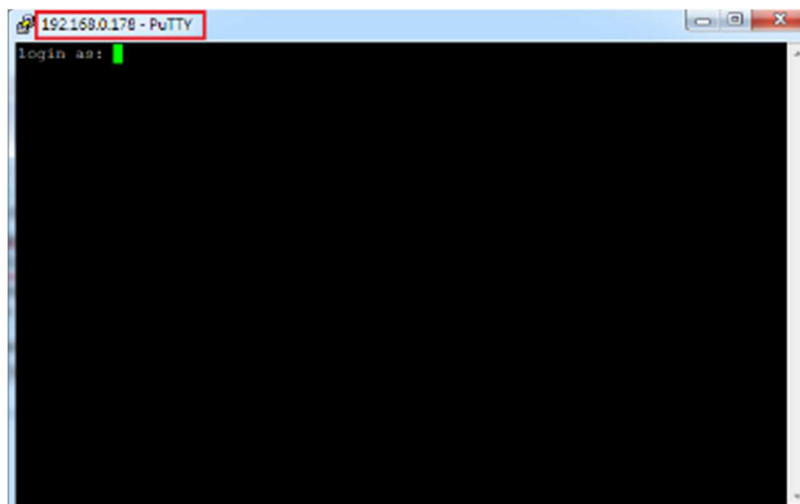
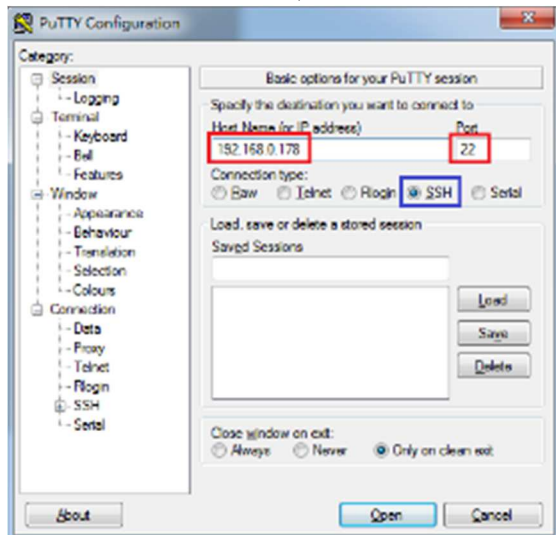
```

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-009 IP address)”

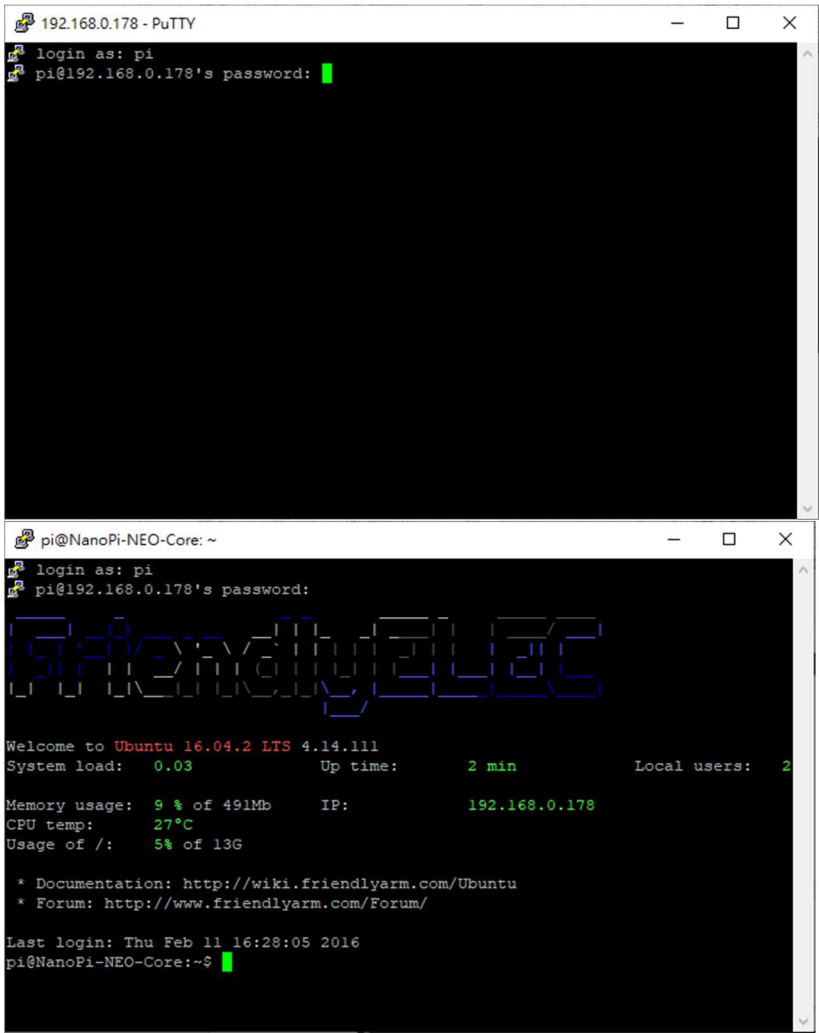


- 進入 EBox-AIO-009 Ethernet console  
輸入 EBox-AIO-009 出廠的一般用戶名及密碼

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

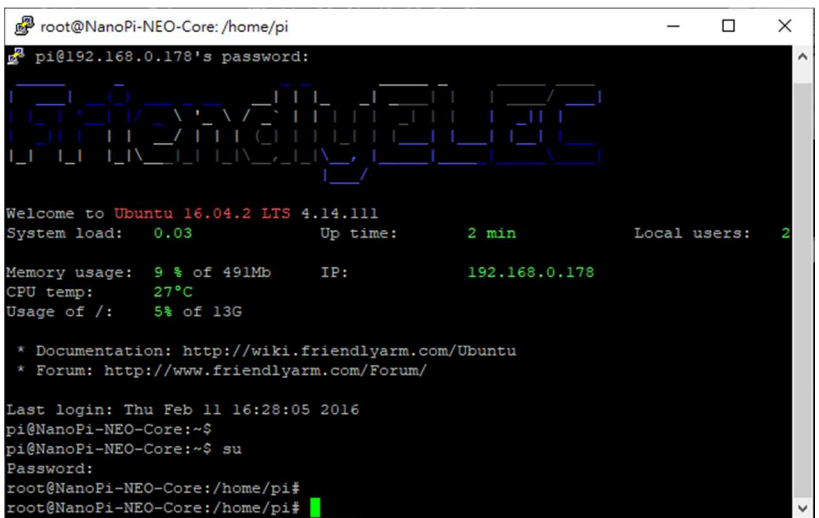


Password	pi
----------	----

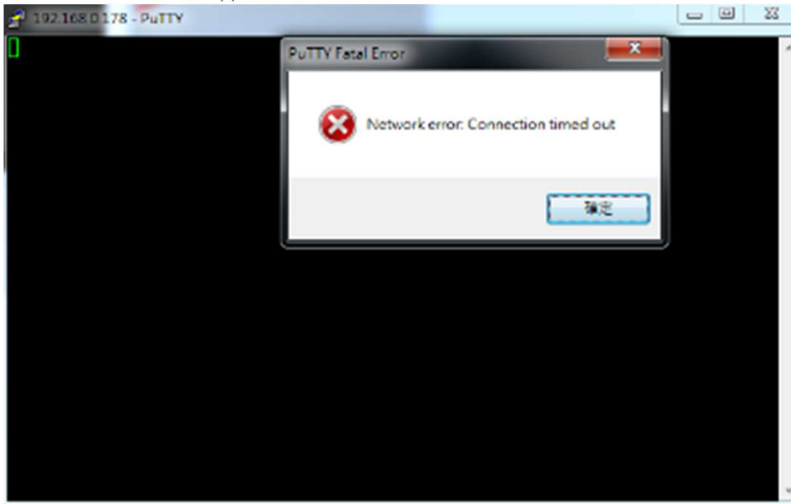


如需升級為管理層級，需執行指令”su”，並輸入 root 的密碼  
輸入 EBox-AIO-009 出廠的一般用戶名及密碼

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



- 無法連入 EBox-AIO-009 Ethernet console



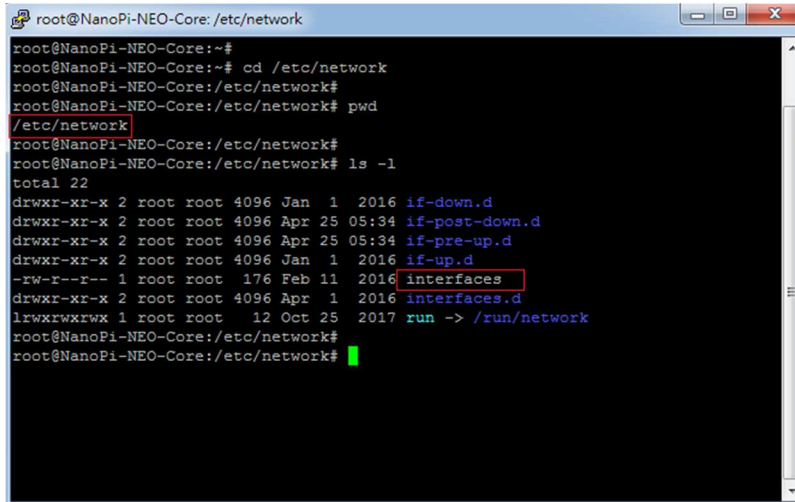
此狀況主要發生的原因常見有 3:

1. 輸入錯誤的 EBox-AIO-009 的 IP address
2. EBox-AIO-009 的網路有問題
  - EBox-AIO-009 的網路線沒接好
  - 不知 EBox-AIO-009 的網路 IP address 設定
3. EBox-AIO-009 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-009 在同一 IP 網段，可參考 ”附錄 A”

### 3 EBox-AIO-009 基本操作

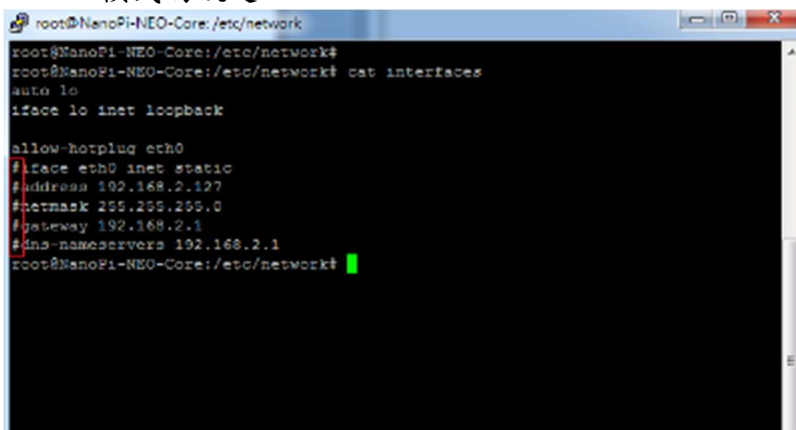
#### 3.1 變更網路設定

EBox-AIO-009 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-009，如要將 EBox-AIO-009 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



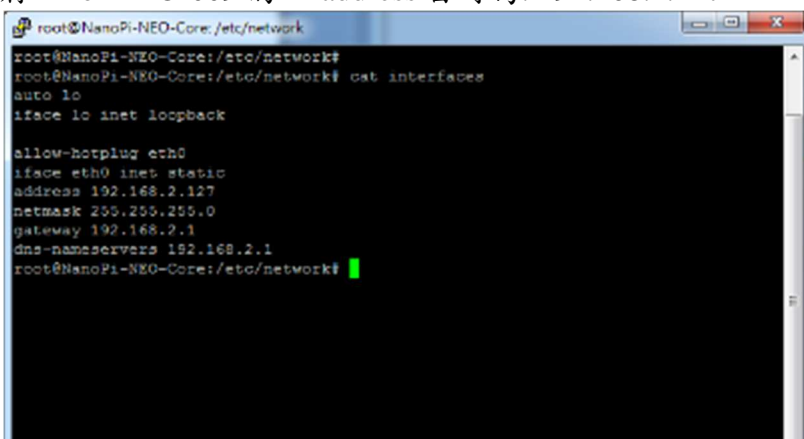
```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# cat interfaces
auto lo
iface lo inet loopback

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-009 的 IP address 自訂為: 192.168.2.127



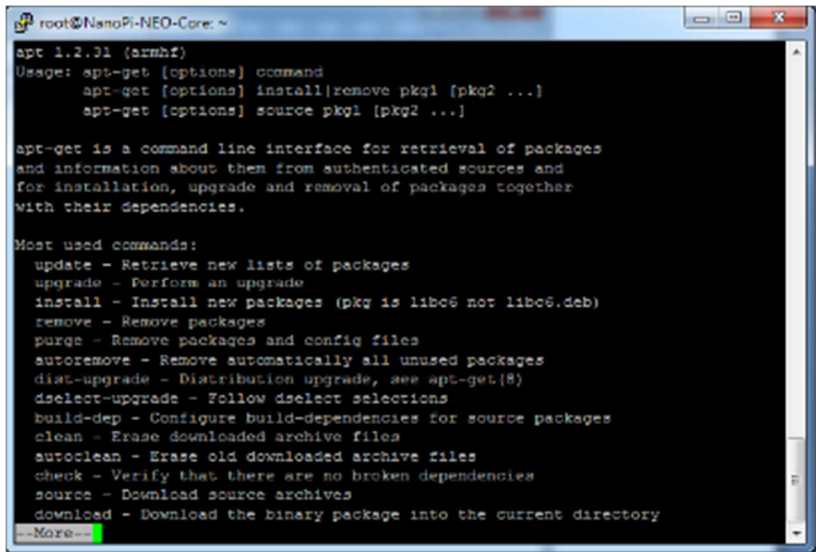
```
root@NanoPi-NEO-Core: /etc/network
root@NanoPi-NEO-Core:/etc/network#
root@NanoPi-NEO-Core:/etc/network# cat interfaces
auto lo
iface lo inet loopback

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

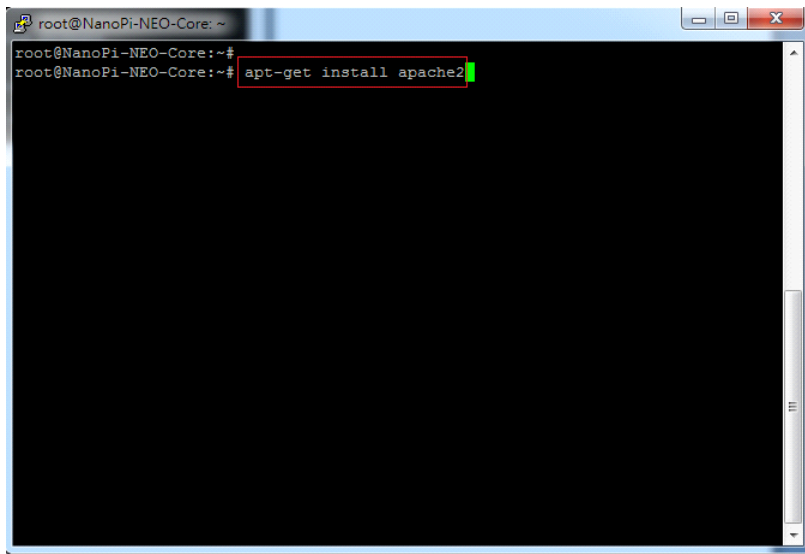
EBox-AIO-009 目前搭載的是 Linux 作業系統，開發或使用 EBox-AIO-009 可能需要安裝其它的開發套件或應用服務。

目前最常使用的方式，就是將 EBox-AIO-009 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 安裝 4G 行動通訊

#### ■ 安裝 4G 模組及 SIM 卡

##### ◆ 安裝 4G 模組

- 建議使用：Quectel 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
- 如無線模只需安裝一支天線，建議安裝於機殼下部天線孔(與模組同側)



◆ 安裝 SIM 卡

打開上蓋的保護套，安裝 SIM Card (只能安裝 Nano Card)



■ 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)

### 3.4 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

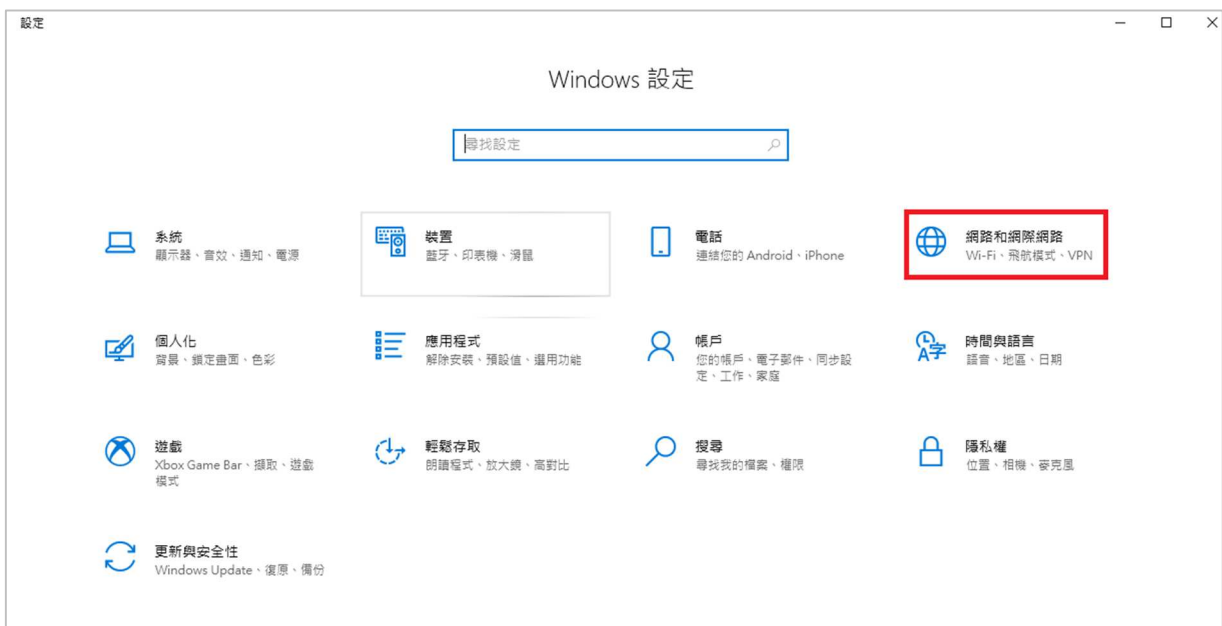
NanoPi NEO Core : [http://wiki.friendlyarm.com/wiki/index.php/NanoPi\\_NEO\\_Core](http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Core)

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。



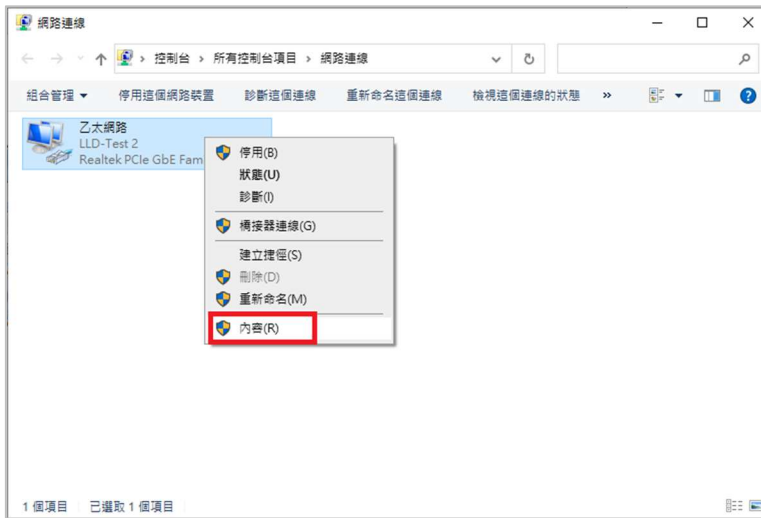
**Step.02** 變更介面卡設定。



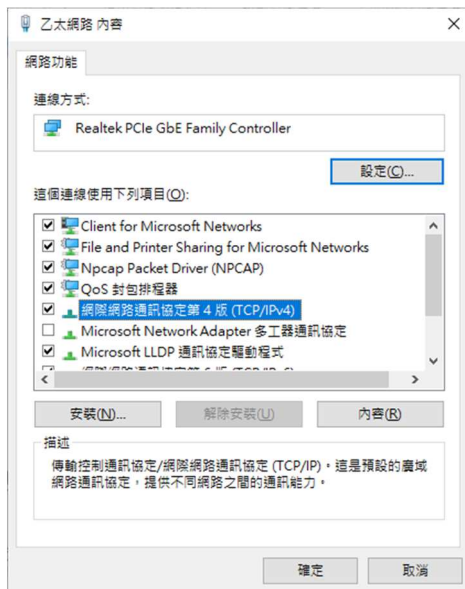




### Step.03 區域連線 → 按右鍵 內容。



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

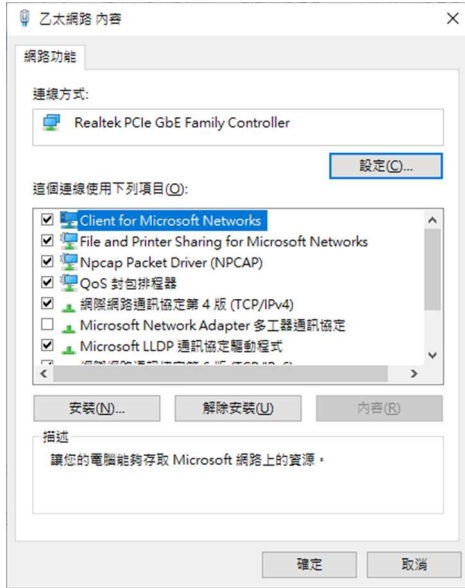


### Step.05 IP 地址：

確認測試主機和 EBox-AIO-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。



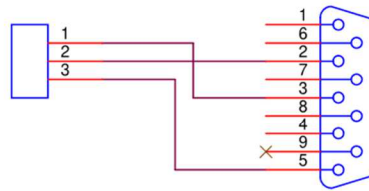
附錄 B. 轉接線

B1. Console 線 (CB-AIOCON-10)



接腳定義

3-pin header 腳位編號	信號定義	DB9 母頭 腳位編號
1 (左-白線)	<b>RXD</b>	<b>3</b>
2	<b>TXD</b>	<b>2</b>
3	<b>GND</b>	<b>5</b>



B2. 電源轉接線

DC Power Jack 轉 Terminal Block



內徑  $\Phi 2.1\text{mm}$  powerjack

## 附錄 C. 外部配件安裝

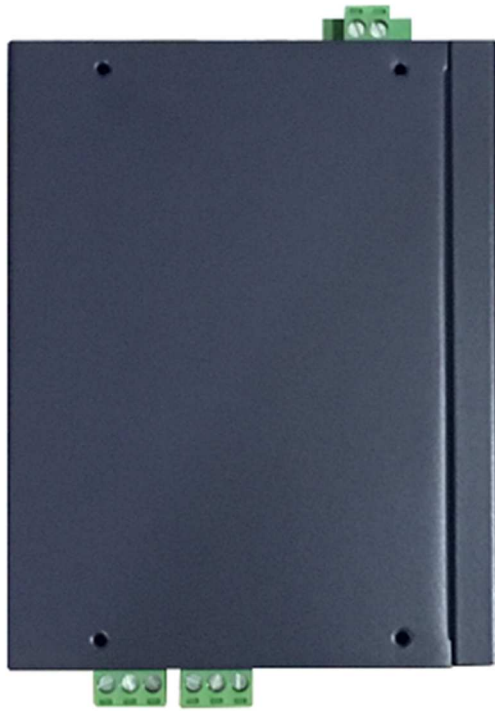
### C1. 導軌配件安裝

專用導軌配件：DK-P02 (附 2 顆 M3 皿頭螺絲)



## C2. 殼外散熱片安裝

專用 EBox-AIO-009 散熱片：Heatsink-02 (每片附 4 顆 M3 螺絲)



## 附錄 D. 測試主機 COM 介面查詢

